



Federal University of Bahia
Programme of Post-graduation in Mechatronics

**Road Detection in Traffic Analysis:
A Context-aware Approach**

Marcelo Mendonça dos Santos

2013

Road Detection in Traffic Analysis: A Context-aware Approach

Marcelo Mendonça dos Santos

*Submitted in partial fulfillment of
the requirements for the degree of
Master in Mechatronics*

Programme of Post-graduation in Mechatronics
Federal University of Bahia

under supervision of
Prof. Dr. Luciano Oliveira (advisor)

Copyright© 2013 by MARCELO MENDONÇA DOS SANTOS. All rights reserved.

S237 Santos, Marcelo Mendonça dos.
Road detection in traffic analysis: a context-aware
approach / Marcelo Mendonça dos Santos. – Salvador, 2013.
60 f. : il. color.

Orientador: Prof. Dr. Luciano Rebouças de Oliveira.
Dissertação (Mestrado) – Universidade Federal da
Bahia. Escola Politécnica, 2013.

1. Levantamentos de trânsito. 2. Trânsito - Controle
eletrônico. 3. Processamento de imagens. 4. Visão por
computador. I. Oliveira, Luciano Rebouças de. II.
Universidade Federal da Bahia. III. Título.


CDD: 629.04

TERMO DE APROVAÇÃO


MARCELO MENDONÇA DOS SANTOS

ROAD DETECTION IN TRAFFIC ANALYSIS: A CONTEXT-AWARE APPROACH

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre em Mecatrônica, Universidade Federal da Bahia, pela seguinte banca examinadora:

Luciano Rebouças de Oliveira – Orientador 
Doutor em Engenharia Elétrica e de Computadores, Instituto de Sistemas e Robótica da Universidade de Coimbra, Portugal
Universidade Federal da Bahia

Perfilino Eugênio Ferreira Júnior – Membo interno 
Doutor em Matemática, Instituto Nacional de Matemática Pura e Aplicada (IMPA)
Universidade Federal da Bahia

Luiz Gustavo Mirisola – Membro externo 
Doutor em Engenharia Elétrica e de Computadores, Instituto de Sistemas e Robótica da Universidade de Coimbra, Portugal
Universidade Federal do ABC

Salvador, 17 de fevereiro de 2014

For those who plant trees

Acknowledgments

I have not found words beautiful enough to congratulate my supporters. Perhaps I will never find them. Anyway, I would like to firm here my compromise in honoring these wonderful people anywhere I go.

Abstract

Correctly identifying the road area on an image is a crucial task for many traffic analyses based on surveillance cameras and computer vision. Despite that, most of the systems do not provide this functionality in an automatic fashion; instead, the road area needs to be annotated by tedious and inefficient manual processes. This situation results in further inconveniences when one deals with a lot of cameras, demanding considerable effort to setup the system. Besides, since traffic analysis is an outdoor activity, cameras are exposed to disturbances due to natural events (e.g., wind, rain and bird strikes), which may require recurrent system reconfiguration. Although there are some solutions intended to provide automatic road detection, they are not capable of dealing with common situations in urban context, such as poorly-structured roads or occlusions due to objects stopped in the scene. Moreover in many cases they are restricted to straight-shaped roads (commonly freeways or highways), so that automatic road detection cannot be provided in most of the traffic scenarios.

In order to cope with this problem, we propose a new approach for road detection. Our method is based on a set of innovative solutions, each of them intended to address specific problems related to the detection task. In this sense, a context-aware background modeling method has been developed, which extracts contextual information from the scene in order to produce background models more robust to occlusions. From this point, segmentation is performed to extract the shape of each object in the image; this is accomplished by means of a superpixel method specially designed for road segmentation, which allows for detection of roads with any shape. For each extracted segment we then compute a set of features, the goal of which is supporting a decision tree-based classifier in the task of assigning the objects as being road or non-road. The formulation of our method — a road detection carried out by a combination of multiple features — makes it able to deal with situations where the road is not easily distinguishable from other objects in the image, as when the road is poorly-structured.

A thorough evaluation has indicated promising results in favour of this method. Quantitatively, the results point to 75% of accuracy, 90% of precision and 82% of recall over challenging traffic videos caught in non-controlled conditions. Qualitatively, resulting images demonstrate the potential of the method to perform road detection in different situations, in many cases obtaining quasi-perfect results.

Contents

1	Introduction	1
1.1	Motivation	3
1.2	Goals	3
1.3	Key contributions	4
1.4	Chapter map	4
2	Background	7
2.1	The need for road detection	8
2.1.1	Calibrating traffic radars	8
2.1.2	Road monitoring by satellites and aircrafts	10
2.1.3	Onboard road detection	15
2.2	Road detection in traffic surveillance applications	19
2.2.1	The flowcharts of detection	20
2.2.2	Background subtraction	21
2.2.3	BGO methods	24
2.2.4	FGO methods	28
2.2.5	HBF methods	31
2.2.6	Summary of related works	35
2.3	Advanced BGS methods	36
2.3.1	Adaptive Background Learning	36
2.3.2	Gaussian-based methods	36
2.3.3	Fuzzy-based methods	38
2.3.4	Multi-layer BGS based on color and texture	39
2.3.5	Neural network with self-organization approach	40
2.4	Supervoxel segmentation	41
2.4.1	Simple linear iterative clustering	42
2.4.2	Graph-based image segmentation	43
2.5	Relation to our work	43
3	Overview of the proposed road detection system	45
3.1	Requirements and constraints	45

3.2	System conception	46
3.3	System design	48
3.4	Dataset	49
3.5	Closure	50
4	Context-aware road detection system	53
4.1	Context-supported road information for background modeling	54
4.1.1	Approximated median BGS	55
4.1.2	Vehicle filtering	56
4.1.3	Road color-based adaptive analysis	57
4.2	Supapixel segmentation based on edge density	59
4.3	Simple-but-efficient strategies for feature extraction	61
4.3.1	Gray amount-based color feature	62
4.3.2	Edge density-based texture feature	63
4.3.3	Horizon line estimation based on edge density	64
4.3.4	Vehicle motion-based features	65
4.4	Object classification	66
4.5	Closure	67
5	Experimental evaluation	69
5.1	Experiments	70
5.1.1	Background modeling performance	70
5.1.2	Supapixel generation performance	73
5.1.3	Road detection performance	79
5.2	Analysis and closure	81
6	Conclusion	83
A	Implementation details	85
A.1	Parameters of CRON	85
A.2	Parameters of the supapixel method	85
	References	87

List of Figures

1.1	The problem of annotating the road manually in traffic analysis systems: minimal changes in the camera position can result in grotesque errors in the analysis.	2
2.1	Traffic radar mounted in elevated side-looking configuration, where intermediate frequencies of emitted signals are proportional to the distances from reflective objects to the radar. Variations in echo power are used to detect road lanes. Image taken from [Zhang <i>et al.</i> 2008].	9
2.2	Interferometric radar at field experiments. The device is used to detect road lanes in order to support a traffic flow analysis process. Image taken from [Felguera-Martin <i>et al.</i> 2012].	10
2.3	Road detection process proposed by [Rathinam <i>et al.</i> 2008]. Following the arrows: an input road image is caught; the input image is rectified; a road profile is obtained by computing image statistics (means and variances); road boundaries are estimated from the obtained profile. Image taken from <i>ibid.</i>	11
2.4	Road detection method proposed by [Li <i>et al.</i> 2009]. (a) Input image, (b) image after exclusion of non-gray areas, (c) final detection result and (d) RGB histograms used to extract road color information. Image adapted from <i>ibid.</i>	13
2.5	Road detection process found in [Gaetano <i>et al.</i> 2011]. (a) Input image with superimposed contours, (b) result of distance function computed from the contours, (c) result of the skeletonization, (d) skeleton after pruning, (e) road skeleton selection and (f) final detection result. Image adapted from <i>ibid.</i>	14
2.6	Predefined patterns of curves used to estimate the road curvature. On the left, the nine predefined curves; on the right, the parameters of the third curve. Image taken from [He <i>et al.</i> 2004].	16
2.7	Road detection process proposed in [He <i>et al.</i> 2004]. (a) Input image, (b) result of projective transformation, (c) extracted edges, (d) edges after shifting by three different predefined curvatures, (e) voting results for each curvature, (f) area selected for computing the road color, (g) road mask and (h) final detection result. Image adapted from <i>ibid.</i>	16

2.8	Examples of illuminant invariant images. First row, input images; second row, generated illuminant invariant models. Images taken from [Alvarez <i>et al.</i> 2008].	17
2.9	Examples of road segmentation produced by [Alvarez <i>et al.</i> 2008]. First row, input images; second row, segmented roads. Images taken from <i>ibid.</i>	18
2.10	Flowchart of the method proposed by [Kuhnl <i>et al.</i> 2011]. Image taken from <i>ibid.</i>	19
2.11	Road detection flowcharts. BGO: detection based only on background information. FGO: detection based only on foreground information. HBF: hybrid detection, based on both background and foreground information.	21
2.12	The Hough transform. (a) Image space, (b) parameter space and (c) accumulator corresponding to (b). Images taken from [Illingworth & Kittler 1988].	25
2.13	Lane detection method proposed by [Gao <i>et al.</i> 2011]. (a) Method flowchart and (b) examples of images corresponding to each step at the detection process. Images adapted from <i>ibid.</i>	26
2.14	Road borders detection algorithm proposed by [Mazaheri & Mozafari 2011]. Left, detected lines. Right, image segmentation into near field and far field. Images taken from <i>ibid.</i>	26
2.15	Lane detection algorithm proposed by [Li & Zhong 2009]. (a) Long lines extracted using HT and the circle centred at the vanishing point revealed by them. (b) Candidate lane regions obtained from flood fill. (c) Detected road lanes. Images taken from <i>ibid.</i>	27
2.16	Flowchart of the lane estimation method proposed by [Lai & Yung 2000]. Image taken from <i>ibid.</i>	28
2.17	Road shape obtained from foreground objects. (a) Input frame. (b), (c), (d) road regions obtained after 2, 5 and 15 seconds, respectively. Image taken from [Luo <i>et al.</i> 2011].	29
2.18	Lane detection method proposed by [Pan <i>et al.</i> 2010]. (a) Input image, (b) histogram of the motion, (c) lane areas and (d) detected lane markings. Image adapted from <i>ibid.</i>	29
2.19	Flowchart of the lane estimation algorithm proposed by [Melo <i>et al.</i> 2006]. Image adapted from <i>ibid.</i>	30
2.20	Road mask generation from the distribution of the pixels intensities. Left, the input frame; middle, the intensities distributions of pixels inside and outside the road; and right, the obtained road mask. Image taken from [Shin <i>et al.</i> 2006].	31

2.21	Road detection method proposed by [Chung <i>et al.</i> 2004]. (a) Input frame, (b) extracted foreground objects, (c) background areas corresponding to foreground objects, (d) major road component, (e) hole filling result, (f) fuzzy map, (g) defuzzified map and (h) final result. Images adapted from <i>ibid.</i>	32
2.22	Examples of road borders detection by the method of [Helala <i>et al.</i> 2012]. Images adapted from <i>ibid.</i>	33
2.23	Lane detection method proposed by [Liu & Wang 2010]. (a) Input image and detected seeds (green squares), (b) region growing segmentation, (c) straight lines extracted by Hough Transform, (d) moving object tracking, (e) detected trajectories, and (f) detected lane markings. Images adapted from <i>ibid.</i>	34
2.24	Adaptive background learning (ABL). (a) Input frames, (b) difference of frames in (a), (c) foreground/background separation, (d) rectified images, and (e) generated background model. Images adapted from [Zhang <i>et al.</i> 2003].	37
2.25	Foreground detection process using fuzzy Choquet integral. Image adapted from [El Baf <i>et al.</i> 2008].	39
2.26	Examples of image segmentation. (a) Original image, (b) superpixels produced by the graph-based approach (images taken from [Felzenszwalb & Huttenlocher 2004]) and (c) superpixels generated by applying SLIC on (a).	41
2.27	Difference in the superpixel search regions of k -means and SLIC. (a) Standard k -means searches the entire image, while (b) SLIC searches a limited region. Image taken from [Achanta <i>et al.</i> 2012].	42
3.1	Rationale behind the proposed road detection method. (a) Input traffic image, (b) segmentation of the objects in (a) after BGS, (c) segments predominantly gray, (d) segments homogeneously textured, (e) segments fully bellow the horizon line, (f) segments where vehicle motion occurs, (g) segments selected as being road and (h) the resulting road mask overlapping (a).	48
3.2	Overview of the proposed road detection system: First, video frames are separated into background and foreground by background subtraction (BGS). Next, superpixel segmentation and feature extraction are performed in the background model, while the foreground model is submitted to a motion analysis. Finally, object classification matches each image object with the corresponding feature vector in order to produce a road mask.	49

3.3	Some instances of our dataset. The images illustrate some of the difficulties imposed to our road detection method, such as roads of different shapes with worn out markings, vehicles parked on roadside, shadows, lighting variations, pedestrians, trees, etc.	50
4.1	Overview of the context-supported road information for background modeling (CRON). The method is comprised of three main tasks: approximated median BGS, which performs background/foreground separation; vehicle filtering, responsible for extracting contextual information from the scene; and color analysis, the process intended to generate background models adaptively.	54
4.2	Vehicle filtering for road sample extraction. Following the arrows: foreground blobs are evaluated by the solidity criterion, which measures the area of intersection among the blob and its convex-hull; aspect ratio of this convex-hull is analyzed: if an object meets this criterion, it is considered car; orientation relative to the x-axis is further considered in order to distinguish vehicles from people; at the end, the road regions are taken as the bottom of the convex-hulls of the objects passed in the criteria chain. . . .	56
4.3	Superpixel segmentation based on edge density. From left to right, columns show intermediate results at iterations 1, 3 and 6 (for a total of 6); T is the threshold pair for a Canny-based edge detection. The rows: (a) input image, (b) detected edges, (c) edge density filtering, (d) thresholding of (c), (e) contour extraction of (d), (f) accumulated contours, and (g) final result after applying morphological operations on (f).	60
4.4	Gray amount-based color feature represented in heat maps. First row, input images; second row, correspondent gray amount images represented in heat maps; on the bottom, heat map scale.	63
4.5	Edge density-based texture feature represented in heat maps. First row, input images; second row, correspondent edge density images represented in heat maps; on the bottom, heat map scale.	63
4.6	Horizon line estimation based on edge density. First row, input images; second row, correspondent edge density images and estimated horizon lines (red).	64
4.7	Vehicle motion-based feature. First row, input images; second row, correspondent motion mapping images represented in heat maps; on the bottom, heat map scale.	65

4.8	Object classification based on decision tree structure. The root (first node) and the branches (intermediate nodes) correspond to the features; the labels shows the decision rules (based on feature values); and the leaves (final nodes) contain the classification result.	66
5.1	Average error in background modeling - Part 1. In (a), the legend for the graph curves; (b) and (c) show the curves related to, respectively, video sequences 1 and 2. Black curve represents the raw video, and the remaining ones are the compared methods.	71
5.2	Average error in background modeling - Part 2. In (a), the legend for the graph curves; (b) and (c) show the curves related to, respectively, video sequences 1 and 2. Black curve represents the raw video, and the remaining ones are the compared methods.	72
5.3	Background modeling: mean and standard deviation of the error. (a), (b), (c) and (d) refers to, respectively, video sequences 1, 2, 3 and 4.	73
5.4	Background models. In the first row, frames of manually extracted background references for each video; in the middle rows, examples of background models given by the best four compared methods; and in the last row, background models generated by CRON.	74
5.5	Number of superpixels generated for each dataset image. On average, SLIC-1 (blue) produces approximately 1000 segments per image; SLIC-2 (red), 245; and the proposed edge density-based method (green), 443.	76
5.6	Boundary recall (BR). On average, SLIC-1 (blue) provides a BR of approximately 0,7; SLIC-2 (red), 0,53; and the proposed edge density-based method (green), 0,73.	76
5.7	Achievable segmentation accuracy (ASA). On average, all evaluated methods — SLIC-1 (blue), SLIC-2 (red) and the proposed edge density-based method (green) — reached approximately 0,99 of ASA.	77
5.8	Undersegmentation error (UE). On average, SLIC-1 (blue) provides a UE of approximately 0,11; SLIC-2 (red), 0,36; and the proposed edge density-based method (green), 0,30.	78
5.9	Examples of superpixel segmentation: (a) input images; (b), (c) and (d) are segmentation results generated by, respectively, SLIC-1, SLIC-2 and the proposed edge density-based algorithm.	79
5.10	Examples of road detection results generated by the proposed method: (a), (c) and (e) are the input images; (b) quasi-perfect results; (d) results with some errors; and (f) results on very challenging scenes.	82

List of Acronyms

ABL	adaptive background learning.
ANN	artificial neural network.
ASA	achievable segmentation accuracy.
BGS	background subtraction.
BR	boundary recall.
CRON	context-supported road information.
FCI	fuzzy Choquet integral.
GMM	Gaussian mixture model.
GPS	global positioning system.
HD	high definition.
HT	Hough transform.
IMU	inertial measurement unit.
KLT	Kanade-Lucas-Tomasi.
LBP	local binary patterns.
LIDAR	laser interferometry detection and ranging.
LUT	lookup table.
NPA	normalized power accumulation.
PDF	probability density function.
PTZ	pan-tilt-zoom.
RANSAC	random sample consensus.
RBF	radial basis function.
ROI	region of interest.

SFA	slow feature analysis.
SLIC	simple linear iterative clustering.
SPCPE	simultaneous partition and class parameter estimation.
SRG	seeded region growing.
SVM	support vector machine.
UAV	unmanned aerial vehicles.
UE	undersegmentation error.

Glossary

- BGS** Background subtraction consists in a technique by which the background of an image is modeled, so that foreground objects can be extracted by subtracting background from the original image.
- BGO** “Background only” are methods that consider only background information in the road detection process.
- CRON** Our method for background modeling based on contex-supported road information.
- FGO** “Foreground only” are methods that consider only foreground information in the road detection process.
- GMM** Gaussian Mixture Model is a background subtraction technique. It proposes to model the pixel dynamics by means of Gaussian distributions.
- HBF** “Hybrid foreground-background” are methods that consider both background and foreground information in the road detection process.
- HT** Hough transform is a feature extraction technique, capable of to identify objects within a certain class of shapes by a voting procedure. The method carries out a transformation from the image space to a parametric space.

- LBP** Local binary pattern operator is defined as a gray-scale invariant texture measure, derived from a general definition of texture in a local neighborhood.
- LUT** Lookup table is a technique that allows for several operations involving arrays by means of indexing.

List of Symbols

BG	Image background.
ED	Edge density filter.
FG	Image foreground.
G	Gray amount metric.
$O(\cdot)$	Big O notation, used to describe the complexity of an algorithm.
i, j, x, y	Image pixel position.
$\#i$	denotes the i th video.
k, t	Video frame index.
RGB	RGB color space.
r, g, b	Indexes denoting each RGB channel.
tp, tn, fp, fn	True positive, true negative, false positive and false negative.

Introduction

*“When everything seems to go wrong, good things happen,
which would not have happened if everything went right before.”*

Renato Russo

Vehicle flow measurement, incident detection, vehicle guidance, driver assistance, parking violation detection are examples of activities that have attracted more and more interest of the research community [Buch *et al.* 2011]. The motivation is straightforward: to automate, as far as possible, the tedious task of interpreting large amounts of image data. Indeed, this amount of data increases every day, mainly due to the dissemination of video-based systems in urban scenarios as well as regular improvements in camera resolution.

A problem, however, is that most of the researches are concentrated in analyzing the traffic itself, while relevant supporting tasks, such as road detection, still need to be properly addressed. It is not uncommon, for example, to find works that propose automatic solutions for issues involving detection, tracking or identification of vehicles [Feris *et al.* 2011, Sobral *et al.* 2013], but that, contradictorily, require manual annotation of the image region where the traffic analysis will be performed.

Apart from the inconvenience of having to manually configure the system for each different scenario — consider a metropolis, where there are usually hundreds of cameras! —, this is also an inefficient option. Figure 1.1 illustrates how changes in camera position could affect a vehicle counter system based on manual road annotation. In applications like that, where the images are taken from considerable distances, even minimal variations in camera angle of view (ϕ_i in Figure 1.1) are amplified due to perspective effect, resulting in significant errors to the analysis process.

The question is, in traffic surveillance systems, where most of the time the cameras are stationary, do we have to worry about changes in their position? Although in such systems the cameras are in fixed spots (usually at roadside), they are not necessarily unyielding. This is so, since traffic monitoring is an outdoor activity and the cameras are subjected to all sort of natural events (e.g., winds, rains, bird strikes, etc.) that can eventually modify

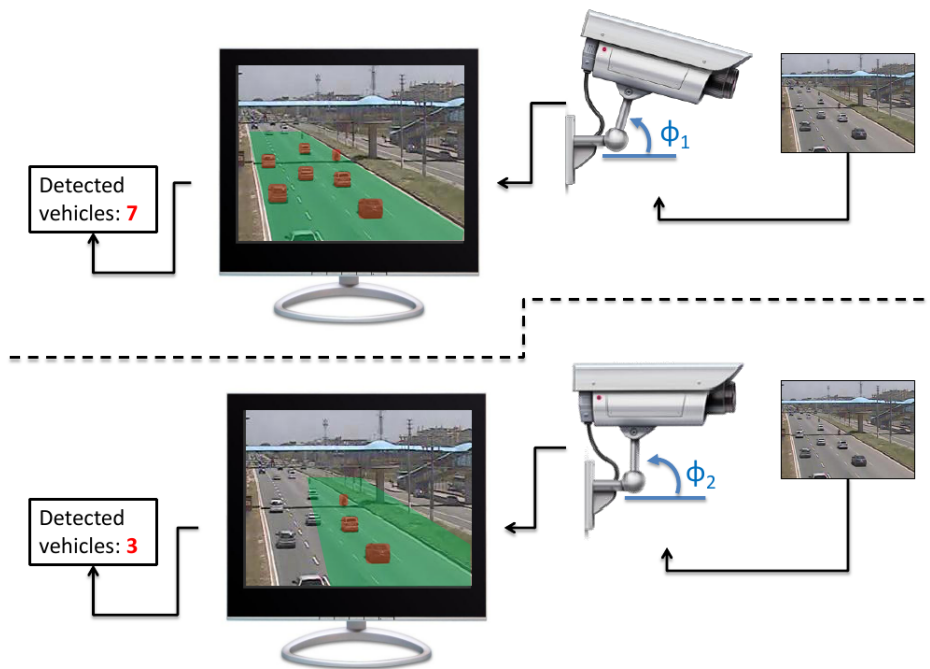


Figure 1.1: The problem of annotating the road manually in traffic analysis systems: minimal changes in the camera position can result in grotesque errors in the analysis.

their original setup. Furthermore, many times these cameras are of pan-tilt-zoom (PTZ) type, which allow changing their angle of view by software or human operators.

It seems that, for a traffic surveillance system intended to be completely autonomous and robust, automatic road detection is of underlying importance. This work addresses this issue, proposing an innovative approach for road detection. The proposed method is constructed upon formulations aimed at solving different problems of computer vision, including image segmentation, background modeling and feature extraction. The system resulting from this effort has been conceived with the computational complexity issue in mind; the goal, thus, is to provide a solution that can be used in conjunction with a traffic analysis system, supporting it.

In the next two sections, we summarize the motivations that have driven this work, as well as the main goals and contributions that we expect to reach. Following, we provide a brief description of which can be found in the remaining chapters.

1.1 Motivation

This work is part of a bigger project called GET-IN¹, which aims at providing a system to infer the actual conditions of the traffic, based on surveillance images. Therefore its initial motivation is to support the traffic analysis module of this system by identifying the region in the image that corresponds to the road. As will be seen in future chapters, automatic road detection is not a novelty. Especially in research areas turned to autonomous vehicles, there are many works intended to provide road detection. Even in traffic surveillance area, where works on road detection are more sparse, some approaches can be found in the literature.

The question is: Why are we motivated to create a new approach for road detection? And the answer: due to our dataset. As we shall see, most of the existing methods for road detection based on surveillance videos deal only with easy datasets. As easy datasets we mean those ones comprised of well-structured quasi-straight roads, with visible markings and curbs, continuous flow of vehicles and without interference of any other moving object. Such characteristics are typical of freeways; indeed, they are absolutely predominant in datasets for road detection. Our project, on the other hand, is intended to deal with a much more challenging dataset. Many of our traffic videos were taken from poorly-structured roads, with worn out markings, some of them very difficult to distinguish from other objects in the scene (even by human eyes). Moreover, in our dataset there are roads of different shapes, ranging from closed curves to intersections. These latter ones give rise to an even bigger problem: road occlusions due to moving objects that stop in the scene. In this case, occluding objects range from vehicles waiting at a red light to pedestrians crossing a crosswalk.

To summarize, overcoming those and other difficulties afforded by our dataset constitutes the main motivation for this work. In this sense, our option for developing a new road detection approach is a consequence of the challenges imposed by the circumstances.

1.2 Goals

Methods intended to perform road detection are a tool, which the final goal is supporting traffic analysis systems. Generally, such systems are comprised of complex processes, demanding great computational effort. Thus, in order to avoid system overhead, it is of paramount importance that the road detection method utilizes, as far as possible, the

¹www.ivisionlab.eng.ufba.br/projects

least amount of processing time. Moreover in order to fully exploit the dissemination of video-based systems in urban areas, rather than their common use in freeway monitoring, traffic analysis systems must be able of dealing with more complex urban scenarios. An important step in this direction is to construct road detection methods capable of detecting roads in such complex scenarios. This work engages these issues, such that it points to the following goals:

1. *To develop a new road detection approach, which allows for a good trade-off between efficiency in detection and computational complexity;*
2. *To provide road detection in any kind of urban scenario, including poorly-structured, highly occluded and any-shaped roads.*

1.3 Key contributions

Rather than a single procedure to be performed over images, object segmentation generally involves a set of steps to reach the final solution. Noise filtering, object shape detection, feature extraction and classification are examples of common tasks related to object segmentation. In our work, each one of these problems had to be addressed, in most cases by means of completely innovative solutions. Among them, we remark as key contributions provided by this work: i) a new dataset of traffic surveillance videos [Santos *et al.* 2013]; ii) a new method for image segmentation based on superpixels; iii) a new approach for background modeling in traffic scenes; iv) some strategies to extract features for road detection [Santos *et al.* 2013]; and v) an innovative road detection method. Besides, this work contributes with the paper *Learning to segment roads for traffic analysis in urban images*, published in the Intelligent Vehicles Symposium (IV), IEEE, 2013.

1.4 Chapter map

The remaining of this work is organized as follows:

- **Chapter 2** presents the background information about the main topics related to our work. Included in this chapter are previous approaches, related concepts and general considerations.
- **Chapter 3** provides an overview of the proposed road detection method, starting from its conception and culminating with the final design of the proposed solution.

-
- **Chapter 4** describes each part of the conceived method, presenting details of implementation.
 - **Chapter 5** evaluates our algorithms by different aspects, including comparisons against other superpixel algorithm, several background modeling methods, as well as the road detection performance over our challenging dataset.
 - **Chapter 6** concludes our work, with discussions and future work.

Background

Contents

2.1	The need for road detection	8
2.1.1	Calibrating traffic radars	8
2.1.2	Road monitoring by satellites and aircrafts	10
2.1.3	Onboard road detection	15
2.2	Road detection in traffic surveillance applications	19
2.2.1	The flowcharts of detection	20
2.2.2	Background subtraction	21
2.2.3	BGO methods	24
2.2.4	FGO methods	28
2.2.5	HBF methods	31
2.2.6	Summary of related works	35
2.3	Advanced BGS methods	36
2.3.1	Adaptive Background Learning	36
2.3.2	Gaussian-based methods	36
2.3.3	Fuzzy-based methods	38
2.3.4	Multi-layer BGS based on color and texture	39
2.3.5	Neural network with self-organization approach	40
2.4	Superpixel segmentation	41
2.4.1	Simple linear iterative clustering	42
2.4.2	Graph-based image segmentation	43
2.5	Relation to our work	43

2.1 The need for road detection

There are several methods intended to detect roads, some of them designed to exploit directly road features, such as color, texture or shape; others search for roads from indirect information, such as vanishing points or vehicle trajectories. Besides these, some methods try to identify road lanes using echo information provided by traffic radars.

Defining a specific approach depends on the application and the type of data that will be processed. Objectively these issues point to the device that is used as sensor, as well as the context of use. Thereby existing methods can be broadly categorized as if they use radar or camera as sensor, whereas camera-based methods can be further differentiated by context: if they take aerial images (from satellites or aircrafts), on-board images (directly from vehicles), or surveillance images (from roadside spots).

In this section, each of these categories are presented as background for the proposed work. The goal is to discuss the needs that have motivated the development of the different methods, and the rationale behind them. In this sense, an overview of some representative works exemplifying each category is given. Since the method proposed in the present work is better related to traffic surveillance applications, a specific section is dedicated to a more detailed discussion concerning such category.

2.1.1 Calibrating traffic radars

Over the past decades several types of devices have been applied in traffic monitoring solutions. It is the case of inductive loops, which are embedded in the roadways, and non-embedded sensors like acoustic, optical and radar ones. Among them, traffic radars have a variety of advantages, such as all-weather operation, no blind area, and low costs of installation and maintenance. Because of that, they serve as a satisfactory replacement of the buried inductive loop detectors, acoustic and optical surveillance equipments [Zhang *et al.* 2008].

Radars have been widely used for speed enforcement on public roadways [Westphal & Kessler 1988, Jendzurski & Paulter 2009]. In this context, accurate calibration of the device is critical, otherwise the radar sensor may stop working or provide inaccurate information leading to improper or even unsafe traffic decision-making. According to Zhang *et al.* (2008), traffic lane boundary identification is the most important task for traffic radar calibration. This is a challenging task since traffic lane boundaries do not necessarily maintain the same position all the time — they can vary with time due to changes in weather and traffic conditions.

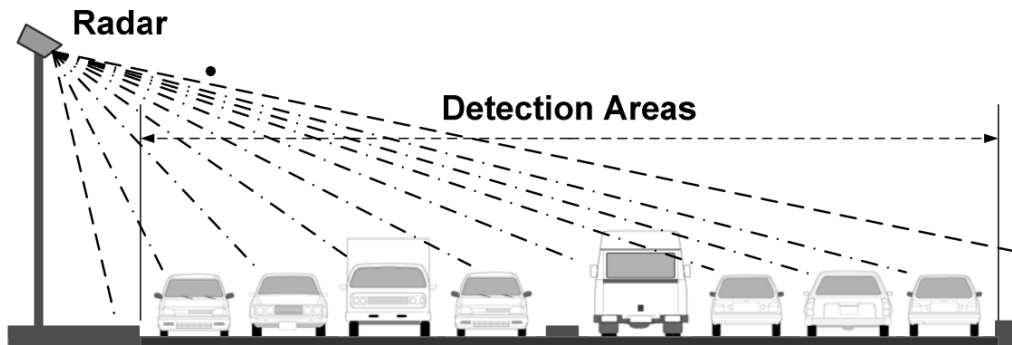


Figure 2.1: Traffic radar mounted in elevated side-looking configuration, where intermediate frequencies of emitted signals are proportional to the distances from reflective objects to the radar. Variations in echo power are used to detect road lanes. Image taken from [Zhang *et al.* 2008].

In order to allow continuously recalibration of traffic radars, that way ensuring the correct operation of these devices, Zhang *et al.* (2008) propose a method to automatically detect road lanes. To this end, the traffic flow detection radar should be mounted in an elevated side-looking configuration on the roadside. The transmitting beam of the radar should cover all traffic lanes and the horizontal beam width should be greater than the length of the vehicles. Figure 2.1 depicts this arrangement. The work explores the fact that intermediate frequencies of emitted signals are proportional to the distances from reflective objects to the radar. Because the distances from the radar to different traffic lanes are not the same, information of each lane is distributed in separated frequency ranges. Thus digital signal processing techniques can be used to analyze the mixed signals. After analog-to-digital conversion, a periodogram is then calculated in order to acquire the received echo power. The authors have verified that the reflected signals of the metallic vehicles with a higher reflection attenuation coefficient are stronger, and the echo power of the road surface — which, in this case, is always made of asphalt or cement — is weaker. The innovation of this method resides in an algorithm based on normalized power accumulation (NPA), which is responsible for analyzing the received echo power signal in order to identify the lane boundaries. The authors report satisfactory practical results in tests performed at roadways in Shanghai, China, outperforming previous approaches based on theoretical calculations and probability density function (PDF).

More recently, [Felguera-Martin *et al.* 2012] propose two different radar-based schemes in order to solve the problem of association of each target with its respective echo response, specially when there are two or more targets in the radar beam. The schemes include down-the-road and across-the-road approaches for radar positioning. According to the



Figure 2.2: Interferometric radar at field experiments. The device is used to detect road lanes in order to support a traffic flow analysis process. Image taken from [Felguera-Martin *et al.* 2012].

authors, the proposed schemes can measure range, speed, and azimuth of several targets in the beam at the same time, differentiating and identifying all of them. For this, interferometry together with a set of geometrical calculations are used to detect the road lanes and, from them, individual vehicles in the traffic flow. The work reports good results using a interferometric radar as the one showed in Figure 2.2. The outcomes were obtained from field trials, where a vehicle was driven through a controlled traffic parking at a speed of approximately 20 km/h.

2.1.2 Road monitoring by satellites and aircrafts

In recent years, the advancement of analytical techniques for processing image data, together with increased computing power, has enabled the deployment of video-based systems in traffic applications. In this context, the usage of cameras provides several advantages: relatively low installation and maintenance costs without requiring traffic disruption; sufficient compactness that allows embedded applications; and the provision of more representative output data (images) than the responses obtained from other sensors commonly used in traffic, like radars or inductive loops.

Nowadays the continuous progress of image acquisition technology has enabled the generation of high definition (HD) images. Such capability allows the images to be taken at greater distances without causing a troublesome blur of the objects. That way, long distance unmanned vehicles, like aircrafts and satellites, find great applicability in earth

observation purposes. This is especially true for tasks involving the extraction of road information, which scope is increasingly extended to complex urban scenarios thanks to the availability of highly detailed images [Gaetano *et al.* 2011].

For instance, [Rathinam *et al.* 2008] propose a control technology for unmanned aerial vehicles (UAV) to enable the monitoring of locally linear structures such as highways, roads, and canals. In this work, a structure is considered as being locally linear if it is approximately linear in each frame caught by the camera on board the UAV. Computer vision is then used to find out and maintain the correct position of the aircraft with respect to the structure being monitored. By doing so, the work tries to overcome a common problem presented by systems based on global positioning system (GPS), in which a sequence of given GPS coordinates (waypoints) defines the trajectory to be followed by the UAV. In such cases, if either precise waypoints along the structure are not known or if they are sparsely spaced, the UAV may be driven to unintended areas, resulting in loss of structure sections in the inspection video.

Alternatively, by adopting an imaging sensor in a closed-loop control scheme, the system of Rathinam *et al.* (2008) controls its location error directly with respect to the structure being inspected. The vision-based detection component consists of a semi-supervised learning algorithm designed to detect the structures. The detection process relies on three stages: At the learning phase — implemented offline — the algorithm should be fed with a sample image, when a cross-sectional profile of the target structure

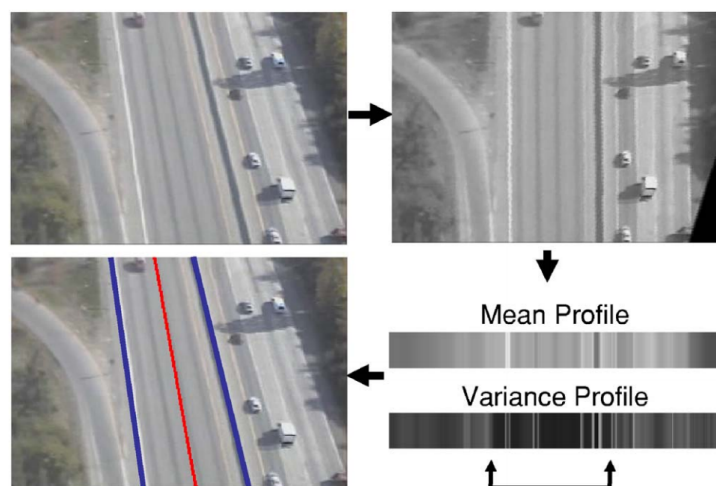


Figure 2.3: Road detection process proposed by [Rathinam *et al.* 2008]. Following the arrows: an input road image is caught; the input image is rectified; a road profile is obtained by computing image statistics (means and variances); road boundaries are estimated from the obtained profile. Image taken from *ibid.*

is produced. The second step matches this cross-sectional profile with horizontal samples of the target image to find the boundaries and the position of the structure in the image. Finally, a curve fitting algorithm uses the cross-sectional profile to fit a cubic-spline curve that models the target structure. Once the equation of the curve to be followed is obtained, the information is then sent to the tracking component responsible to guide the UAV. According to the authors, the second and the third steps can run in real time for each target image. In Figure 2.3 an example of the described road detection process can be seen.

Also using aerial images taken from aircrafts, but with the proposal of calculating traffic parameters, [Li *et al.* 2009] describe a method to detect and track vehicles on highways. The method performs road extraction in order to determine a region of interest (ROI) in the image. The goal is to rule out image portions (estimated to be approximately 60% of the pixels) that are not relevant to the traffic analysis. Assuming that the color of highways is closer to gray color, as well as most of the lane markings and road signs are painted white, the authors have constructed their road detection method based on a set of rules:

- a. RGB components of the road image should be approximately equivalent;
- b. There should be white stripes in the road area;
- c. Road should be a connected region;
- d. Road surface should appear most often in the image frame;
- e. The color of every pixel in the road should be similar;
- f. Road width is within a certain range;
- g. Road position should not change significantly in two successive frames.

In order to implement those rules the method of [Li *et al.* 2009] attempts to eliminate non-gray areas in the image by doing

$$G(x, y) = \begin{cases} F(x, y), & |r - b| < \tau_1 \quad \text{and} \quad |g - b| < \tau_1 \quad \text{and} \quad |r - g| < \tau_1 \\ 0, & \text{otherwise} \end{cases} \quad (2.1)$$

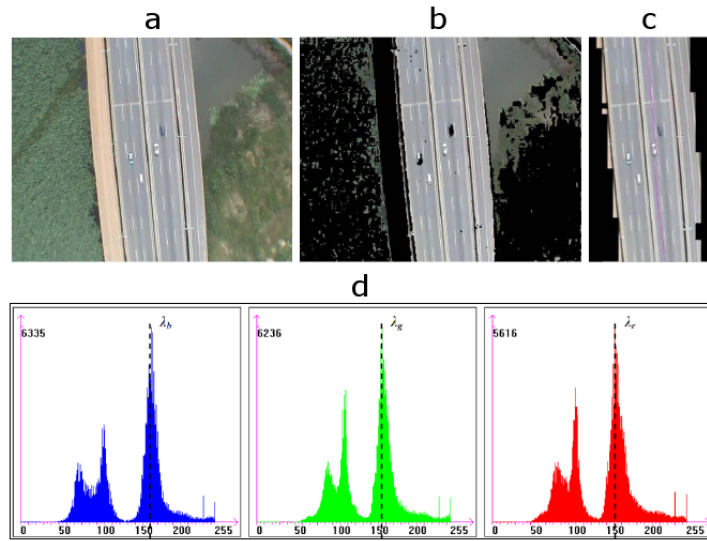


Figure 2.4: Road detection method proposed by [Li *et al.* 2009]. (a) Input image, (b) image after exclusion of non-gray areas, (c) final detection result and (d) RGB histograms used to extract road color information. Image adapted from *ibid.*

where F is an input frame with pixel coordinates (x, y) , G is the obtained gray similar image, r , g and b represent each RGB channel, and τ_1 is a threshold defined by the authors as 20 at the experiments.

Next, a histogram is calculated for each RGB component of G , and the peak value of each histogram is registered. The assumption is that road color information will be accumulated in these peaks. After, the image is divided into blocks of size 15×15 pixels, where the inner pixels are compared with the peak values, as follows

$$S(x, y) = \begin{cases} 1, & |r - \lambda_r| < \tau_2 \quad \text{and} \quad |g - \lambda_g| < \tau_2 \quad \text{and} \quad |b - \lambda_b| < \tau_2 \\ 0, & \text{otherwise} \end{cases} \quad (2.2)$$

where S is a block mask in which 1 denotes a pixel that is similar to the extracted road color, $\lambda_r, \lambda_g, \lambda_b$ are the peak values and τ_2 is the similarity threshold.

Finally, the method of [Li *et al.* 2009] selects as being road those blocks where at least 60% of the pixels are similar to the histogram peak values. Figure 2.4 depicts this detection process.

Another example of method intended to perform road extraction from aerial images is found in [Gaetano *et al.* 2011]. Dealing with HD satellite images, in that work Gaetano *et al.* (2011) use morphological analysis for road extraction in complex urban scenarios.

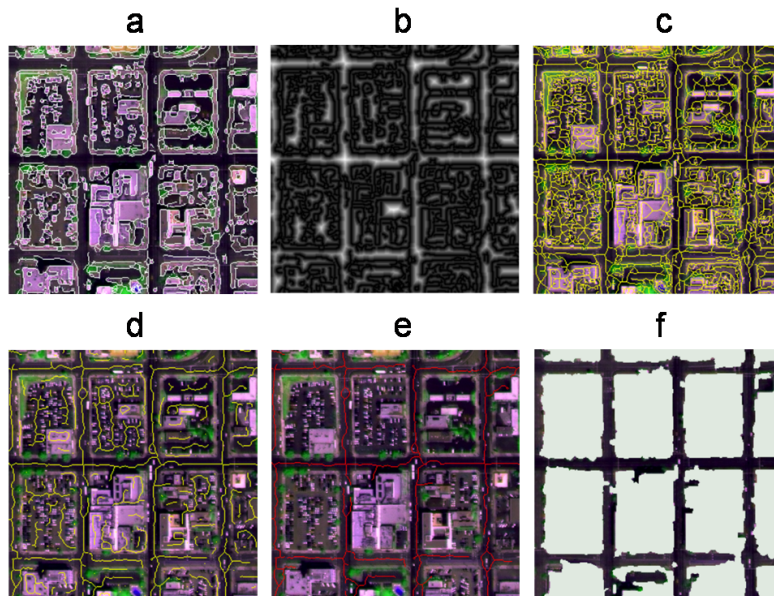


Figure 2.5: Road detection process found in [Gaetano *et al.* 2011]. (a) Input image with superimposed contours, (b) result of distance function computed from the contours, (c) result of the skeletonization, (d) skeleton after pruning, (e) road skeleton selection and (f) final detection result. Image adapted from *ibid.*

The strategy is to generate skeletons that work as descriptors for road objects. The skeletonization procedure picks up the linear structure of road segments, allowing the extraction of road objects by first detecting their skeletons and then associating each of them with a region in the image.

The process starts by means of an edge detector, which provides a “global” skeletonization of the edge map. This will be the basis for the morphological analysis of the image. Next, road sub-skeletons are extracted under morphological criteria, mainly based on the idea that the points belonging to the road structures stand close to their skeletons. By doing so, entire road regions can be segmented by means of morphological reconstruction.

According to Gaetano *et al.* (2011), in real cases road edges present many discontinuities; however, their effects on the skeletonization process are minimal when dealing with objects like roads. In such cases, they state, the branches caused by edge discontinuities are usually small appendices connected to the skeleton backbone. Thus, a technique based on the detection of *crest lines* over a distance function can be used to refine the skeletons. The whole process is depicted in Figure 2.5, and the steps can be summarized as follows:

- a. Edges are extracted from the input image as starting points for morphological analysis;
- b. Contours are used to compute a distance function of the edge map;

- c. From the edge map, the skeleton of the source image is extracted;
- d. Skeleton reduction based on morphological criteria is performed, resulting in the most significant skeletons;
- e. The connected segments are analysed under shape criteria in order to extract road skeletons;
- f. Watershed-based reconstruction is applied on the skeleton segments resulting in a road map.

2.1.3 Onboard road detection

In the last decades, increasingly investments have been directed to researches aimed to develop intelligent vehicles. Driver assistance systems capable of alerting drivers in abnormal situations or even assuming partial or completely the control of the car are gradually being incorporated into vehicles. According to [Bar Hillel *et al.* 2012], the development of such systems depends on the overcoming of two main perceptual problems: road and lane identification, and obstacle detection.

Different solutions have been tried to address these problems, including several sensing modalities, such as monocular or stereo vision-based systems, laser interferometry detection and ranging (LIDAR) and vehicle dynamics information obtained from car odometry or inertial measurement unit (IMU). Bar Hillel *et al.* (2012) consider that computer vision is the most prominent research area due to the fact that road and lane markings are made for human vision, while LIDAR, global positioning provided by GPS and digital maps are important complements. In [He *et al.* 2004], the authors address the problem of identifying the road from an onboard camera in real-time. They provide a vision-based detection method, which combines edge and color features. The goal of that work is to cope with the problem of detecting roads in both urban or rural environments. In rural, the task is more complicated since there may be no road markings or any other structured traffic information.

The method extracts each feature by means of two different modules. The first one analyzes an edge image of the scene in order to obtain the candidates for the left and right road borders. By assuming that the color components of road surfaces obey a Gaussian distribution, the found borders are then used to delimit the area where the mean and variance of this distribution will be subsequently computed. Besides, the borders are further used to estimate the road curvature. This is made by means of a projective trans-

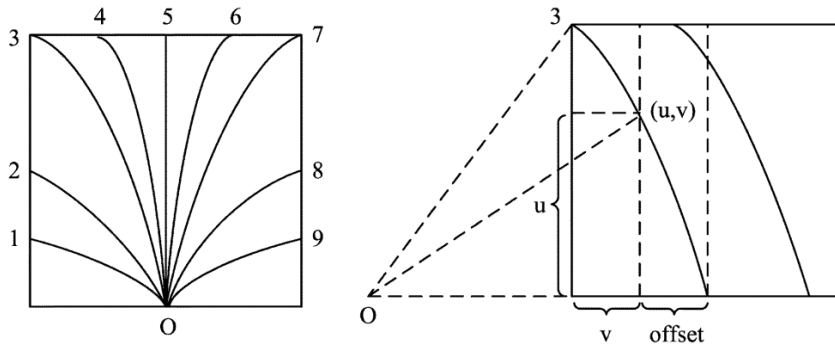


Figure 2.6: Predefined patterns of curves used to estimate the road curvature. On the left, the nine predefined curves; on the right, the parameters of the third curve. Image taken from [He *et al.* 2004].

formation that leads the edge image from the road space (3D) to the image space (2D). Once transformed, the curvature is estimated through a voting scheme that compares the edges with nine different patterns of curves, as showed in Figure 2.6.

After road border extraction, the second module uses the delimited area to compute

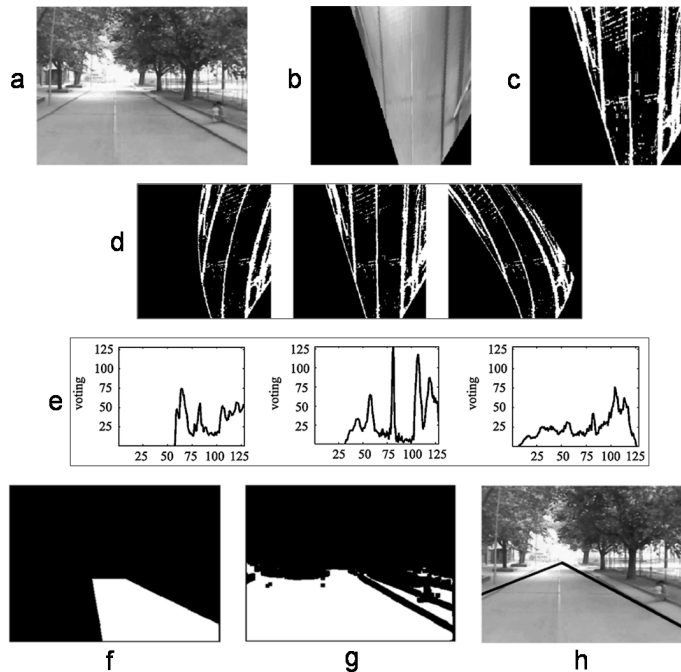


Figure 2.7: Road detection process proposed in [He *et al.* 2004]. (a) Input image, (b) result of projective transformation, (c) extracted edges, (d) edges after shifting by three different predefined curvatures, (e) voting results for each curvature, (f) area selected for computing the road color, (g) road mask and (h) final detection result. Image adapted from *ibid.*

a color reference of the road surface. The algorithm then searches in the remainder of the image, assigning as being road those areas where the color matches with the computed color reference. The final detection result is given by computing the boundaries of these selected areas and then reconverting them to the road space. In Figure 2.7 each step of this method is depicted.

Another example of system designed to segment road regions from on board images is presented in [Alvarez *et al.* 2008]. In this work, Alvarez *et al.* (2008) try to deal with the problems arising from lighting variations and shadows. The road images are taken from a single color camera placed on the windshield of a car. The rationale of the proposal resides in the combination of two approaches. Firstly, an illuminant-invariant feature space is computed to attenuate shadow effects. This space consists of a gray-scale image that is obtained by projecting log-log pixel values onto the direction orthogonal to lighting change, within and outside the umbra. Obtaining the orthogonal direction depends on camera calibration. This is reached by collecting real images from a monocular color sensor, which are used to linearize the sensor response and lens distortion. The calibration is result of an entropy minimization technique based on the information in each image. Figure 2.8 shows some examples of illuminant invariant images generated by this method.

In a second step, a classifier separates the pixels into two classes: road and non-road. For that, for each new frame a road model is constructed by means of a seeded region growing (SRG) technique. Making the assumption that the bottom of the image is road (the authors have verified that the lowest row corresponds to a distance of about 4 meters away from the vehicle), SRG is then fed with a set of seeds collected on this part of



Figure 2.8: Examples of illuminant invariant images. First row, input images; second row, generated illuminant invariant models. Images taken from [Alvarez *et al.* 2008].



Figure 2.9: Examples of road segmentation produced by [Alvarez *et al.* 2008]. First row, input images; second row, segmented roads. Images taken from *ibid.*

the image. A remark is that this segmentation process does not use shape information, that way avoiding limitations when addressing unstructured roads. In Figure 2.9 some examples of segmentation obtained from this method are showed.

An on-board monocular camera is also the sensor used in the system proposed by [Kuhnl *et al.* 2011]. Their offline trained system deals with the problem of detecting roads by means of two complementary components. The first one consists in an initial patch-based segmentation, where texture and appearance features are used to describe road and non-road regions on the patch level. At this point, slow feature analysis (SFA) is performed over the class specific appearance descriptors, producing a low order feature set. The goal is to improve the patch-based road classification process, where the descriptors, combined with color and texture features, are used to train a boosting-based classifier. By doing this, the classifier is made capable of roughly segment road portions from the image, specially those ones closer to the center of the roads. On the other hand, regions closer to the borders are hard to distinguish on the patch level, resulting in a high misclassification rate. In order to overcome this problem, a complementary post-processing step is implemented. The strategy consists in using a specialized classifier that was trained specifically for image boundary regions. This new classifier differs from the former by an increased number of tree splits and boosting iterations. The feature vector acquires a higher dimensionality by the addition of new texture, color and SFA-produced features. The final result is then obtained by fusing the segmentations of both system steps. An overview of this method is given in Figure 2.10.

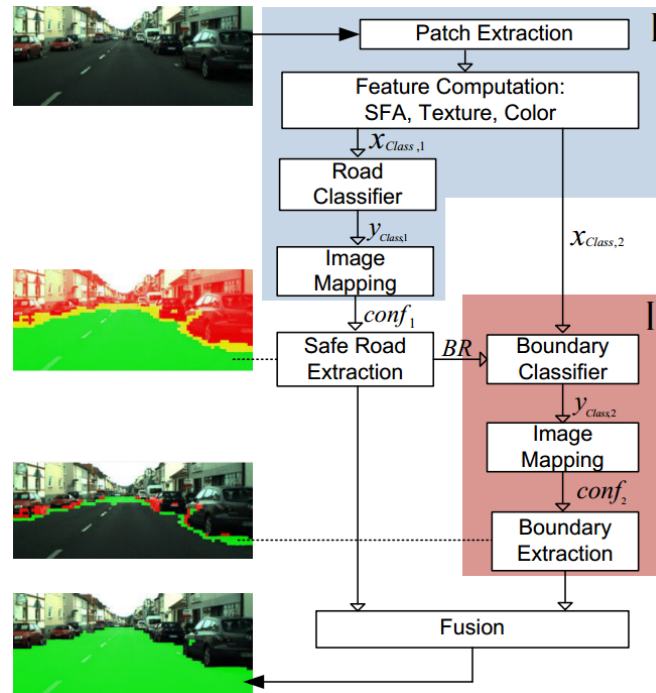


Figure 2.10: Flowchart of the method proposed by [Kuhnl *et al.* 2011]. Image taken from *ibid.*

2.2 Road detection in traffic surveillance applications

Camera-based traffic surveillance systems are already a reality in many cities around the globe. Their purpose, among others, is to provide real-time statistical data on traffic conditions [Feris *et al.* 2011, Sobral *et al.* 2013, Unzueta *et al.* 2012], as well as to alert potentially abnormal events, such as traffic infractions [Vijverberg *et al.* 2007], accident occurrence [Hwang *et al.* 2010], parking violations [Lee *et al.* 2007], dangerous driving [Zhou *et al.* 2011], just to cite a few. Consequently, there is an increasing need of development of automation solutions in order to substitute humans in the tedious task of interpreting large amounts of image data. In this sense, a number of challenges concerning computer vision and image processing still needs to be addressed. An example is automatic road detection. Correctly identifying roads in a scene is a critical task for many traffic analysis systems; however, there are several difficulties in trying to do this automatically. Firstly, despite in such applications the cameras are commonly positioned in fixed points on the side or above the roads (e.g., in lampposts or viaducts), they may not be absolutely stationary. In many cases they provide pan-tilt-zoom (PTZ) technology, which allow changing their angle of view by software or human operators. Besides, since traffic monitoring is an outdoor activity, the cameras are subject to all sort of distur-

bances due to environment events, as for instance strong winds, rains and collisions with birds. It is noteworthy that, owing to perspective effects, even small changes in camera view angle may affect the system behavior. This is also true for natural lighting variation, as it occurs due to weather changes. Therefore, the ability for automatically detecting the road, without the need for camera calibration, is a meaningful advantage to a system intended to analyze the traffic. Moreover, it facilitates the installation and maintenance of the equipment due to the system capability for self-configuration.

An imperative matter in the development of such an automatic system is adaptivity. According to [Kastrinaki *et al.* 2003], the ability of reacting to changes in the scene, while carrying out a variety of goals, is a key point in designing methods for traffic data collection. With that in mind, existing methods try to fix the adaptivity issue by means of different approaches: BGS, lane markings detection, color and texture features extraction, vanishing point analysis, moving objects mapping, are examples of strategies recurrently found in the literature. In this section, we will discuss these and other computer vision techniques, addressing their application in the context of road detection. In this sense, we provide a brief survey of the existing works on road detection, in which a classification scheme for the methods is suggested, according to their flowcharts of detection.

2.2.1 The flowcharts of detection

Concerning traffic surveillance systems, road detection methods can be generally divided into three types, depending on the source from which the data used in the detection is extracted. This aspect is better noticeable by looking at the flowcharts in Figure 2.11, where each system type is depicted. As can be seen, the starting point is common to all of the proposed types; it consists in to perform BGS for each new input frame. Henceforth, however, the systems differ to each other by the output of the BGS that is regarded. This way, “background only” (BGO) methods only consider data extracted from the background, discarding all foreground information. In “foreground only” (FGO), conversely, only foreground objects are taken into account while the background is disregarded. Finally, “hybrid background-foreground” (HBF) systems are the hybrid of the previous ones, i.e., the detection process in this type of system is based on the combination of both background and foreground features.

Next, we will discuss the ways adopted by each related method to accomplish the tasks involved in the flowcharts presented in Figure 2.11.

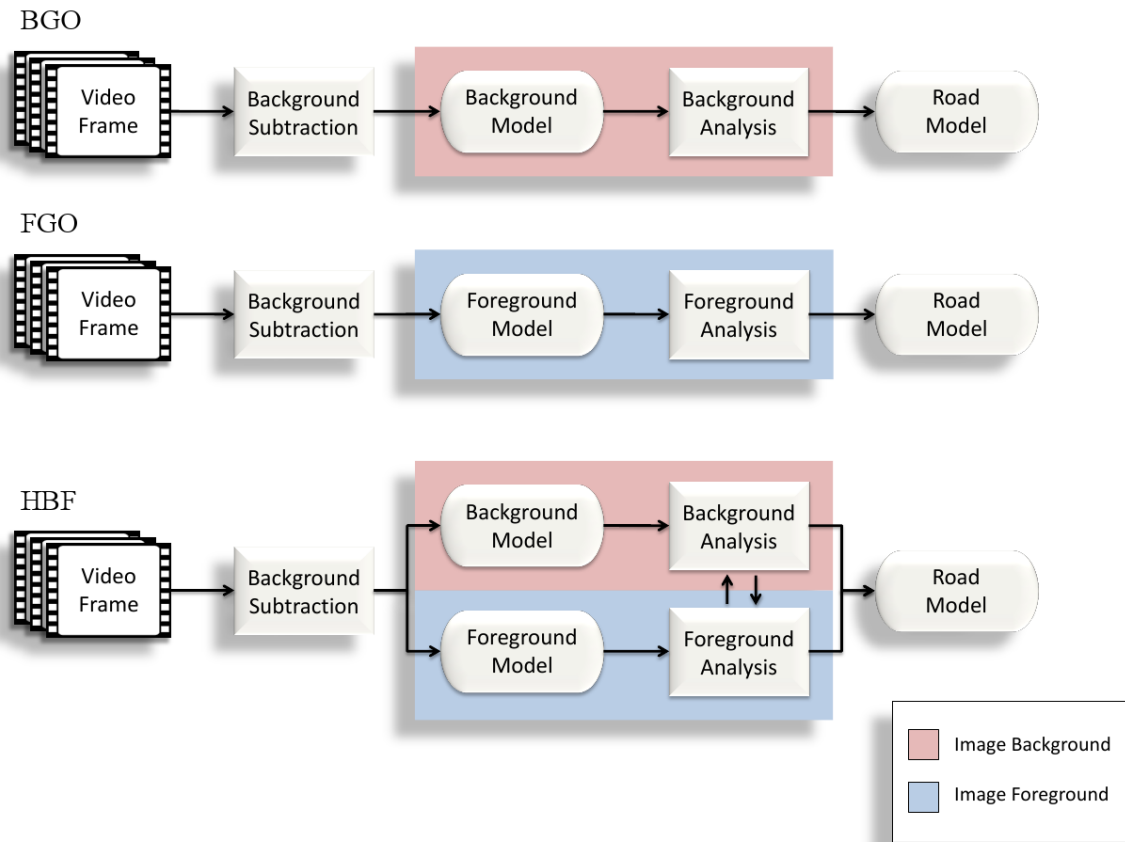


Figure 2.11: Road detection flowcharts. BGO: detection based only on background information. FGO: detection based only on foreground information. HBF: hybrid detection, based on both background and foreground information.

2.2.2 Background subtraction

Broadly speaking, the objects that appear in a video sequence can be divided into foreground — which consists in the objects that can acquire motion at some point — and background — the objects that are always stationary in the scene. In the case of traffic surveillance videos, common foreground objects are vehicles and pedestrians going through the scene, whereas the background is basically the road and its surroundings, such as curbs, sidewalks, trees, buildings, etc. This way, to perform background subtraction before the detection itself seems to be a timely first step for a road detection system. Once accomplished, the subtraction allows to separate foreground objects from the background. Hence, the task of identifying the road becomes easier, since some occlusions due to moving objects can be filtered. Furthermore, motion information can be exploited in order to contribute to the detection process.

Among the methods related to the present section, [Luo *et al.* 2011] perform BGS by

the simplest manner: frame differencing. Such technique consists in calculating the inter-frame difference of pixel intensities. If a pixel intensity differs above a given threshold from the current frame to the next, then it is taken as foreground; otherwise, the pixel is assumed as background. This strategy, however, only works in cases where all foreground pixels are moving relatively fast and all background pixels are static.

[Shin *et al.* 2006] model the background by taking the pixels with the highest intensity after accumulating data of each pixel from the first frame to the current frame. [Gao *et al.* 2011], in turn, obtain a background model by using the accumulated data to calculate the mean of each pixel. Both techniques are very simplistic and produce poor results in real images.

The method of [Pan *et al.* 2010] differs from the aforementioned by dealing only with edge information. Instead of accumulating pixel intensities, it uses Sobel edge detector to obtain an edge image for each new input frame. Because edge images only contain binary data, they can be accumulated by just incrementing the pixels. This way, since background objects appear more frequently in the scene than foreground objects, they usually originate higher accumulation values. A threshold is then used to separate these two classes of objects. The drawback of this method is the loss of valuable information due to the conversion of the input images to edges.

[Chung *et al.* 2004] perform BGS by means of an own histogram-based technique. It is carried out by progressively recording the changes in the intensities of each pixel along the frames. For that, a histogram table is built by selecting the intensity values that have maximum counting frequencies in the histograms. In that table, each pixel is mapped in a row containing its most frequent intensity and a frequency counter. Then, background models can be constructed by directly taking the pixel values registered in the histogram table.

The road detection method of [Melo *et al.* 2006] relies on the background modeling algorithm proposed by [Gutchess *et al.* 2001]. From this, background models are generated based on adaptive smoothness. The process is initially offline, and consists in locating, for each pixel in a video sequence, periods of stable intensity, i.e., time intervals where the pixel intensity does not change above a given threshold. These intervals represent a set of candidate hypotheses, one of which will be selected as the background model. This selection will be done at the online stage of the algorithm, and the criterion is the likelihood of each candidate model. To evaluate the likelihood, the method performs a set of calculations based on optical flow information extracted from the neighborhood around the pixel. The foreground, in turn, is extracted by taking the difference from the input

frame to the produced background model.

[Mazaheri & Mozaffari 2011] also estimate a background model, but from a different manner. They segment the input frame in many small non-overlapped blocks, which are saved as a codebook. Next, they calculate the histogram of the input frame, along with the mean intensity for each image block. The result is then submitted to an adaptive filter, based on recursive least squares, which outputs a background model after a number of iterations. It is noteworthy that this method does not address foreground extraction, since it is not used in the road detection process.

This is not the case of the system proposed by [Helala *et al.* 2012], for example. Here, foreground information supports the road detection process. To access it, the BGS technique proposed by [Kim *et al.* 2004] is used in order to extract moving objects from the background. The method works by quantizing values of background samples into codebooks, this way constructing a background model. Samples at each pixel are clustered into the set of codewords. The background is then encoded on a pixel-wise basis, and the foreground is estimated by computing the distance from the sample to the nearest cluster mean.

The BGS technique that closes this subsection is called Gaussian mixture model (GMM). This is a well known method in the computer vision area, for which several implementations and improvements exist. In respect to the present section, three works on road detection make use of GMM as support. They are [Huang 2010], [Li & Zhong 2009] and [Liu & Wang 2010]. The first two use the original GMM version proposed by [Stauffer & Grimson 1999], while the third refers to a version presented in a PhD thesis [Borenstein 1987]. Here we will focus on the first option, i.e., the GMM of Stauffer and Grimson (1999).

The rationale behind this method consists in modeling the pixel behavior by means of Gaussian distributions. The reason for using more than one distribution — a mixture of them — is to fully catch the pixel dynamic, which includes changes due to noise, lighting variations and others. That way, the recent history of each pixel, $\{X_1, \dots, X_t\}$, can be modeled by a mixture of K Gaussian distributions. The probability of current pixel value X_t is then

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (2.3)$$

where K is the number of distributions, and $\omega_{i,t}$, $\mu_{i,t}$ and $\Sigma_{i,t}$ are, respectively, the weight, the mean and the covariance matrix of the i^{th} Gaussian distribution (η) in the mixture

at time t . In order to produce a background model, the parameters $\omega_{i,t}$, $\mu_{i,t}$ and $\Sigma_{i,t}$ need to be determined. For that, Stauffer and Grimson (1999) suggest that the first B distributions are the best choice to model the background, and are given by

$$B = \operatorname{argmin}_b \left(\sum_{K=1}^b \omega_k > T \right) \quad (2.4)$$

where T is a constant to determine the minimum portion of the background in the scene. Again, the strategy to detect foreground objects consists in taking the difference from the input frame to the background model (here, obtained from Equation 2.4).

We remark that the method proposed by [Lai & Yung 2000] is the only one among all that we found which does not address the BGS issue. Even so, the authors suggest that some procedure should be previously performed in order to filter moving objects from the frames submitted to their method.

2.2.3 BGO methods

BGO methods are the most straightforward attempts to solve the road detection problem. Because all the road information is contained in the background, such systems focus the detection process on directly extracting road features from this part of the image. In this context, road borders and lane markings are the most commonly used features, and their extraction is usually based on the Hough transform (HT). HT is a technique designed to isolate features of a particular shape within a binary image. It works by determining specific values of parameters which characterize the wanted shape [Illingworth & Kittler 1988]. The technique is most commonly used for the detection of regular curves such as lines, circles and ellipses, since they are easier to deal with in a parametric form. This way, the difficult problem of detecting the shapes in the image space is converted into a more easily solved local peak detection problem in a parameter space. For example, considering the straight line detection problem, a set of collinear points (x, y) in the image space (Figure 2.12.a) can be defined by the usual relation

$$y = m.x + c \quad (2.5)$$

where the parameters m and c are the line slope and y -intercept, respectively. However, by looking at the parameter space (Figure 2.12.b) in Equation 2.5, the line is represented by a single point, corresponding to a unique set of parameters (m, c) . The idea of HT is to implement an accumulator (Figure 2.12.c), where the accumulation value represents

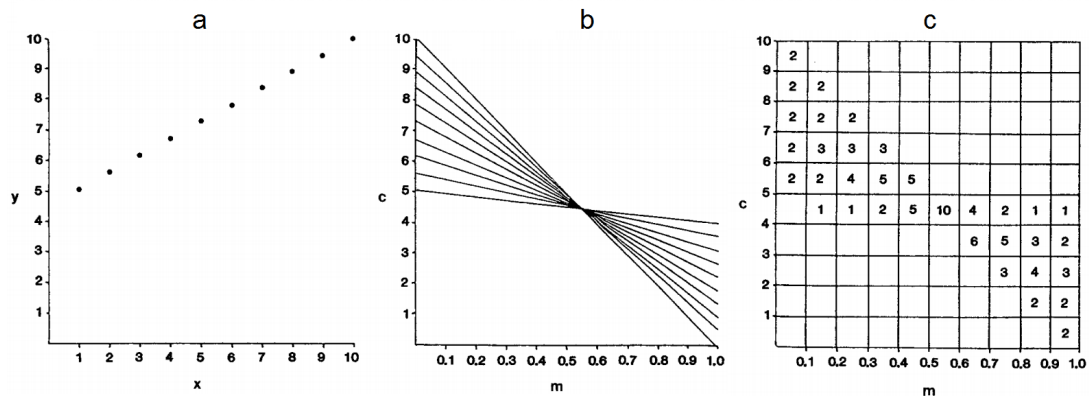


Figure 2.12: The Hough transform. (a) Image space, (b) parameter space and (c) accumulator corresponding to (b). Images taken from [Illingworth & Kittler 1988].

the many-to-one relation from the points in the image space to the corresponding points in the parameter space. By doing so, the problem of detecting straight lines is reduced to the problem of finding the highest values in the accumulator.

An example of BGO method based on HT is [Huang 2010]. The vehicle counting system presented in this work is carried out with the help of an automatic lane markings detection method. It is performed by applying HT on the background of freeway monitoring images. Selected straight lines are then sorted according to their positions and directions from the left to the right of the image, resulting in clusters of lines. For each cluster, a lane marking is taken as the straight line with the highest HT accumulation value.

Another BGO system based on HT is found in [Gao *et al.* 2011]. This work implements a lane detection module in order to support a crossing road monitoring system. The detection starts by doing the binarization of the background image in order to obtain edges. Next, the edges corresponding to straight lines are selected by HT. Road borders and lane markings are then assumed to be the lines that attend some predefined angular restrictions. Figure 2.13 depicts this process.

The method proposed by [Mazaheri & Mozaffari 2011] attempts to find the road borders by combining HT with the clustering technique K-means (see Section 2.4.1 for an explanation about K-means). After image lines extraction from a vertical gradient filter, the HT is applied to detect the straight lines. Next, these lines are clustered with the help of K-means. This way, the image can be segmented into two areas (taking the camera position as reference): near field and far field (see Figure 2.14). The goal is to focus the subsequent steps of the method (aiming vehicle classification) only on the near field, that way saving processing time and memory.

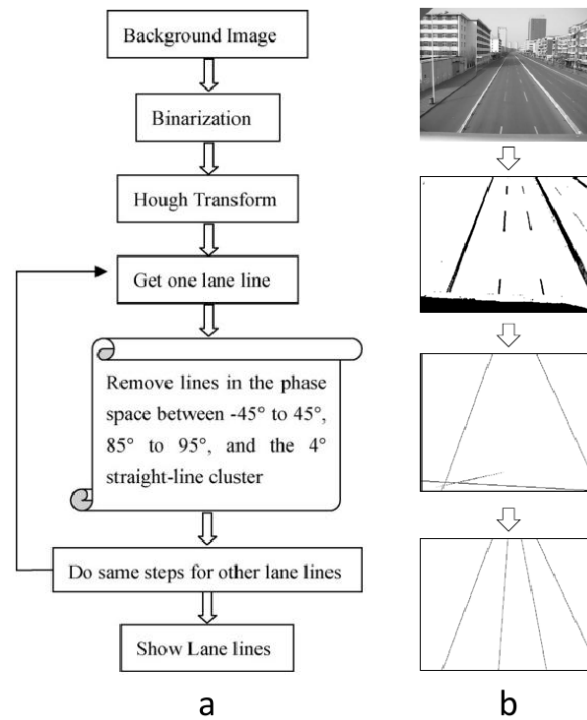


Figure 2.13: Lane detection method proposed by [Gao *et al.* 2011]. (a) Method flowchart and (b) examples of images corresponding to each step at the detection process. Images adapted from *ibid.*

HT is also used by [Li & Zhong 2009] to detect lane markings. The process consists in selecting long lines by means of HT. In this case, let $L = \{L_i | i = 1, \dots, n\}$ be the n detected lines; $P = \{P_i | i = 1, \dots, n(n-1)/2\}$ represent all existing intersections among these lines. The method then calculates vanishing points by doing

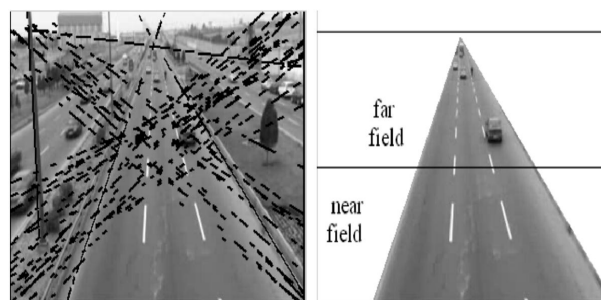


Figure 2.14: Road borders detection algorithm proposed by [Mazaheri & Mozaffari 2011]. Left, detected lines. Right, image segmentation into near field and far field. Images taken from *ibid.*

$$VP = \frac{2}{n(n-1)} \sum_{i=1}^{n(n-1)/2} P_i \quad (2.6)$$

The next step is to select seed points $S = \{Seed_i | i = 1, \dots, m\}$ for each closed region formed by the detected lines and circles generated by the system. These circles have the center at different vanishing points, and their radius are the average of the distances between the VP and the mid-point of all detected lines. Flood fill technique is then performed on each seed point $Seed_i$, resulting in m connected regions $RG = \{Rg_i | i = 1, \dots, m\}$. At the end, the regions Rg_i corresponding to real road lanes are determined by taking their average color

$$C = \frac{1}{num} \sum I(x, y) \quad (2.7)$$

where I is the intensity of a pixel with coordinates (x, y) , belonging to the region Rg_i , and num is the number of pixels inside this region. From this, when C is in a color range $[C_{min}, C_{max}]$, the area of Rg_i is larger than a threshold and there are two or more line segments pointing to the vanishing point, then this region is assumed to be a lane. Examples of intermediate images produced by this method are showed in Figure 2.15.

Among BGO methods, [Lai & Yung 2000] is an exception since it does not use HT. This method is based on the detection and discrimination of edge lines by orientation and length, and works as follows: The starting point is to extract the edges from a traffic image background; for that, Sobel edge detector is used. Then, the edges are thinned and approximated by straight lines, each line with an associated orientation and length. The next step consists in using K-means to cluster these lines according to their orientation. Since 2D camera space is not appropriate to correctly extract the orientation feature, the

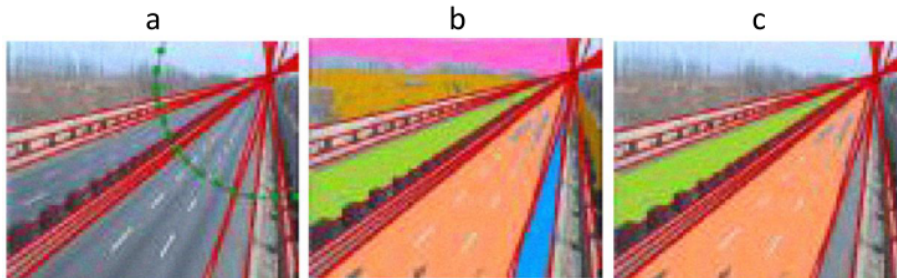


Figure 2.15: Lane detection algorithm proposed by [Li & Zhong 2009]. (a) Long lines extracted using HT and the circle centred at the vanishing point revealed by them. (b) Candidate lane regions obtained from flood fill. (c) Detected road lanes. Images taken from *ibid*.

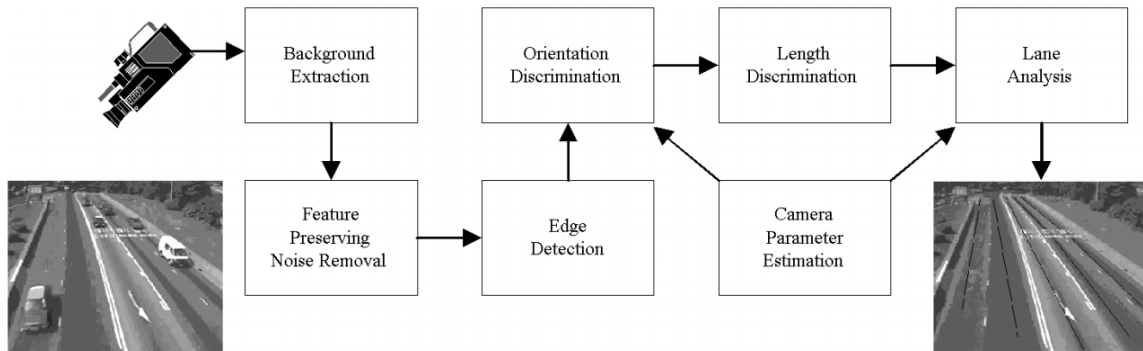


Figure 2.16: Flowchart of the lane estimation method proposed by [Lai & Yung 2000]. Image taken from *ibid.*

clustering is performed after 2D to 3D image transformation. This process requires that camera parameters are known, including focal length, mounting height, and its distance to the focused point on the road. In the sequence, K-means is applied again in order to further cluster the lines according to their lengths. The process ends with the lane estimation, taking into account an assumed road width and the separation between two detected lines adjacent to each other. Figure 2.16 illustrates the flowchart of this method.

2.2.4 FGO methods

As aforementioned, the road detection process of FGO methods is based only on information extracted from the foreground part of the image. In this case, it is assumed that all moving objects correspond to vehicles passing over the road. Therefore if the vehicle trajectories are detected, they can be used to reveal the position and shape of the road in the image.

Three methods were identified as being of FGO type. The first one, proposed by [Luo *et al.* 2011], uses the foreground obtained by means of inter-frame difference to extract the road borders. This is reached by extracting and accumulating the moving objects present in each foreground frame along a sequence. After a given time (empirically estimated), it is expected that the accumulated objects produce a mask which matches with the road shape (see Figure 2.17). Then, the borders of the mask are extracted and submitted to a HT. With this, the method can find the continuous lines that best delimit the region where the motion occurs in the image. Then, these lines are assumed as to be the road borders.

The second example of FGO method is that one proposed by [Pan *et al.* 2010]. That work proposes a traffic surveillance system for vehicle counting. However instead of man-

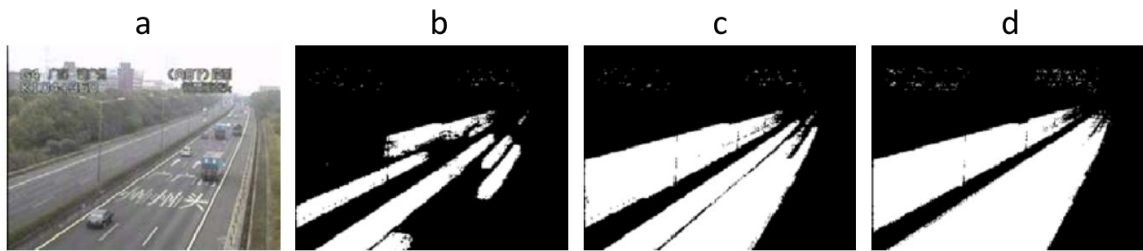


Figure 2.17: Road shape obtained from foreground objects. (a) Input frame. (b), (c), (d) road regions obtained after 2, 5 and 15 seconds, respectively. Image taken from [Luo *et al.* 2011].

usually fixing the windows where the vehicles are detected, Pan *et al.* (2010) provide a lane detection module from which the windows can be automatically positioned in the image. The lane detection starts by taking the blobs of moving objects previously extracted. The center of these blobs are accumulated in order to produce a histogram of the image motion. The histogram is then smoothed, binarized and submitted to connected-component analysis, resulting in lane areas. Finally, the detected lane markings are composed by the pixels at each image row that divides a given lane area. Figure 2.18 shows images obtained from this process.

The last FGO example is the method found in [Melo *et al.* 2006]. This work is aimed to detect and classify highway lanes using the vehicle motion trajectories. For this, several tasks are performed by the system, as described in the flowchart shown in Figure 2.19.

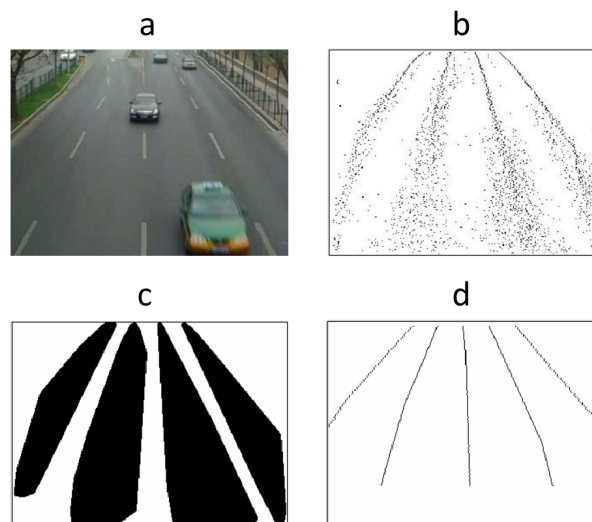


Figure 2.18: Lane detection method proposed by [Pan *et al.* 2010]. (a) Input image, (b) histogram of the motion, (c) lane areas and (d) detected lane markings. Image adapted from *ibid.*

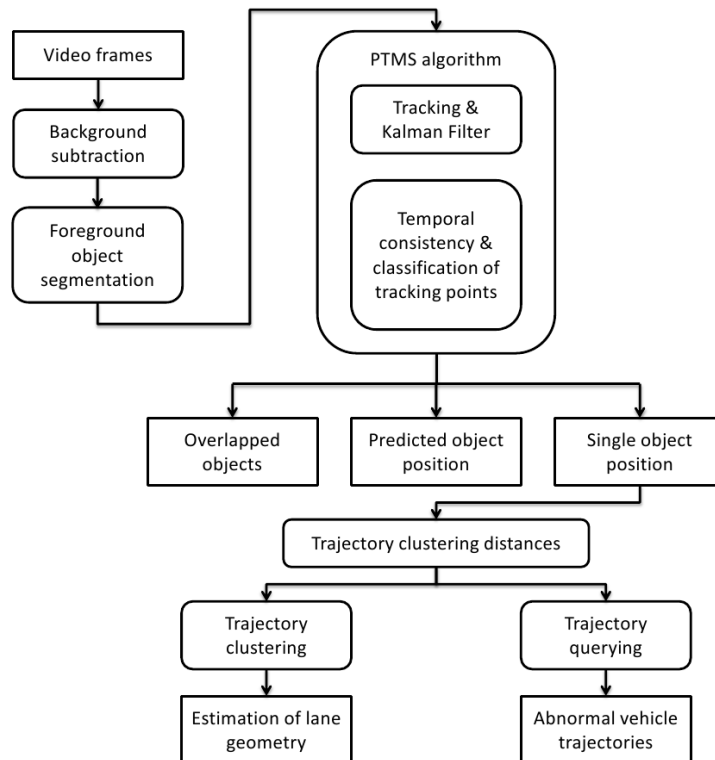


Figure 2.19: Flowchart of the lane estimation algorithm proposed by [Melo *et al.* 2006]. Image adapted from *ibid.*

First, foreground objects are segmented by taking as reference the background model produced by means of background subtraction. Then, an algorithm called “Predictive Trajectory Merge-and-Split” (PTMS) is used to infer the positions of the detected vehicles, as well as to merge and split the blobs in situations of partial or complete occlusion. After using Kalman filter to track the objects, PTMS performs a time-consistent analysis to detect occlusion occurrence. The goal is to identify when the blobs are composed of multiple vehicles, since they should not be used as input to the trajectory clustering algorithm in order to avoid errors at the position estimation. The remaining of the method consists of the application of a number of techniques in order to model the road lanes from the detected vehicle trajectories. These techniques include random sample consensus (RANSAC), aiming at noise filtering; K-means, to cluster the detected trajectories; and least squares, used to model the lanes by cubic polynomials.

2.2.5 HBF methods

As seen before, BGO methods try to detect the road only by means of features extracted from the road itself, such as borders and markings. This strategy, however, limits the applicability of the method since it requires high quality images, taken from restrict viewing angles and regarding only well structured roads. FGO methods, on the other hand, are more robust to deal with scenes where the road markings cannot be viewed (as when they are very deteriorated). However this only works for those parts of the road where sufficient motion occurs. In this context, the hybrid approach of HBF methods appears as an improvement on the road detection robustness, by combining strategies of both BGO and FGO methods. From this idea, foreground objects could be taken as reference to extract some kind of road information, which, by extension, could aid on the background processing to detect the road.

This is exactly what does the lane detection system proposed by [Shin *et al.* 2006]. In principle, this method is similar to a simple BGO method based on HT. However, in order to improve the detection, information about motion is used to better filter the image edges that are candidate for lane markings. For that, the algorithm accumulates data of each pixel from the first frame to the current frame. Then, by analyzing the distribution of the pixels intensities, the standard deviation can be used to distinguish pixels belonging to the road from the remaining ones. This is so because, due to the motion on the inside of the road, the pixels in this part of the image change their intensity much more than the pixels on the outside. Thus a road mask can be created (see Figure 2.20) and applied on the edges detected by a Sobel detector. HT is then used to detect the straight lines among the resulting edges, and these lines are assumed as lane markings.

[Chung *et al.* 2004] propose a method to detect the road by means of fuzzy sets. For

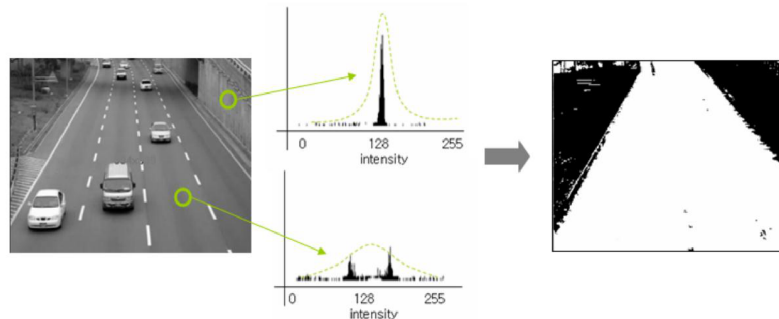


Figure 2.20: Road mask generation from the distribution of the pixels intensities. Left, the input frame; middle, the intensities distributions of pixels inside and outside the road; and right, the obtained road mask. Image taken from [Shin *et al.* 2006].

that, moving objects are firstly extracted from the foreground, and then submitted to morphological operations. Next, connected-component analysis is used aiming to noise reduction. The position of the detected foreground objects are then used to reveal patches of the background, which are cumulatively pasted in an image, called “road image”. From this process, after a number of frames, it is supposed that a major component of the road area will be constructed in the road image. However, since some holes and noise are still expected on the recovered road area, a morphological hole filling technique is then applied. The final step of this method has the goal of locating the correct road boundaries. A color feature is extracted from the road image by averaging the chromatic characteristics of the dominant pixels in the area. Let R and A be, respectively, the road image and an image obtained by doing $A = \text{Background} - R$; then a fuzzy version of the road image can be constructed by doing

$$\mu_B(x, y) = \omega \frac{\sum_{S=r,g,b} |S_R(x, y) - S_A(x, y)|}{\max \{D_i\}} + (1 - \omega) \frac{d(x, y)}{\max \{d_i\}} \quad (2.8)$$

where ω is a weighting factor, S_R and S_A are the respective chromatic characteristics of R and A ; $d(x, y)$ is the distance from the point (x, y) to the nearest point of R ; $\max \{D_i\}$ and $\max \{d_i\}$ are normalization terms. Equation 2.8 is a fuzzy membership function that specifies the degree of a background pixel (x, y) belonging to the road region. This way, the problem of locating the road boundaries is then summarized as selecting an adequate α -cut for this function. Figure 2.21 shows the results of each step of this method.

The next example is the system proposed by [Helala *et al.* 2012]. This work, intended

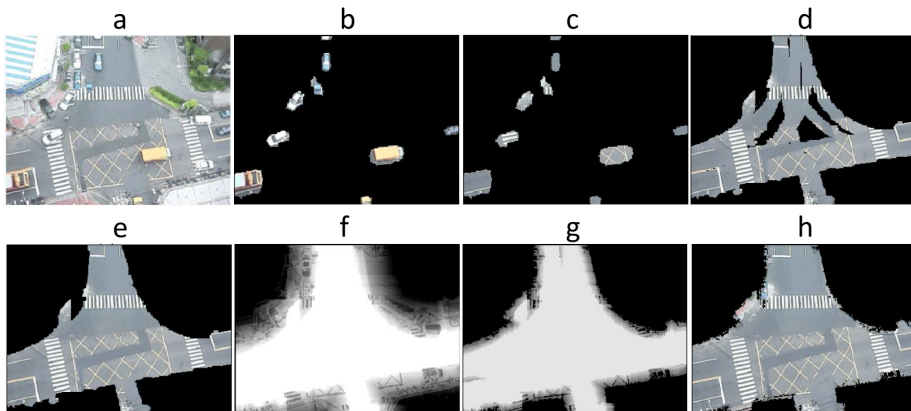


Figure 2.21: Road detection method proposed by [Chung *et al.* 2004]. (a) Input frame, (b) extracted foreground objects, (c) background areas corresponding to foreground objects, (d) major road component, (e) hole filling result, (f) fuzzy map, (g) defuzzified map and (h) final result. Images adapted from *ibid.*



Figure 2.22: Examples of road borders detection by the method of [Helala *et al.* 2012]. Images adapted from *ibid.*

to detect dominant road borders in traffic scenes, consists of five steps: (i) superpixel segmentation, (ii) contour approximation, (iii) hierarchical bottom up clustering, (iv) confidence assignment, and (v) pairwise ranking. As a HBF method, the first four steps are based only on data contained in the image background; they are aimed at selecting a set of candidate lines. The latter step, in turn, extracts foreground information in order to support the detection of real road borders among these candidate lines. The five steps are accomplished as follows: First, superpixels are used to find stable edges in the traffic image. By performing the segmentation over a sequence of frames, Helala *et al.* (2012) expect that superpixel boundaries will match, most of the time, with the road borders. Then, these edges are accumulated in order to identify the road region. The next step is contour approximation, consisting in approximating the superpixels boundaries with polygons. For that, a technique called adaptive sampling is used. This technique works by sampling and selecting points at the superpixel contour that attend to a collinearity criterion. After that, the edges of the polygons are clustered in a hierarchical bottom up method, in which pairs of clusters that are closer to each other are merged to form a single cluster as one moves up the hierarchy. The process continues until only a single cluster is left. The confidence assignment phase attributes a confidence level for each cluster, penalizing the ones with high variance or small number of segments. Finally, in the pairwise ranking step, pairs of clusters are constructed and ranked according to perspective and motion cues. The dominant road boundary is then taken as the lines corresponding to the top ranked cluster. Specifically, this is reached by calculating the vanishing point

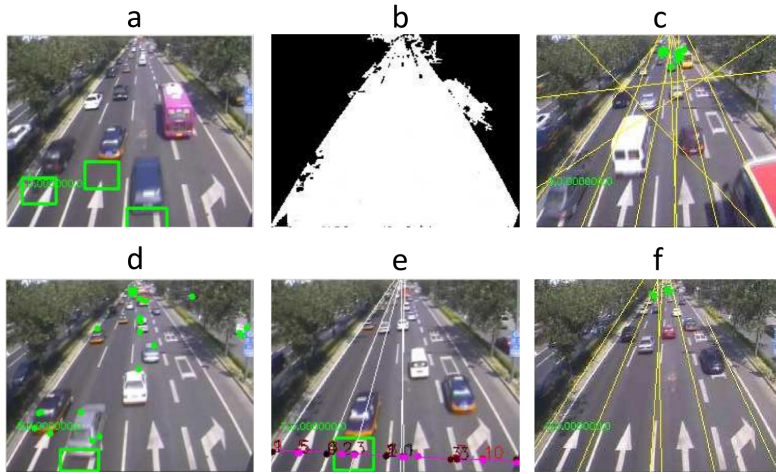


Figure 2.23: Lane detection method proposed by [Liu & Wang 2010]. (a) Input image and detected seeds (green squares), (b) region growing segmentation, (c) straight lines extracted by Hough Transform, (d) moving object tracking, (e) detected trajectories, and (f) detected lane markings. Images adapted from *ibid*.

for each pair of lines, which is used for perspective analysis. For motion analysis, moving objects are segmented from the foreground and their centroids are calculated. Figure 2.22 shows some results obtained by this method.

Our last example of road detection in traffic surveillance is [Liu & Wang 2010]. This HBF method addresses the lane segmentation problem by means of probability maps. Just as occurs in some previously discussed works, here the blobs corresponding to moving objects are accumulated, frame-by-frame, in order to reveal parts of the road. Because this accumulation process gives rise to an image where the higher values indicate the road localization in the image background, it is called probability map. Once road parts are found, samples of them can be extracted and passed as seeds to a SRG segmentation technique. After a number of frames, it is expected that the amount of obtained seeds will allow the SRG to segment the entire road region. The next step is to detect the lane markings. Canny detector is used to extract the edges from the segmented road region, and HT is applied to these edges in order to select straight lines. Finally, To distinguish which lines match with the real lane markings, the method compares the orientation of the lines with the trajectories of the moving objects. In this case, the trajectories are obtained by submitting the foreground blobs to a optical flow method based on Kanade-Lucas-Tomasi (KLT) technique. Figure 2.23 illustrates this method.

2.2.6 Summary of related works

This subsection provides a table that summarizes the main information about the road detection methods discussed so far.

Table 2.1: Summary of road detection methods for traffic surveillance

Method	Type	Purpose	Background Subtraction	Main detection techniques	Applicability (Dataset)
[Gao <i>et al.</i> 2011]	BGO	Lane markings detection	Temporal mean of pixel intensities	Image binarization, Hough Transform	Well-structured straight roads
[Huang 2010]	BGO	Lane markings detection	Gaussian Mixture Model	Hough Transform	Well-structured straight roads (freeways)
[Mazaheri & Mozaffari 2011]	BGO	Lane markings detection	Adaptive background estimation	Vertical gradient filter, Hough Transform, K-means	Well-structured straight roads (highways)
[Li & Zhong 2009]	BGO	Road borders detection	Gaussian Mixture Model	Hough Transform, vanishing point estimation, color analysis	Well-structured straight roads (highways)
[Lai & Yung 2000]	BGO	Lane markings detection	Not addressed	Sobel edge detector, 2D-3D coordinate transformation, K-means	Well-structured straight roads (freeways)
[Pan <i>et al.</i> 2010]	FGO	Lane markings detection	Edge based background estimation	Moving objects histogram, connected-component analysis	Urban roads
[Luo <i>et al.</i> 2011]	FGO	Road borders detection	Inter-frame difference	Moving objects mapping, Hough Transform	Straight roads (freeways)
[Melo <i>et al.</i> 2006]	FGO	Lane markings detection	Adaptive smoothness	Kalman filter, cubic polynomial functions, RANSAC, K-means	Highways
[Chung <i>et al.</i> 2004]	HBF	Road area segmentation	Histogram-based progressive technique	Moving objects mapping, color analysis, fuzzy, shadowed sets	Urban roads of different shapes
[Shin <i>et al.</i> 2006]	HBF	Lane markings detection	Occurrence analysis of pixel intensity	Sobel edge detector, moving objects mapping, Hough Transform	Well-structured straight roads
[Liu & Wang 2010]	HBF	Lane markings detection	Gaussian Mixture Model	Moving objects mapping, region growing, Canny edge detector, Hough Transform, KLT tracking algorithm	Well-structured straight roads
[Helala <i>et al.</i> 2012]	HBF	Road borders detection	Codebook construction	Superpixel segmentation, hierarchical bottom up clustering, vanishing point estimation, motion analysis	Straight roads (highways) at different lighting conditions

2.3 Advanced BGS methods

In Section 2.2.2, we discussed the BGS techniques that compose road detection methods. However, with exception of GMM, the solutions that we have seen so far are too simplistic and outdated. Actually, they only work well under ideal conditions, i.e., when the background is absolutely static and foreground objects move fast enough. Obviously when it comes to real traffic videos such conditions cannot be ensured, as for instance when a traffic jam occurs.

On the other hand, BGS is a topic of extensive research, so that a variety of general-purpose BGS techniques can be found in the literature. In this section we investigate some of the more advanced BGS methods. Our goal is to prepare the basis for a comparative analysis among BGS methods, which will be provided in future chapters.

2.3.1 Adaptive Background Learning

In [Zhang *et al.* 2003] a framework to perform unsupervised detection and spatio-temporal tracking of vehicles is proposed. This framework relies on a BGS technique called adaptive background learning (ABL). Such technique is based on a video segmentation method called simultaneous partition and class parameter estimation (SPCPE), which is applied to separate the pixels in a traffic video sequence into foreground and background. Figure 2.24 illustrates the process of background learning. The first step consists in computing the difference images by subtracting successive frames and applying linear normalization (Figure 2.24.b). Next, SPCPE is used to classify the pixels into foreground and background (Figure 2.24.c), followed by a rectification procedure to eliminate noise (Figure 2.24.d). The rectified segmentation maps are then used to generate a background image (Figure 2.24.d) based on histogram analysis, where the peaks in dominant bins are assumed as background pixels. A remark is that, according to the authors themselves, when a vehicle stops moving or moves slowly in a traffic monitoring sequence the framework may deem it as part of the background.

2.3.2 Gaussian-based methods

As mentioned in Section 2.2.2, there are many different implementations for BGS based on Gaussian distributions. An example is the method of [Zivkovic 2004], where an improved GMM-based algorithm using recursive equations is presented. In this case, recursion is responsible for constantly updating the parameters of the model, as well as simultaneously

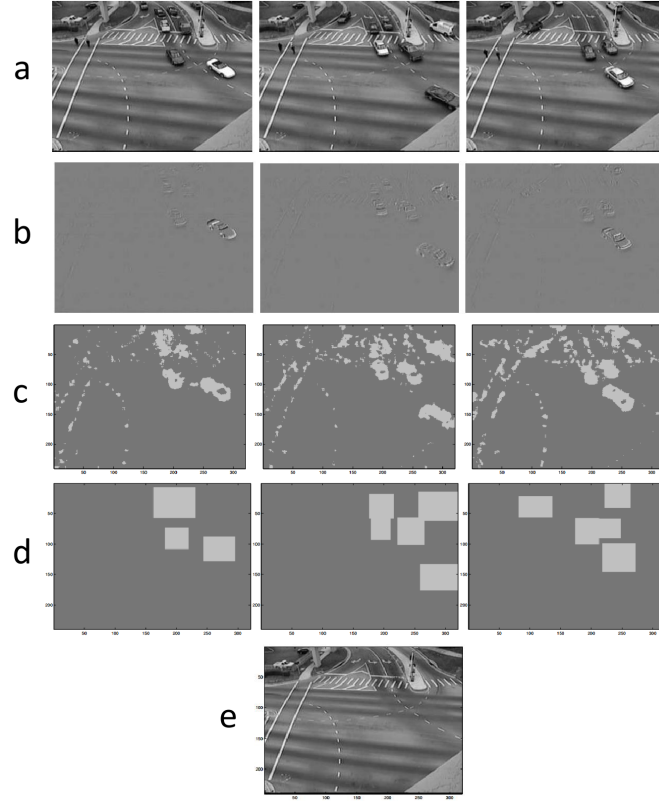


Figure 2.24: Adaptive background learning (ABL). (a) Input frames, (b) difference of frames in (a), (c) foreground/background separation, (d) rectified images, and (e) generated background model. Images adapted from [Zhang *et al.* 2003].

selecting the appropriate number of distributions for each pixel. Thus, let $\hat{\pi}_m$ be the mixing weights, $\hat{\mu}_1, \dots, \hat{\mu}_M$ the estimates for the means and $\hat{\sigma}_1^2, \dots, \hat{\sigma}_M^2$ the estimates for the variances of the GMM model, Zivkovic proposes to update these parameters by doing

$$\begin{aligned}
 \hat{\pi}_m &\leftarrow \hat{\pi}_m + \alpha(o_m^{(t)} - \hat{\pi}_m) \\
 \hat{\mu}_m &\leftarrow \hat{\mu}_m + o_m^{(t)}(\alpha/\hat{\pi}_m)\vec{\delta}_m \\
 \hat{\sigma}_m^2 &\leftarrow \hat{\sigma}_m^2 + o_m^{(t)}(\alpha/\hat{\pi}_m)(\vec{\delta}_m^T \vec{\delta}_m - \hat{\sigma}_m^2)
 \end{aligned} \tag{2.9}$$

where $\vec{\delta}_m = \vec{x}^{(t)} - \hat{\mu}_m$, $\vec{x}^{(t)}$ is a given pixel model, $o_m^{(t)}$ is an ownership function based on Mahalanobis distance, T is the time interval and α is approximately $1/T$. The background model, B , is now given by

$$B = \operatorname{argmin}_b \left(\sum_{m=1}^b \hat{\pi}_m > (1 - c_f) \right) \tag{2.10}$$

where c_f is a measure of the maximum portion of the data that can belong to foreground objects without influencing the background model. According to Zivkovic, if a foreground object remains static long enough, its weight becomes larger than c_f and it can be considered to be part of the background. Additionally, from the analysis of the first line of Equation 2.9 is possible to infer that the object should be static for approximately $\log(1 - c_f)/\log(1 - \alpha)$ frames for this to happen.

Another possibility is to use a single Gaussian distribution to model the background pixels, as in [Wren *et al.* 1997]. Such approach, which is precursor of GMM-based methods, is an alternative to save memory and processing time since the load to compute a single Gaussian distribution is significantly lower than in the case of many distributions.

2.3.3 Fuzzy-based methods

In order to overcome some weaknesses of the original running average-based BGS, [Sigari *et al.* 2008] presented a new approach including fuzzy concepts, so called fuzzy running average. The method is based on the following running average formulation for background updating

$$BG_t = \begin{cases} \alpha BG_{t-1} + (1 - \alpha)I_t, & \text{if } |I_t(i, j) - BG_{t-1}(i, j)| > th_u \\ BG_{t-1}, & \text{otherwise} \end{cases} \quad (2.11)$$

where α is a factor ranging from 0 to 1. If the difference between a pixel in the current frame $I_t(i, j)$ and its corresponding in the last background $BG_{t-1}(i, j)$ is higher than a threshold th_u , then the current background BG_t is updated by running average, given by $\alpha BG_{t-1} + (1 - \alpha)I_t$. Otherwise, the last background model, BG_{t-1} , is kept. Instead of an overall value for α , Sigari *et al.* (2008) propose to compute it for each pixel based on the current value of the fuzzy background subtraction, given by

$$\alpha_{i,j} = 1 - (1 - \alpha_{min})exp(-5 * FBGS(i, j)) \quad (2.12)$$

where the fuzzy background subtraction is according to

$$FBGS(i, j) = \begin{cases} 1, & \text{if } |I_t(i, j) - BG_{t-1}(i, j)| < th_s \\ \frac{|I_t(i, j) - BG_{t-1}(i, j)|}{th_s}, & \text{otherwise} \end{cases} \quad (2.13)$$

Another fuzzy-based approach is the method in [El Baf *et al.* 2008]. This method uses

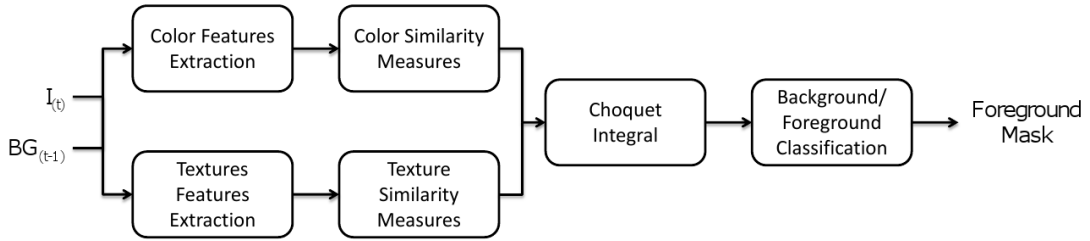


Figure 2.25: Foreground detection process using fuzzy Choquet integral. Image adapted from [El Baf *et al.* 2008].

fuzzy Choquet integral (FCI) to aggregate color and texture features aiming to support the foreground/background classification task. Broadly speaking, this BGS system consists of three stages: background initialization, background maintenance and foreground detection. Particularly, an initial background model is obtained by taking the average of the first N video frames. Next, this model is updated based on the following rules

$$BG_t(i, j) = \begin{cases} (1 - \alpha)BG_{t-1}(i, j) + \alpha I_t(i, j), & \text{if } (i, j) \text{ is background} \\ (1 - \beta)BG_{t-1}(i, j) + \beta I_t(i, j), & \text{if } (i, j) \text{ is foreground} \end{cases} \quad (2.14)$$

where α and β are learning rates to control the speed to adapt to illumination changes and the incorporation of motionless foreground objects into the background, respectively. The foreground detection process occurs as depicted in Figure 2.25. Following the flowchart: Color and texture features are extracted from the background image BG_{t-1} and the current frame I_t ; similarity measures are computed for each feature; the features are aggregated by the FCI; background/foreground classification is done by thresholding Choquet integral's results.

2.3.4 Multi-layer BGS based on color and texture

[Yao & Odobez 2007] propose a multi-layer BGS method which extracts local texture features represented by local binary patterns (LBP) and photometric invariant color measurements in RGB color space. LBP is defined as a gray-scale invariant texture primitive statistic, obtained by the following operator

$$\begin{aligned} \mathbf{LBP}_{P,R}(\mathbf{x}) &= \left\{ LBP_{P,R}^{(p)}(\mathbf{x}) \right\}_{p=1,\dots,P}, \\ LBP_{P,R}^{(p)}(\mathbf{x}) &= s(\mathbf{I}^g(\mathbf{v}_p) - \mathbf{I}^g(\mathbf{x}) + n), \quad s(x) = \begin{cases} 1 & x \geq 0, \\ 0 & x < 0, \end{cases} \end{aligned} \quad (2.15)$$

where $\mathbf{I}^g(\mathbf{x})$ corresponds to the gray value of the pixel \mathbf{x} in the image \mathbf{I} , and $\{\mathbf{I}^g(\mathbf{v}_p)\}_{p=1,\dots,P}$ to the gray values of the P equally spaced pixels $\{\mathbf{v}_p\}_{p=1,\dots,P}$ on a circle of radius R with center at \mathbf{x} . The parameter n accounts for the noise. Equation 2.15 labels the pixels of an image region by thresholding the neighborhood of each pixel with the center value and considering the result as a binary number (binary pattern). According to the authors, LBP features can work robustly on background modeling in most cases. However it should fail when both the background image and the foreground objects share the same texture information. To tackle this problem, they have used photometric color features in RGB color space. For that, the method compares the color difference between a foreground pixel and a background pixel using their relative angle in RGB color space with respect to the origin and the minimal and maximal values for the background pixel (obtained in a background learning process). Then, at the end, a background model is constructed from the combination of both texture and color features.

2.3.5 Neural network with self-organization approach

In [Maddalena & Petrosino 2008], artificial neural network (ANN) has been used to perform background modeling through a self-organizing approach. This method consists in organizing the ANN as a 2D flat grid of neurons (or nodes), where each node computes a function of the weighted linear combination of incoming inputs. The weights, in this case, resemble the ANN learning, and each node is represented by a weight vector. A neuronal map representing the background model is then constructed by taking weight vectors for each color pixel in the input image. To update the model, each pixel of the incoming frames are fed to the ANN, where they are compared to the current pixel model in order to analyze if they match. If a best matching occurs, it means that the pixel belongs to the background; otherwise, it is assumed that the pixel is in a shadowed region or belongs to a moving object.

2.4 Superpixel segmentation

The term *superpixel* describes the process of oversegmenting an image into homogeneous regions that align well with object boundaries. This allows to represent an image with only a couple of hundred segments that work as atomic building blocks instead of tens of thousands of pixels [Schick *et al.* 2012]. Another point is that, due to their tendency to adhere to object boundaries, superpixels can further be used in the task of recognizing object shapes.

Many different approaches to segment an image into superpixels can be found in the literature. As showed in [Schick *et al.* 2012], state-of-the-art methods produce relatively close results with respect to segmentation accuracy and boundary recall. The main difference among them, in this case, resides in the shapes of the superpixels generated by each method. That way, some methods provide more regular and compact superpixels, while others favour a good matching with the object shapes. In the first case, the drawback is the undersegmentation issue, which means that sometimes much more superpixels than the necessary are generated. In the second case, the problem refers to computational load, since these are usually graph-based methods. To illustrate these scenarios, Figure 2.26 compares segmentations produced by two methods that are representative of each aforementioned case. They are, respectively, simple linear iterative clustering (SLIC)¹ and graph-based image segmentation; both will be discussed in the following.

¹Available at www.vlfeat.org/.

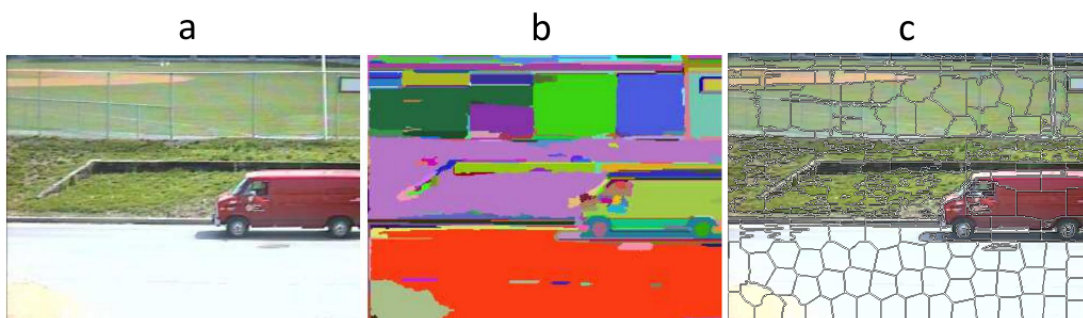


Figure 2.26: Examples of image segmentation. (a) Original image, (b) superpixels produced by the graph-based approach (images taken from [Felzenszwalb & Huttenlocher 2004]) and (c) superpixels generated by applying SLIC on (a).

2.4.1 Simple linear iterative clustering

SLIC is an adaptation of k -means for superpixel generation, proposed by [Achanta *et al.* 2012]. To understand how SLIC works, let us firstly see the rationale behind k -means. K -means is a very simple clustering algorithm, which became quite popular in image processing applications. It works by assigning each pixel in the image to the nearest cluster, considering a priori k clusters. In a general manner, the method is composed by the following steps:

1. Place k points into the space represented by the objects that are being clustered. These points represent initial group centroids;
2. Assign each object to the group that has the closest centroid;
3. When all objects have been assigned, recalculate the positions of the k centroids;
4. Repeat steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

The metric to be minimized, in this case, is the within-cluster sum of squares. SLIC is based on the same idea, however it presents two modifications aiming to save computer time and to provide more regularity and compactness for the superpixels. The first modification consists in constraining the search space to a region proportional to the superpixel size, that way reducing the number of distance calculations in the optimization process (see Figure 2.27). The second change consists in using, instead of the original K -means metric, a weighted distance measure that combines color and spatial proximity

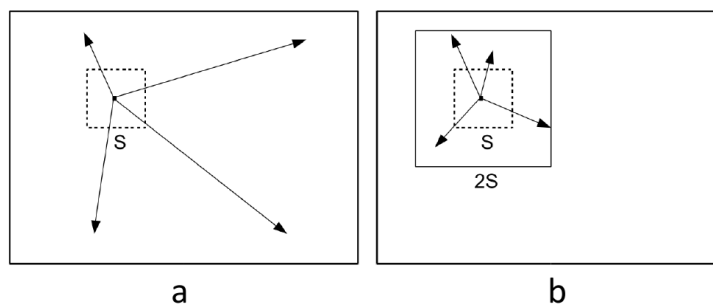


Figure 2.27: Difference in the superpixel search regions of k -means and SLIC. (a) Standard k -means searches the entire image, while (b) SLIC searches a limited region. Image taken from [Achanta *et al.* 2012].

$$D = \sqrt{d_c^2 + \left(\frac{d_s}{S}\right)^2 m^2} \quad (2.16)$$

where d_c denotes the Euclidean distance between pixel and cluster intensities in $[l \ a \ b]^T$ color space, d_s is the Euclidean distance between pixel and cluster spatial coordinates, S is the sampling interval and m is a factor which provides control over size and compactness of the generated superpixels.

2.4.2 Graph-based image segmentation

[Felzenszwalb & Huttenlocher 2004] propose to segment an image through a graph-based approach. The method selects edges from a graph, where each pixel in the image corresponds to a node in the graph, and certain neighboring pixels are connected by undirected edges. Weights on each edge measure the dissimilarity between pixels. In order to avoid over-segmentation of regions with high variability but that belong to the same object, the method adaptively adjusts the segmentation criterion based on the degree of variability in adjacent regions of the image. Moreover the evidence for a boundary between two regions is inferred by comparing two measures: intensity differences across the boundary and intensity differences between neighboring pixels within each region. From the example of segmentation produced by this method (showed in Figure 2.26.b), it is possible to notice that, differently from SLIC (Figure 2.26.c), the superpixels do not have any format or size restriction, varying from small round (e.g., the vehicle wheels) to big rectangular-shaped (e.g., the road area) segments.

2.5 Relation to our work

In Section 2.1, it was mentioned that road detection is a required task in many systems that develop tasks related to the traffic. It is the case of traffic speed enforcement based on radars, where information about the lanes is used to calibrate the sensors. With the advent of the HD imaging, images could be taken from large distances, as from aircrafts or even satellites. This have allowed the use of such devices for monitoring purposes, including the inspection of road structures, as well as the analysis of traffic conditions. Image-based systems have also been employed directly on vehicles, as a way to provide driver assistance or, more ambitiously, to enable autonomous driving.

Although these researches are not directly related to our work — which is turned to

traffic surveillance applications from cameras in fixed roadside spots —, some solutions presented by them have inspired the development of the present work. It is the case, for example, of the image color analysis proposed by [Li *et al.* 2009]. As we shall see in future chapters, our system is based on a color feature called *gray amount*, which takes into account gray pixels in the image similarly to the method of Li *et al.* (2009).

In Section 2.2, we discussed the methods that are directly related to our work. According to the classification scheme that we have adopted, BGO methods are the ones which focus only on background features; FGO focus only on motion information extracted from the foreground; and HBF take advantage of both image components. By this criterion, as it will be clear later, our method fits into the HBF category.

Relevant improvements compared to state-of-the-art methods are presented. First, our road segmentation process is superpixel-driven, which allows for segmentation of structures with any shape. Second, rather than using just one feature, we extract a set of them, regarding different aspects — color, texture and horizon line — from the image background. By doing so, our algorithm is able to carry out the road segmentation in a greater range of situations. And third, we deal with motion information from a contextual manner, which allows us to know when a moving object is or is not a vehicle. It contributes to the robustness of the method, since in real traffic scenes usually exist other moving objects besides cars.

Other important remark concerns the BGS. In Section 2.2.2 we investigated the solutions proposed by the existing road detection methods to fix the problem of separating background and foreground information. As observed, the BGS methods that have been adopted are too simplistic and outdated. Indeed, there are other more elaborate techniques, as approached in Section 2.3. However even those advanced techniques do not provide a background modeling good enough for our purposes. This is because they are specially sensible to foreground objects that move slowly in the scene. To overcome this problem, we propose a innovative background modeling approach, which takes into account contextual information during the modeling process.

Closing this chapter, we have discussed about segmentation. As already said, our system segments the road based on superpixel generation. Again, an owner solution was developed to cope with this issue. This is because the over-segmentation problem presented by methods as in [Achanta *et al.* 2012], as well as the computational cost of graph-based methods as in [Felzenszwalb & Huttenlocher 2004] are not compatible with our purposes. This point will be better comprehended in the next chapters, when we will present the details of our road detection method.

Overview of the proposed road detection system

Contents

3.1	Requirements and constraints	45
3.2	System conception	46
3.3	System design	48
3.4	Dataset	49
3.5	Closure	50

As we have seen, rather than a specific procedure to be applied on an image, road detection methods are comprised of a set of smaller subtasks, each one involving different image processing techniques. Besides the methods behind each technique, the final result depends on how these techniques are arranged to compose the bigger system.

This chapter introduces our road detection method by providing an overview of the entire system, mainly approaching how its components are organized. At this moment, the proposed method is addressed in a general manner, starting from the requirements and constraints that have driven the conception of the method to the general design of the proposed solution.

3.1 Requirements and constraints

The road detection system presented here has been developed according to the following guidelines:

- **Detection of roads with any shape.** The proposed system must be able to detect urban paved roads with any shape, such as straight or curved road segments,

intersections, roundabouts etc.. curved roads, and not necessarily well-structured (i.e., with visible and well-defined curbs and markings).

- **Detection of poorly-structured roads.** The system must provide detection even at the presence of poorly-structured roads (i.e., without visible and well-defined curbs and markings).
- **Robustness to motionless foreground objects.** The proposed method must include a strategy to deal with foreground objects that move slowly or even stop in the scene.
- **Robustness to partial lack of features.** In order to be applicable in a wider range of urban scenarios, the proposed method must be robust to situations where the features are not fully present (e.g., when there is no motion in the scene or the horizon line is not visible).
- **Low computational complexity.** The goal of the road detection method is to support a traffic analysis system in an *on-the-fly* manner. Thus, it is imperative that such a method save the maximum memory and processing time to the traffic analysis module.

Moreover the following constraints limit the scope of this system:

- **Natural daylight illumination.** At this point, the system will not be able to operate at night conditions.
- **Gray asphalt.** The detection is restricted to roads paved with common gray asphalt. Although in very specific cases the asphalt is painted another color, this constraint attends the vast majority of the urban situations.
- **Perspective images.** Although it could eventually work for aerial taken images, the system has been designed regarding only traffic images taken in perspective. This is the usual type of image in traffic monitoring.

3.2 System conception

Based on the points listed in the last section, we conceived our road detection method upon the following rationale, illustrated in Figure 3.1. Taking a traffic image (Figure 3.1.a), after BGS if we can segment the entire objects present in this image (Figure 3.1.b), then we can find out which of them correspond to road parts by doing some assumptions:

- **Gray color.** Despite some variations in the color intensity and tonality, roads are generally gray. That way, if most of the pixels of a given object are not gray, then this object is not a road.
- **Homogeneous texture.** In order to reach a large view of the roads, traffic surveillance cameras are usually placed in elevated spots in roadside. Because the texture is not distance invariant, the road pavement roughness is not caught by the cameras. Thereby the road texture presents a high homogeneity in this kind of image.
- **Position below the horizon line.** Given a road in a traffic scene, taken in perspective, the road area will always appear below the horizon line.
- **Vehicle motion occurrence.** Under normal conditions, vehicles in a traffic scene always move on the road surface. Therefore if the position of a given vehicle is known, then the position of the corresponding road part is also known.

Each assumption gives rise to a respective feature: *color*, *texture*, *position relative to the horizon line* and *vehicle motion occurrence*. In a traffic image, if an object is predominantly gray (Figure 3.1.c), has homogeneous texture (Figure 3.1.d), appears fully below the horizon line (Figure 3.1.e) and is where vehicle motion occurs (Figure 3.1.f), then there is a high probability of this object to be a road. By expanding this reasoning we conclude: the probability of a given object to be road is proportional to how much it matches with the conceived features.

However it is noticeable that the first three features are not sufficient to determine if an object is a road. In this case, working as exclusion criteria. In other words, the sky in a cloudy day could present a gray color along with a homogeneous texture. But since the sky never appears below the horizon line, then it would be discarded.

For a few cases, it is true, this rationale will not work. Indeed a sidewalk could be gray, homogeneously textured and fully below the horizon line. Thus, in such case motion information is required to disambiguate the classification. It is noteworthy that, although the proposed system has been designed to work when a partial lack of feature occurs, its efficiency depends on the quantity and the quality of the available features. Nevertheless, there are too few objects that, like a sidewalk, could be mistaken for a road in an usual traffic scene. That said, by inferring and weighting the probability of each pointed feature, we can determine the global probability of an object to be or not a road. This way, by taking the objects with suitable probability rank, we can compose a road mask (Figure 3.1.g) that matches to the real road area (Figure 3.1.h).

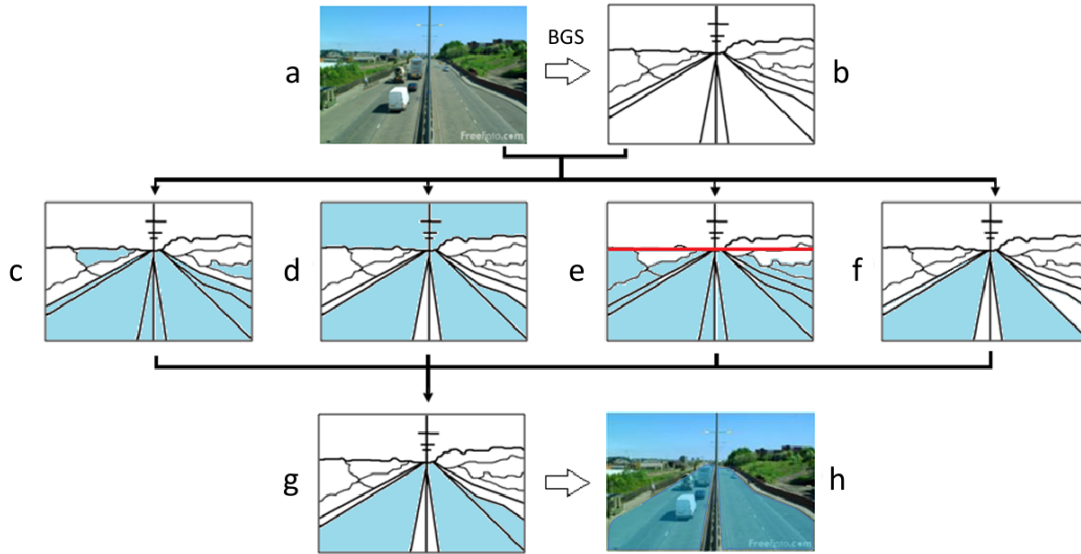


Figure 3.1: Rationale behind the proposed road detection method. (a) Input traffic image, (b) segmentation of the objects in (a) after BGS, (c) segments predominantly gray, (d) segments homogeneously textured, (e) segments fully below the horizon line, (f) segments where vehicle motion occurs, (g) segments selected as being road and (h) the resulting road mask overlapping (a).

3.3 System design

Figure 3.2 shows an overview of the system that we have designed to accomplish the methodology proposed in the last section. Like all related methods discussed so far, ours begins by doing BGS for each new frame of a given traffic sequence. However, in our case, BGS is comprised of a novel background modeling process, which is responsible for making the system robust to motionless foreground objects. As a HBF method, both generated models of background and of foreground are exploited at the detection phase. From the background, besides extracting color, texture and horizon line features, our system segments each object in the image by generating superpixels. The foreground, in turn, is submitted to a motion analysis process, which is intended to find out the objects corresponding to vehicles; these objects give rise to the motion feature. The last stage refers to a classification process, where each superpixel (image object) is matched with the correspondent feature vector. This process is responsible for separating the objects into road and non-road classes, yielding a road mask.

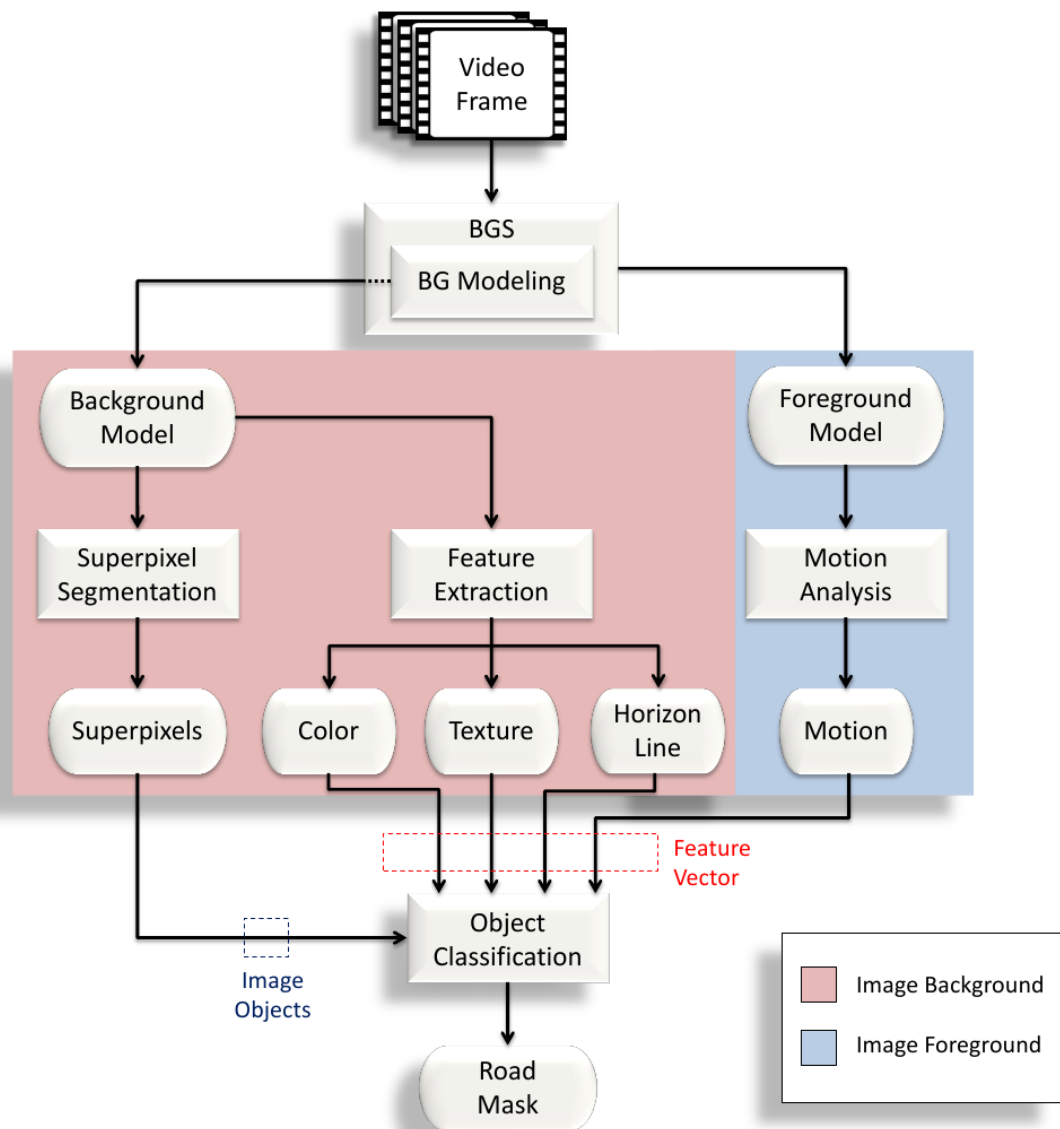


Figure 3.2: Overview of the proposed road detection system: First, video frames are separated into background and foreground by BGS. Next, superpixel segmentation and feature extraction are performed in the background model, while the foreground model is submitted to a motion analysis. Finally, object classification matches each image object with the corresponding feature vector in order to produce a road mask.

3.4 Dataset

There are very few available datasets comprised of traffic surveillance videos. Even those which can be found, most of the time are intended to traffic analysis applications. In that case, since the goal is not forward road detection but for evaluating traffic status, videos are taken from the same road at different traffic conditions. Therefore they are not

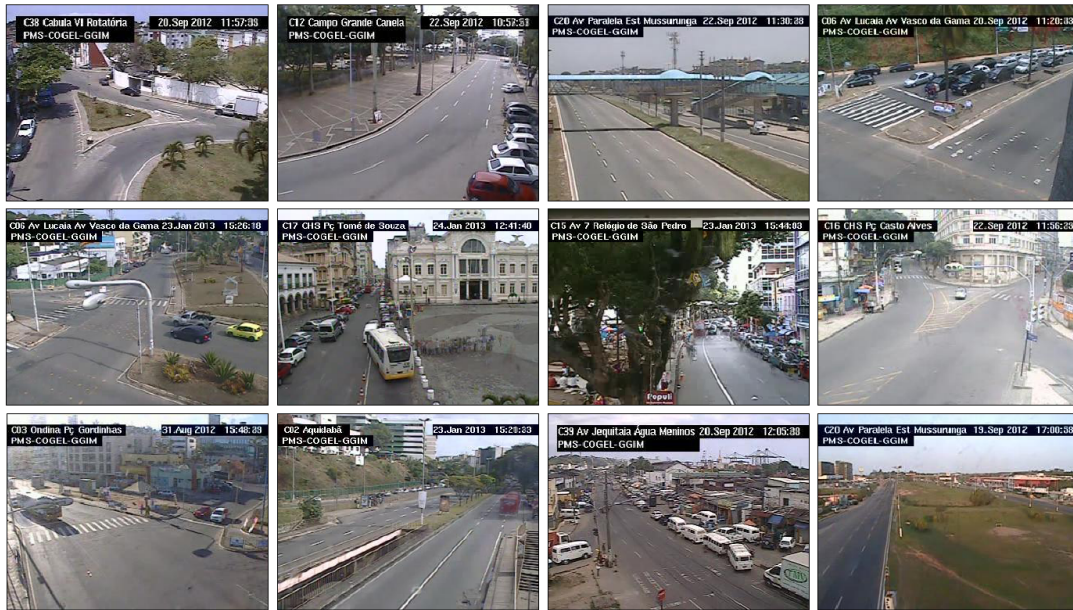


Figure 3.3: Some instances of our dataset. The images illustrate some of the difficulties imposed to our road detection method, such as roads of different shapes with worn out markings, vehicles parked on roadside, shadows, lighting variations, pedestrians, trees, etc.

suitable for our purposes. Another option would be using the same dataset as the related works, however none of them has provided it. That way, we needed to collect our own traffic videos.

Our dataset is comprised of 26 videos of traffic surveillance, each one taken from a different urban road¹. Figure 3.3 shows some frames extracted from these videos. As can be seen, this is a very challenging dataset, including roads of different shapes with worn out markings, vehicles parked on roadside, shadows, lighting variations, pedestrians, trees, among other difficulties.

3.5 Closure

The road detection method outlined in this chapter is based on a number of tasks, dealt as black boxes so far. Background modeling, superpixel segmentation, motion analysis, feature extraction are all issues that still need to be addressed.

In the next chapter we will present our solution to cope with each one of those issues.

¹Videos were provided by COGEL (Companhia de Governança Eletrônica do Salvador), which is the local agency responsible for the video surveillance system of Salvador city. There are a total of 26 videos, each one with approximately 50 seconds, resolution 320 x 240 and frame rate 25 frames per second.

As will be seen, they are novel in many aspects in order to provide the functionalities specified here. This is the case of our background modeling method, which takes into account contextual information extracted from the scene in order to deal with motionless foreground objects. Our segmentation method, in turn, overcomes the over-segmentation problem by generating superpixels that match, as much as possible, with the shapes of the objects in the image. Differently from graph-based techniques, this is done without significant computational cost. Similarly, our road detection method is comprised of simple-but-efficient strategies to perform feature extraction. This way, we have reached a good trade-off between performance on the detection and computational complexity.

Context-aware road detection system

Contents

4.1	Context-supported road information for background modeling	54
4.1.1	Approximated median BGS	55
4.1.2	Vehicle filtering	56
4.1.3	Road color-based adaptive analysis	57
4.2	Superpixel segmentation based on edge density	59
4.3	Simple-but-efficient strategies for feature extraction	61
4.3.1	Gray amount-based color feature	62
4.3.2	Edge density-based texture feature	63
4.3.3	Horizon line estimation based on edge density	64
4.3.4	Vehicle motion-based features	65
4.4	Object classification	66
4.5	Closure	67

As previously discussed, road detection is not the final purpose for a traffic analysis system. Instead it is a tool to aid the module in charge of extracting traffic data. Therefore computational complexity is a crucial issue for road detection algorithms, since an excessive load could compromise the performance of those processes directly related to the traffic analysis.

The implementation of our road detection method has been carried out with this matter in mind. As will be shown next, we have developed solutions that conjugate low computational complexity with effectiveness. Such an accomplishment has been reached by means of multifunctional processes. For example, during the background modeling

stage we perform vehicle detection. The goal, in that case, was to use contextual information obtained from vehicles to improve the modeling process. However, since the vehicles were detected, information about them could also be used by the motion analysis process in order to extract an improved motion feature. Likewise, the color feature extraction is based on an aforementioned color metric (called gray amount), which has been shared with the background modeling process. Multifunctionality is also present in our segmentation algorithm, which is based on a concept called *edge density*, computed to determine the superpixels. It turns out that edge density is also a good texture descriptor, so that it can be used to extract texture feature. But not only that: edge density is also the basis for horizon line estimation, as will be seen later.

4.1 Context-supported road information for background modeling

Figure 4.1 shows an overview of the proposed background modeling method, so called context-supported road information (CRON), which is based on the following strategy. Firstly, BGS based on approximated median is performed to extract a foreground mask, as well as to generate an earlier color background model. Next, the mask is analyzed

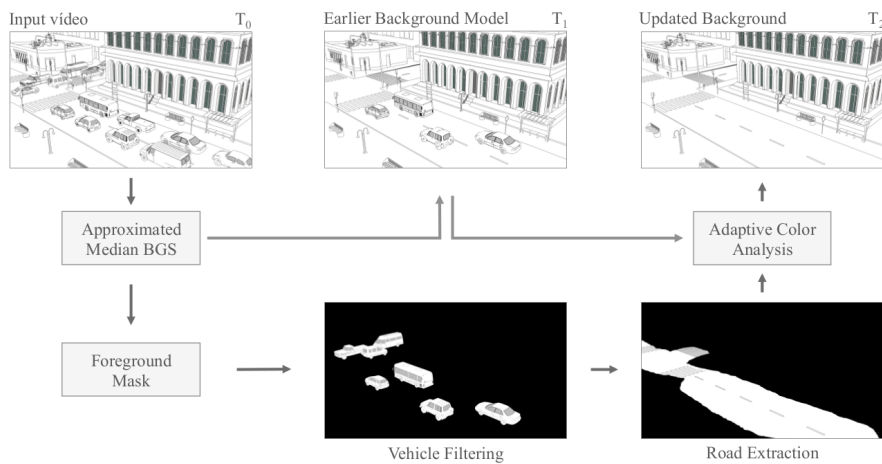


Figure 4.1: Overview of the context-supported road information for background modeling (CRON). The method is comprised of three main tasks: approximated median BGS, which performs background/foreground separation; vehicle filtering, responsible for extracting contextual information from the scene; and color analysis, the process intended to generate background models adaptively.

in order to filter out those objects most likely to correspond to vehicles, since vehicle localization mostly reveals the presence of the road. These revealed road parts provide information about the road color, which is used adaptively to update the background model by maximizing the road color in the image.

4.1.1 Approximated median BGS

The rationale of CRON resides in using road color information as reference to adaptively model the background, in order to better distinguish between background and foreground objects, when dealing with traffic videos. For that aim, an initial requirement is to extract some road information, as well as to obtain a starting background model which will be posteriorly updated. To tackle this problem, we use the approximated median BGS, proposed by [McFarlane & Schofield 1995], to separate the foreground from the background at the beginning of the processing (by now, without concerning for the motionless foreground objects problem). The approximated median algorithm is an improvement of the original median based BGS, which addresses the memory requirement problem. In the approximated form, instead of storing each pixel value along the frames to calculate the median, the background is obtained by incrementing or decrementing the pixel intensities, depending on whether its value becomes higher or lower from the current frame to the next. This is given by

$$BG_k = \begin{cases} BG_{(k-1)} + u, & \text{if } Frame_k - BG_{(k-1)} > 0 \\ BG_{(k-1)} - u, & \text{otherwise} \end{cases} \quad (4.1)$$

where BG denotes pixels in background, u is a background updating factor, and $k = 1, 2, \dots, N$ is the number of frames to be analyzed. Additionally, a foreground mask can be obtained by doing

$$FG_k = \begin{cases} 1, & \text{if } |Frame_k - BG_{(k-1)}| > \rho \\ 0, & \text{otherwise} \end{cases} \quad (4.2)$$

where FG denote pixels in the foreground, and ρ is the threshold to determine the foreground.

The motivation to use the approximated median BGS as support for our background modeling method, was mainly because it attends the on-the-fly requisites of our system. Besides, when in presence of relatively fast moving objects, by adequately choosing u and ρ , the approximated median yields a background model good enough for our purposes at this point. Indeed the quality of the background model produced at this point is not

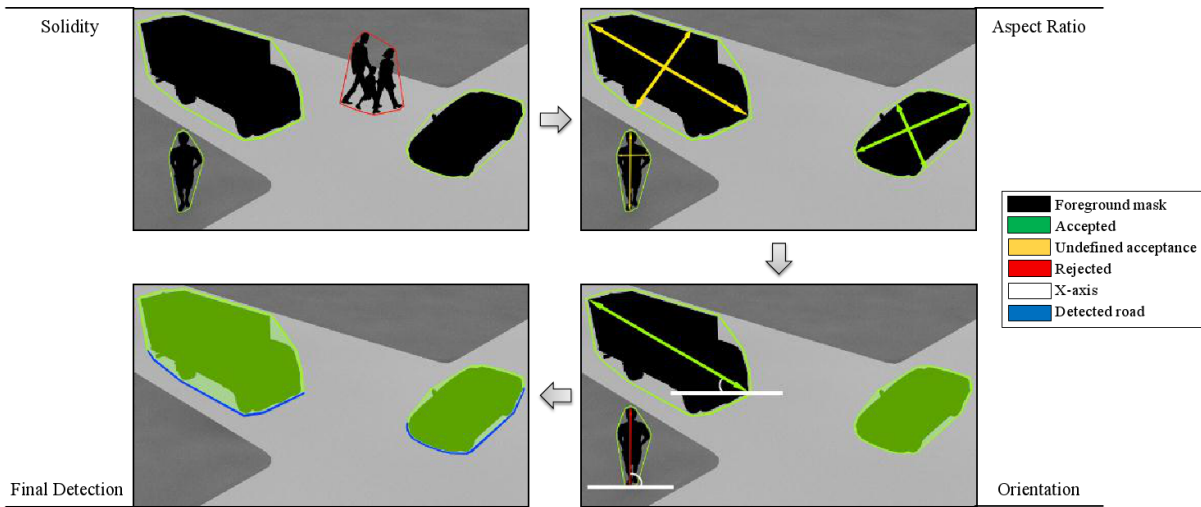


Figure 4.2: Vehicle filtering for road sample extraction. Following the arrows: foreground blobs are evaluated by the solidity criterion, which measures the area of intersection among the blob and its convex-hull; aspect ratio of this convex-hull is analyzed: if an object meets this criterion, it is considered car; orientation relative to the x-axis is further considered in order to distinguish vehicles from people; at the end, the road regions are taken as the bottom of the convex-hulls of the objects passed in the criteria chain.

critical, since it will be posteriorly updated by CRON, as will be described in the next sections.

4.1.2 Vehicle filtering

Methods in the same category of CRON, which provide background modeling, usually do not properly deal with foreground objects when they become slow or static in the video sequence. This mainly happens due to their generalist approaches, focused on just temporal aspects of the pixel behaviour. In other words, since such methods do not use any contextual information in order to learn how foreground and background pixels are, they are not capable of distinguishing among them if these two kinds of pixels behave similar for a while.

To address that question, CRON adopts a more context-aware approach. The method starts by analyzing the foreground mask provided by Equation 4.2 in order to filter those objects more likely to be vehicles. For that, each blob in the mask (black regions) is analyzed by following a chain of criteria, as in Figure 4.2.

1. **Solidity** consists in a measure of how much pixels in the blob are also pixels of the convex hull which envelops that blob. The idea is that cars tend to be convex objects

and, therefore, if a given object presents low solidity, it is probably not a car. This first criterion mainly serves to eliminate those blobs derived from groups of people that are walking near to each other. However, the solidity alone is not enough to determine if an object is a car. For example, people walking alone with arms near to the body appear to be reasonably convex (as seen with the person inside the green hull in Figure 4.2). Then, to deal with such situations, the algorithm analyzes the blob under the next criterion.

2. **Aspect ratio.** In traffic surveillance videos, foreground is normally comprised of vehicles passing on the roads, as well as people crossing these roads or walking on the sidewalks. In this context, a discriminant feature to distinguish cars from isolated people is the aspect ratio, i.e., the relation between height and width of an object. Since the height of a common person is not smaller than three times the width, in situations where it is not true the object is probably a car (green arrows in Figure 4.2). This rule, however, is not valid for long vehicles, like bus or trucks, or even small cars queued (yellow arrows in Figure 4.2). For those cases, the blobs are further analyzed under the next criterion.
3. **Orientation.** Taking the x-axis as reference (white segments in Figure 4.2), the angle formed by the longest axis of the blobs is somewhat different when it comes from people (red arrow in Figure 4.2) or vehicles. While the former presents nearly 90 degrees, the latter follows the orientation of the roads, which are almost never vertical from the common viewpoints of surveillance cameras. Thus, by choosing the correct angles, the orientation criterion can help us to further identify vehicles in the foreground mask.

After passing throughout this triage process, the remaining objects — considered to be vehicles — are used to detect road regions. For that, only the pixels at the bottom of the vehicle convex hulls (blue regions in Figure 4.2) are selected. The goal is to avoid misclassification, since, in the images, the lowest parts of the vehicles are always overlapping road regions, while the same cannot be said for the upper parts. At the end of the vehicle filtering, the selected pixels are then accumulated along the frames to support the adaptive road color analysis, as will be described in the next section.

4.1.3 Road color-based adaptive analysis

In order to analyze the color aspect of the road pixels, we conceived a descriptor called gray-amount (Algorithm 1, Line 9), which is computed by taking the mean of two mea-

sures:

- **Pixel grayness.** This measure is reached by calculating, for each pixel, the average of the RGB inter-channel square differences (Line 7).
- **Color pixel distance to the road color.** This measure consists in the euclidean distance from the pixel intensity to the road color (Line 8).

Algorithm 1 CRON: context-supported road information for background modeling

Input: Traffic video

Output: Background model $newBG$

```

1: /* Main Routine */
2: for k = 1 to K frames do
3:    $[BG_{(k)}, FG_{(k)}] \leftarrow approximatedMedianBGS(k)$ 

4:    $Rmap_{(k)} \leftarrow vehicleFiltering(FG_{(k)})$  ▷ Mapped road pixels

5:    $roadColor \leftarrow weightedMean(Rmap_{(k)}, BG_{(k)})$ 

6:   /* Pixelwise operations */
7:    $G'_{(k)} \leftarrow \frac{\sum_{i=1, C_i \in \{R, G, B\}}^2 \sum_{j=i+1, C_j \in \{R, G, B\}}^2 (BG_{C_i} - BG_{C_j})^2}{3}$ 

8:    $G''_{(k)} \leftarrow \left| \frac{BG_{R(k)} + BG_{G(k)} + BG_{B(k)}}{3} - roadColor \right|$ 

9:    $G_{(k)} \leftarrow \frac{G'_{(k)} + G''_{(k)}}{2}$  ▷ Gray Amount

10:  /* Update */
11:  if  $(G_{(k)} > G_{(k-1)})$  and  $(FG_{(k)} == 0)$  then
12:     $newBG_{(k)} \leftarrow BG_{(k)}$ 
13:  end if
14: end for

```

The first measure — adapted from Equation 2.1 [Li *et al.* 2009] — aims to favour gray pixels, since they have similar values in each channel. On the other hand, the greater the inter-channel difference, the lower this measure. It is in line with the system constraints presented in last chapter, where we have limited the scope to only paved gray roads. In respect to the second measure, pixel intensity is computed by taking the mean of its RGB values¹. The road color is then adaptively obtained by extracting samples of road pixels

¹Equivalent to convert the pixel to grayscale.

from the earlier background model. This is possible due to the road portions revealed at the filtering process presented in Section 4.1.2. In this case, we calculate the weighted average of the sampled road pixels, where the weights are given by the correspondent accumulation of the detected vehicles along the frames. The idea is to give more weight to the colors of the road regions where the heavier traffic occurs.

Algorithm 1 delineates the whole background modeling process presented here. As can be seen from Line 10, CRON controls the updating of the earlier background model by maximizing the gray amount for each of their pixels. This way, if a previous static foreground object starts moving, then the gray amount in that region increases due to the road pixels being revealed, which results in a fast background updating. On the other hand, if a moving object suddenly stops, the gray amount in that region gradually decreases, which make the updating process stops there.

4.2 Superpixel segmentation based on edge density

In our discussion about superpixel segmentation, in Chapter 2, we pointed out some drawbacks of existing approaches. As seen, there is usually a trade-off between over-segmentation and computational load. The problem is that both effects are incompatible with our road detection system as designed. Indeed, whilst low computational complexity is an explicit system requirement, a segmentation process that excessively divides the road area in many small parts would turn difficult the posterior classification of these parts. This is because the features that we conceived sometimes present local variations in relation to their global aspect. For example, although the road color is generally gray, it can vary locally due to shadows and lighting variations. The same is true for the texture, mainly affected by markings on the road, or the motion, that does not occur over every minimal part of the road, but just next to its center. Thus, to overcome these problems, we developed a different superpixel approach, which is very simple and runs fast, while providing a satisfactory matching with object shapes. This technique is based on the concept called *edge density*, which ultimately denotes the amount of edges in a given image region. Figure 4.3 depicts how this concept is useful to generate superpixels, step by step.

The first step is to compute the image gradients (Figure 4.3.a); then, an iterative process begins. Like an edge detector (e.g., Canny, Sobel), thresholding is used to obtain the image edges. Initially we use a fewer restrictive threshold, so that a great amount of edges are detected. As the process iterates, the threshold becomes higher, yielding

increasingly less edges (Figure 4.3.b). The rationale consists in progressively extracting the contour of the objects in a top-down manner. In this case, detecting a great amount of

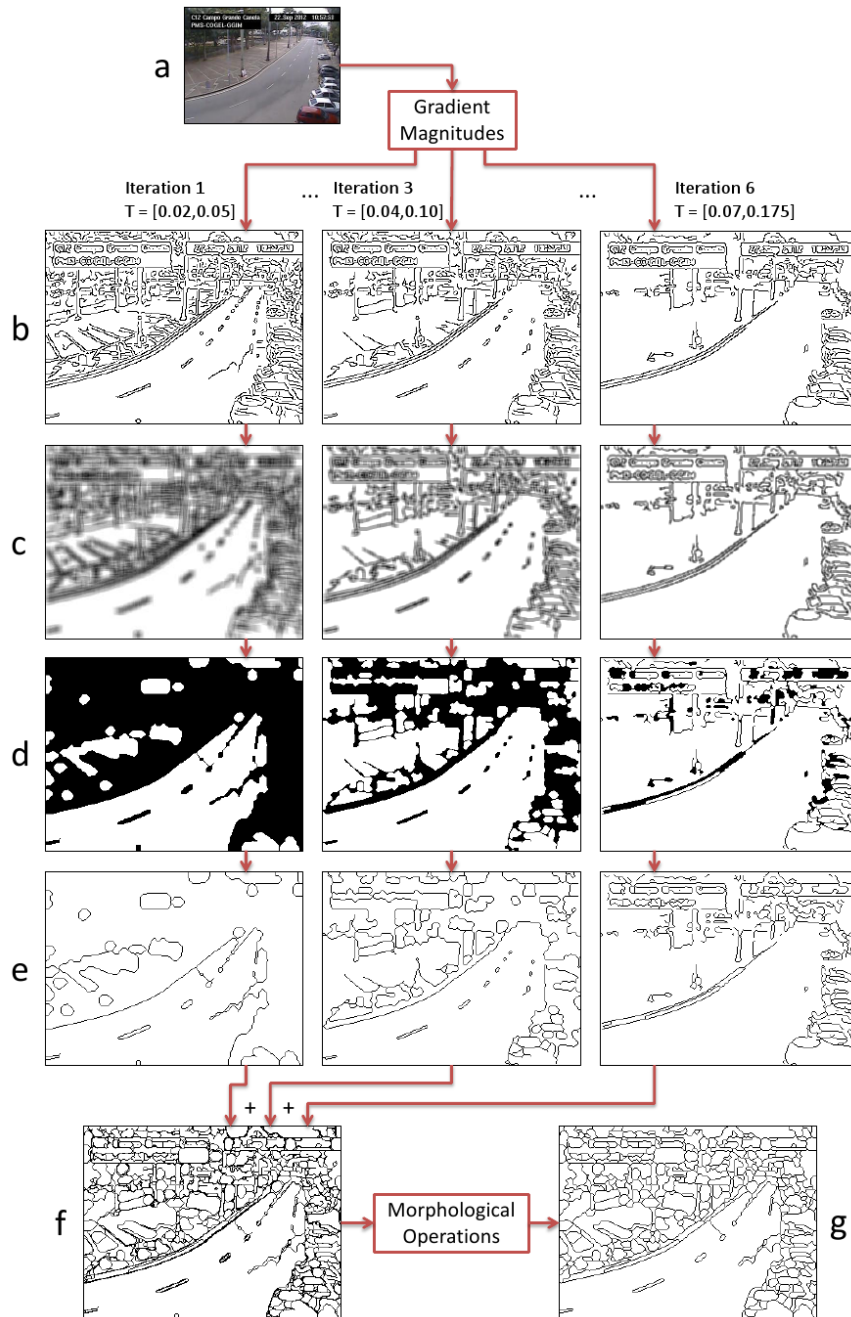


Figure 4.3: Superpixel segmentation based on edge density. From left to right, columns show intermediate results at iterations 1, 3 and 6 (for a total of 6); T is the threshold pair for a Canny-based edge detection. The rows: (a) input image, (b) detected edges, (c) edge density filtering, (d) thresholding of (c), (e) contour extraction of (d), (f) accumulated contours, and (g) final result after applying morphological operations on (f).

edges allow us to extract the outer contour of the object, since the edges tend to compose a cluster with the same shape of the object that originate them. As the number of edges decreases at each new iteration, we are able to extract the contour of smaller and smaller internal parts of the objects.

The next step is to perform a spatial linear filtering (Figure 4.3.c) in the edge image by means of so-called edge density filter

$$ED = \frac{1}{S^2} \sum_{i=1}^{S^2} p_i \quad (4.3)$$

where the edge density ED is the arithmetic median calculated in the neighborhood $S \times S$ of the pixels p_i .

The idea behind this filter is that, although the edge clusters are discontinuous, i.e., composed by edge and non-edge pixels, their local edge density is not null. This way, by applying a suitable threshold over the filter kernel, we can expand the edges to fill the area covered by the filter. On the other hand, the expanded edges lose the adherence with the real contour of the objects in the original image. To cope with this situation, a thinning morphological operation is then applied in order to refine the borders, but without generating new discontinuities. Since the edges are expanded by $S - 1$ pixels, the thinning needs to be performed $(S - 1)/2$ times to correctly return them to their original position in the image² (Figure 4.3.d).

Once repositioned, we then use lookup table (LUT) technique to extract the contour of the expanded edges (Figure 4.3.e), so that by this process we obtain only perfectly closed segments, which are suitable to be superpixels. At the end, after having accumulated the contours generated at each algorithm iteration (Figure 4.3.f), we then apply morphological operations in order to merge adjacent contours and remove isolated pixels (Figure 4.3.g). Algorithm 2 summarizes the entire superpixel segmentation process described here.

4.3 Simple-but-efficient strategies for feature extraction

In this section we present the processes used to extract the features that we have conceived at the system design: color, texture, position relative to the horizon line and vehicle motion occurrence. As previously discussed, in an effort to ensure low computational

²Considering that the thinning operation occurs symmetrically in both sides of the edges.

Algorithm 2 Superpixel segmentation based on edge density

Input: Grayscale image I , initial filter size S , lower and higher thresholds $[t, T]$ **Output:** Binary image containing superpixels SPX

```

1: /* Initialization */
2:  $GM \leftarrow \text{gradientMagnitudes}(I)$ 
3:  $SPX \leftarrow 0$ 
4: /* Main Routine */
5: for  $k = t$  to  $T$  do
6:    $E \leftarrow \text{edgeSelection}(GM, k)$ 
7:   for each pixel  $p_i$  in a region  $S \times S$  of  $E$  do
8:      $ED \leftarrow \frac{1}{S^2} \sum_{i=1}^{S^2} p_i$  ▷ Edge Density
9:   end for
10:  Perform binarization of  $ED$ 
11:  Perform thinning of  $ED$   $((S - 1)/2)$  times
12:   $B \leftarrow \text{blobContourExtraction}(ED)$ 
13:  /* Update */
14:   $SPX \leftarrow \text{OR}(SPX, B)$ 
15:  if  $S \geq 5$  then
16:     $S \leftarrow S - 2$ 
17:  end if
18: end for
19: /* Post-processing */
20: Perform morphological operations on  $SPX$ 

```

load, we make use of multifunctional strategies so that the features are extracted in a simple-but-efficient manner.

4.3.1 Gray amount-based color feature

The gray amount concept presented in Section 4.1.3 is directly used here as a pixel color descriptor. The motivation is that, since this measure has been calculated based on the road color itself, the closer the pixel color in relation to the road color, the higher the gray amount. This way, the feature is computed for each superpixel by taking the average gray amount of all of its pixels. To illustrate its relative importance to the road detection process, Figure 4.4 depicts the heat map of the gray amount descriptor, where the higher the gray amount, the hotter the image pixel. Notice how the road areas have the hotter values of the map. As expected, it is not sufficient to distinguish the road because other objects also present high gray amount, especially sidewalks and buildings.

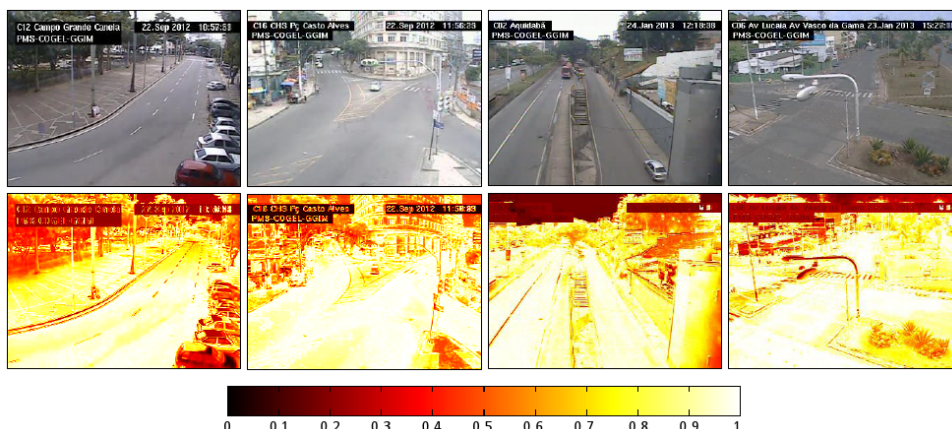


Figure 4.4: Gray amount-based color feature represented in heat maps. First row, input images; second row, correspondent gray amount images represented in heat maps; on the bottom, heat map scale.

4.3.2 Edge density-based texture feature

Similar to the color feature, the texture feature is also obtained by reusing a descriptor previously computed; in this case, we compute the average edge density of the superpixel.

When filtered by an edge detector, objects with homogeneous texture yield fewer edges than non-homogeneous ones. This way we can further use the edge density provided by our superpixel method as a texture homogeneity metric. The heat map in Figure 4.5 evidences how much this feature can be meaningful for our segmentation purpose, substantially highlighting the road region around the other objects. Again, it is noticeable that the

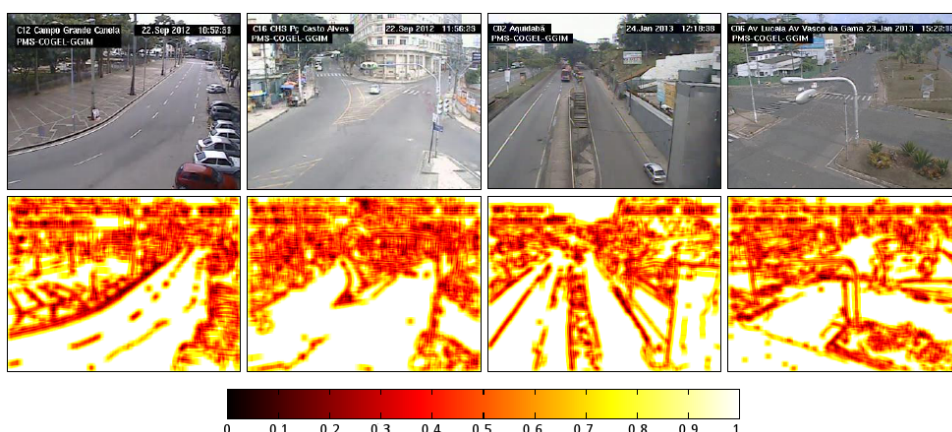


Figure 4.5: Edge density-based texture feature represented in heat maps. First row, input images; second row, correspondent edge density images represented in heat maps; on the bottom, heat map scale.

feature alone is not sufficient to detect the road, since there are other objects in the images with homogeneous texture (e.g., sidewalks and the sky).

4.3.3 Horizon line estimation based on edge density

Herdtweck and Wallraven (2010) evaluated several methods to detect horizon lines in both urban and non-urban scenarios [Herdtweck & Wallraven 2010]. Among them, a particular approach stands out because, unlike what is usually done, it is not based on vanishing point analysis. Instead, it uses Gabor filters — a technique commonly applied to encode texture information. By this method, the horizon line is estimated as the image line that produces the maximum response to the filters. Despite the authors do not deepen into the hypothesis that has motivated this method, it is expectable that the horizon is the most likely place for containing the highest concentration of objects in the image due to perspective effects. Consequently, this will also be the region with the highest texture concentration, which ultimately will maximize the response of the filters.

Our horizon line estimation method is inspired by the same idea. However, rather than spending more processing time to perform Gabor filtering, we assume the horizon line to be the image line with the maximum edge density, which has already been computed in some previous steps. It is straightforward because, once a region has the highest texture concentration, it also will be more likely to contain edges. Figure 4.6 shows some examples of horizon line estimated based on this idea. Once the horizon line is found, the corresponding feature is computed by calculating the vertical position of the superpixel related to this line (normalized in the interval $[0;1]$). In other words, since the road is expected to be always below to the horizon line, the feature is proportional to the

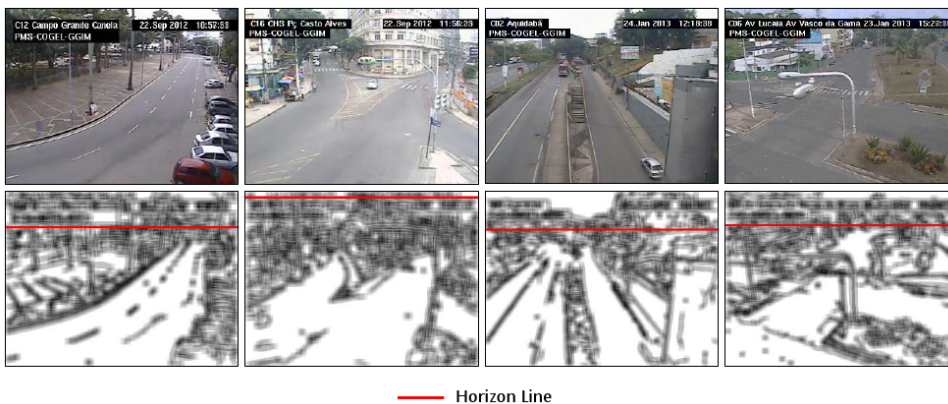


Figure 4.6: Horizon line estimation based on edge density. First row, input images; second row, correspondent edge density images and estimated horizon lines (red).

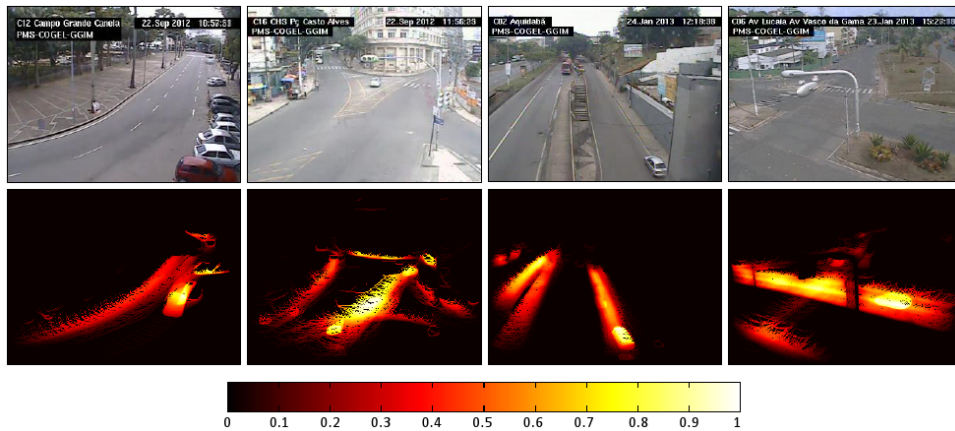


Figure 4.7: Vehicle motion-based feature. First row, input images; second row, correspondent motion mapping images represented in heat maps; on the bottom, heat map scale.

superpixel part that meets such condition. Thus, a superpixel fully below the horizon line has maximum prior value, while a fully above one has prior value equal to zero.

4.3.4 Vehicle motion-based features

While cars are moving in a traffic scene, their trajectories are an obvious cue to identify the road region. Since our road detection system has performed a vehicle detection process in a previous stage (see Section 4.1.2), it is straightforward to use the data extracted at that moment in favour of identifying the road pixels. We reach this by projecting on each superpixel the map containing the accumulated data of vehicle motion along the frames.

Figure 4.7 shows some heat maps representing the accumulated motion data, where the hotter the pixel, the higher the frequency of motion occurred on it. In this case, we compute the feature regarding two dimensions: (i) motion intensity, which measures the average intensity of the traffic in the superpixel region and is calculated by normalizing the values of the motion map in the interval $[0;1]$ and taking their average; and (ii) motion range, which measures the covering of the traffic area over the superpixel area; this is done by binarizing the motion map and dividing the sum of all 1-valued pixels by the total of pixels within the superpixel.

4.4 Object classification

The described feature extraction process outputs a 5-D vector — color, texture, position relative to horizon line and two dimensions for motion — for each generated superpixel. The object classification stage provides the rule by which the superpixels are judged as being road or non-road, depending on the corresponding feature values.

In the literature there are several techniques intending to solve classification problems, such as ANN, support vector machine (SVM), Bayesian classifiers, just to cite a few. Among them, one of the most simple and fast are decision trees. According to [Murthy 1997], decision trees perform classification by a sequence of simple and easy-to-understand tests whose semantics are intuitively clear to domain experts.

In our work, the classification problem is solved by means of a structure similar to a decision tree. However, instead of using a supervised method to automatically learn the decision rules, we have constructed them manually, in an empirical manner. This was possible because, by the way that our system was designed, the relationship among

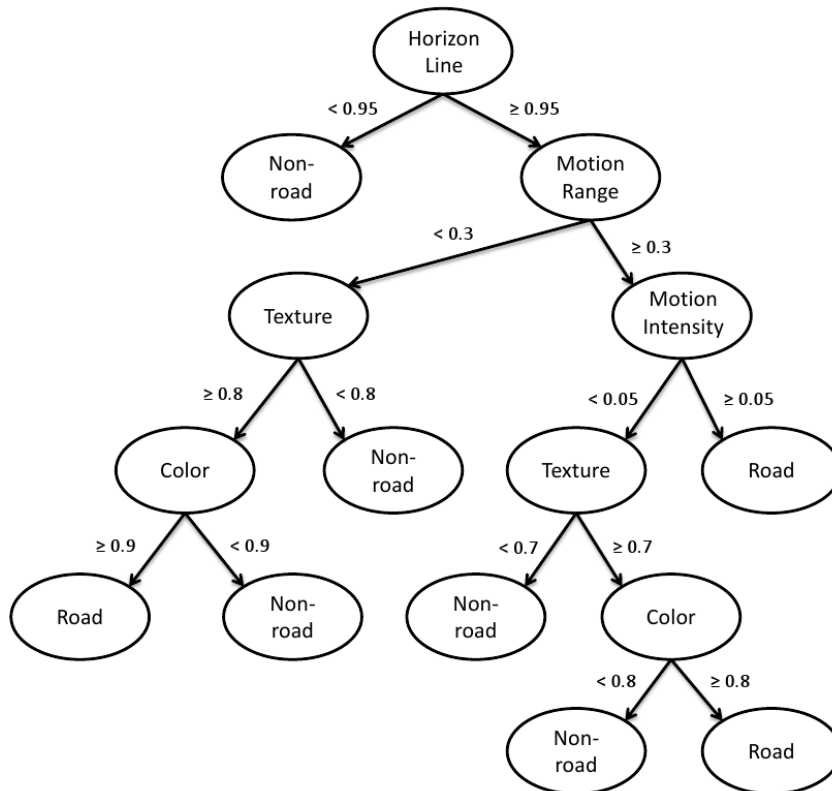


Figure 4.8: Object classification based on decision tree structure. The root (first node) and the branches (intermediate nodes) correspond to the features; the labels shows the decision rules (based on feature values); and the leaves (final nodes) contain the classification result.

the features and the classification problem is easy to be inferred. It is advantageous since the method does not require an extra effort to perform training sets, which further facilitates the system setup. On the other hand, classifiers based on supervised learning are driven by consolidated optimization techniques, which could lead our system to improved results. Nevertheless, as the experimental evaluation in next chapter will show, even this simplistic classification strategy reaches satisfactory results, which ultimately confirms the effectiveness of our method as a whole. Figure 4.8 depicts the structure of the used classifier, as well as the feature values for each proposed rule.

4.5 Closure

This chapter presented, in details, the implementation of the road detection method that we have designed in Chapter 3. As showed, the methods were developed taking into account not only functional effectiveness, but also low computational complexity. Indeed, in order to provide a good trade-off between these qualities, novel solutions concerning image processing tasks had to be created. To evaluate them, in the next chapter we will perform a set of tests comparing some of the related works against ours.

Experimental evaluation

Contents

5.1	Experiments	70
5.1.1	Background modeling performance	70
5.1.2	Superpixel generation performance	73
5.1.3	Road detection performance	79
5.2	Analysis and closure	81

This chapter presents a thorough evaluation of the proposed road detection method. Although the key question to be answered is “how well this method detects roads?”, there are other issues that also deserve attention. It is the case of the developed background modeling technique. Much more than a simple tool to help a road detection task, this method shows interesting capabilities that let open the possibility to use it in other traffic surveillance applications. Section 5.1.1 is dedicated to evaluate this method, comparing it against some of the advanced BGS techniques previously approached.

Likewise, the method that we propose for generating superpixels goes beyond the applicability toward traffic images, appearing as a feasible alternative to segment any kind of image. This conclusion emerges from the comparative analysis presented in Section 5.1.2.

Finally, Section 5.1.3 tries to answer the question raised initially. This is carried out by confronting our road detection method with the dataset presented in Chapter 3. As we shall demonstrate, the performance presented over such challenging images confirms the potential of the method.

5.1 Experiments

5.1.1 Background modeling performance

A comparative analysis is presented, comparing CRON with some implementations¹ of BGS methods previously discussed. They are identified by the following shorthands:

ABL adaptive background learning [Zhang *et al.* 2003].

WMM weighted moving mean.

FCI fuzzy Choquet integral [El Baf *et al.* 2008].

NN self organization through artificial neural networks [Maddalena & Petrosino 2008].

ML multi-layer based on color and texture features [Yao & Odobez 2007].

FG an adapted version of fuzzy running average [Sigari *et al.* 2008] using Gaussian.

SG simple Gaussian [Wren *et al.* 1997].

GMM Gaussian mixture model, version of [Zivkovic 2004].

GMM2 original GMM [Stauffer & Grimson 1999], implemented with Mahalanobis distance.

To assess the performance of the methods in producing reliable background models, we have gathered four traffic videos, with each video containing a different situation. Table 5.1 summarizes the characteristics of each video. The comparative analysis was made as follows: i) Firstly, a background model to be used as reference was manually extracted from each video sequence; ii) after that, binary masks were generated with respect to the road regions for each background reference; iii) next, for each input video frame, a background model was produced by each BGS method to be evaluated; iv) at the end, the generated models were then compared with the corresponding background reference, considering only the road regions delimited by the road masks. The average error is then taken as the mean of these differences.

The results obtained from this analysis are depicted in Figures 5.1 and 5.2, where the average error presented for each method has been plotted. To better view the plots, the initial parts of the curves are not completely shown, since it corresponds to the stabilization phase, where too high errors usually occur.

¹Available at code.google.com/p/bgslibrary/.

Table 5.1: Characteristics of the videos used in the background modeling evaluation.

Video	Characteristic
#1	Vehicles run uninterruptedly with normal traffic conditions.
#2	An abnormal situation occurs from frame 700, when a car slows down until stopping, parks on the roadside, and then reverses looking for a parking spot.
#3	An intersection where cars stop in one of the roads, while cars on the other road are running; from frame 445, the traffic light switches and the situation inverts.
#4	A moderate traffic jam occurs between frames #1 and #250, forcing some vehicles to stop, while others remain moving forward; at frame 500, all vehicles stop for a pedestrian crossing.

Observing the plots of video #1 (Figure 5.1.b), most of the methods performs slightly better than CRON. This was expected since in the cases where the foreground is always moving fast, those methods based only on temporal analysis can easily distinguish foreground from background pixels (FCI, NN, FG, GMM, GMM2, ML). CRON, on the other hand, accumulates some small errors due to foreground pixels that eventually present a gray amount higher than the background color. Despite that, the performance of CRON is better than WMM, ABL (which appear to just follow the input) and SG, which presents a very slow background updating.

In video #2 (Figure 5.1.c), when the abnormal situation occurs, almost all methods are prone to incorporating motionless foreground pixels as background ones. Only SG

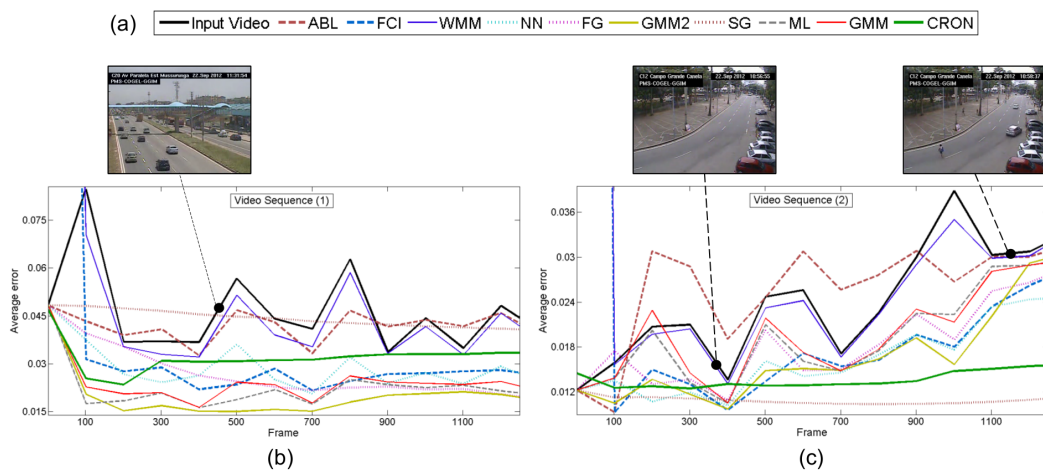


Figure 5.1: Average error in background modeling - Part 1. In (a), the legend for the graph curves; (b) and (c) show the curves related to, respectively, video sequences 1 and 2. Black curve represents the raw video, and the remaining ones are the compared methods.

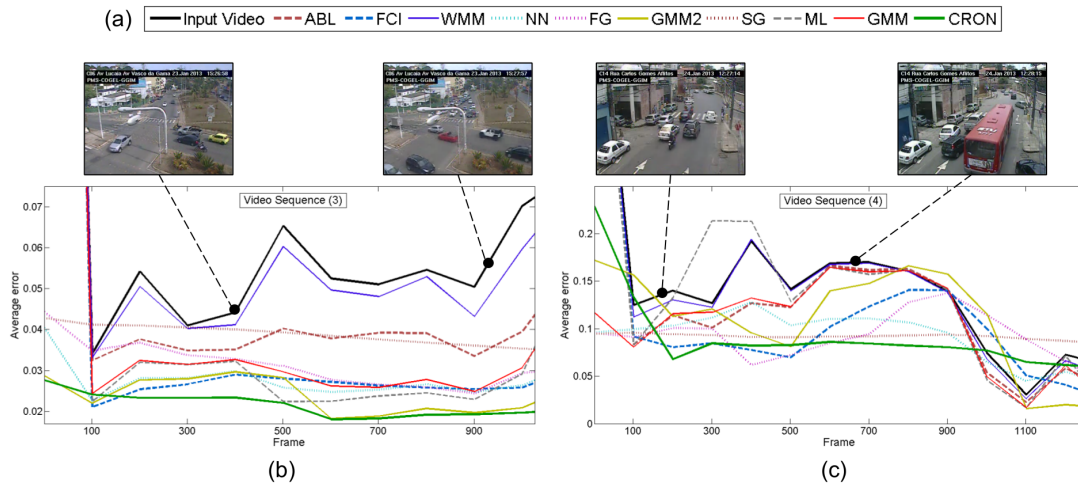


Figure 5.2: Average error in background modeling - Part 2. In (a), the legend for the graph curves; (b) and (c) show the curves related to, respectively, video sequences 1 and 2. Black curve represents the raw video, and the remaining ones are the compared methods.

and CRON yield stable background models in this situation. However the former reaches this result by means of a poor background updating, as happens in video #1.

From video #3, in Figure 5.2.b, it is noteworthy that the other methods suffers from the intersection problem, when the cars on a road start moving and the others become static: leading these methods to always misclassifying foreground as background. After the stabilization phase, CRON's performance is always the best.

The plots corresponding to the video #4 in Figure 5.2.c show that some methods, like FCI and FG, are sometimes capable of dealing with motionless foreground objects for a short time interval, as is the case in the initial part of the video. However, if the foreground becomes static for a longer period, as when the vehicles stop on the crosswalk in the middle part of the video, then these methods also fail in the background modeling. CRON, on the other hand, has demonstrated to be able to produce reliable background models along all situations. The stability presented in this modeling process is an interesting advantage, specially for those applications aimed to segment background objects, as road detection methods.

The characteristics of CRON are further evidenced by the summarized data shown in Figure 5.3, where CRON's mean and standard deviation of the error remain low throughout all the frames, demonstrating the stability of the method. Moreover Figure 5.4 compares the (manually extracted) background references with the models generated by the best four competing methods (GMM2, SG, ML and FCI), besides CRON. It is noteworthy how CRON's models are always robust to motionless foreground objects.

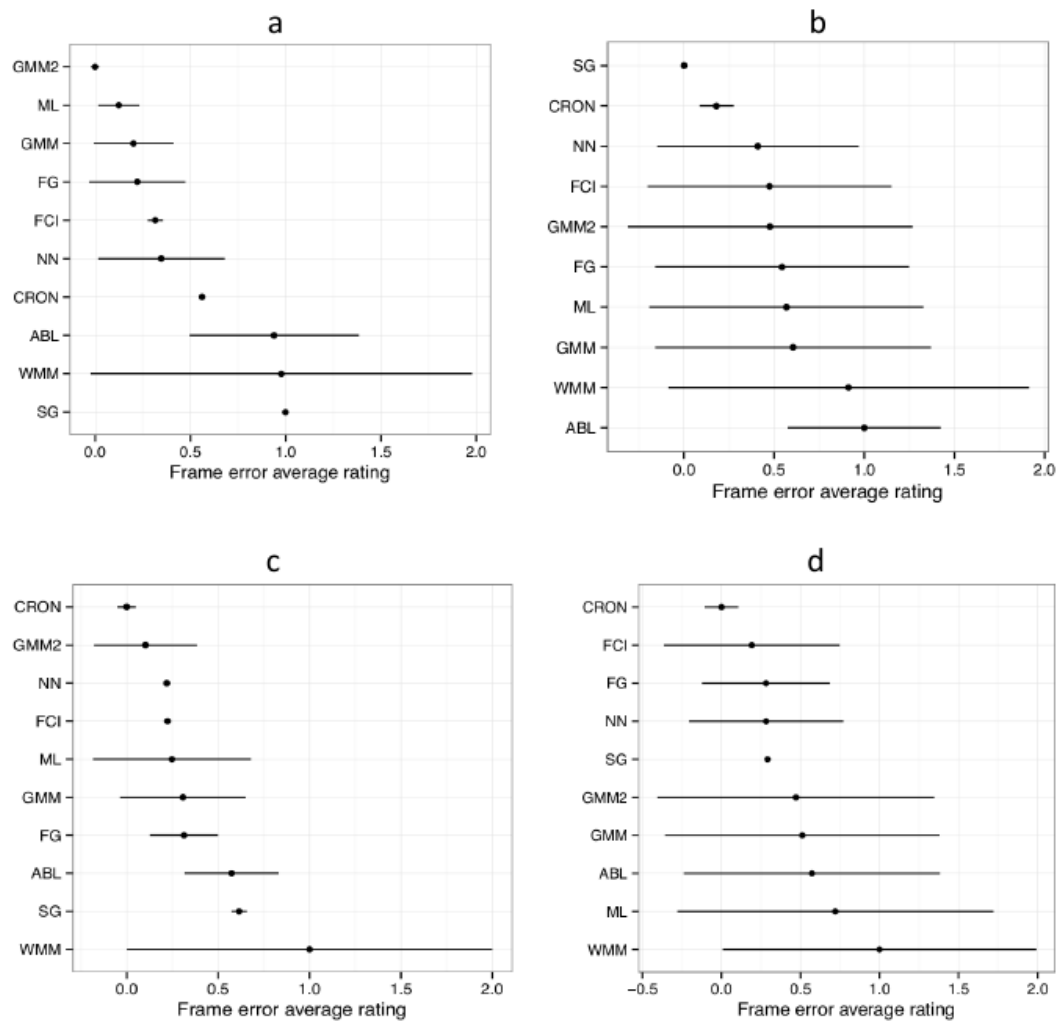


Figure 5.3: Background modeling: mean and standard deviation of the error. (a), (b), (c) and (d) refers to, respectively, video sequences 1, 2, 3 and 4.

5.1.2 Superpixel generation performance

In this section we compare our edge density-based superpixel method with the state-of-the-art SLIC. The comparison was performed over the manually extracted background models used in the last section. As ground truth, we used the contours of the binary masks that were annotated, since they are reliable references for road segmentation. We then assess the performance of the methods by means of metrics commonly used to evaluate superpixel algorithms: boundary recall (BR), achievable segmentation accuracy (ASA) and undersegmentation error (UE). However, in the case of the experimental evaluation presented here, some adjustments had to be done.

First, most of the methods requires as input the (user-defined) parameter that specifies

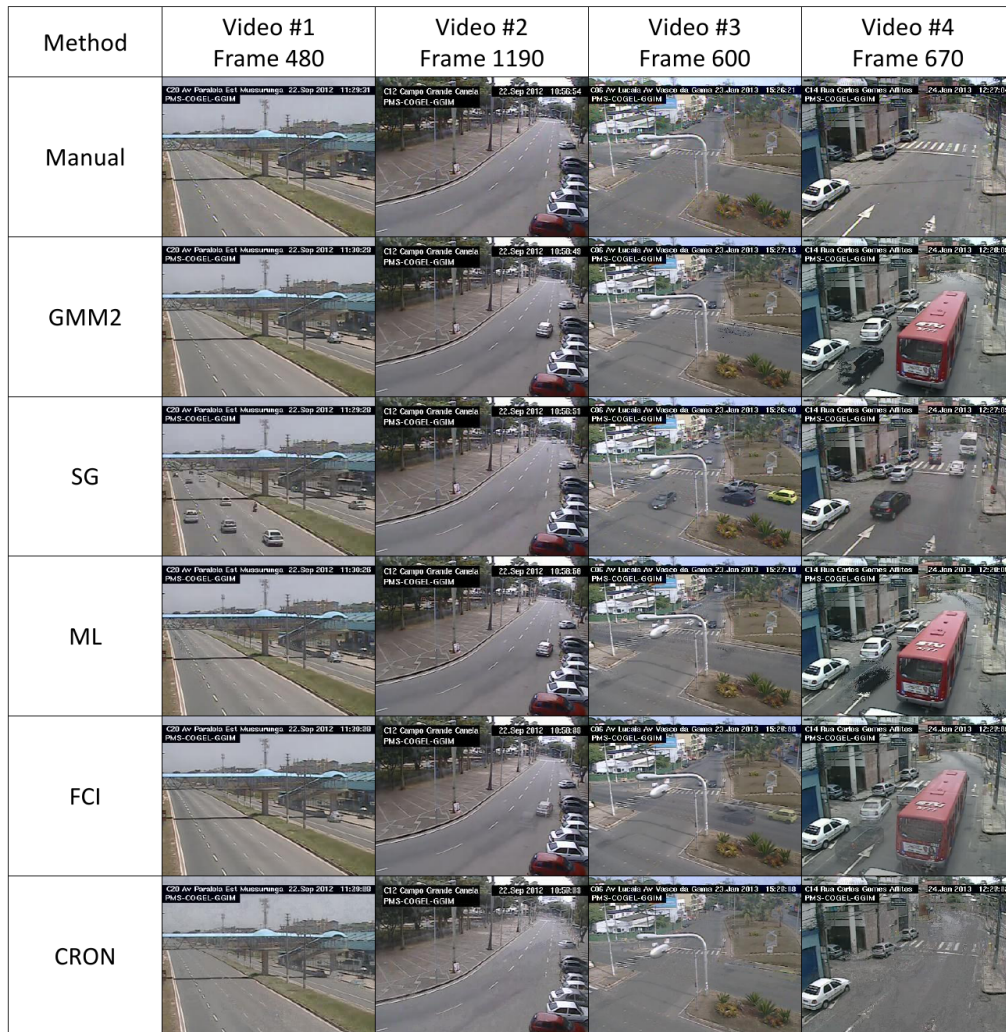


Figure 5.4: Background models. In the first row, frames of manually extracted background references for each video; in the middle rows, examples of background models given by the best four compared methods; and in the last row, background models generated by CRON.

the number of superpixels to be generated. Thus, the evaluation metrics are usually computed as function of this number — starting with few segments and incrementing the number iteratively. Our method, on the other hand, defines the number of superpixels adaptively, so that the user does not have control over the number of segments that are generated. Thus, our tests could not be performed by varying the number of superpixels. Instead, we have simply computed the metrics for each dataset instance, and then plotted the results as a function of the image indexes. In other words, each value in the X-axis of the graphs presented in this section corresponds to a dataset image (in this case, the

X-axis is numbered from 1 to 26, the total of dataset instances).

The adopted methodology is compatible with the SLIC implementation² that we have evaluated, which is a modified version where the number of segments depends on two input parameters: the minimum size for the segments and a factor that determines how much regular their shapes must to be. Against our method, we have tested two configurations of SLIC, identified by the following shorthands:

ED denotes the proposed edge density-based method, configured with initial filter size 9x9 and lower and higher thresholds 0.05 and 0.175, respectively.

SLIC-1 denotes SLIC configured with superpixel area of at least 15 pixels and regularity factor 0.1.

SLIC-2 denotes SLIC configured with superpixel area of at least 50 pixels and regularity factor 0.01.

Next, we discuss the graphs comparing the methods and showing the calculations behind each evaluation metric.

Number of segments. Figure 5.5 shows the number of superpixels generated by each method, for each image of the dataset. As expected, SLIC-1 generates much more segments than SLIC-2 due to its smaller segment area. On average, SLIC-1 produces approximately 1000 segments per image, while SLIC-2 produces 245 segments per image. Our method, in turn, produces an intermediate quantity: 443 segments per image, on average.

Boundary recall (BR) [den Bergh *et al.* 2013]. This metric evaluates the percentage of borders from the ground truth that coincides with the borders of the superpixels. It is formulated as

$$BR(s) = \frac{\sum_{p \in B(g)} I[\min_{q \in B(S)} \|p - q\| < \varepsilon]}{|B(g)|} \quad (5.1)$$

where $B(g)$ and $B(S)$ are the union sets of superpixel boundaries of the ground-truth and the computed superpixels, respectively; p and q are pixels belonging to each of these sets; and $I[\cdot]$ is an indicator function that returns 1 if the nearest pixel is within ε distance. We set $\varepsilon = 2$ as in [den Bergh *et al.* 2013].

Figure 5.6 shows the results of computing BR for each image of the dataset. Our method provides BR higher than SLIC-1, even generating less than half of the superpixels. SLIC-2 performance, on the other hand, shows that SLIC algorithm provides poor

²Available at www.vlfeat.org/.

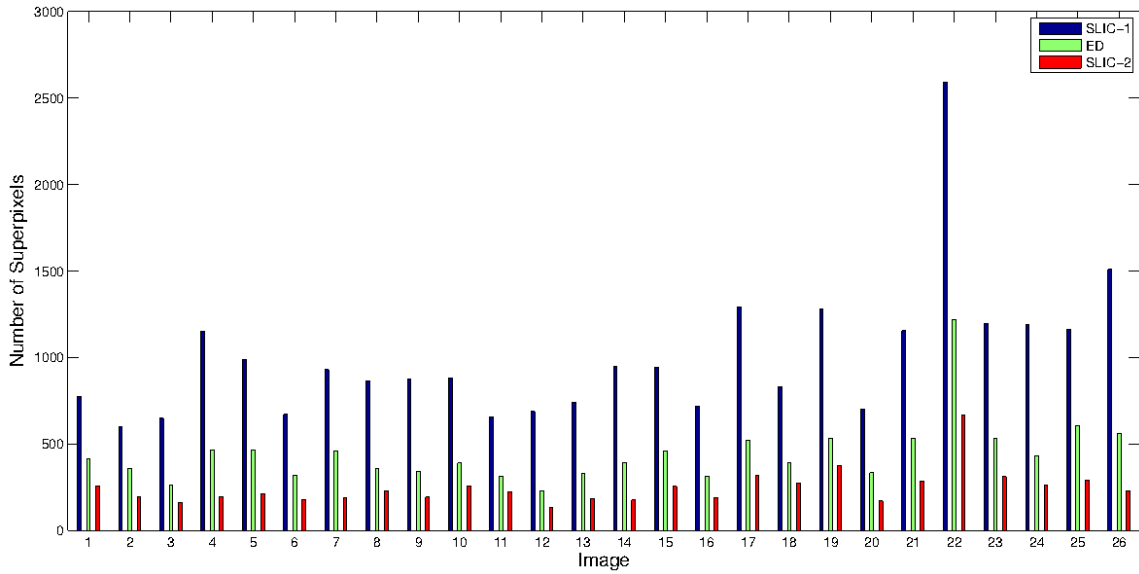


Figure 5.5: Number of superpixels generated for each dataset image. On average, SLIC-1 (blue) produces approximately 1000 segments per image; SLIC-2 (red), 245; and the proposed edge density-based method (green), 443.

adherence to the object borders when configured to generate a few number of segments. On average, our method reaches a BR of approximately 0,73; and SLIC-1 and SLIC-2

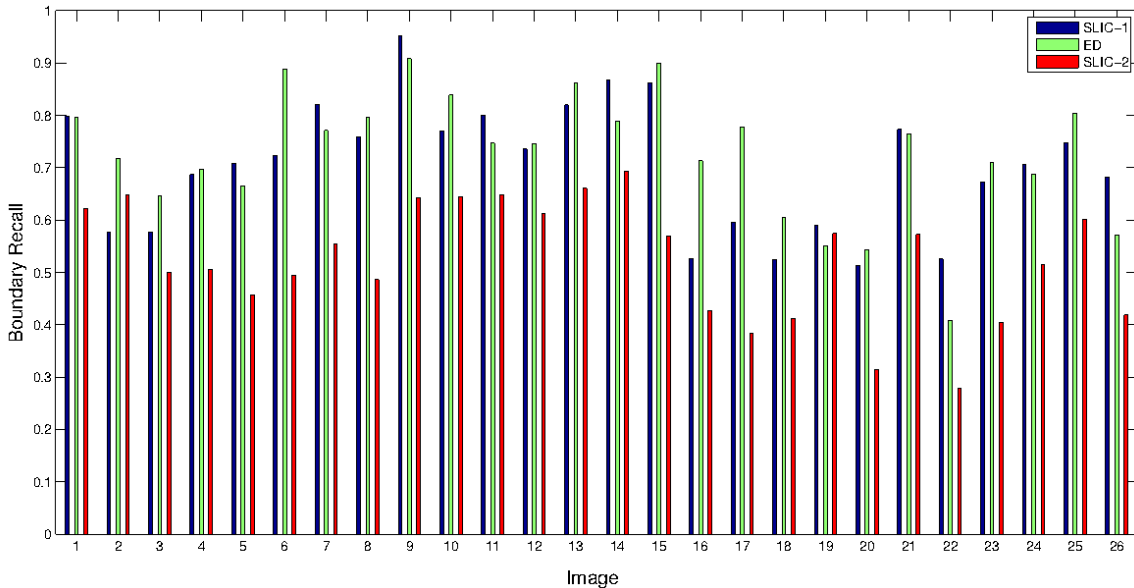


Figure 5.6: Boundary recall (BR). On average, SLIC-1 (blue) provides a BR of approximately 0,7; SLIC-2 (red), 0,53; and the proposed edge density-based method (green), 0,73.

reach, respectively, 0,7 and 0,53.

Achievable segmentation accuracy (ASA) [den Bergh *et al.* 2013]. It is an upper bound measure that gives the maximum performance when taking superpixels as units for object segmentation, and is computed as

$$ASA(s) = \frac{\sum_k \max_i |s_k \cap g_i|}{\sum_i |g_i|} \quad (5.2)$$

where g_i are the ground truth segments and s_k denotes the output segments of the evaluated method. By this metric, the superpixels are labeled with the label of the ground truth segment which has the largest overlap.

As showed in Figure 5.7, all methods perform very similarly regarding ASA metric. All of them have reached an average of 0,99 for this metric.

Undersegmentation Error (UE) [den Bergh *et al.* 2013]. It measures that a superpixel should not overlap more than one object. The standard formulation is

$$UE(s) = \frac{\sum_i \sum_{k:s_k \cap g_i \neq \emptyset} |s_k - g_i|}{\sum_i |g_i|} \quad (5.3)$$

where g_i and s_k are the same as in Equation 5.2, and $|s_k - g_i|$ indicates the size of the pixel leaks. We then sum the pixel leaks over all the segments and normalize it by the

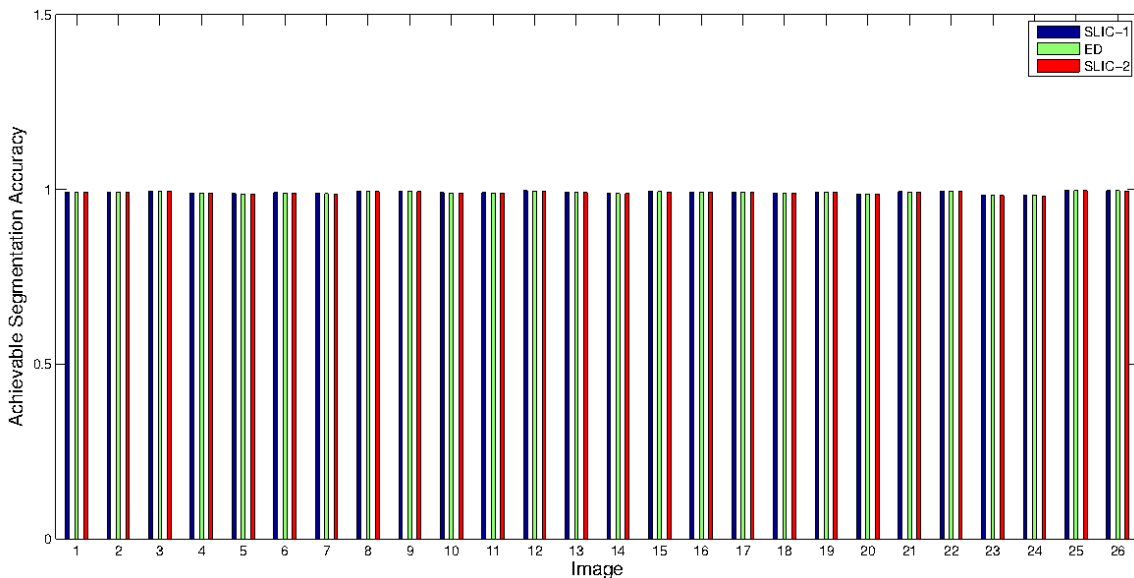


Figure 5.7: Achievable segmentation accuracy (ASA). On average, all evaluated methods — SLIC-1 (blue), SLIC-2 (red) and the proposed edge density-based method (green) — reached approximately 0,99 of ASA.

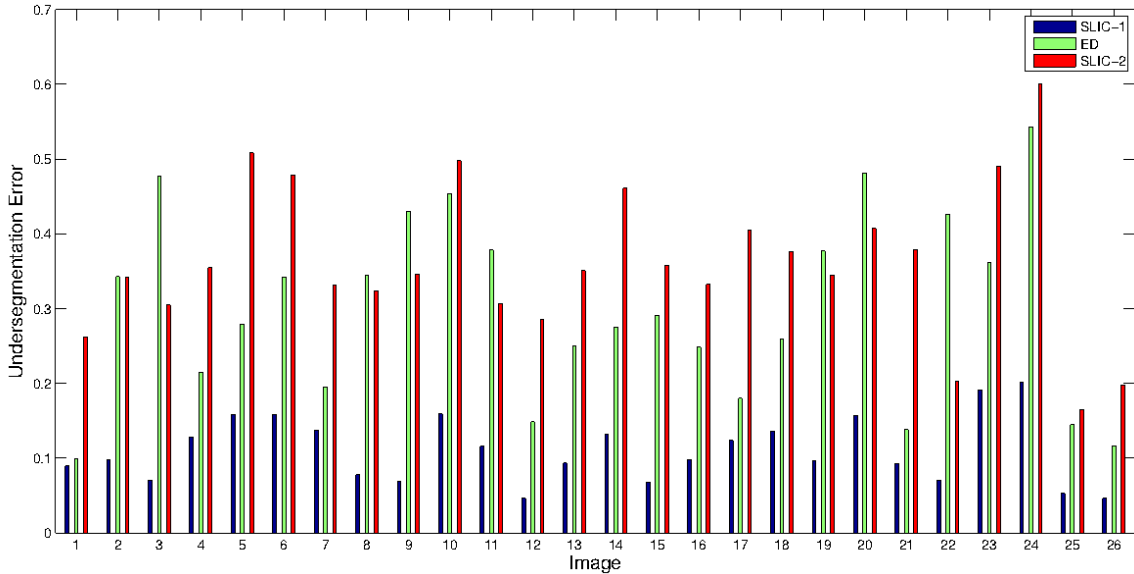


Figure 5.8: Undersegmentation error (UE). On average, SLIC-1 (blue) provides a UE of approximately 0,11; SLIC-2 (red), 0,36; and the proposed edge density-based method (green), 0,30.

image size $\sum_i |g_i|$. This metric has different implementations in the literature, since it is not clear in how to treat the pixels that lie on a border between two labels. We take the metric exactly as used in [Liu *et al.* 2011].

Figure 5.8 shows the UE presented by each method. Differently from BR and ASA metrics, the lower the UE, the better the segmentation result. In this case, our method and SLIC-2 perform significantly poorer than SLIC-1. The results show an average of 0,11 of UE for SLIC-1, 0,30 for ED and 0,36 for SLIC-2. These results can be explained by considering that UE metric compares segment areas to measure to what extend superpixels flood over the ground truth segment borders. Obviously, methods that generate big segments, such as ours and SLIC in the second configuration, are more prone to generate higher UE than methods that only produce small superpixels, as SLIC-1.

Figure 5.9 shows some segmentations provided by each evaluated method. Notice how the proposed method is the best in avoiding oversegmentation of road regions, while ensuring a good adherence to the object borders.

With respect to the processing time, although SLIC is faster than our method, both methods run very quickly. On average, SLIC takes 27 milliseconds per image, while our method takes 52 milliseconds per image. It is further significant if we take into account that these algorithms are implemented in MATLAB. Moreover, as SLIC, our method is $O(N)$ complex. This is reasonably better than the graph-based method discussed in

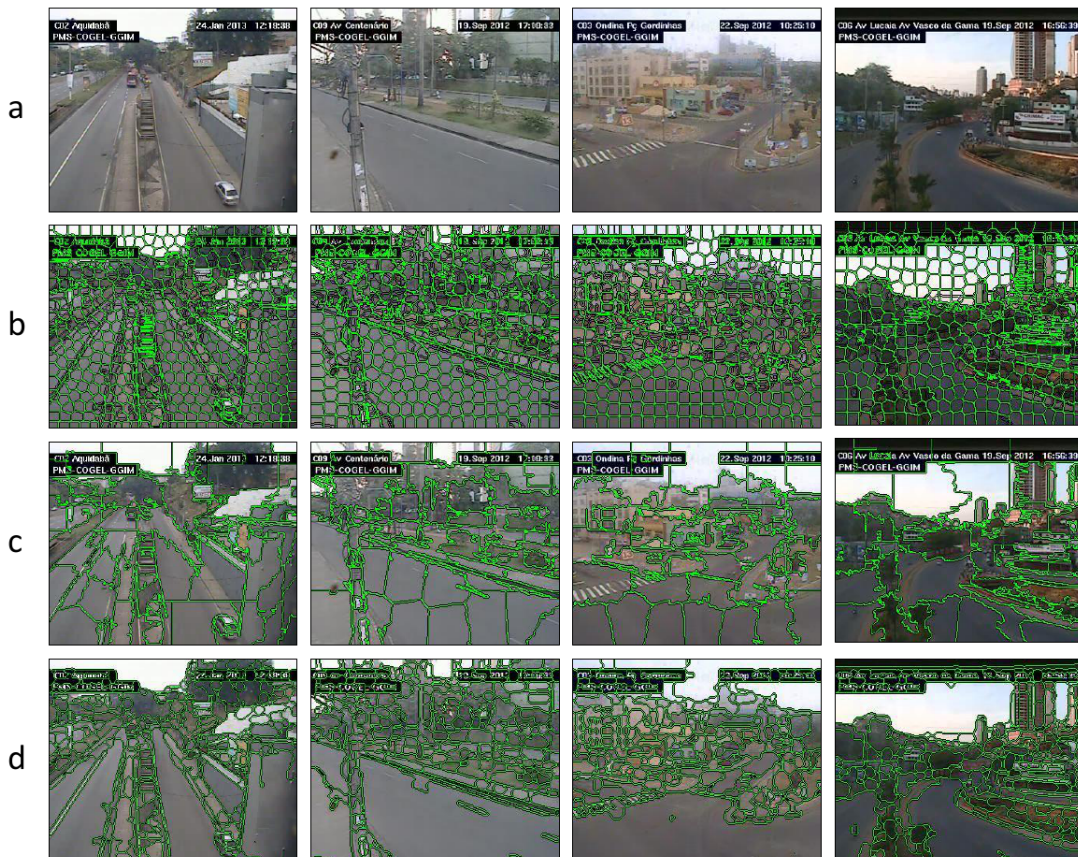


Figure 5.9: Examples of superpixel segmentation: (a) input images; (b), (c) and (d) are segmentation results generated by, respectively, SLIC-1, SLIC-2 and the proposed edge density-based algorithm.

Section 2.4.2, which, according to the authors, is $O(N \log N)$ complex [Felzenszwalb & Huttenlocher 2004].

5.1.3 Road detection performance

In this section we evaluate the performance of our method in road detection. Unfortunately, none of the related works that we have discussed make their code or dataset available for comparison. Thus we decided to compare our method with its earlier iteration, presented in [Santos *et al.* 2013].

In that earlier version, although the general design of the method is the same, some improvements had not been developed yet. It is the case of the background modeling process, which in the earlier version was accomplished by an edge-based approach. Furthermore, the actual version presents improvements in superpixel technique, mainly due

to the use of LUT to extract the contour of the objects. Finally, instead of the actually used decision tree structure, the earlier version was based on a SVM classifier.

To assess the performance of the method we perform a pixel-wise comparison of the generated road masks with the ground truth. From this comparison, we compute the following metrics:

Accuracy. This measure is the overall correctness of the model and is calculated as the sum of correct classifications divided by the total number of classifications

$$\frac{tp + tn}{tp + tn + fp + fn} \quad (5.4)$$

where tp , tn , fp and fn denote the rates of, respectively, true positive, true negative, false positive and false negative.

Precision is a measure of the accuracy provided that a specific class has been predicted. It is defined by

$$\frac{tp}{tp + fp} \quad (5.5)$$

Recall is a measure of the ability of a prediction model to select instances of a certain class from a data set. It is also called sensitivity, and corresponds to the true positive rate. It is defined by

$$\frac{tp}{tp + fn} \quad (5.6)$$

Table 5.2 compares the results presented by our method (actual version) with that found in [Santos *et al.* 2013] (earlier version). Moreover earlier version was evaluated under four different SVM configurations: using polynomial (degree 2 and 3), linear and radial basis function (RBF) kernels.

According to the results, the performance of the proposed method was significantly improved in both accuracy and precision; on the other hand, recall decreased in compar-

Table 5.2: Performance evaluation in road detection

Classifier	Accuracy	Precision	Recall
SVM Polynomial-3	0.65	0.76	0.81
SVM Polynomial-2	0.67	0.89	0.74
SVM Linear	0.66	0.71	0.90
SVM RBF	0.70	0.76	0.89
Decision Tree-based	0.75	0.90	0.82

ison to SVM-based version with linear and RBF kernels. In Figure 5.10 some examples of these results are showed, including quasi-perfect segmentations (5.10.b), results with some errors (5.10.d) and results obtained from very challenging scenes (5.10.f), that is, scenes with poorly-structured roads and high occlusion.

With respect to runtime speed, the evaluation was performed by taking 2000 frames per video, for each video of our dataset (see Section 3.4). The entire dataset was processed in approximately 36 minutes, which gives an average of 42 milliseconds per frame³.

5.2 Analysis and closure

Along with the description of our road detection method, several innovative concepts and techniques were introduced aiming at solving specific problems on image processing. In this chapter a thorough evaluation of those solutions was accomplished in order to assess their performance in practice.

The experiments have pointed to positive results. Particularly, regarding the background modeling, CRON proved to be the most stable among the evaluated methods, being the only one able to deal with the problem of motionless foreground objects. Likewise, the proposed edge density-based superpixel algorithm was successful in providing a good trade-off between matching with object shapes and computational complexity. When compared to the state-of-the-art SLIC, our superpixel method was superior in boundary recall, even generating less than half of the segments produced by SLIC.

Finally, the system comprised of these methods appeared to be a promising approach to perform road detection. Experiments accomplished in the last section revealed interesting results: 75% of accuracy, 90% of precision and 82% of recall. Such results are even more significant if one takes into account the difficulties imposed by the dataset. Furthermore it is noteworthy the fact that the system runs really fast, taking just over 50 milliseconds to process each frame.

³Although it could seem strange the entire road detection process spending less time (42ms) than superpixel generation (52ms, as mentioned in Section 5.1.2), it must be noticed that superpixel generation is performed only once per video instead of for each frame.

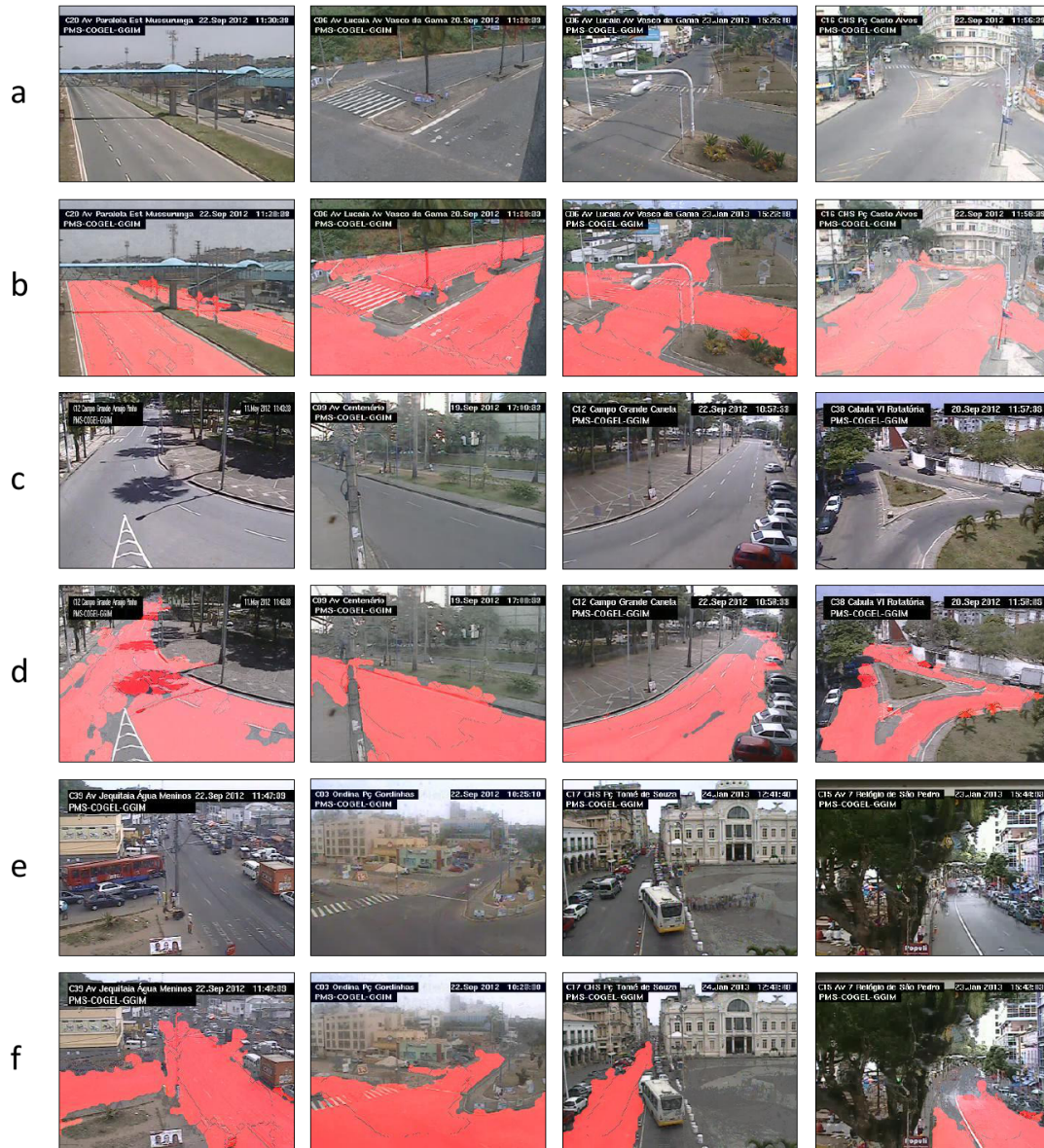


Figure 5.10: Examples of road detection results generated by the proposed method: (a), (c) and (e) are the input images; (b) quasi-perfect results; (d) results with some errors; and (f) results on very challenging scenes.

Conclusion

This work presented a novel approach to perform road detection in traffic surveillance images. The proposed method was designed to work in challenging urban scenarios, including poorly-structured roads, with different shapes. For that, several problems concerning computer vision had to be addressed, sometimes requiring completely new solutions.

Along with this work, we have conceived a new approach for background modeling in traffic videos, so called context-supported road information (CRON). It showed to be a very stable method, producing reliable background models even in presence of occlusions caused by moving objects stopped in the scene. Such robustness was reached by means of an adaptive process, which extracts contextual information from the scene in order to preserve the background model consistency.

Other important result of this work was a new superpixel method, which is based on edge information. This algorithm takes advantage from the fact that edges are reliable descriptors of local dissimilarities in an image, so that they are good candidates to superpixel borders. By handling iteratively the amount of edges extracted from an image, the proposed method is then able to generate superpixels with high adherence to object borders in a fast way.

In addition, simple-but-efficient strategies were formulated, allowing us to extract multiple features — color, texture, position relative to the horizon line and vehicle motion occurrence — without processing overhead. This way, we could properly deal with partial lack of features, which ultimately made our road detection method robust to deal with different situations.

Indeed, we were able to perform road detection efficiently by using a simple unsupervised classifier based on a decision tree structure. This is what indicated a thorough evaluation, where our method reached about 75% of accuracy, 90% of precision and 82% of recall over really challenging traffic videos, where in many cases the roads were so poorly-structured and highly occluded that were difficult to be distinguished from the other objects even by humans.

In the future, we plan to go forward the following directions in order to further enhance the results obtained by our method:

For short term, to investigate the use of supervised probabilistic classifiers, as for instance the Bayesian classifier;

For long term, to design new features, with illumination-invariant properties, in order to reduce the effects of lighting changes on the detection performance, as well as to enable the system operation also at night conditions.

Implementation details

This appendix brings complementary details about the road detection method proposed in this work. Basically, we provide parameter values used at the experiments. The goal is to facilitate eventual attempts of reproducing the method exactly as described.

A.1 Parameters of CRON

Approximated median parameters were defined as $u = 0,002$ (Equation 4.1) and $\rho = 0,098$ (Equation 4.2).

Vehicle filtering blobs were accepted in each filtering criterion when presenting solidity $> 65\%$, aspect ratio $> 70\%$ and orientation < 65 degrees.

Road color-based adaptive analysis was performed by means of a weighted mean (Algorithm 1), calculated as follows. Let R be a matrix of counters for vehicle motion occurrence at each pixel p in the image, G the corresponding matrix containing the gray level of these pixels, N the total of pixels and $N_{(R \neq 0)}$ the number of pixels where motion occurrence is not null. Then, the road color is estimated by doing

$$roadColor = \frac{\sum_{i=1}^N R(p_i) \cdot G(p_i)}{N_{(R \neq 0)}}. \quad (\text{A.1})$$

A.2 Parameters of the superpixel method

Edge density filter size was initialized as $S = 9$, and decremented by 2 at each new iteration, to the minimum size of 3.

Low and high thresholds for edge detection were defined as 0,05 and 0,175, respectively, with step of 0,025.

The morphological operations applied at the post-processing phase (Algorithm 2) were area opening — to remove isolated particles smaller than 25 pixels of area — and spur — to remove branches with any size, inside the segments.

References

- [Achanta *et al.* 2012] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Susstrunk. *SLIC Superpixels Compared to State-of-the-Art Superpixel Methods*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 34, no. 11, pages 2274–2282, 2012.
- [Alvarez *et al.* 2008] J.M. Alvarez, A. Lopez and R. Baldrich. *Illuminant-invariant model-based road segmentation*. In Intelligent Vehicles (IV), IEEE Symposium on, pages 1175–1180, 2008.
- [Bar Hillel *et al.* 2012] Aharon Bar Hillel, Ronen Lerner, Dan Levi and Guy Raz. *Recent progress in road and lane detection: a survey*. Machine Vision and Applications, pages 1–19, 2012.
- [Borenstein 1987] Johann Borenstein. *The nursing robot system*. PhD thesis, Technion – Israel Institute of Technology, 1987.
- [Buch *et al.* 2011] N. Buch, S.A. Velastin and J. Orwell. *A Review of Computer Vision Techniques for the Analysis of Urban Traffic*. Intelligent Transportation Systems, IEEE Transactions on, vol. 12, no. 3, pages 920–939, 2011.
- [Chung *et al.* 2004] Yun-Chung Chung, Jung-Ming Wang, Chyang-Lih Chang and Sei-Wang Chen. *Road segmentation with fuzzy and shadowed sets*. In Computer Vision (ACCV), 6th Asian Conference on, pages 294–299, 2004.
- [den Bergh *et al.* 2013] Michael Van den Bergh, Xavier Boix, Gemma Roig and Luc J. Van Gool. *SEEDS: Superpixels Extracted via Energy-Driven Sampling*. CoRR, vol. abs/1309.3848, 2013.
- [El Baf *et al.* 2008] F. El Baf, T. Bouwmans and B. Vachon. *Fuzzy integral for moving object detection*. In Fuzzy Systems (FUZZ-IEEE), IEEE International Conference on, pages 1729–1736, 2008.
- [Felguera-Martin *et al.* 2012] D. Felguera-Martin, J.T. Gonzalez-Partida, P. Almorox-Gonzalez and M. Burgos-Garcia. *Vehicular Traffic Surveillance and Road Lane Detection Using Radar Interferometry*. Vehicular Technology, IEEE Transactions on, vol. 61, no. 3, pages 959–970, 2012.
- [Felzenszwalb & Huttenlocher 2004] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. *Efficient Graph-Based Image Segmentation*. Int. J. Comput. Vision, vol. 59, no. 2, pages 167–181, September 2004.
- [Feris *et al.* 2011] R. Feris, J. Petterson, B. Siddiquie, L. Brown and S. Pankanti. *Large-scale vehicle detection in challenging urban surveillance environments*. In Applications of Computer Vision (WACV), IEEE Workshop on, pages 527–533, Jan 2011.

- [Gaetano *et al.* 2011] R. Gaetano, J. Zerubia, G. Scarpa and G. Poggi. *Morphological road segmentation in urban areas from high resolution satellite images*. In Digital Signal Processing (DSP), 17th International Conference on, pages 1–8, 2011.
- [Gao *et al.* 2011] Tao Gao, Zhengguang Liu, Shiguo Lian, Shihong Yue and Jun Zhang. *Crossing road monitoring system based on adaptive decision for illegal situation*. Applied Soft Computing, vol. 11, no. 7, pages 4399 – 4412, 2011.
- [Gutchess *et al.* 2001] D. Gutchess, M. Trajkovics, E. Cohen-Solal, D. Lyons and A. K. Jain. *A background model initialization algorithm for video surveillance*. In Computer Vision (ICCV), 8th IEEE International Conference on, volume 1, pages 733–740 vol.1, 2001.
- [He *et al.* 2004] Yinghua He, Hong Wang and Bo Zhang. *Color-based road detection in urban traffic scenes*. Intelligent Transportation Systems, IEEE Transactions on, vol. 5, no. 4, pages 309–318, 2004.
- [Helala *et al.* 2012] M.A. Helala, K.Q. Pu and F.Z. Qureshi. *Road Boundary Detection in Challenging Scenarios*. In Advanced Video and Signal-Based Surveillance (AVSS), IEEE 9th International Conference on, pages 428–433, 2012.
- [Herdtweck & Wallraven 2010] Christian Herdtweck and Christian Wallraven. *Horizon Estimation: Perceptual and Computational Experiments*. In Applied Perception in Graphics and Visualization, Proceedings of the 7th Symposium on, APGV '10, pages 49–56, New York, NY, USA, 2010. ACM.
- [Huang 2010] Lili Huang. *Real-time multi-vehicle detection and sub-feature based tracking for traffic surveillance systems*. In Informatics in Control, Automation and Robotics (CAR), 2nd International Asia Conference on, volume 2, pages 324–328, 2010.
- [Hwang *et al.* 2010] Ju-Won Hwang, Young-Seol Lee and Sung-Bae Cho. *Hierarchical Probabilistic Network-Based System for Traffic Accident Detection at Intersections*. In Ubiquitous Intelligence Computing and 7th International Conference on Autonomic Trusted Computing (UIC/ATC), 7th International Conference on, pages 211–216, Oct 2010.
- [Illingworth & Kittler 1988] J. Illingworth and J. Kittler. *A survey of the Hough transform*. Comput. Vision Graph. Image Process., vol. 44, no. 1, pages 87–116, August 1988.
- [Jendzurski & Paulter 2009] John Jendzurski and Nicholas G. Paulter. *Calibration of Speed Enforcement Down-The-Road Radars*. Journal of Research of the National Institute of Standards and Technology, vol. 114, page 137, 05 2009.
- [Kastrinaki *et al.* 2003] V. Kastrinaki, M. Zervakis and K. Kalaitzakis. *A survey of video processing techniques for traffic applications*. Image and Vision Computing, vol. 21, pages 359–381, 2003.

- [Kim *et al.* 2004] Kyungnam Kim, T.H. Chalidabhongse, D. Harwood and L. Davis. *Background modeling and subtraction by codebook construction*. In Image Processing (ICIP), International Conference on, volume 5, pages 3061–3064 Vol. 5, 2004.
- [Kuhnl *et al.* 2011] T. Kuhnl, F. Kummert and J. Fritsch. *Monocular road segmentation using slow feature analysis*. In Intelligent Vehicles (IV), IEEE Symposium on, pages 800–806, 2011.
- [Lai & Yung 2000] A. H S Lai and N. H C Yung. *Lane detection by orientation and length discrimination*. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 30, no. 4, pages 539–548, 2000.
- [Lee *et al.* 2007] Jong Taek Lee, M. S. Ryoo, M. Riley and J.K. Aggarwal. *Real-time detection of illegally parked vehicles using 1-D transformation*. In Advanced Video and Signal Based Surveillance (AVSS), IEEE Conference on, pages 254–259, Sept 2007.
- [Li & Zhong 2009] Zongmin Li and Liangliang Zhong. *An efficient framework for detecting moving objects and structural lanes in video based surveillance systems*. In Pervasive Computing (JCPC), Joint Conferences on, pages 355–358, 2009.
- [Li *et al.* 2009] Qingquan Li, Bo Lei, Yang Yu and Rui Hou. *Real-time highway traffic information extraction based on airborne video*. In Intelligent Transportation Systems (ITSC), 12th International IEEE Conference on, pages 1–6, 2009.
- [Liu & Wang 2009] Juncheng Liu and Miaoxin Wang. *An Approach for Lane Segmentation in Traffic Monitoring Systems*. In Multimedia Information Networking and Security (MINES), International Conference on, volume 1, pages 239–243, 2009.
- [Liu & Wang 2010] Juncheng Liu and Miaoxin Wang. *Lane segmentation in traffic monitoring systems based on probability map*. In Intelligent Control and Automation (WCICA), 8th World Congress on, pages 6245–6249, 2010.
- [Liu *et al.* 2011] Ming-Yu Liu, O. Tuzel, S. Ramalingam and R. Chellappa. *Entropy rate superpixel segmentation*. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 2097–2104, June 2011.
- [Luo *et al.* 2011] D. Luo, J. Zhao, X. Li and Z. Wang. An extraction method of road structure for surveillance video, chapitre 151, pages 1533–1542. 2011.
- [Maddalena & Petrosino 2008] L. Maddalena and A. Petrosino. *A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications*. Image Processing, IEEE Transactions on, vol. 17, no. 7, pages 1168–1177, 2008.
- [Mazaheri & Mozaffari 2011] M. Mazaheri and S. Mozaffari. *Real time adaptive background estimation and road segmentation for vehicle classification*. In Electrical Engineering (ICEE), 19th Iranian Conference on, pages 1–1, 2011.

- [McFarlane & Schofield 1995] N.J.B. McFarlane and C.P. Schofield. *Segmentation and tracking of piglets in images*. Machine Vision and Applications, vol. 8, no. 3, pages 187–193, 1995.
- [Melo *et al.* 2006] J. Melo, A. Naftel, A. Bernardino and J. Santos-Victor. *Detection and classification of highway lanes using vehicle motion trajectories*. Intelligent Transportation Systems, IEEE Transactions on, vol. 7, no. 2, pages 188–200, 2006.
- [Murthy 1997] Sreerama K. Murthy. *Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey*. Data Mining and Knowledge Discovery, vol. 2, pages 345–389, 1997.
- [Pan *et al.* 2010] Xinting Pan, Yunlong Guo and Aidong Men. *Traffic Surveillance System for Vehicle Flow Detection*. In Computer Modeling and Simulation (ICCMS), 2nd International Conference on, volume 1, pages 314–318, 2010.
- [Rathinam *et al.* 2008] S. Rathinam, Z. Kim and R. Sengupta. *Vision-Based Monitoring of Locally Linear Structures Using an Unmanned Aerial Vehicle*. Journal of Infrastructure Systems, vol. 14, no. 1, pages 52–63, 2008.
- [Santos *et al.* 2013] M. Santos, M. Linder, L. Schnitman, U. Nunes and L. Oliveira. *Learning to segment roads for traffic analysis in urban images*. In Intelligent Vehicles (IV), IEEE Symposium on, pages 527–532, 2013.
- [Schick *et al.* 2012] A. Schick, M. Fischer and R. Stiefelhagen. *Measuring and evaluating the compactness of superpixels*. In Pattern Recognition (ICPR), 21st International Conference on, pages 930–934, 2012.
- [Shin *et al.* 2006] Wook-Sun Shin, Doo-Heon Song and Chang-Hun Lee. *Vehicle Classification by Road Lane Detection and Model Fitting Using a Surveillance Camera*. JIPS, vol. 2, no. 1, pages 52–57, 2006.
- [Sigari *et al.* 2008] Mohamad Hoseyn Sigari, Naser Mozayani and Hamid Reza Pourreza. *Fuzzy running average and fuzzy background subtraction: concepts and application*. International Journal of Computer Science and Network Security, vol. 8, no. 2, pages 138–143, 2008.
- [Sobral *et al.* 2013] Andrews Sobral, Luciano Oliveira, Leizer Schnitman and Felipe De Souza. *Highway Traffic Congestion Classification Using Holistic Properties*. In Signal Processing, Pattern Recognition and Applications (SPPRA), 10th IASTED International Conference on, 2013.
- [Stauffer & Grimson 1999] Chris Stauffer and W. E L Grimson. *Adaptive background mixture models for real-time tracking*. In Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, volume 2, pages –252 Vol. 2, 1999.

- [Unzueta *et al.* 2012] L. Unzueta, M. Nieto, A. Cortes, J. Barandiaran, O. Otaegui and P. Sanchez. *Adaptive Multicue Background Subtraction for Robust Vehicle Counting and Classification*. Intelligent Transportation Systems, IEEE Transactions on, vol. 13, no. 2, pages 527–540, June 2012.
- [Vijverberg *et al.* 2007] J.A. Vijverberg, N.A.H.M. de Koning, J. Han, P.H.N. de With and D. Cornelissen. *High-Level Traffic-Violation Detection for Embedded Traffic Analysis*. In Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on, volume 2, pages II-793–II-796, April 2007.
- [Westphal & Kessler 1988] R. Westphal and A. Kessler. *35-GHz-Doppler radar for law enforcement agencies in Europe*. In International Microwave Digest (MTT-S), IEEE Symposium on, pages 1031–1033 vol.2, 1988.
- [Wren *et al.* 1997] C.R. Wren, A. Azarbayejani, T. Darrell and A.P. Pentland. *Pfinder: real-time tracking of the human body*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 19, no. 7, pages 780–785, 1997.
- [Yao & Odobez 2007] Jian Yao and J. Odobez. *Multi-Layer Background Subtraction Based on Color and Texture*. In Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pages 1–8, 2007.
- [Zhang *et al.* 2003] Chengcui Zhang, Shu ching Chen, Mei ling Shyu and Srinivas Peeta. *Adaptive background learning for vehicle detection and spatio-temporal tracking*. In IEEE Pacific-Rim Conf. on Multimedia (PCM), pages 15–18, 2003.
- [Zhang *et al.* 2008] Hao Zhang, Wen Yu and Xiaowei Sun. *Adaptive Traffic Lane Detection Based on Normalized Power Accumulation*. In Intelligent Transportation Systems (ITSC), 11th International IEEE Conference on, pages 968–973, 2008.
- [Zhou *et al.* 2011] Kai Zhou, K.M. Varadarajan, M. Vincze and Fuqiang Liu. *Driving behavior inference from traffic surveillance data*. In Intelligent Transportation Systems (ITSC), 14th International IEEE Conference on, pages 600–605, Oct 2011.
- [Zivkovic 2004] Z. Zivkovic. *Improved adaptive Gaussian mixture model for background subtraction*. In Pattern Recognition (ICPR), Proceedings of the 17th International Conference on, volume 2, pages 28–31 Vol.2, 2004.



TERMO DE AUTORIZAÇÃO PARA PUBLICAÇÃO DIGITAL NA BIBLIOTECA DIGITAL DA UFBA

1 Identificação do tipo de documento

Tese [] Dissertação [x] Monografia [] Trabalho de Conclusão de Curso []

2 Identificação do autor e do documento

Nome completo: Marcelo Mendonça dos Santos

CPF: 030.892.615-37

Telefone: (71) 9283-9204 e-mail: marceloms@ufba.br

Programa/Curso de Pós-Graduação/Graduação/Especialização: Mestrado em Mecatrônica

Título do documento: Road Detection in Traffic Analysis: A Context-aware Approach

Data da defesa: 17/02/14

3 Autorização para publicação na Biblioteca Digital da UFBA

Autorizo com base no disposto na Lei Federal nº 9.610, de 19 de fevereiro de 1998 e na Lei nº 10.973, de 2 de dezembro de 2004, a Universidade Federal da Bahia (UFBA) disponibilizar gratuitamente sem ressarcimento dos direitos autorais, o documento supracitado, de minha autoria, na Biblioteca Digital da UFBA para fins de leitura e/ou impressão pela Internet a título de divulgação da produção científica gerada pela Universidade.

Texto completo [x] Texto parcial []

Em caso de autorização parcial, especifique a (s) parte(s) do texto que deverão ser disponibilizadas:

Salvador, 15 de 04 de 2014
Local Data

Marcelo Mendonça dos Santos
Assinatura do (a) autor (a) ou seu representante legal

4 Restrições de acesso ao documento

Documento confidencial? [x] Não
[] Sim Justifique: _____

Informe a data a partir da qual poderá ser disponibilizado na Biblioteca Digital da UFBA:

__/__/__ [x] Sem previsão

Assinatura do Orientador: Almir (Opcional)

O documento está sujeito ao registro de patente? Não []
Sim [x]

O documento pode vir a ser publicado como livro? Sim [x]
Não []

Preencher em três vias. A primeira via deste formulário deve ser encaminhada ao Sistema de Bibliotecas da UFBA/Biblioteca Central; a segunda deve ser enviada para a Biblioteca de sua Unidade, juntamente com o arquivo contendo o documento; a terceira via deve permanecer no Programa de Pós-Graduação para o registro do certificado de conclusão do Curso.

Universidade Federal da Bahia
Sistema de Biblioteca da UFBA
Grupo Técnico da Biblioteca Digital da UFBA



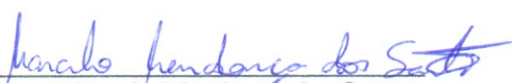
CADASTRO DE INFORMAÇÕES PARA PUBLICAÇÃO DIGITAL
NA BIBLIOTECA DIGITAL DA UFBA

1. Identificação do tipo de material	
Tese () Dissertação (x) Monografia () Trabalho de Conclusão de Curso ()	
2. Colegiado do Curso de Pós-Graduação:	
Título: Road Detection in Traffic Analysis: A Context-aware Approach	
Autor(a): Marcelo Mendonça dos Santos	
CPF: 030.892.615-37	E-mail: marceloms@ufba.br
Orientador(a):	
Nome: Prof. Dr. Luciano Rebouças de Oliveira	
CPF: 673.235.295-49	E-mail: lreboucas@ufba.br
Co-Orientadores	
Nome:	
CPF:	E-mail
Membros da Banca	
Nome: Prof. Dr. Luciano Rebouças de Oliveira (Orientador)	
CPF: 673.235.295-49	E-mail: lreboucas@ufba.br
Nome: Perfilino Eugênio Ferreira Júnior	
CPF: 565.450.885-72	E-mail: perfeuge@ufba.br
Nome: Luiz Gustavo Mirisola	
CPF: 251.154.508-01	E-mail: luiz.mirisola@gmail.com
Nome:	
CPF:	E-mail
Data de Homologação Pós Graduação:	
Financiadores:	
Data:	
Assinatura:	

Salvador, 15/04/2014

DECLARAÇÃO

Declaro para os devidos fins que o texto final apresentado para a conclusão do meu curso de Mestrado em Mecatrônica da Universidade Federal da Bahia é de minha autoria. Declaro também que quaisquer informações utilizadas neste texto, mas que sejam provenientes de outros trabalhos tem fonte claramente expressa e, quando for o caso, foram devidamente autorizadas pelo(s) respectivo(s) autor(es).


Nome: Marcelo Mendonça dos Santos
CPF: 030.892.615-37