Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Programa de Pós-Graduação em Informática

Rodrigo Ferreira Berriel

Vision-Based Ego-Lane Analysis System: dataset and algorithms

Vitória, ES

2016

Rodrigo Ferreira Berriel

Vision-Based Ego-Lane Analysis System: dataset and algorithms

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Informática.

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Programa de Pós-Graduação em Informática

Orientador Prof. Dr. Thiago Oliveira dos Santos

Co-orientador Prof. Dr. Edilson de Aguiar

Vitória, ES

2016

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Setorial Tecnológica,
Universidade Federal do Espírito Santo, ES, Brasil)

B533v

Berriel, Rodrigo Ferreira, 1992-
Vision-based ego-lane analysis system : dataset and algorithms / Rodrigo Ferreira Berriel. – 2016.
100 f. : il.

Orientador: Thiago Oliveira dos Santos.
Coorientador: Edilson de Aguiar.
Dissertação (Mestrado em Informática) – Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Processamento de imagens. 2. Kalman, filtragem de. 3. Banco de dados. 4. Veículos autônomos. 5. Filtro de partículas. 6. Sistemas avançados de assistência ao motorista (ADAS). 7. Faixa de trânsito. I. Santos, Thiago Oliveira dos. II. Aguiar, Edilson de. III. Universidade Federal do Espírito Santo. Centro Tecnológico. IV. Título.

CDU: 004

Vision-Based Ego-Lane Analysis System: dataset and algorithms

Rodrigo Ferreira Berriel

Dissertação submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Mestre em Informática.

Aprovada em 03 de agosto de 2016 por:

Prof. Dr. Thiago Oliveira dos Santos - PPGI/UFES

Prof. Dr. Edilson de Aguiar - CEUNES/UFES

Prof. Dr. Alberto Ferreira De Souza - PPGI/UFES

Prof. Dr. William Robson Schwartz - UFMG/MG

O julgamento dessa dissertação foi realizado com a participação por meio de videoconferência do **membro externo** o Prof. Dr. William Robson Schwartz seguindo as normas prescritas na portaria normativa no. 1/2016. Desse modo, as assinaturas do membro externo são representadas neste documento pelas respectivas assinaturas do presidente da comissão julgadora, o Prof. Dr. Thiago Oliveira dos Santos.

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
Vitória-ES, 03 de agosto de 2016.

To my loving parents, Luci and Jorge; and to my brother, Bruno

Acknowledgements

First off, I would like to thank God, for being on my side along the way at all times. I was never alone!

Secondly, but not least, I would like to thank my mother, Luci, certainly the most important person in my life. Her boundlessly love and incomparable dedication were essential to all achievements of my life, including this one. Thanks!

In memory of my father, Jorge, I would like to say that just like my mother, his everlasting love and wise advices were imperative so that I could keep going. It will be amazing if I come to be like the person you were. I promise to do my best to you!

I would like to thank my brother, Bruno. I know that I can count on you, whenever I come to need.

I must say that it would be much more difficult without the constant guidance and help of my advisor, Prof. Dr. Thiago Oliveira dos Santos. I must also acknowledge the valuable time of my co-advisor, Prof. Dr. Edilson de Aguiar. Thank you both for teaching me and guiding me towards this goal!

I would like to thank Fundação de Amparo à Pesquisa e Inovação do Espírito Santo (FAPES) for the scholarship.

Lastly, I would like to thank all the friends I have made on the LCAD. To those who I had the opportunity work more closely, really thanks! Also, I would like to thank you, reader, for taking your time to read this dissertation.

"I have fought the good fight to the end;
I have run the race to the finish;
I have kept the faith;"
2 Timothy 4:7

Resumo

A detecção e análise da faixa de trânsito são tarefas importantes e desafiadoras em sistemas avançados de assistência ao motorista e direção autônoma. Essas tarefas são necessárias para auxiliar veículos autônomos e semi-autônomos a operarem com segurança. A queda no custo dos sensores de visão e os avanços em *hardware* embarcado impulsionaram as pesquisas relacionadas a faixa de trânsito – detecção, estimativa, rastreamento, etc. – nas últimas duas décadas. O interesse nesse tópico aumentou ainda mais com a demanda por sistemas avançados de assistência ao motorista (ADAS) e carros autônomos. Embora amplamente estudado de forma independente, ainda há necessidade de estudos que propõem uma solução combinada para os vários problemas relacionados a faixa do veículo, tal como aviso de saída de faixa (LDW), detecção de troca de faixa, classificação do tipo de linhas de divisão de fluxo (LMT), detecção e classificação de inscrições no pavimento, e detecção da presença de faixas ajdacentes.

Esse trabalho propõe um sistema de análise da faixa do veículo (ELAS) em tempo real capaz de estimar a posição da faixa do veículo, classificar as linhas de divisão de fluxo e inscrições na faixa, realizar aviso de saída de faixa e detectar eventos de troca de faixa. O sistema proposto, baseado em visão, funciona em uma sequência temporal de imagens. Características das marcações de faixa são extraídas tanto na perspectiva original quanto em images mapeadas para a vista aérea, que então são combinadas para aumentar a robustez. A estimativa final da faixa é modelada como uma *spline* usando uma combinação de métodos (linhas de Hough, filtro de Kalman e filtro de partículas). Baseado na faixa estimada, todos os outros eventos são detectados. Além disso, o sistema proposto foi integrado para experimentação em um sistema para carros autônomos que está sendo desenvolvido pelo Laboratório de Computação de Alto Desempenho (LCAD) da Universidade Federal do Espírito Santo (UFES).

Para validar os algorítmos propostos e cobrir a falta de base de dados para essas tarefas na literatura, uma nova base dados com mais de 20 cenas diferentes (com mais de 15.000 imagens) e considerando uma variedade de cenários (estrada urbana, rodovias, tráfego, sombras, etc.) foi criada. Essa base de dados foi manualmente anotada e disponilizada publicamente para possibilitar a avaliação de diversos eventos que são de interesse para a comunidade de pesquisa (i.e. estimativa, mudança e centralização da faixa; inscrições no pavimento; cruzamentos; tipos de linhas de divisão de fluxo; faixas de pedestre e faixas adjacentes). Além disso, o sistema também foi validado qualitativamente com base na integração com o veículo autônomo. O sistema alcançou altas taxas de detecção em todos os eventos do mundo real e provou estar pronto para aplicações em tempo real.

Palavras-chave: Processamento de Imagens, Análise da Faixa de Trânsito, Sistema de Assistência ao Motorista, Filtro de Partículas, Filtro de Kalman, Base de Dados

Abstract

Lane detection and analysis are important and challenging tasks in advanced driver assistance systems and autonomous driving. These tasks are required in order to help autonomous and semi-autonomous vehicles to operate safely. Decreasing costs of vision sensors and advances in embedded hardware boosted lane related research – detection, estimation, tracking, etc. – in the past two decades. The interest in this topic has increased even more with the demand for advanced driver assistance systems (ADAS) and self-driving cars. Although extensively studied independently, there is still need for studies that propose a combined solution for the multiple problems related to the ego-lane, such as lane departure warning (LDW), lane change detection, lane marking type (LMT) classification, road markings detection and classification, and detection of adjacent lanes presence.

This work proposes a real-time Ego-Lane Analysis System (ELAS) capable of estimating ego-lane position, classifying LMTs and road markings, performing LDW and detecting lane change events. The proposed vision-based system works on a temporal sequence of images. Lane marking features are extracted in perspective and Inverse Perspective Mapping (IPM) images that are combined to increase robustness. The final estimated lane is modeled as a spline using a combination of methods (Hough lines, Kalman filter and Particle filter). Based on the estimated lane, all other events are detected. Moreover, the proposed system was integrated for experimentation into an autonomous car that is being developed by the High Performance Computing Laboratory of the Universidade Federal do Espírito Santo.

To validate the proposed algorithms and cover the lack of lane datasets in the literature, a new dataset with more than 20 different scenes (in more than 15,000 frames) and considering a variety of scenarios (urban road, highways, traffic, shadows, etc.) was created. The dataset was manually annotated and made publicly available to enable evaluation of several events that are of interest for the research community (i.e. lane estimation, change, and centering; road markings; intersections; LMTs; crosswalks and adjacent lanes). Furthermore, the system was also validated qualitatively based on the integration with the autonomous vehicle. ELAS achieved high detection rates in all real-world events and proved to be ready for real-time applications.

Keywords: Ego-Lane Analysis, Lane Estimation, Kalman Filter, Particle Filter, Driver Assistance Systems, Dataset, Image Processing

List of Figures

Figure 1 – Driving scenarios may contain a lot of variation. These variations may occur due to uncountable factors, including (a) weather, (b) road texture, (c) shadows, (d) abrupt illumination change (e.g. tunnels) and (e) pavement markings among many others. 25

Figure 2 – Preprocessing strategies: (a) Region of Interest (RoI), (b) Inverse Perspective Mapping and (c) temporal integration. In most cases, grayscale conversion is also applied. 32

Figure 3 – The Inverse Perspective Mapping attempts to convert from camera plane coordinates to the ground plane coordinates (or world coordinate system). 33

Figure 4 – Different types of crosswalks. 38

Figure 5 – There are different Lane Markings Types (LMT), and they might be in different conditions, such as fading and occluded. 39

Figure 6 – Different types of Arrow Markings. Arrows markings, such as lane markings, can also be fading after years of usage without maintenance. 40

Figure 7 – Road Signs and Crosswalk. Arrow markings in the upper row. Lower row: stop-line (left) and crosswalk (right). These symbols are defined in the Brazilian Manual of Traffic Signs (DENATRAN, 2007) 40

Figure 8 – Overview of the Ego-Lane Analysis System (ELAS). The system processes a sequence of images, one at a time. Firstly, general features are extracted. Subsequently, each module is responsible to a task. Finally, the individual outputs of all modules are combined into the output of the system. 41

Figure 9 – Overview of General Feature Extraction. The input image is preprocessed and general features are extracted in the form of four binary maps, i.e. feature maps. These maps and the processed images are used by the subsequent modules. 42

Figure 10 – Overview of Crosswalk Detection. I^{DOG} is analyzed in order to detect crosswalk candidates in the frame. A region of interest (red) is used to search for houghs (green). If a set of hough lines is detected, the region is rotated using the angle of the houghs. The final crosswalk detection is based on the analyses of the convolution between the projected crosswalk region and its inverted version. 44

Figure 11 – This figure illustrates the dominant angle calculation (Equation 3.3). $H^{2D}_{x,y}$ is represented by the blue bars. Green histogram represents H^{1D}_{i} and α is its maximum. Used in further processing, the red histogram illustrates the projection of the selected Hough lines into the horizontal axis (Equation 3.8). 46

Figure 12 – Road Signs Detection and Classification Overview. The lane region is scanned and trimmed, generating symbol candidates. A two-step scheme is used to classify these candidates: if a symbol candidate has the proportion of a stop-line, then it is classified as one, otherwise the classification is done through a Template Matching scheme. 47

Figure 13 – Previously detected crosswalks and road signs are given as input and this module removes them from the feature maps. The goal is to reduce noise and increase robustness of the lane estimation module. 48

Figure 14 – Overview of the lane measurement generation. The process receives I^{SRF} as input, generates some lane candidates, filters and validates them and outputs a lane measurement (in green). 49

Figure 15 – Lane base estimation overview. Lane measurements are put through a Finite-State Machine. The state machine works like a hysteresis, while the Kalman filter works to smooth the measurements. The lane base estimate showed in dark green is given as output. 51

Figure 16 – The function $W_1^{'}$, described in the Equation 3.9, assigns greater weights to those particles with more evidences close to their derived estimates, even when there are no lane markings in one of the sides. 54

Figure 17 – Particle filter overview. The particle is defined by three values: x_2, x_3 and w_2. During the prediction, the goal is to approximate the particle previous state to the current estimate given by the linear model. On the correction phase, the filter aims to assign for each particle a weight. Roughly, the weight is defined by how precise is the estimate, given a feature map, and how clean is the inner space of the estimated lane. . . 55

Figure 18 – Lane marking types (LMT) as detected and classified according to a threshold-based method. These are the LMTs: white-single-solid (WSS), white-single-dashed (WSD), yellow-single solid (YSS), yellow-single-dashed (YSD), yellow-mixed-solid (YMS), yellow-mixed-dashed (YMD) and yellow-double-solid (YDS). Decision processes (a, b, c, d and e) are described in Section 3.6 56

Figure 19 – Adjacent Lane Detection. The dark blue bars shows the histogram counts for the hough lines generated by the figure at the bottom of the image. The green region is the estimated lane region. The blue region is the searching region calculated based on the lane width of the current estimation. The red region is the space between the ego-lane and the adjacent lane. 57

Figure 20 – *Time-Sliced* images from a single scene. For each scene, four TS images were generated based on the four selected rows to annotate the lane position ground truth. Annotating these images is equivalent of annotating over time. In this figure, TS images were rotated for visualization purposes. 60

Figure 21 – Satellite view of some of the scenes in the dataset. Vitória: (a) BR_S01, (b) VIX_S05 and (c) VIX_S11; Guarapari: (d) GRI_S01 Vila Velha: (e) VV_S02 and (f) VV_S03. A more detailed overview and camera images can be seen at Appendix B. 61

Figure 22 – Camera setup used in the dataset captured using the IARA. In (a), both of the available Bumblebee models. In (b), the GPS antenna that is in front of the chosen camera can be seen, and in (c) an image captured by the left camera from the stereo. 62

Figure 23 – Satellite view of the two highway scenes. The first one (a) is of the 3.1km route and the second (b) is of the 23.2km route. 62

Figure 24 – Lane Position Metric. A variation to the LPD introduced in (SATZODA; TRIVEDI, 2014b) is proposed. The error is computed in the IPM image as the average distance (in red) of the lane estimate (blue lines) to the ground truth (background scene) divided by the lane width. Also, the lane center deviation (in purple) is calculated based on the closest point as the distance between the actual (green circle) and the estimated (blue circle) lane center. 63

Figure 25 – With this technique, quality loss causes a degradation of the image, in special lane markings. From left to right: (a) 100%, (b) 75%, (c) 50% and (d) 25%. 65

Figure 26 – Satellite view of the scenes recorded using IARA: (a) Av. Fernando Ferrari and (b) Av. Nossa Senhora da Penha (also known as Reta da Penha) 66

Figure 27 – Qualitative results on the annotated dataset. 67

Figure 28 – Qualitative results on the dataset recorded using IARA. 67

Figure 29 – The mean absolute error (%) of the near and far region are reported. Additionally, execution time (ms) is shown. In this case, particle filter was tested with 50, 100, 200 e 400 particles (PF50, PF100, PF200 and PF400, respectively). 68
Figure 30 – Frames per Second. ELAS was tested with 30, 20, 15 and 10 frames per second (FPS30, FPS20, FPS15, FPS10, respectively). All these tests were performed using PF400. In fact, FPS30 was reported again for completeness, because it is the same as PF400. 69
Figure 31 – Image Quality. Lower image qualities were simulated based on the image resolutions where 100% (QUA100) represents the original image and the others represent the relative resolutions: 75%, 50% and 25% (QUA75, QUA50, QUA25, respectively). 70
Figure 32 – Comparison between ELAS and Berriel et al. (2015) 70
Figure 33 – Some types of failures of the system, based on naive assumptions and side effects of particular techniques. 71

Figure 34 – Intelligent Autonomous Robotic Automobile — IARA 87
Figure 35 – Some of the sensors of IARA. Blue and red arrows represent the 3 Bumblebee cameras. The red one was the forward-looking camera used as input sensor. The green arrow points to one of the GPS antenna that is in front of the chosen camera. 88
Figure 36 – Sample images of the Viewer3D module. The trail of the lane position is represented by the cyan dots and lines. The actual lane estimation is rendered by the yellow dots and lines. 89

List of Tables

Table 1 – Performance measurement of the classification tasks. 69
Table 2 – Execution Time Performance of ELAS with 400 particles (in ms) 70

List of abbreviations and acronyms

ELAS	Ego-Lane Analysis System
LDW	Lane Departure Warning
ROI	Region of Interest
VP	Vanishing Point
ACO	Ant Colony Optimization
SIR	Sequential Importance Resampling
LiDAR	Light Detection and Ranging
IARA	Intelligent Autonomous Robotic Automobile
LCAD	Laboratório de Computação de Alto Desempenho (High Performance Computing Laboratory)

Contents

1 Introduction . . . 25
1.1 Motivation . . . 26
1.2 Contributions . . . 27
1.3 Structure . . . 28

2 Theoretical Background . . . 31
2.1 Preprocessing . . . 31
2.2 Lane Feature Extraction . . . 34
2.3 Lane Estimation and Tracking . . . 35
2.4 Lane Analysis . . . 38

3 Ego-Lane Analysis System (ELAS) . . . 41
3.1 General Feature Extraction . . . 41
3.1.1 Preprocessing . . . 41
3.1.2 Feature Maps Generation . . . 42
3.2 Crosswalk Detection . . . 44
3.3 Road Signs Detection and Classification . . . 45
3.4 Crosswalk and Road Signs Removal . . . 47
3.5 Lane Estimation . . . 48
3.5.1 Lane Measurement Generation . . . 48
3.5.2 Lane Base Estimation . . . 51
3.5.3 Lane Curvature Estimation . . . 52
3.5.4 Lane Departure Measure . . . 55
3.6 Lane Marking Type (LMT) Classification . . . 56
3.7 Adjacent Lane Detection . . . 57
3.8 Integration into an Autonomous Driving Framework . . . 57

4 Materials and Experimental Methodology . . . 59
4.1 Database . . . 59
4.2 Metrics . . . 63
4.3 Experiments . . . 64
4.3.1 Kalman with/without Particle Filter . . . 65
4.3.2 Image quality . . . 65
4.3.3 Frame rate . . . 65
4.3.4 Qualitative Experiments . . . 66
4.4 Setup . . . 66

5 Results and Discussion **67**
5.1 Qualitative Results 67
5.2 Quantitative Results 68
5.3 Discussion 70

6 Conclusions **73**
6.1 Future Works 73

Publications **75**

Bibliography **77**

Appendix **85**

APPENDIX A Integration into CARMEN **87**

APPENDIX B Annotated Dataset **91**

APPENDIX C Scenes from IARA **95**

APPENDIX D Other Scenes **97**

APPENDIX E Qualitative Results **99**

1 Introduction

> “Begin at the beginning," the King said gravely, “and go on till you come to the end: then stop."
>
> — Lewis Carroll, Alice in Wonderland

Lane detection and analysis are key steps for advanced driver assistance systems (ADAS) and self-driving cars. ADAS and self-driving cars are being investigated for a long time (EUREKA, 1987). While the former intends to help drivers and enhance their efficiency, the latter intends to remove drivers from the equation. Those solutions have been thought because driving is a dangerous and expensive activity for the global community. Traffic accidents were responsible for more than 1.25 million of deaths worldwide in 2015 (WHO, 2015). Human drivers are the main cause of traffic accidents (LUM; REAGAN, 1995), due to inherited limitations (limited field-of-view, tiredness, etc.) or irresponsible actions (drunk driving, traffic infractions, etc.). According to World Health Organization (WHO, 2013), traffic accidents cost about US$1,855 billion worldwide (R$40 billion in Brazil alone (IPEA, 2015)).

Lane is a division of a road marked off with painted lines. It intends to separate single lines of traffic according to speed or direction. Human drivers are able to detect and recognize lanes while driving in an effortless way. Despite the huge research effort spent in the last decades trying to make computers to enclose human performance on this task, computers are not there yet. One of the reasons is that roads and lane markings can be presented in a wide range of scenarios. Figure 1 shows roads with painted lane markings. They are presented varying in texture, color, lightness, relative position, etc. Likewise, they can be captured by different cameras. As a result, lane features are really hard to extract.

(a) (b) (c) (d) (e)

Figure 1 – Driving scenarios may contain a lot of variation. These variations may occur due to uncountable factors, including (a) weather, (b) road texture, (c) shadows, (d) abrupt illumination change (e.g. tunnels) and (e) pavement markings among many others.

Lane detection systems can be clustered in regards to sensors and lane models.

Sensors are equipment used to acquire data for the system. The most common sensors used in the literature are: monocular (SATZODA; TRIVEDI, 2014a) and stereo cameras (BERTOZZI; BROGGI, 1998), LiDAR (WIJESOMA; KODAGODA; BALASURIYA, 2004) and sensor-fusion (HUANG et al., 2008). Each sensor presents its drawbacks and advantages. Camera-based approaches have the advantage of inherently capturing the far-field of vision and, in comparison to the others, they are usually cheaper. As a drawback, cameras are very sensitive to light variation. On the other hand, laser-based approaches have the advantage of better handling lighting variations, such as shadows, dark environments, direct light and sudden illumination change (tunnels) at the cost of being usually more expensive than camera-based ones. Sensor-fusion based approaches get the best of both worlds at the expense of a huge amount of data to process and the high cost of integration of the sensors. Despite the chosen sensor, lane must be modeled somehow. Models are used to represent the lane. The most common models used in the literature are: linear, parabolic and spline-based models. Virtually, any combination of sensors and lane models can be used to tackle the lane detection problem.

Lane analysis, particularly ego-lane analysis, comprises the multiple tasks related to the host lane: lane detection and estimation (LE), lane departure warning (LDW), lane change detection, lane marking type (LMT) classification, pavement markings detection and classification, detection of adjacent lanes, etc. Host lane is the lane where the vehicle is localized. The core task in lane analysis is the lane detection. Ego-lane analysis gives some understanding about the car environment. It allows control systems to make better decisions and help the vehicle to keep itself safe.

1.1 Motivation

Lane related research is an active research topic and there has been an increase of the interest around it. There were several factor that motivated this increase of interest: recent advances in embedded hardware, decreasing costs of sensors, the increase in demand for ADAS technologies and the buzz with self-driving cars. Academia, industries and governments are working towards a solution for safe transportation. In the lane related topics, hand-crafted features are being developed to help solving the lane detection problem. Also, various lane models are being studied to cover all possible lane representations, as reported by (MCCALL; TRIVEDI, 2006) in a recent survey. In the literature, in general, camera-based approaches are gaining more traction over other sensors (SHIN; XU; KLETTE, 2014). The other tasks involved in lane analysis are also being broadly studied.

Although widely studied, there is still need for improvements. Camera-based solutions often suffers from its major drawback: light variation. It is not different in lane related research. As these algorithms and systems are intended to be used in critical

systems, real-time operation is a requirement. Also, as the lane analysis tasks are highly correlated and somehow dependent, there is still need for studies that propose a combined solution. Besides that, as (HILLEL et al., 2014) reported in recent survey, there is a need for public datasets to enable fairer comparisons and boost advances in this topic.

Therefore, given the potential impact of lane analysis systems and their applications, we were motivated to investigate vision-based algorithms that could tackle the actual limitations. Inspired by the fact the human drivers mostly use vision to perform this task, we want to investigate a vision-based system that could perform these tasks with a high accuracy among them all.

Besides these factors, one of the motivations for this work was the research that is being conducted in the High Performance Computing Laboratory (LCAD) about autonomous vehicles. In our laboratory, we are developing an autonomous car: IARA (Intelligent Autonomous Robotic Automobile). One of the goals of IARA is to go from our university to a neighbor city autonomously: it comprises a more than 50km route that goes through urban cities and highways producing a diversity of scenarios. The work hereby presented could also help our autonomous vehicle to achieve this goal.

1.2 Contributions

This dissertation presents a system (ELAS) for ego-lane analysis. The system is based on a combination of methods (Hough lines, Kalman filter and Particle filter) and estimates the ego-lane of the vehicle. The system requires a sequence of frames as input, and processes one at a time. An inverse perspective mapping is applied and features are extracted from both perspectives. After that, lane candidates are extracted with the use of Hough transform. Then, the candidates are filtered using a mechanism based on a two dimensional histogram, where there main goal is to filter the candidates based on their dominant angle. The filtered candidates are given to a buffer mechanism, that ensures temporal coherence, and to state machine that controls a Kalman filter, responsible to smooth the estimates. Finally, a spline-based Particle filter is used to estimate the curvature of the ego-lane. Besides that, the system is able to detect and classify road markings (e.g. arrows and crosswalks), measure lane departure, detect lane changes, classify lane markings type and detect the presence of adjacent lanes. The proposed system is able to perform all these tasks in real-time using a standard PC. A study about the robustness of the system simulating different frame rates and sensor inputs of different qualities was conducted. Moreover, the proposed system was integrated into a real autonomous vehicle: IARA (Intelligent Autonomous Robotic Automobile) from LCAD (High Performance Computing Laboratory), where it is meant to be used. Although ELAS is completely independent of the autonomous vehicle, its robustness could be improved if other sensors were fused.

Additionally, a novel dataset with more than 20 scenes (15,000+ frames) was generated, manually annotated and made publicly available to enable future fair comparisons. Also, the implementation of ELAS is publicly available.

Summarizing, the main contributions are:

- A system that is able to properly estimate the lane in real-time with high accuracy in terms of positioning. The system combines linear and spline-based model to accurately estimate the lane position. For the linear part, a combination of Hough transform, two-dimensional histogram and Kalman filter is used. For the curved part, a spline-based Particle filter is used;
- A study of the robustness of the proposed system regarding some technical aspects, such as frame rate and image quality;
- An overview of the integration process into a real autonomous vehicle, IARA;
- A public dataset containing more than 15,000 frames of Brazilian roads that were manually annotated for the events of interest to the research community.;
- Both, source code and datasets are publicly available at: http://www.lcad.inf.ufes.br/wiki/index.php/Ego-Lane_Analysis_System

1.3 Structure

The structure of this dissertation is as follows:

- After this introduction, Chapter 2 presents a brief review of the literature about common steps, such as: Preprocessing, Lane Feature Extraction, Lane Estimation and Tracking and Lane Analysis. A general overview is given, but most of the related works thereby discussed use techniques closely related with those used in this work;
- Chapter 3 introduces a novel real-time Vision-Based Ego-Lane Analysis System. A combination of methods (Hough lines, Kalman Filter and Particle filter) is used to perform lane estimation. The lane is modeled by a spline allowing the proper estimation of various lane models. Also, it describes the integration process of this system into a framework used in a real autonomous vehicle;
- In the Chapter 4, a novel dataset is presented. This dataset comprises over 20 scenes (in more than 15,000 frames), where the main features and events of interest were manually annotated. Besides that, the experimental setup is described;
- Chapter 5 reports the results achieved by the proposed system, describes a study of its robustness regarding some aspects (e.g. frame rate and input image quality),

discusses the difficulties of experimentation in this topic and presents the limitations of the proposed system. Furthermore, qualitative results of the integration process are shown;

- Chapter 6 outlines key points, presents the conclusions and indicates directions for future developments.

2 Theoretical Background

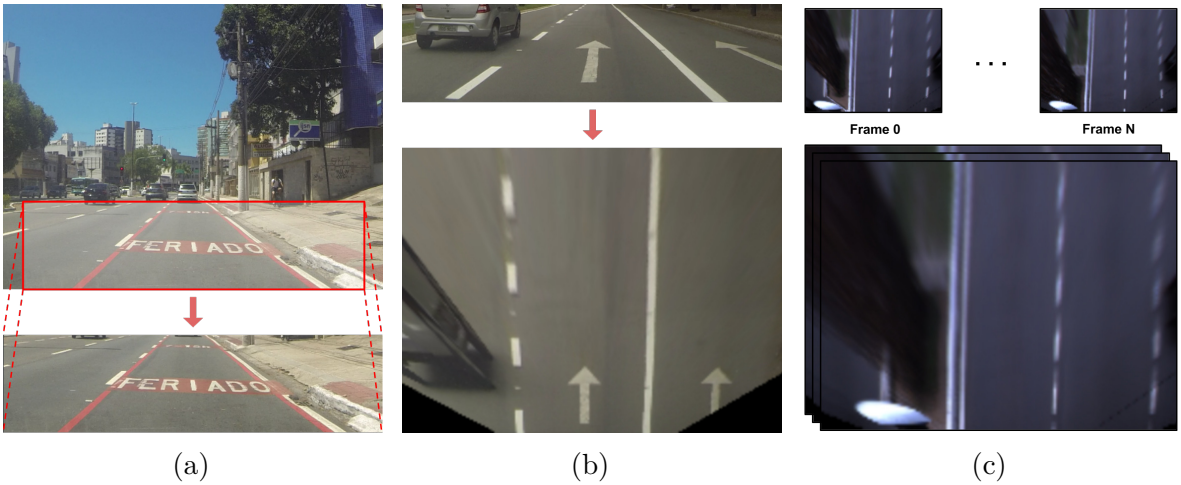
This chapter contains the theoretical background related to this work. In general, the so-called lane detection algorithms usually share a common pipeline (YENIKAYA; YENIKAYA; DÜVEN, 2013): preprocessing, feature extraction, lane detection, model fitting, and tracking. Therefore, firstly, the literature around preprocessing techniques and feature extraction for lane is presented. After that, two key tasks are reviewed: lane estimation (which comprises lane detection and model fitting) and tracking. Finally, the lane analysis problem is described alongside with an overview of the most recent developments in the topic. Most of the related works hereby referenced are vision-based approaches, just like the proposed work.

2.1 Preprocessing

There is virtually an unlimited number of scenarios in which people can drive a vehicle. Each scene itself may vary based on any combination of landscape, position of the sun (time of the day), weather, occlusions, shadows, fog, road texture, etc. Also, besides the fact that images can be captured by different cameras (which can lead to different field-of-view, aperture, ISO, etc.), images can also be captured from different camera positions, direction and rotations. Besides that, in the most common camera pose (forward-looking mounted in the top of a vehicle), there is a lot of irrelevant content (i.e. sky, building, etc.) to the problem in the captured image. Due to these variations, it is a very common strategy to preprocess the input image and further apply a feature extraction technique.

Preprocessing strategies have been applied to road images in order *i)* to reduce the region to be processed, avoiding clear mistakes at the same time of increasing computational performance; *ii)* to correct the perspective distortion, allowing some techniques that would not work otherwise; and *iii)* to increase reliability of the incoming data, assuming lane markings are not expected to change abruptly in terms of spatiality when compared to other objects in a scene. Furthermore, a conversion to grayscale is usually applied in order to reduce the amount of data to be processed.

In the first case (Figure 2a), it is assumed that a given input image has a region of interest (RoI). A RoI comprises mainly the road, discarding sky, buildings, parts of the host car and other less relevant contents. The definition of a RoI allows for increased computational performance, since there will be less data to process, and avoids clear mistakes, as irrelevant objects may represent or add noise that will be further processed. The selection of the RoI can be done statically (LI et al., 2014) or dynamically (SATZODA;



(a) (b) (c)

Figure 2 – Preprocessing strategies: (a) Region of Interest (RoI), (b) Inverse Perspective Mapping and (c) temporal integration. In most cases, grayscale conversion is also applied.

TRIVEDI, 2013b). In the former, there is a-prior knowledge of the image captured by the camera (or the camera pose), then the region is set based on this knowledge. In the latter, processing techniques are applied in order to determine this region. These techniques vary from sky detection (ZUO; YAO, 2013), horizon analysis (WANG; SHEN; TEOH, 2000), vanishing point estimation (RASMUSSEN, 2004) and others. Although it may sounds better to have a dynamic selection, the processing time spent on this task may be prohibitive for real-time systems. Also, in a real system, camera pose is expected to be known (even though a calibration may be required after its installation, or it could be defined in the manufacturing process). Therefore, defining the region of interest beforehand can be easily done at a low cost.

In the second case (Figure 2b), often the perspective distortion is considered an additional noise that must be removed or softened. The image captured by a forward-looking camera suffers inherently from a perspective distortion. This distortion is responsible for the breaking of a variety of assumptions: parallel lines (such as lane markings) do not appear parallel, constant width elements (such as lane markings width) present a varying width proportional to the distance from the camera and others. Many of the lane detection algorithms work better under the assumption of a flat road, with parallel lanes that contains constant width. Based on this assumption, an Inverse Perspective Mapping (IPM) (Eq. 2.1) is applied (Figure 3). The IPM is a retinotopic mapping applied to the image and, in most cases, assumes a fixed flat road. The result of the IPM is often called "bird's eye view". To apply the IPM, a 3×3 homography matrix H has to be defined:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.1}$$

where $H = [h_{ij}]$ represents the homography. In the literature, some authors (BÓDIS-

SZOMORÚ; DABÓCZI; FAZEKAS, 2007) calculate H offline, based on a simple calibration; and other authors (PAULA; JUNG, 2015) calculate the matrix on-the-fly. This online calculation of the homography matrix also requires extra processing time which cannot always be done to achieve real-time operation. Automatic calibration allowing the proper calculation of H has been explored by (PAULA; JUNG, 2015), but it demands a specific driving sequence, which may not always be available or possible to be acquired.

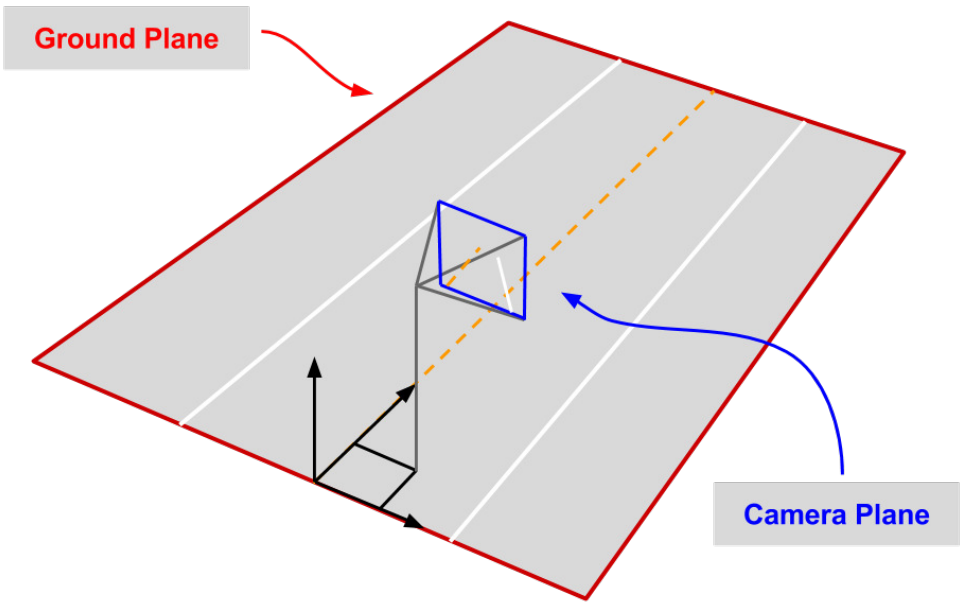


Figure 3 – The Inverse Perspective Mapping attempts to convert from camera plane coordinates to the ground plane coordinates (or world coordinate system).

In the third case (Figure 2c), a set of previous frames are averaged. As lane markings are assumed not to change abruptly, mainly when compared to other objects (other vehicles, pedestrians, etc.), a reasonable amount of previous frames is averaged. It is expected that this procedure will enhance lane markings and weaken moving objects, decreasing the effect of possible noise sources in the further process. Likewise, this technique is also applied in order to give dashed lane markings the appearance of a long and continuous line (BORKAR et al., 2009). Input frame rate and the speed of the vehicle may vary, therefore their impact should be considered for temporal integration.

In this work, it was chosen to use static preprocessing techniques. Grayscale color conversion is applied, as well as a static selection of the region of interest, based on the position of the camera. Also, a static IPM is performed, assuming a fixed flat road, based on an offline calibration process. It was also used a temporal integration, but unlike the temporal integration of the image, temporal coherence is ensured by a custom mechanism.

2.2 Lane Feature Extraction

After preprocessing steps, there is usually a feature extraction step. In this phase, the goal is to extract visual cues that most likely will help a given algorithm to detect the lane in the image. Commonly, the preprocessed image is segmented based on the extracted features. For this, there are many features that may be considered: edges, colors, texture, structure, etc.

Edges are defined by changes in the intensity function (MARR; HILDRETH, 1980). Lane markings are meant to have clear edges, because they are designed to be easily distinguished by the human vision. The problem arises when real-world comes in: lane markings are left fading after years, cars can occlude them, other objects can likely generate edges and shadows can reduce this expected contrast. Nevertheless, edge detection algorithms have been extensively used and, as it can be seen in recent surveys (HILLEL et al., 2014; YENIKAYA; YENIKAYA; DÜVEN, 2013), it certainly can be considered as the most common technique for this task. Common edge-detection algorithms used are, but not limited to: Sobel (HAGA; SASAKAWA; KURODA, 1995; WANG; DAHNOUN; ACHIM, 2012), Canny (WANG; HUANG; FU, 2005; NEDEVSCHI et al., 2004), steerable filters (MCCALL; TRIVEDI, 2006; GUO; MITA, 2009) and hand-crafted filters (ALY, 2008; NIETO et al., 2012). Most of these manually-engineered filters in the literature rely on the Low-High-Low intensity pattern caused by a lane marking on the road.

Another approach would be a color-based. Human visual system helps us to use the richness of color information in a way that seems to be an effortless task. However, for machine vision, especially for systems where real-time performance is a requirement, the extra processing time required to process the additional data usually restrain its use. It is worth saying that this approach has proved to be more effective on rural roads, where road color may be a better discriminative feature than on urban environments. Most commonly, color information is used to cluster pixels to further classification. Some authors use supervised algorithms, such as neural networks based on RGB histograms (FOEDISCH; TAKEUCHI, 2004), support vector machines (SVM) based on RGB pixels (ZHANG; HOU; ZHOU, 2005) and Bayesian classification (CRISMAN; THORPE, 1993). Other authors use unsupervised methods, such as using ISODATA algorithm based on hue and saturation histograms (SOQUET; AUBERT; HAUTIERE, 2007), a modified ISODATA named UNSCARF based on RGB pixels (CRISMAN; THORPE, 1991), a modified UNSCARF named RSURD based on HSV values (GAO; LUO; MOLI, 2007) among others.

Many approaches combine features (LI et al., 2015; SATZODA; TRIVEDI, 2015; GAIKWAD; LOKHANDE, 2015). Texture-based approaches are not common for urban and structured roads, but alongside with color-based, they are fairly used for rural and

unstructured environments. Multiscale Gabor wavelet filters were used in (RASMUSSEN, 2004) for finding vanishing point. Neural networks were employed by (FERNANDEZ-MALOIGNE; BONNET, 1995) for road and nonroad classification based on texture clusters.

After feature extraction, segmentation is usually performed, mainly throughout a binarization technique, resulting in what is called a "feature map" or "evidence map" as in (BERRIEL et al., 2015). This includes an additional threshold. Thresholds are spread all over these methods and their fine tuning are a known limitation. Thresholds usually restrict these systems to work under specific conditions (such as the ones with strong shadows, abrupt illumination change, fading lane markings, etc.) requiring additional tuning or recalibration. Adaptive threshold techniques were also explored (BROGGI, 1995; WU; LIN; CHEN, 2009). More recently, deep learning is also being used to both extract features and perform lane segmentation (HUVAL et al., 2015) and to understand a scene and give feedback to a control system (CHEN et al., 2015). Even though, the virtually unlimited amount of scenes forbids researchers to opt in for a specific feature extractor.

In this work, different feature extractors were used to generate multiple feature maps. This feature maps are then combined, on demand, by each processing module, according to the combinations of features that will most likely to help in its task.

2.3 Lane Estimation and Tracking

Lane estimation is the process of fitting a lane model into an input (e.g. monocular camera image). In the literature, the term Lane Estimation can be found with two different meanings: a broader one, which refers to the whole process of preprocessing, feature extraction and model fitting (SATZODA; TRIVEDI, 2014b); and a narrower one, referring to the model fitting only (HUANG, 2010). In this work, we will refer to Lane Estimation in its narrower meaning: model fitting. Interchangeably, lane detection have been used in the same sense of lane estimation in the literature (WANG; TEOH; SHEN, 2004).

In the previous steps, the image was preprocessed and features were extracted (resulting in one or more feature maps). In general, the lane estimation is usually performed on this feature map. This process can be done either in the original perspective or in the IPM-based feature map. As fitting lane model in the inverse perspective has proved to be an effective solution (YENIKAYA; YENIKAYA; DÜVEN, 2013), it has been used widely. Lane itself has been modeled into a wide range of models. In (KUK et al., 2010), a purely linear model was used. Most of the linear models are based in the Hough transform. Linear-based techniques have the advantage of usually being faster and they explore the assumption that the lane near the car is approximately linear, even when there is curvature. As Hough transform is sometimes seen as an expensive algorithm (SATZODA;

SUCHITRA; SRIKANTHAN, 2009), both of these works propose faster implementation of the transformation. The drawback of linear-based models, is the fact that the complexity of lanes in real world cannot be simplified to this point. In (YU; HAN; HAHN, 2008), a quadratic function is used to model the lane. Parabolic models are more flexible than linear and can be as fast as linear. Yet, these models are still restrictive in complex scenes. Many other models have been designed based on these limitations. Although single line is too prohibitive for modeling complex scenes, a combination of line segments, as in (LIPSKI et al., 2008), have been used with success. This can be done because lane markings are assumed to be locally linear, at least approximately. A circular shape model was proposed by (MA; LAKSHMANAN; HERO, 2000), under the assumption that lane boundaries are laid as concentric circles, at least over small segments. Although the authors claim that these models are better than previous parabolic models, they did not perform experimentation to support this statement. The same assumption is used by models based on circular arcs, such as in (KREUCHER; LAKSHMANAN; KLUGE, 1998) and (CHENG et al., 2006). Besides these parametric models, a semi-parametric model has been commonly used: spline-based models. Splines are a piecewise polynomial approximation (SCHOENBERG, 1946). Different spline models have been used to model lane boundaries, such as B-splines (WANG; TEOH; SHEN, 2004), Bezier-splines (ALY, 2008), Cubic splines (KIM, 2006) and others. Spline-based models share a common core: the model is defined by control points. The advantage of having these control points is that a small change in the points results in a proportionally small change in the resulting lane estimate. This property allows for more control of the result generated by the model. Also, to a certain extent, splines can fit more complex lane and road models by just adding control points. In addition to the previous proposed models, non-parametric models were also proposed. In (BROGGI; CATTANI, 2006), an evolutionary approach (Ant Colony Optimization – ACO) was used in order to detect and estimate the road boundaries. The individuals (ants) modeled a non-parametric boundary, which enabled a model more suitable for non-urban scenes, such as deserts.

In most works, domain constraints are incorporated to either increase robustness (through a validation of the lane candidates), avoid outliers or increase computational performance. Some authors use the vanishing point (VP) to support their models. Vanishing point was used in (KUK et al., 2010) not only to speedup the feature extraction, but also to constraint the lane model. The author considered only line candidates that pass through a circle centered on the vanishing point (they named this region ROIDPD – Region Of Interest in Parametric Domain). Although the benefits are clear (improvement of computational performance of the Hough transform), the accuracy of their algorithm is constrained by the accuracy of the vanishing point estimator. As the previous vanishing points are temporally integrated, a false positive vanishing point location could confine the lane estimates in a wrong position over time. Other authors have used VP to constraint

models based on the assumption of parallel lane markings and road boundaries (NIETO et al., 2008); and that the road follows the overall image flow (RASMUSSEN, 2004). Many other constrains were applied by other authors, such as: lane width do not change abruptly between subsequent frames (DAHLKAMP et al., 2006), lane markings orientation can be constrained based on the areas of the image (BORKAR et al., 2009), lane markings have relatively constant width (ROTARU; GRAF; ZHANG, 2004), and other constraints.

As a result of the lane estimation process, a lane estimate is generated. Generally, lane estimation is performed in a sequence of images and the lane estimate is expected to be consistent between frames. Therefore, lane tracking is a common procedure. Lane tracking is performed with the same intent of domain constrains: to reduce computational cost, to deal with outliers and to improve overall accuracy and precision. Mainly, temporal integration is applied to improve the consistency of the estimates. Most works use either Kalman filter (MCCALL; TRIVEDI, 2006; BORKAR; HAYES; SMITH, 2009) or Particle filter (NEDEVSCHI et al., 2004; NIETO et al., 2016; SHIN; TAO; KLETTE, 2015). Some authors perform tracking on the original input, but certainly most of them track lane estimates on the transformed image (IPM). In the IPM image, the coordinates of the lane estimate can be assumed to be real world coordinates, given relative position of the car to the vehicle. Therefore, the motion of the vehicle can be used to adjust and constraint lane estimates. Vehicle dynamics can be derived from image, but most of the time, other sensors are used to give this input (e.g. GPS, IMU, odometry, etc.). The IPM, as it is commonly used (a static matrix, assuming flat surface), introduces another problem. Differences in road inclination, casual bumps and others introduce an undesired distortion, causing lane markings to lose their width constancy, among other noises. This is a problem especially for tracking, due to the assumption that lane estimates change smoothly over time. In general, model parameters are adjusted based on the car motion. In the case of spline-based models, control points are moved according to the dynamic of the vehicle, as in (JIANG et al., 2009). In (SHIN; TAO; KLETTE, 2015), particle filters are used to perform the tracking. Lane borders, modeled by the particles, are updated based on an approximation of the distance between the positions of the car in two subsequent frames. This approximation is made by a fixed number based on a given vehicle speed. Although not done, the authors claim the real vehicle speed could be received from the car or estimated using some visual odometry technique and this parameter could be derived from it. In (NIETO et al., 2016), both Kalman and Particle filters are used. The state vector is divided into a linear and non-linear part. Kalman filter is applied to the linear part, using a constant-velocity model and a Sequential Importance Resampling (SIR) Particle Filter is used on the non-linear part. This allows for a significant dimensionality reduction for the particle filter, therefore reducing the computational cost.

It is worthy saying that many of these works are dated around 2004-2007, because of the influence of the DARPA Grand Challenges (DARPA, 2007) promoted by the Defense

Advanced Research Projects Agency in 2004 and 2005, and the DARPA Urban Challenge in 2007. Back then, these competitions were the major incentives (US$1 milling funding per team in 2007) and paved the way to nowadays developments and investments. In this work, mainly, it was used a combination of two filters to both help estimating and tracking the lane estimate: Kalman and Particle filters. Besides them, custom mechanism were designed to help on both tasks.

2.4 Lane Analysis

Besides lane estimation, this work addresses a series of other tasks related to the lane. The awareness required from a driver to keep a vehicle on a lane involves much more than knowing the lane boundaries. The driver is expected to keep the vehicle in the lane, preferably on the safest position. Also, the driver must follows traffic laws. In addition, there are many visual cues that help drivers to understand what are the possible actions that may take place in a lane, such as: crosswalks indicate the need of additional caution, arrows show allowed moves, lane marking types define (at least in Brazil) whether you can overtake or not and whether there might have an adjacent lane, among other visual cues. Because of that, beyond lane estimation, many works have been done in the lane related tasks. Here, the combination of the lane estimation and these lane related tasks will be referenced as Lane Analysis.

One of these tasks is often named LDW: Lane Departure Warning. It comprises estimating the deviation of the car from the centerline, assuming both sides of the lane are parallel to the centerline. In (LEE, 2002), the author was able to detect lane departure without detecting the lane markings. Based on an edge distribution function, the author estimates a symmetry axis (centerline). Then, this symmetry axis is used to estimate the lane departure. In general, lane center deviation (lane departure measure) is straightforwardly estimated based on lane estimation (detection), as in (SON et al., 2015).

Figure 4 – Different types of crosswalks.

Besides LDW, crosswalk detection is a quite important task, since crosswalks indicates that drivers should be more cautious with the presence of pedestrians. Crosswalks are presented (Figure 4) in many different ways. They can also be fading. There has been many studies on this task, but most of them focus on crosswalk detection for the

visually impaired people (IVANCHENKO; COUGHLAN; SHEN, 2008; AHMETOVIC et al., 2015). This usually puts detection on a very different perspective, which makes impractical the use of such algorithms in a car. In (MASCETTI et al., 2016), rotation and position of a mobile device is used in order to compute the horizon line. This computation is a requirement for their system to work. A line detector is used in the feature extraction phase. Line segments clustering followed by a zebra crossing validation are responsible for detecting crosswalks. In (HASELHOFF; KUMMERT, 2010), a forward-looking camera was used and features were extracted based on adaptive filtering, using Haar-like features. After that, row-wise line segments are connected and the result is analyzed. A classifier is used to detect the crosswalks and Kalman filter is used to track the ground-plane and refine crosswalk detection. In a more general perspective, (WANG et al., 2014) uses color and depth (RGB-D) information to detect crosswalks.

Figure 5 – There are different Lane Markings Types (LMT), and they might be in different conditions, such as fading and occluded.

Lane markings type (Figure 5) detection and recognition is also a valuable task. In Brazil, lane marking types are helpful to distinguish various situations, such as: the presence of adjacent lanes, traffic in the opposite direction and if overtaking is allowed or not. In (PAULA; JUNG, 2015), an automatic on-the-fly calibration procedure is applied based on previous lane estimation. To perform this on-the-fly calibration, a specific type of driving sequence is required. After that, intensity is used to extract road markings. To classify, a two-step scheme is used. Firstly, a Bayesian classifier based on Gaussian distributions is used to classify some of the lane markings type. The second step of the classification is performed by analyzing the variance of the local maximum and minimum of the intensity profile of the scan lines on the IPM. In (SATZODA; TRIVEDI, 2013a), scan bands with dynamic width are used to predict the existence of lane markings. After that, steerable filters are used to extract horizontal variations, then evidences of lane markings are detected for further classification.

Arrow markings (Figure 6) detection and classification are other tasks of interest for advanced driver assistance systems. In (SUCHITRA; SATZODA; SRIKANTHAN, 2012), arrow signatures, including parts of arrows, are derived from the edge maps. Edge maps are processed and analyzed in the Hough space, then used in the detection and classification scheme. Authors claim that these signatures are invariant to scale and rotation. In (LIU et

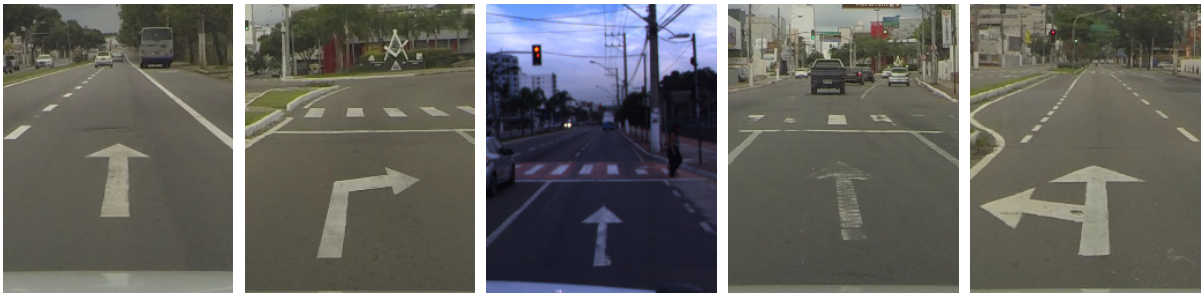


Figure 6 – Different types of Arrow Markings. Arrows markings, such as lane markings, can also be fading after years of usage without maintenance.

al., 2015), high brightness slice filtering is applied on the IPM image. A two-step scheme is used to detect and classify the road markings: Adaboost classifier with Haar-like feature (in the coarse recognition) and followed by an Extreme Machine Learning with histogram of gradients features (in the fine recognition). After that, temporal information is used to enhance overall accuracy. These arrow markings, as well as other pavement markings, may change in appearance from country to country. In Brazil, they were defined as in the Figure 7 by (DENATRAN, 2007).

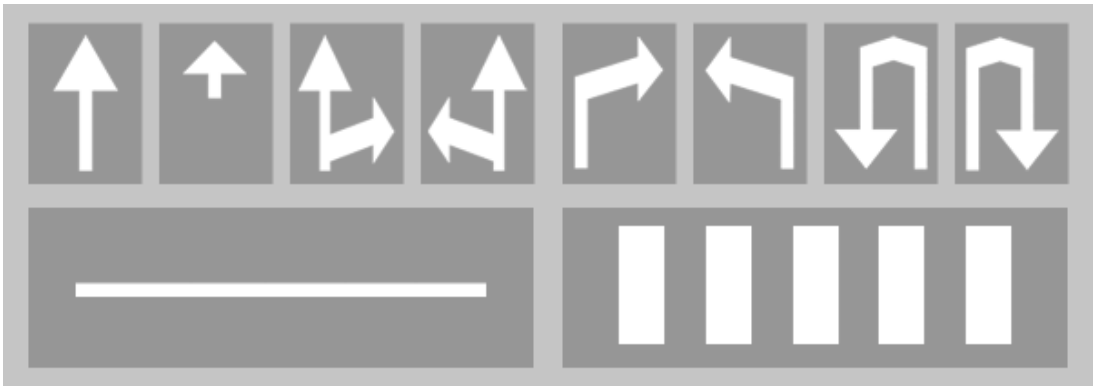


Figure 7 – Road Signs and Crosswalk. Arrow markings in the upper row. Lower row: stop-line (left) and crosswalk (right). These symbols are defined in the Brazilian Manual of Traffic Signs (DENATRAN, 2007)

Most of these studies focus only on a particular task. Some of them assume the lane estimation is performed by another algorithm. Most of them also keep their implementation and datasets private, turning comparisons into a fairly difficult problem.

3 Ego-Lane Analysis System (ELAS)

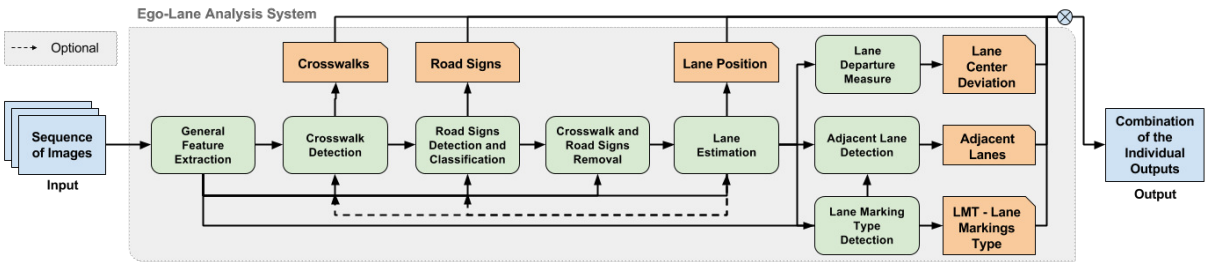


Figure 8 – Overview of the Ego-Lane Analysis System (ELAS). The system processes a sequence of images, one at a time. Firstly, general features are extracted. Subsequently, each module is responsible to a task. Finally, the individual outputs of all modules are combined into the output of the system.

In this section, the Vision-Based Ego-Lane Analysis System (ELAS) is detailed. ELAS processes a temporal sequence of images, analyzing each of them individually. These images are expected to come from a monocular forward-looking camera mounted on a vehicle. The general work-flow is described in Figure 8. Firstly, general road markings features are extracted and stored in feature maps. Based on these features, pavement markings (i.e. crosswalks, arrow markings, etc.) are detected and removed from the maps. Subsequently, lanes are estimated by using a combination of Hough lines and Kalman filter (referred from now on as Kalman) for the base of the lane, and spline-based particle filter for the curvature of the lane. This combination was used to take advantage of the lane stability and linearity near the car (Hough and Kalman), while reducing the number of parameters of the particle filter responsible for the curvature. Finally, based on the estimated lane position, the remaining tasks are performed: LMT classification, adjacent lanes detection and deviation from the lane center. All these tasks are performed in real-time (i.e. more than 30 frames per second) and are explained in details in the next subsections. As a result, for each image the system outputs information describing the lane position, lane markings type, crosswalks, road signs, presence of adjacent lanes and deviation of the car related to the lane center.

3.1 General Feature Extraction

A general set of feature maps is extracted from each frame and the maps are used by each module according to its task. An overview of this process is shown in Figure 9.

3.1.1 Preprocessing

Before generating the feature maps, the input image is converted to grayscale, since only pixel intensities are analyzed for generating the feature maps. Secondly, a

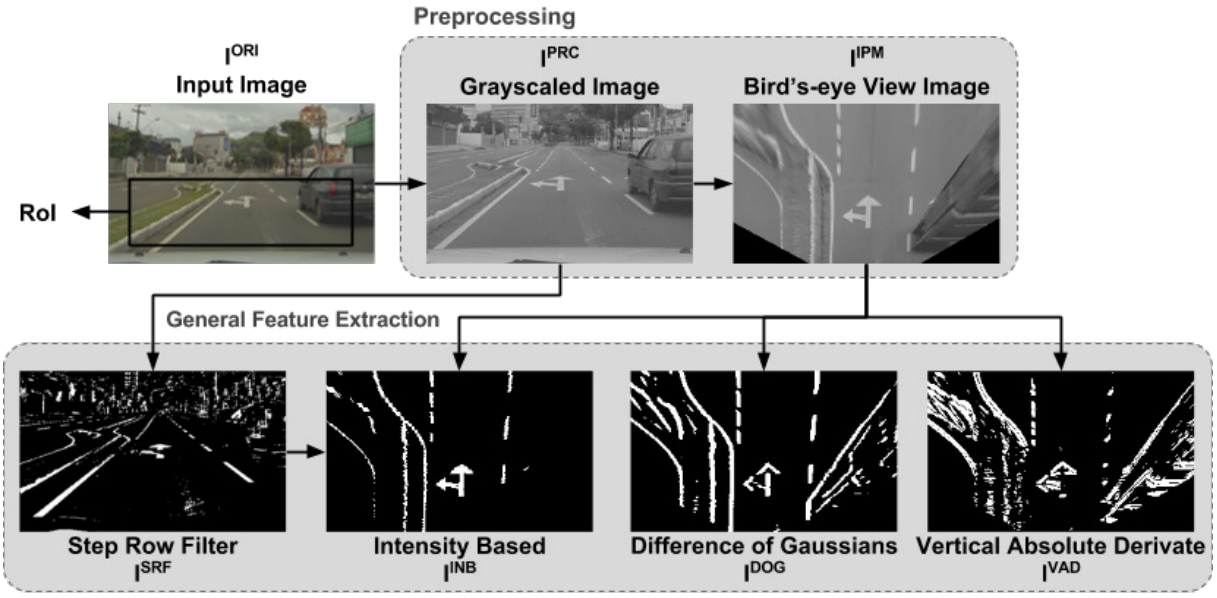


Figure 9 – Overview of General Feature Extraction. The input image is preprocessed and general features are extracted in the form of four binary maps, i.e. feature maps. These maps and the processed images are used by the subsequent modules.

region of interest (RoI) is set in order to remove irrelevant parts of the image. Finally, an Inverse Perspective Mapping (IPM) (MALLOT et al., 1991) is applied in order to reduce perspective distortion, resulting in a top-view image (also called bird's-eye view). To find the homography matrix, an offline calibration is performed. To apply the IPM, ELAS assumes a constant ground plane along the input frames, using a static homography matrix. In this bird's-eye view image, lane markings tend to have constant width. However, due to road inclination and casual bumps, lane markings width constancy can be temporarily lost because fixed ground plane assumption may fail. As explained later, this issue is addressed by the lane model used.

Offline Calibration To find the homography matrix required for the IPM, an offline process is performed. For that, a given image where the car is in the center of the lane is used. In this image, four points are selected: they comprise the four points where the lane boundaries (or their line extensions, in case of an image with dashed lane markings) intersect the region of interest. The homography matrix is calculated so that these four points are mapped into a quadrilateral (constant ground plane).

3.1.2 Feature Maps Generation

Four feature maps are generated using threshold-based techniques. Each feature map is a binary image. White pixels in these maps are called evidences. An overview of this process is described in Figure 9.

Step Row Filter Map (I^{SRF}) Lane markings are usually areas brighter than its surroundings (asphalt). In this feature map, lane marking evidences are detected on the original grayscale image using the step row filter defined by Equation 3.1, as presented in (NIETO; LABORDA; SALGADO, 2010).

$$y_i = 2x_i - (x_{i-\tau} + x_{i+\tau}) - |x_{i-\tau} - x_{i+\tau}| \tag{3.1}$$

Here, τ is assumed large enough to surpass the size of double lane markings. Additionally, this filter was applied in the original image, linearly adjusting τ according to the vertical axis. As a result, this feature map is expected to capture lane markings, as well as being robust against shadows, potholes and asphalt resurfacing. It worth noticing that depending of τ some objects, e.g. white cars, may generate undesired features on this map.

Horizontal Difference of Gaussians Map (I^{DOG}) Based on the same assumption (lane markings are brighter than asphalt), in this map, evidences are found using a horizontal Difference of Gaussians (MARR; HILDRETH, 1980) (DoG), equivalent to a horizontal Mexican hat with central opening with the average size of a lane width. DoG is applied in the IPM grayscale image in order to take advantage of lane width invariability, and is followed by a threshold. For this feature map, it is expected to capture the lane markings in the IPM image, but noise is also expected from shadows, cars, curbs, and others. Even though, this feature map is expected to increase overall robustness when combined with other feature maps.

Vertical Absolute Derivative Map (I^{VAD}) Some objects of interest (i.e. vehicles, stop lines, etc.) contains horizontal edges in the bird's-eye view image. To extract these features, vertical changes are calculated using the absolute value of the y-image derivative followed by a threshold. In this feature map, objects that are unlikely to be lane markings are expected to generate features. This feature map is expected to be useful when detecting obstacles ahead of the vehicle.

Intensity Based Map (I^{INB}) This map aims to enhance the lane markings detection in I^{SRF} by analyzing the intensity of the detected evidences. Therefore, it receives the I^{SRF} and the grayscale input image as input. The inverse of I^{SRF} is assumed to be composed of asphalt and others. Hence, the mean (μ_A) and standard deviation (σ_A) in the grayscale input image considering pixels that are not evidences in I^{SRF} are used to filter wrongly detected lane markings in the original I^{SRF}. This process generates an intermediary feature map ($I^{INB'}$) comprising detected evidences in I^{SRF} that have corresponding intensities in the grayscale input image with values higher than $\mu_A + 2\sigma_A$. The same procedure is applied in the $I^{INB'}$ in order find out an optimal threshold for the lane markings. Therefore, the mean (μ_{LM}) and standard deviation (σ_{LM}) in the grayscale input image considering pixels that are evidences in $I^{INB'}$ are used to calculate a better threshold for the lane markings

in the grayscale input image, i.e. I^{INB} comprises pixels in the grayscale input image with intensities values higher than $\mu_{LM} - \sigma_{LM}$. This process is summarized in Equation 3.2, where $\wedge$ denotes a pixel-wise AND operation.

$$
\begin{aligned}
I^{A} =& \neg(I^{PRC} \wedge I^{SRF}) \\
\mu_{A} =& \overline{I^{A}}, \sigma_{A} = \sqrt{\overline{I^{A} - \mu_{A}}} \\
I^{INB'}_{x,y} =& \begin{cases} 1 & \text{if } (I^{PRC}_{x,y} > \mu_{A} + 2\sigma_{A}) \wedge (I^{SRF}_{x,y} > 0) \\ 0 & \text{otherwise} \end{cases} \\
\mu_{LM} =& \overline{I^{INB'}}, \sigma_{LM} = \sqrt{\overline{I^{INB'} - \mu_{LM}}} \\
I^{INB}_{x,y} =& \begin{cases} 1 & \text{if } I^{PRC}_{x,y} > \mu_{LM} - \sigma_{LM} \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
\tag{3.2}
$$

3.2 Crosswalk Detection

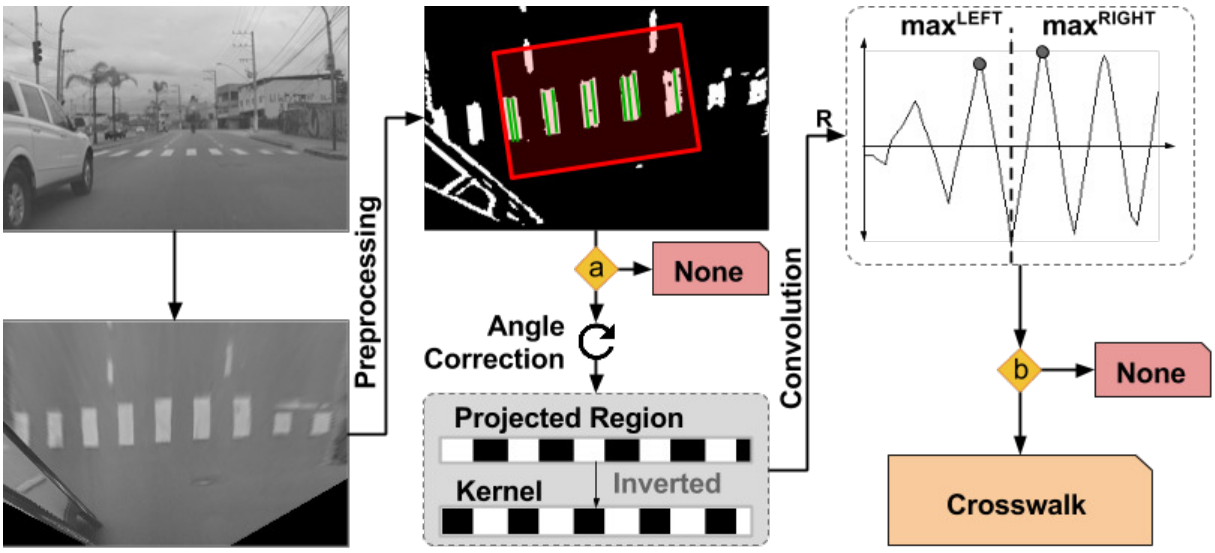


Figure 10 – Overview of Crosswalk Detection. I^{DOG} is analyzed in order to detect crosswalk candidates in the frame. A region of interest (red) is used to search for houghs (green). If a set of hough lines is detected, the region is rotated using the angle of the houghs. The final crosswalk detection is based on the analyses of the convolution between the projected crosswalk region and its inverted version.

Crosswalks are series of large vertical strips (Figure 7) to mark a place where pedestrians can safely cross a street. Therefore, the goal of this module is to detect crosswalks ahead of the vehicle. It uses the DoG-based feature map (I^{DOG}) as input to emphasize the vertical strips, and it takes advantage from the previous Lane Position (LP), if available. Firstly, a morphological closing is applied to remove small holes in I^{DOG}. Subsequently, a morphological erode is applied in order to degrade road markings that are

expected to have smaller width than crosswalks. These preprocessing operations are based on the assumption that crosswalks are usually uniform in the IPM image. After that, a predefined region of the frame that is located in front of the vehicle is chosen as a crosswalk searching region. If a previous frame LP is available, lane direction is considered to properly define the crosswalk searching region according to the lane orientation. In order to detect crosswalks, features are extracted from this region using a Hough Transform. Further crosswalk detection processing is only carried on if hough lines are detected, otherwise the region is assumed to have no crosswalk (Figure 10 (a)). Extracted Hough Lines are grouped into a 2-dimensional histogram (see Figure 11) considering their angle and intersection with the bottom of the IPM image (with bin sizes equal to 1 and 3, respectively). This 2D histogram is denoted as $H^{2D}_{x,y}$, where $0 \leq y < M$ and $0 \leq x < N$ (in this case, $M = 360$ and $N = 640$). It is assumed that the angle of the crosswalk in relation to the image can be calculated by the dominant angle of the extracted Hough lines. As crosswalks could be rotated in relation to the car or lane itself, the search region is rotated in the direction of the dominant angle α of the extracted Hough lines. The dominant angle α is calculated (Figure 11) as the maximum number of hits considering the sum of a region along the axis of the angles:

$$\begin{aligned} H^{1D}_i &= \sum_{y=i-d}^{i+d} \sum_{x=0}^{N} H^{2D}_{x,y} \\ \alpha &= \max\{H^{1D}_i \mid 0 \leq i < M\} \end{aligned} \tag{3.3}$$

In order to emphasize the locations where most strips are, the binary rotated region is projected (vertical maximum) into the horizontal axis, reducing it to a binary line. This binary line is then normalized to $\{-1, +1\}$. Crosswalks are assumed to have a distinguishable periodicity. Therefore, a convolution with its inverse should generate a minimum value in the center and maximum values as it shifts to the left or to the right. Based on this assumption, crosswalks are detected using a convolution of the projected crosswalk with its inverse. Considering R as the result of this convolution, crosswalk detection for a projected region of size n is defined by Equation 3.4 and illustrated in Figure 10 (b).

$$\begin{aligned} f(R) &= \max\{R_0, \cdots, R_{\frac{n}{2}-1}\} + \max\{R_{\frac{n}{2}}, \cdots, R_n\}) \\ \text{detected} &= \begin{cases} \textit{crosswalk} & \text{if } f(R) > 0 \\ \textit{none} & \text{otherwise} \end{cases} \end{aligned} \tag{3.4}$$

3.3 Road Signs Detection and Classification

Road signs are visual marks added to the road to inform drivers about what is and what is not allowed in that pathway and to control its flow. Each country can implement its own symbols and use them based on their own rules, turning general detection and

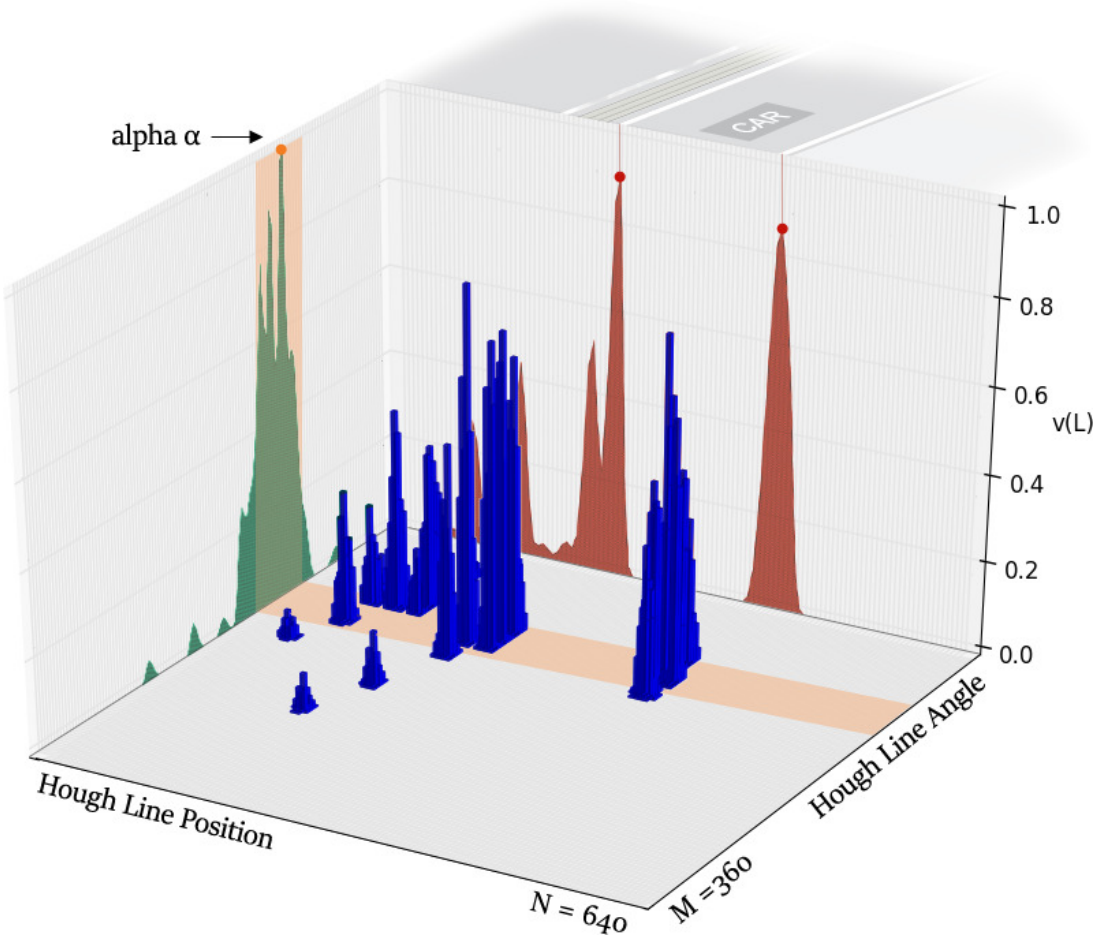


Figure 11 – This figure illustrates the dominant angle calculation (Equation 3.3). $H^{2D}_{x,y}$ is represented by the blue bars. Green histogram represents H^{1D}_{i} and α is its maximum. Used in further processing, the red histogram illustrates the projection of the selected Hough lines into the horizontal axis (Equation 3.8).

classification of road signs into a complex problem. This module (Figure 12) combines a simple candidate extraction and classification procedure to detect and identify the different arrow types, the stop line (Figure 7), or any other "unknown" symbols.

Candidates Extraction First of all, the input of this module is a combination of two feature maps (one for vertical and one for horizontal features) assumed to enhance pavement marking-like features (i.e. edges of markings). This combination is defined by $I^{DOG} \vee I^{VAD}$, where $\vee$ denotes a pixel-wise OR. Thereafter, if the previous lane position is known, lane region is corrected based on its angle. Subsequently, lane region is scanned. The scanning process performs a continuity check over the projection of the region (horizontal maximum) into the vertical axis. As a result of this scanning process, a set of non-overlapping road sign candidates are extracted. These candidates are further filtered by a scanning process performed in their projection (vertical maximum) into the horizontal axis. Signs not having evidences in the center of the lane are ignored (road signs are assumed to be close to the center). The remaining signs are trimmed using the

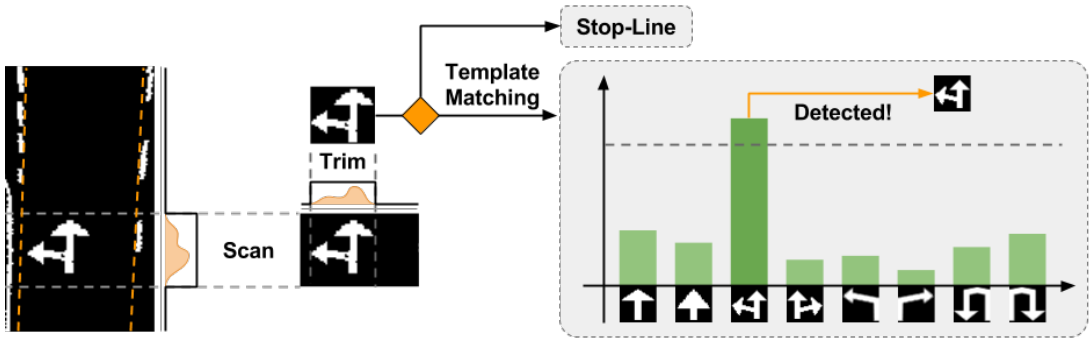


Figure 12 – Road Signs Detection and Classification Overview. The lane region is scanned and trimmed, generating symbol candidates. A two-step scheme is used to classify these candidates: if a symbol candidate has the proportion of a stop-line, then it is classified as one, otherwise the classification is done through a Template Matching scheme.

horizontal axis projection. In addition, pavement markings are expected to be brighter than asphalt. Therefore, the intensity of the sign candidates are compared with the intensity of the remaining surrounding region inside the lane. Finally, only candidates (signs) having average intensity higher than the average intensity plus one standard deviation of its surrounding region (asphalt) are considered.

Classification Basically, classification is done through a two-step process: stop line detection and template matching scheme. Stop line detection is performed through a proportion ($\frac{\text{width}}{\text{height}}$) threshold. Subsequently, a template matching scheme is applied in the remaining candidates. For this process, binary templates were generated using all 8 arrows from Brazilian Manual of Traffic Sign (DENATRAN, 2007) (see Figure 7). Candidates and patches are normalized to the same size (32×32 pixels). Then, a template matching function, in this case Normalized Cross-Correlation (LEWIS, 1995), is calculated giving each candidate a set of coefficients $C = \{c_1, \cdots, c_n\}$, where $n = 8$ is the number of templates. Road sign candidates are classified into one of the arrows following Equation 3.5:

$$f(C) = \begin{cases} i & \text{if } \max\{C\} > \text{threshold} \\ unknown & \text{otherwise} \end{cases} \tag{3.5}$$

where i is the index of the greatest coefficient, therefore $1 \leq i \leq 8$. More than one road sign can be detected in a given frame. The result of this module is a set of road signs classified into 1 of 10 classes: stop line, one of 8 arrows or unknown.

3.4 Crosswalk and Road Signs Removal

Although lane markings are meant to be visually distinguishable, roads have these other elements with similar properties that may confuse or degrade overall detection quality. Because at this point the presence of some of these symbols (i.e. crosswalk and

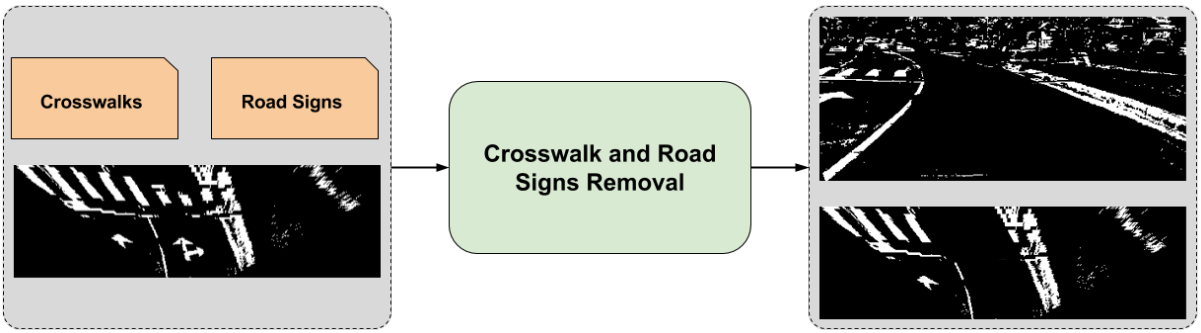


Figure 13 – Previously detected crosswalks and road signs are given as input and this module removes them from the feature maps. The goal is to reduce noise and increase robustness of the lane estimation module.

road signs) is already known, they can be removed (Figure 13) from the feature maps used in further modules. This is expected to enhance lane markings detection, therefore increasing lane estimation accuracy.

3.5 Lane Estimation

The lane estimation process is a difficult problem for many reasons (e.g. lanes vary in shape, brightness, color and texture; traffic; etc.). In ELAS, a combination of Hough lines and Kalman filter is used for the base of the lane and a spline-based particle filter is used to model the curvature of the lane and distortions from the IPM. The estimation is performed in two phases because it is assumed that the lane base (i.e. lane region near the vehicle) can be linearly approximated, which simplifies the complexity of the particle filter used to model the lane as a spline. At first, individual lane candidates and lane measurements are generated based on Hough lines. Subsequently, Kalman is used to estimate the lane base. Finally, lane position (mainly curvature) is estimated by using a particle filter in which the lane estimation is a spline-based model.

3.5.1 Lane Measurement Generation

The goal of this task (Figure 14) is to generate a valid pair of Hough lines (i.e. lane measurement). It is assumed that most of the extracted Hough lines have the same orientation as the lane. Initially, a morphological skeleton (MARAGOS; SCHAFER, 1986) of I^{SRF} is used to extract a set of lines (i.e. Hough lines) using Probabilistic Hough Transform (KIRYATI; ELDAR; BRUCKSTEIN, 1991). This thinning is applied because large blobs tend to generate too many Hough lines with some of them misaligned with the lane markings. A Hough line can be represented as a set of points discretized in relation to the pixels in the vertical axis ($L = \{p_1, \cdots, p_n\}$). To remove outliers and properly choose a representative candidate for each lane, a 2D histogram (H^{2D}) is computed in the same

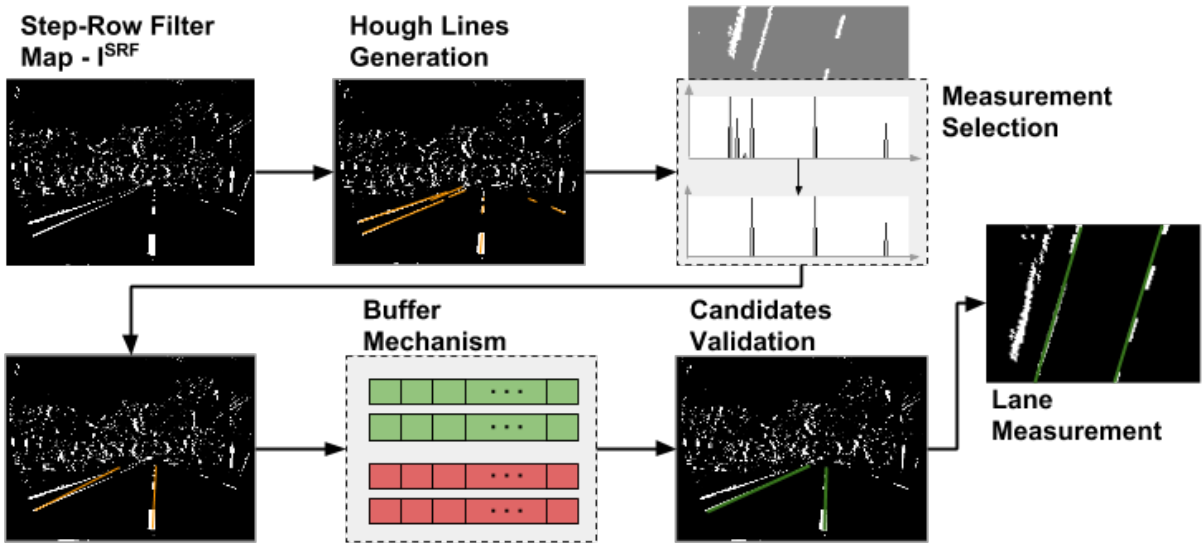


Figure 14 – Overview of the lane measurement generation. The process receives I^{SRF} as input, generates some lane candidates, filters and validates them and outputs a lane measurement (in green).

way as in Section 3.2, but with a different bin counting. The contribution of each Hough line ($v(L)$) to the histogram is based on the number of evidences (i.e. white pixels in the feature map) under this line. Hough lines with evidences closer to it are preferred over those which evidences are farther, considering a maximum search length b. Therefore, $v(L)$ is calculated using Equation 3.6, where $dist(p)$ is the distance between p and the closest evidence in the same horizontal line of p.

$$v(L) = \sum_{p \in L} \max\{0, b - dist(p)\} \tag{3.6}$$

Evidences are counted here, and in all the modules that uses the evidence counting process, on a combined map (I^{CMB}) defined by Equation 3.7.

$$I^{CMB} = I^{SRF} \wedge I^{INB} \tag{3.7}$$

where $\wedge$ is a pixel-wise AND operation.

Given this 2D histogram, the dominant angle α is calculated following Equation 3.3 using $d = 5$. Based on the dominant angle α, H^{2D} is masked to keep only Hough lines with angle closer than a threshold δ, using $\delta = 15$. Then, masked 2D histogram is divided based on the vehicle's position into left and right histograms, and reduced to 1D normalized histograms using Equation 3.8.

$$H_x^{1D'} = \max\{H_{x,\alpha-\delta}^{2D}, \cdots, H_{x,\alpha+\delta}^{2D}\} \tag{3.8}$$

Punishment Mechanism In the case of double lane markings, the innermost lane marking is expected to be chosen in the lane estimation process. Curbs and lane markings

partially occluded by dust may also generate Hough lines in a certain amount that could lead to wrong estimates. To overcome this problem, a punishment mechanism was implemented. Basically, this mechanism punishes the outer Hough lines by using a punishment factor γ on $H_x^{1D'}$. In fact, outer values are multiplied by $1 - (\gamma \times H_x^{1D'})$ in order to reduce them proportionally to inner values of $H_x^{1D'}$, resulting in H^{1D}. To avoid overweight of values around the center of the image, a neutral region in the center of the image (around the center of the vehicle) is created where this punishment is not applied. For each half (left and right) of H^{1D}, maximums are found. Each maximum has a Hough line associated with it. The output of this step is up to two Hough lines that generated those maximums in the current frame. Each Hough line can also be denoted as $L^c = \{\rho, \theta\}$, in which ρ is the position in the horizontal axis where a given candidate L^c intersects the bottom of the image and θ is the direction angle of this line. Since the feature map contains noise, the output can be one of these: pair of independent lines; one of the lines, left or right; none of them.

Buffer Mechanism Individual lane candidates are noisy. This mechanism acts as a temporal reinforcement factor. It ensures temporally coherent measures over previously accepted estimates. In real world, lane shapes are not expected to vary abruptly from one frame to another. Based on this fact, four buffers are created, two for each side: one for correct ($B^{correct}$) and another for incorrect ($B^{incorrect}$) candidates. The size of both buffers was empirically chosen: 10. A candidate is accepted into its $B^{correct}$ if buffer is not full or if it is close to the candidates in this buffer. A candidate $L^c = \{\rho, \theta\}$ is close to a buffer if the euclidean distance between its ρ and θ to the mean ρ and θ of the elements $B^{correct}$ are less than a threshold (empirically defined as 15) respectively.

If a candidate for a side is accepted, its correspondent $B^{incorrect}$ is cleared. Otherwise, if a given candidate is rejected, it is stored into $B^{incorrect}$. If $B^{incorrect}$ is full, these sequentially rejected candidates are understood as a temporally coherent measure, and the current candidate should be accepted as correct. In this case, buffers are swapped in order to force a change on the estimates. Additionally, both, Kalman and Particle filters (described later) are reset. This reset is performed to avoid smooth transitions in these swap situations that the estimation should adapt fast.

Candidates Validation At this point, the following sets are possible: a pair of independent lines; one of the lines, left or right; none of them. Many scenarios might lead to missing line for one of the sides: true absence of one or both lane markings, a vehicle in front of the lane markings, noise, rejection by the buffer mechanism, etc. If at least one of the expected lane candidates is missing, its pair has to be generated by the system. A missing candidate is generated by projecting the only candidate available into the other side of the lane, using lane width from the previous frame. If both lane candidates are

missing, the previous estimation (if available) is used. Subsequently, they are combined (i.e. averaged) to compose a lane measurement. A measurement is represented by a base point, a lane width and an angle. This technique also reinforces the expected parallelism of lanes.

Finally, a lane measurement (i.e. a pair of lines) needs to be valid to be reported, otherwise it is reported as missing (i.e. none). The lane measurement is considered valid when $v(L^{left}) + v(L^{right})$ is greater than a threshold, where $v(L)$ is defined by Equation 3.6. There are 3 possible outputs for this step: lane measurement derived either from a pair of lines or the previous estimation, lane measurement derived from one line and the previous lane width, or no measurement.

3.5.2 Lane Base Estimation

Using the lane measurement given by the previous step, a Kalman filter (KALMAN, 1960) corrects the observations (i.e. lane measurements) based on a series of measurements over time. It acts like a smoother, since abrupt changes are unexpected most of the time, thus undesired. Nevertheless, in some cases, abrupt changes are desired (e.g. where there is no lane to be tracked, lane changes, etc.), but should not be confused with noise. Therefore, lane base estimate is controlled by a finite-state machine based on the presence or not of a lane measurement. The output of this module is a lane base (i.e. base point, lane width and lane direction) estimated by the Kalman, or no estimate.

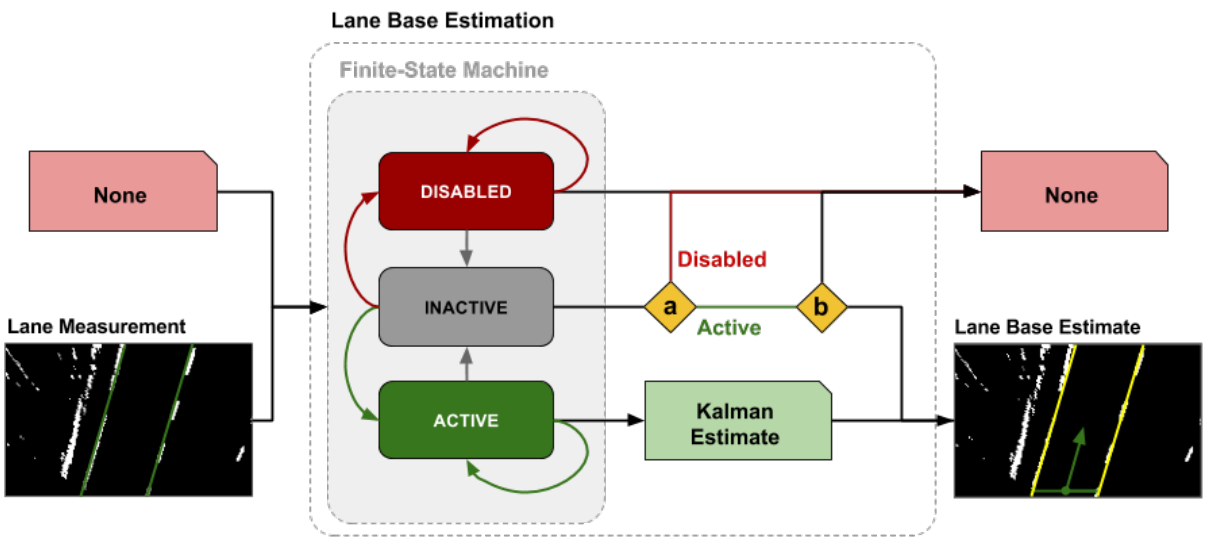


Figure 15 – Lane base estimation overview. Lane measurements are put through a Finite-State Machine. The state machine works like a hysteresis, while the Kalman filter works to smooth the measurements. The lane base estimate showed in dark green is given as output.

Kalman Filter The Kalman is used to correct a lane measurement based on previous estimates. Lane measurement (a pair of lines, L^{left} and L^{right}) is modeled in the observation

vector as $z = \{p_b, p_t, w\}$, where p_b and p_t are intersections of the measurement with bottom and top of the image, respectively; and w is the lane width.

Finite-State Machine It works like a hysteresis, where Kalman is the controlled component of this finite-state machine (Figure 15). It controls the activation of the Kalman based on the presence or not of lane measurements. Basically, this state machine has 3 states:

- Active: if the measurement was derived from a pair of lines, it is passed to the Kalman to be predicted and corrected. If it was derived from one line (and the other side is just the projection based on the previous lane width), transition matrix of the Kalman is modified in order to ensure that the lane width is not updated (i.e. corrected). In both cases, the output is Kalman's corrected estimate and the filter remains active. If there was no measurement, the state is changed to inactive;

- Inactive: this is meant to be a transitory state, because when there were lane measurements for 10 sequential frames, Kalman should be activated again. Otherwise, if there were no lane measurements for the same amount of time, Kalman should be disabled. The output of this state is the same of the previous state (i.e. previous lane base estimate if it was active and no estimate if it was disabled, as shown in Figure 15 (a)). However, the previous lane base estimate must be valid to be reported (Figure 15 (b)). The lane base estimate is considered valid when $v(L^{left}) + v(L^{right})$ is greater than a threshold, where L^{left} and L^{right} are the lines derived (shown in yellow in Figure 15) from the current lane estimation, and $v(L)$ is defined by Equation 3.6;

- Disabled: the output of this state is no lane. Some situations are expected to lead to this state, such as: very unstable areas (e.g. lane markings constantly faded for a long period of time, intersections, etc.).

3.5.3 Lane Curvature Estimation

Previous modules rely on Hough lines to provide an initial estimate. Since it cannot represent nonlinear models, this estimate is not suitable for curves. In this module, curvature is estimated using a spline-based particle filter. With the purpose of modeling curvature and adding robustness against deformations in the IPM, a lane estimate (i.e. particle) is defined. The quality of a particle is measured as a combination of evidences of lane marking and cleanness in the lane center. One particular issue with lane estimation is when there is a limited trustworthy zone ahead of the vehicle for estimating a lane (e.g. a car in front of the vehicle or a road intersection). In such cases, particle's quality should not be influenced by what is beyond this area. Therefore, the result of this module is a

final lane estimation, bounded by a trustworthy area. This lane estimation is a virtual particle resulting from the weighted average of the current set of particles of the filter.

Trustworthy Area ELAS considers detected stop lines as a delimiter of such area. Also, a search is performed in the middle of the previous lane estimation looking for evidences in I^{VAD} (i.e. cars, motorcycles, or other obstacles are expected to produce evidences on this map). If there are evidences and they do not belong to one of the recognized road signs — except stop lines — they are taken as delimiters too. To apply this technique, the search space in the weight function is bounded to the vertical axis of this delimiter, since it defines a region to be considered by the particles of the filter.

Particle Filter ELAS uses a Particle filter (GORDON; SALMOND; SMITH, 1993) in order to estimate the curvature of the lane. Therefore, lane estimates are defined by two lane widths, to be robust against deformations in the IPM; and three control points uniformly distributed in the vertical axis, based on (BERRIEL et al., 2015). Then, a particle is defined as $\{w_1, w_2, x_1, x_2, x_3\}$. Nonetheless, the dimensionality of the particle filter is only 3, because w_1 and x_1 (lane width and position near vehicle, respectively) are calculated by the Kalman and not estimated by the particle filter. Only w_2, x_2 and x_3 are estimated by the filter, and x_2 and x_3 are sufficient as control points because their vertical positions are kept fixed (i.e. control points are uniformly distributed in the vertical axis).

Particle filter is a predictor-corrector method. Particles are randomly initialized around the lane measurement, using its position and direction. In the prediction phase, the particle filter aims to estimate a lane in the actual frame based on a previous set of particles. In our setup, a particle can move x_2 and x_3 randomly based on the direction of the lane, derived from the lane base estimate, and lane width w_2 can vary randomly with fixed standard deviation. In the correction phase, each particle has a weight assigned to it. This weight is correlated to its survival probability on the re-sampling process. At this point, all estimated particles have the same weight. Firstly, each particle has its weight (W) updated according to Equation 3.11, where each spline derived from the particle can be represented as a set of points discretized in relation to the pixels in the vertical axis ($S = \{p_1, \cdots, p_h\}$), bounded by the trustworthy area height h.

Basically, this weight function gives higher weights to those particles with more evidences under their correspondent lane markings (Equation 3.9) and with a cleaner lane space between both sides of the estimated lane (Equation 3.10), where $\mu = 0$, $\sigma_1 = 1/3$, $\sigma_2 = 1/12$ and $G(x \mid \mu, \sigma_1)$ is the Gaussian function. The main motivation for the Equation 3.9 was the fact that sometimes only one side of the lane markings are visible, or some particles may have evidences only under one side. For these cases, Equation 3.9 ensures that the weights are only close to one when both sides are far from the evidences. The behavior of the Equation 3.9 can be seen in the Figure 16. Likewise, Equation 3.10

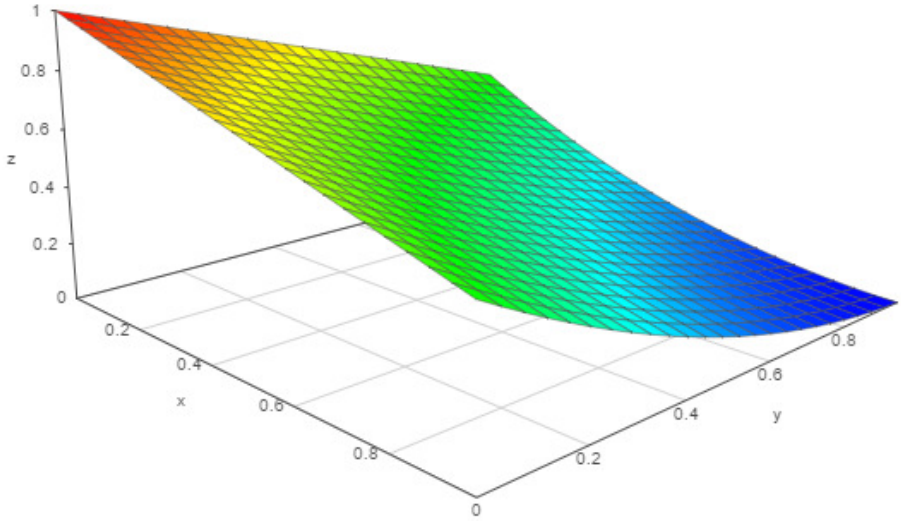


Figure 16 – The function $W_1^{'}$, described in the Equation 3.9, assigns greater weights to those particles with more evidences close to their derived estimates, even when there are no lane markings in one of the sides.

was motivated by the fact that the inner part of the estimated lane is expected to be clean of evidences on both sides.

$$l = \frac{1}{h} \sum_{p \in S^{left}} I_p^{CMB} \text{ and } r = \frac{1}{h} \sum_{p \in S^{right}} I_p^{CMB}$$

$$W_1^{'} = 1 - \left(l \times r + (1 - l \times r) \left(\frac{l + r}{2} \right) \right) \tag{3.9}$$

$$W_2^{'} = \sum_{p \in S^{left}} \sum_{j=0}^{d} I_{p(x+j,y)}^{CMB} + \sum_{p \in S^{right}} \sum_{j=0}^{d} I_{p(x-j,y)}^{CMB} \tag{3.10}$$

$$W = G(W_1^{'} |\, \mu, \sigma_1) \times G(W_2^{'} |\, \mu, \sigma_2) \tag{3.11}$$

After updating the weight of each particle, a resampling technique is used. ELAS uses the Low Variance Sampling (THRUN; BURGARD; FOX, 2005) to resample the particles based on their weights: higher weights mean higher chances to be resampled. Final lane estimation is a spline-based model represented by a virtual particle that is generated based on the weighted average of the current set of particles. For more details on the particle filter, see Figure 17.

In the Figure 17, a particle ($\{w_1, w_2, x_1, x_2, x_3\}$) is illustrated on the left, where the components w_1 and x_1 (in green) are defined by the Kalman in the previous module and the components w_2, x_2 and x_3 (in orange) are estimated by the filter. The Prediction and Correction steps are illustrated on the right. In the prediction, x_2 (blue cross) is randomly chosen following a combination of gaussian distributions as follows. A projection (green square) of the base point x_1 is calculated using the angle of the lane. A reference point

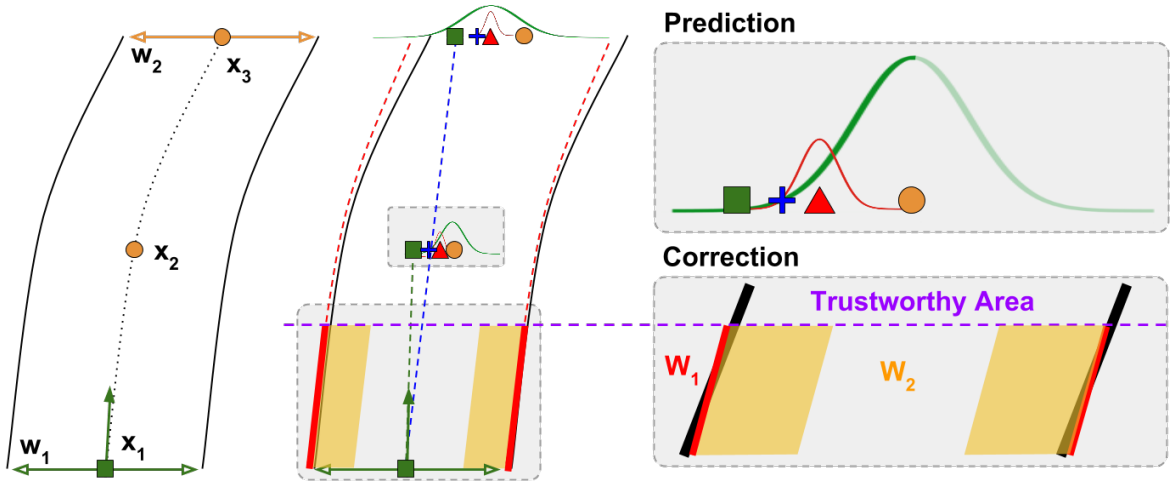


Figure 17 – Particle filter overview. The particle is defined by three values: x_2, x_3 and w_2. During the prediction, the goal is to approximate the particle previous state to the current estimate given by the linear model. On the correction phase, the filter aims to assign for each particle a weight. Roughly, the weight is defined by how precise is the estimate, given a feature map, and how clean is the inner space of the estimated lane.

(red triangle) is randomly chosen using a gaussian (in green) with a dynamic standard deviation and centered in the previous particle position (orange circle). The gaussian is mirrored to ensure an estimation in the direction of the green square. The standard deviation is calculated using 1/3 of the distance between the orange circle and the green square. The new location (blue cross) of the particle for the control point x_2 is finally randomly chosen from a fixed size normal distribution (in red) centered on the red triangle. The same procedure is applied to predict x_3. However, to allow for more mobility, the dynamic standard deviation is not divided by 3. Additionally, the base point x_1 is not projected with the Kalman estimated angle, but in the direction of the newly estimated x_2 (blue cross of the middle). In the Correction step, evidences under the estimated lanes (red lines) and the evidences in the inner space of the lane (orange regions) are counted (Equation 3.9 and Equation 3.10, respectively) considering the trustworthy area.

3.5.4 Lane Departure Measure

To enable lane departure warning (LDW), ELAS estimates the position of the vehicle related to the host lane. It assumes a forward-looking camera mounted on the top of this vehicle. The car position is expected to be in the center of the image. Using the known car position and lane center estimation, ELAS can calculate vehicle's position related to the center of the lane.

3.6 Lane Marking Type (LMT) Classification

Each lane side is classified into one of 7 types of LMTs (Figure 18). The lane estimation is derived into both sides, left and right lanes, and a region along the lane is analyzed considering the evidences in the I^{INB} map. Firstly, the number of yellow lane marking evidences is counted. The count is based on whether the pixel value is within a specific range $((30°, 31\%, 31\%)$ and $(50°, 78\%, 78\%))$ in the HSV color-space. If less than half of the evidences under the lane are considered yellow, than the lane is assumed be one of the 2 white LMTs (Figure 18 (a)). Otherwise, it is assumed to be one of the 5 yellow LMTs. Subsequently, two counts are made: one for calculating the percentage of evidences to distinguish between "dashed" or "solid"; and, one to distinguish between "single" or "double". The counts are made in the I^{INB} map considering a region along the lane. The first counts the number of evidences along lane considering the horizontal maximum projection for each point of the lane (i.e. a point in the lane has an evidence if any of its horizontal neighbors has). The second counts the number of horizontal High-Low-High intensities patterns, also known as "white-black-white" (WBW), for each point of the lane (i.e. a point in the lane has a WBW if there was a white to black to white transition along the horizontal axis). White lanes with less than 30% evidences are assumed WSD, otherwise it is assumed WSS (Figure 18 (b)). Yellow lanes with less than 20% WBW patterns are assumed "single" (Figure 18 (c)) and are further classified in YSS and YSD using the same procedure of the white lanes (Figure 18 (b)). Otherwise, they are assumed "double" and need to be further classified between "mixed" and "solid" (Figure 18 (d)). Lane is assumed YDS if it has more than 80% of WBW patterns, otherwise it is assumed "mixed", and need to be further classified between YMS and YMD (Figure 18 (e)). This distinction is made by analyzing the portions of the lane without WBW patterns to identify which side has the solid line.

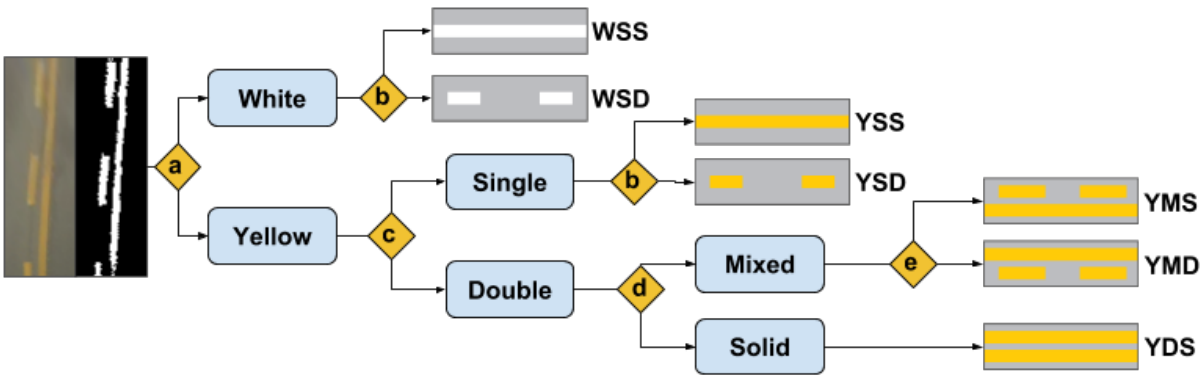


Figure 18 – Lane marking types (LMT) as detected and classified according to a threshold-based method. These are the LMTs: white-single-solid (WSS), white-single-dashed (WSD), yellow-single solid (YSS), yellow-single-dashed (YSD), yellow-mixed-solid (YMS), yellow-mixed-dashed (YMD) and yellow-double-solid (YDS). Decision processes (a, b, c, d and e) are described in Section 3.6

To avoid unstable reports caused by noisy lane markings (fading, covered by dust,

occluded by an obstacle or even during transition areas between different LMTs), a buffer mechanism was added. In these buffers (size 30), a winner-takes-all approach is used to report the LMT. The output of this module is a pair of LMT, one for the left and another for the right lane side.

3.7 Adjacent Lane Detection

Based on the final lane estimation and classified LMTs, ELAS looks for adjacent lanes. According to the Brazilian Manual of Traffic Sign, only “white single solid" (WSS) lane markings may not have an adjacent lane. Therefore, this is the only case the system needs to take a decision. In order to decide about the presence or not of adjacent lanes in these cases, a searching area taking the estimated lane width in account is considered. Given this area, the system checks two conditions (Figure 19): if there are Hough lines with angle close to the ego-lane angle (i.e. less than δ of difference, where $\delta = 15$) in the expected area of the adjacent lane; and if the space between this area and the ego-lane has a very small amount of Hough lines. For each lane side individually, when LMT is WSS and if both conditions are met, this module reports the existence of such adjacent lane based on this process. Otherwise, if LMT is other than WSS, the existence of an adjacent lane is presumed, based on the Brazilian Manual of Traffic Sign (DENATRAN, 2007). The lane marking color allows for deciding between same or opposite direction lane.

Figure 19 – Adjacent Lane Detection. The dark blue bars shows the histogram counts for the hough lines generated by the figure at the bottom of the image. The green region is the estimated lane region. The blue region is the searching region calculated based on the lane width of the current estimation. The red region is the space between the ego-lane and the adjacent lane.

3.8 Integration into an Autonomous Driving Framework

As one of the motivations for this research was the autonomous vehicle that is being developed by our laboratory, the proposed system was integrated into an autonomous driving framework. Currently, the High Performance Computing Laboratory (LCAD) of the Universidade Federal do Espírito Santo (UFES) is developing an autonomous car

named Intelligent Autonomous Robotic Automobile (IARA). To interact with IARA, the systems have to be deployed into CARMEN (Carnegie Mellon Robot Navigation Toolkit), the framework used by IARA. As the original CARMEN project was discontinued, LCAD maintains its own version of the framework. In order to avoid dependency of CARMEN to use the proposed system, a wrapper was developed as a module of CARMEN. To interact with other modules, ELAS dispatch messages with the information acquired in the Lane Analysis process using the protocol used by the framework. A more detailed description of the integration process is available in the Appendix A.

4 Materials and Experimental Methodology

In order to validate our system, we ran experiments using a novel dataset. Besides common performance measurements, a set of experiments were conducted to analyze the impact of some unexpected setups (e.g. low processing power and low image quality). To evaluate the integration of ELAS into CARMEN, a new dataset was created and qualitative experiments were performed.

4.1 Database

To validate our system where it is expected to run, we recorded and annotated a novel dataset. This dataset was recorded using a GoPRO HERO 3 camera in Brazil in different days at 29.97 frames per second. It comprises more than 20 scenes (i.e. more than 15,000 frames) containing all sorts of challenges we expect to encounter in the real world: highways, urban road, traffic, intersections, lane changes, writings on the road, shadows, wobbly capture, different weather, slightly different camera positions, faded lane markings, multiple LMT transitions, different vehicle speed, etc. Scenes were recorded in three cities: Vitória, Vila Velha and Guarapari on the state of Espírito Santo, Brazil (see Figure 21). Each frame has 640×480 pixels. This dataset was manually annotated for relevant events to the research community, such as: lane estimation, change, and centering; road markings; intersections; LMTs; crosswalks and adjacent lanes. Lane position ground truth generation was done following (BORKAR; HAYES; SMITH, 2012). During LMT transitions, the same strategy of (PAULA; JUNG, 2015) was adopted, where both types were annotated. The dataset will be made publicly available[1]. A detailed overview of the dataset can be seen at Appendix B.

Annotation Process While generating the database for quantitative analysis, the most time consuming task of generating was the annotation process. In particular, of all annotation tasks, annotating the lane position is the most expensive, in terms of time. As already said, to generate the ground truth of the lane position, the technique proposed by (BORKAR; HAYES; SMITH, 2012) was used. In contrast of individually annotating each frame, which is very slow, this procedure comprises of annotating a set of *Time-Sliced* images. A *Time-Sliced* (TS) image is generated by stacking a specific row of pixels from a video. Therefore, for a video that contains F frames with images having dimensions $M \times N$, a *Time-Sliced* image will have dimensions equal to $F \times N$. Aiming to increase the accuracy of this process, differently than proposed in the original technique, we chose four

[1] www.lcad.inf.ufes.br/wiki/index.php?title=Ego-Lane_Analysis_System

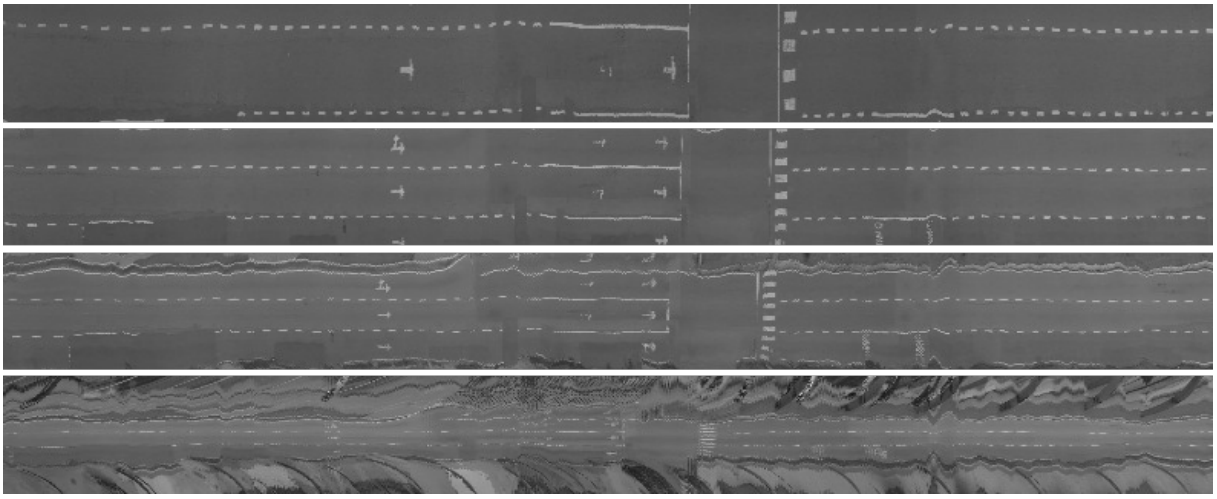


Figure 20 – *Time-Sliced* images from a single scene. For each scene, four TS images were generated based on the four selected rows to annotate the lane position ground truth. Annotating these images is equivalent of annotating over time. In this figure, TS images were rotated for visualization purposes.

rows instead of three, therefore generating four *Time-Sliced* images (Figure 20). Based on that, annotating a TS image is equivalent to annotating a given point of the images of a dataset over time. While annotating multiple TS images would be likely to annotate multiple points of each image over time. Based on these points, for each image, a cubic interpolation, as in the original technique, was applied to generate the ground truth. After that, each frame is individually validated by a human and, if required, they are adjusted to increase the accuracy of the ground truth.

Dataset with IARA Besides this annotated dataset, another one was captured with IARA. To generate this dataset, one of the Bumblebee cameras available on IARA was used: Bumblebee XB3, a 3-sensor stereo camera (Figure 22a). To create this dataset, we drove with IARA for an 8.4km route in the city of Vitória, Brazil. It was recorded in a cloudy week-day with normal traffic. To enable evaluation, we ensured that there were lane changes, lane marking type transitions, pavement signs, crosswalks, intersection, sometimes stopped at a traffic light, curves and many other things. To capture this database, we used an already existent module of CARMEN to capture the images from the camera and save them to disk. Although the maximum frame rate of this camera is 15Hz, we could only achieve an unstable frame rate varying from 10 to 14 frames per second. This instability was caused by this module and the fact that many other sensors data were also being recorded. Even though unstable frame rates should be avoided, it was also good in this case, because it enabled us to evaluate our system in an even more different scenario that it might need to work with. Alongside the difficulty added by unstable frame rate, this setup is very different from previous ones. In this case, there is a GPS antenna in front of the camera (Figure 22b), reducing its visibility and the camera was out of the car

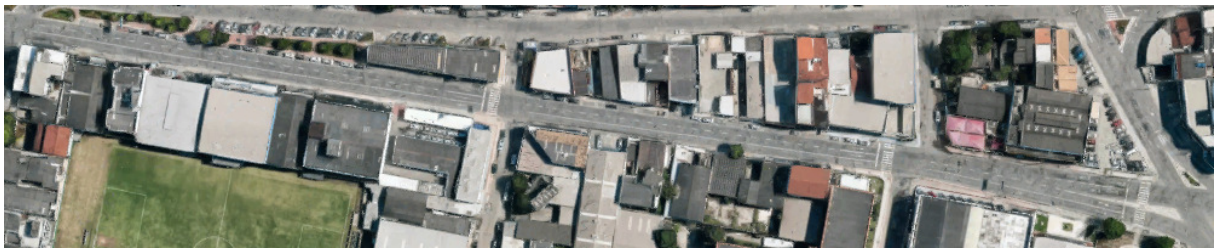
(a)

(b)

(c)

(d)

(e)



(f)

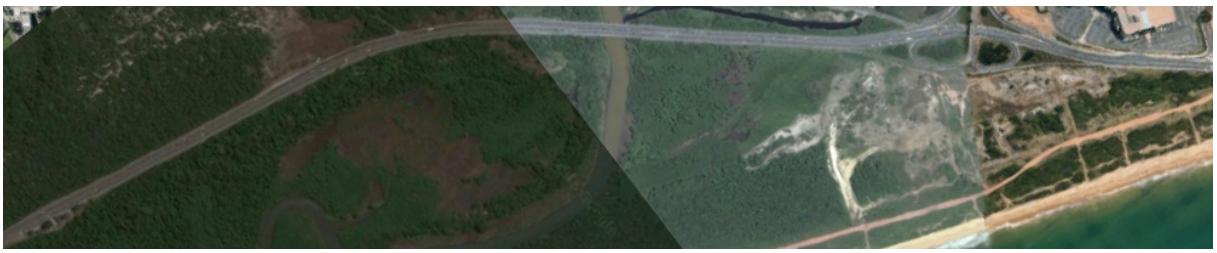
Figure 21 – Satellite view of some of the scenes in the dataset. Vitória: (a) BR_S01, (b) VIX_S05 and (c) VIX_S11; Guarapari: (d) GRI_S01 Vila Velha: (e) VV_S02 and (f) VV_S03. A more detailed overview and camera images can be seen at Appendix B.

(Figure 22c).

(a) (b) (c)

Figure 22 – Camera setup used in the dataset captured using the IARA. In (a), both of the available Bumblebee models. In (b), the GPS antenna that is in front of the chosen camera can be seen, and in (c) an image captured by the left camera from the stereo.

Qualitative Datasets In addition to these datasets, two highway scenes were used for qualitative evaluation. These two scenes (Figure 23) comprises of a 3.1km route of a rainy day in a highway in the city of Vila Velha and a long scene of 23.2km in a highway that connects the city of Vila Velha to Guarapari. Also, these long scenes were used to assess the stability and reliability of the system during long executions, and to verify its performance in different scenarios, such as tolls. Sample images from these datasets can be seen in the Appendix D.



(a)

(b)

Figure 23 – Satellite view of the two highway scenes. The first one (a) is of the 3.1km route and the second (b) is of the 23.2km route.

4.2 Metrics

There is no consensus on performance evaluation metrics when it comes to lane estimation. In (SATZODA; TRIVEDI, 2014b), the authors discuss about this problem and expose the different metrics used in the literature. The only common part on the evaluation protocols is the visual output analysis, i.e. qualitative analysis. In terms of quantitative evaluation, the authors discuss three metrics: accuracy of lane feature extraction, ego-vehicle localization and lane position deviation. The first metric, accuracy of lane feature extraction, is a measure of how accurate the extracted feature are from the actual estimate. For that, if the difference between them are below an acceptable error (usually correlated to the lane markings width), the estimate is assumed to be correct, otherwise it is wrong. The second metric, ego-vehicle localization (also known as lane center deviation), measures how far from the actual lane center the estimated lane center is. In practical terms, it reports the error in the ego-vehicle localization in the lane. It is worth mentioning that it is possible to have a low error on the lane center deviation despite of the accuracy of the lane features and lane estimator, although they are expected to have some correlation. Also, this metric is mainly based on the estimate near the vehicle. The third metric, lane position deviation (LPD), was introduced by the authors. In summary, it computes the average absolute distance in the x-direction between the ground truth and the estimate on the original image. Based on that, we propose a variation of this metric, shown in the Figure 24.

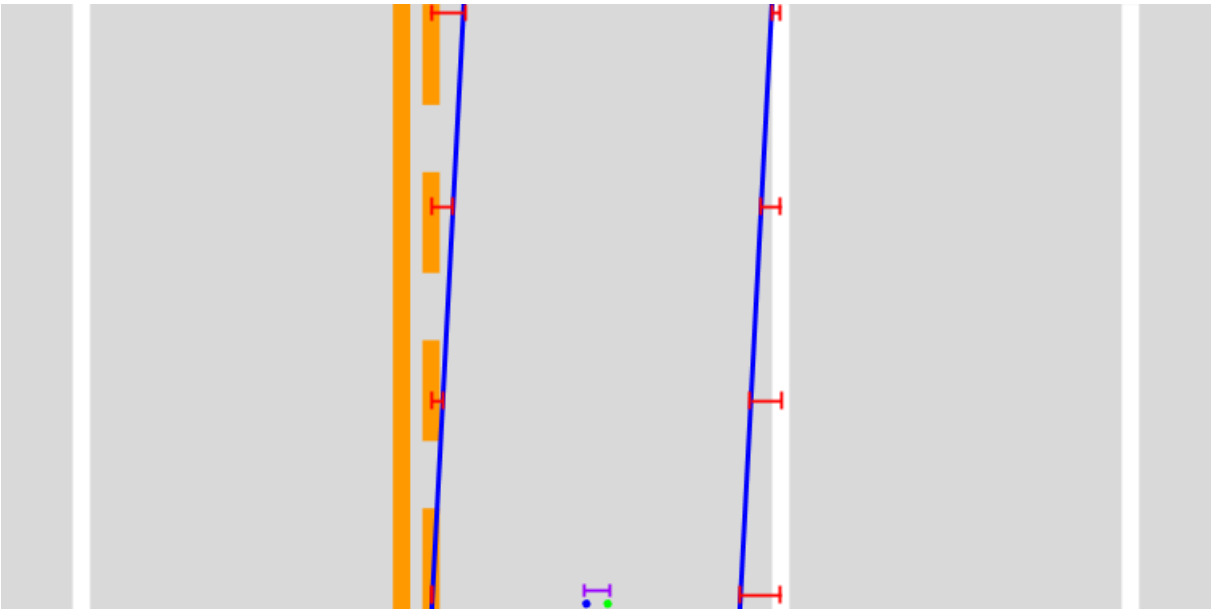


Figure 24 – Lane Position Metric. A variation to the LPD introduced in (SATZODA; TRIVEDI, 2014b) is proposed. The error is computed in the IPM image as the average distance (in red) of the lane estimate (blue lines) to the ground truth (background scene) divided by the lane width. Also, the lane center deviation (in purple) is calculated based on the closest point as the distance between the actual (green circle) and the estimated (blue circle) lane center.

To evaluate the lane estimation, four evaluation points were chosen. These points are equally distributed in the region of interest in respect to their distances from the car. In this way, the three initial points are considered as near and the last one as far. The absolute mean error of each lane side is reported. This error is the distance (in pixels) of the estimated position and the ground truth, and is reported in terms of percentage of the lane width. When compared to the original metric, there are some changes that worth noting. Firstly, this variation is computed in the IPM image instead of on the original image. In the original image, near and far regions are unbalanced. In absolute values, the same errors in the world coordinate system would lead to smaller errors on the far field, due the perspective distortion. Secondly, the error is split into two: near and far. This separation is useful because errors near the car are assumed to have more impact in the effectiveness of the systems that rely on this information. Finally, the error is reported relative to the width of the lane. In Brazil, lanes may vary from 2.4 up to 4.3 meters. Given the fact that a 10 centimeter error is relatively different — 4.1% to 2.3% (56% of difference) — depending on the width of the lane, it seems more appropriate to report a relative measure, instead of the absolute value as proposed in the original metric. Lane center deviation is derived from lane estimation. As lane departure warning (LDW) system rely on this information, the mean of the absolute lane center deviation is reported.

For the detection tasks (i.e. crosswalk, pavement signs, lane marking types and adjacent lanes), three metrics were reported: accuracy, precision and recall. Accuracy can be defined as the sum of the correct predictions over the total population, as defined in Equation 4.1. Precision (Equation 4.2) measures how many instances of the ones that the classifier took as correct are really correct, while recall (Equation 4.3) measures how many instances that are correct were correctly classified.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.3}$$

where TP, TN, FP and FN represent the number of true positives, true negatives, false positives and false negatives respectively. Additionally, execution time is reported in milliseconds.

4.3 Experiments

Each module was evaluated to measure their performances and understand their behaviors in various scenarios. For that, quantitative and qualitative experiments were

designed. The parameters used on the system were empirically defined using other scenes where the camera was at similar position.

4.3.1 Kalman with/without Particle Filter

Roughly, lane estimation is performed in two steps in our system: hough-based and particle filter-based steps. The former is required, but it is intrinsically limited in curved roads. The latter comes to fulfill this gap, estimating the curvature and giving robustness against the fixed ground plane assumptions that naturally fails sometimes. On the other hand, the latter also requires more processing capacity. This experiment consists on measuring the impact of the particle filter, analyzing the trade-off involved with the use of the particle filter and checking if it is viable to rely only in the hough-based output. As particle filter overall performance is directly related to the number of particles, we also experimented different number of particles $\{50, 100, 200, 400\}$.

4.3.2 Image quality

Our dataset was annotated in 640×480 pixels and was captured using a high quality camera. However, this is not always the case, and, this experiment measures ELAS performance with lower quality images. For quality here, we do not mean the image dimension, but quality in terms of noise. The main goal of this experiment is to assess the robustness of the system on images with noisier images, e.g. edges less sharp. To simulate this lower quality capture (i.e. to add noise), we down-sampled frames to 480×360 (75%), 320×240 (50%) and 160×120 (25%) using nearest-neighbor as the interpolation method and up-sampled back to 640×480 using a linear interpolation. This process causes loss of data (Figure 25). These experiments aim to test robustness against lower quality images without having to capture or annotate a dataset all over again.

(a) (b) (c) (d)

Figure 25 – With this technique, quality loss causes a degradation of the image, in special lane markings. From left to right: (a) 100%, (b) 75%, (c) 50% and (d) 25%.

4.3.3 Frame rate

Our dataset was captured at 29.97 frames per second (FPS). Nevertheless, the system might need to operate with lower frame rates due to a limitation in the camera, to

limited processing power or even to limited hardware (e.g.: slow hard drive). To simulate these scenarios of lower frame rates, some of the frames of our datasets were dropped in order to achieve 20, 15 and 10 frames per second. Lower the frame rate, weaker is the assumption of small variations and smooth transitions between two consecutive frames.

4.3.4 Qualitative Experiments

Other Scenes To assess the robustness of the proposed system over long runs — i.e. to verify its execution stability and robustness —, it was tested qualitatively on two long scenes, that together sums up to more than 25 kilometers.

Integration into CARMEN The dataset captured with IARA (Figure 26) was used to qualitatively evaluate the performance of the proposed system. The captured database contains a typical day. This was chosen because that is the scenario expected to be encountered by the IARA with the current version.

(a)

(b)

Figure 26 – Satellite view of the scenes recorded using IARA: (a) Av. Fernando Ferrari and (b) Av. Nossa Senhora da Penha (also known as Reta da Penha)

4.4 Setup

All experiments were performed on a desktop with Intel Core i7-4770 (3.40GHz) and 16GB RAM. ELAS was implemented in C++ using the open source library OpenCV. Despite this setup, ELAS used up to 15% of the processing power and up to 40MB of RAM during experiments.

5 Results and Discussion

Both qualitative and quantitative results are reported. The annotated dataset is used to provide quantitative evaluation and both, the annotated and the other dataset, were used to qualitatively evaluate the performance of the proposed system.

5.1 Qualitative Results

To qualitatively evaluate the proposed system a visual output was designed. The visual output has many elements: recognized lane marking types are pictured on the upper left and upper right; on the upper center of the output image, there is a gray box, where the lane center deviation is displayed as a filled box (green box means low deviation and red box means high deviation); on the image, lane orientation is denoted as a white arrow; the green overlay area denotes the area between the lane estimates; arrow markings and stop lines are displayed as a blue overlay area alongside their icons; crosswalks are displayed below the lane center deviation box.

Figure 27 – Qualitative results on the annotated dataset.

Figure 28 – Qualitative results on the dataset recorded using IARA.

Qualitative results were reported in both datasets: the annotated dataset (Figure 27) and the dataset recorded using IARA (Figure 28). More qualitative results on the annotated dataset can be seen in the Appendix E and more results on the dataset recorded using IARA can be seen in the Appendix C. Demonstration videos of ELAS are also available[1,2]. In the video, ELAS is executed on the annotated datasets and on the qualitative sequences (more than 25km).

[1] https://www.youtube.com/watch?v=NPU9tiyA8vw
[2] https://www.youtube.com/watch?v=R5wdPJ4ZI5M

5.2 Quantitative Results

The performance and accuracy impact of both scenarios (with and without Particle filter) can be seen in Figure 29. As expected, when the Particle filter is disabled (i.e. only Kalman output is being used and reported), there is a decrease in the execution time. As it can be seen, Kalman-only presents lower accuracy in the far region, where curvature happens. Additionally, a higher number of particles increases Particle filter accuracy at the cost of an increase in execution time. As all these configurations allow real-time execution (more than 30 frames per second), the best was chosen (PF400) to benchmark other experiments. PF400 configuration achieved mean absolute error of 1.3% (± 0.2) in the near region and 3.6% (± 0.3) in the far region executing in 27.2ms (± 2.9).

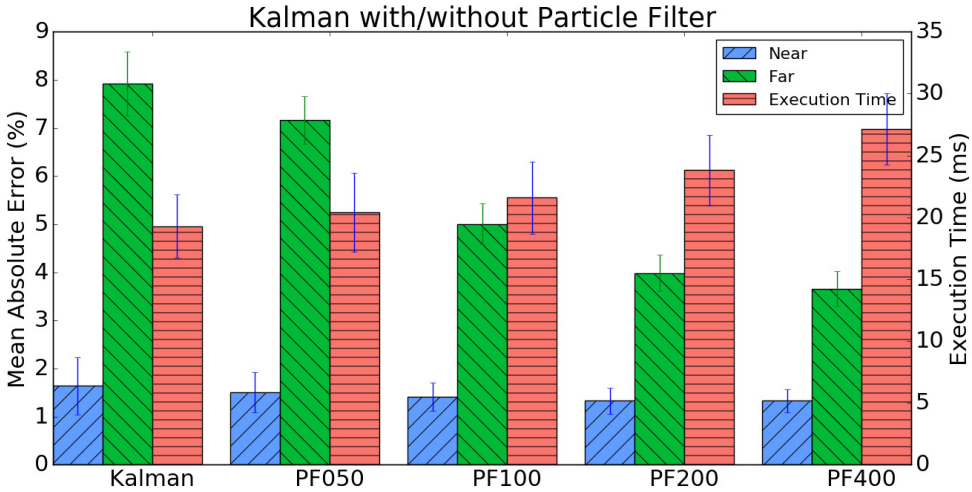


Figure 29 – The mean absolute error (%) of the near and far region are reported. Additionally, execution time (ms) is shown. In this case, particle filter was tested with 50, 100, 200 e 400 particles (PF50, PF100, PF200 and PF400, respectively).

As can be seen in Table 1, ELAS achieves an average of 86.7% (± 2.8) in the classification accuracy (90.9% of precision and 83.3% of recall) of road signs and crosswalks. As an example of misclassification, a crosswalk was being intermittently classified between "unknown" and crosswalk because of a crosswalk variation containing arrows. Our LMT detector overall classification rate was 93.1% (± 1.9), with the lowest accuracy in the rainy scene. The precision of the LMT detector was 95.2% with 90.9% recall. It was also noted that, usually, LMT misclassifications are from yellow-double to yellow-single-solid and white-dashed to white-solid. This is less dangerous than misclassifying in the opposite direction, since these cases can lead control systems to take dangerous decisions (e.g. allow a lane change while it should not). Also, one of the advantages of our approach is that it does not need training (PAULA; JUNG, 2015). On the other hand, one limitation is the color detection in cases where overall color is expected to be changed (e.g. snow and dusty windshield in a sunny day). By design, our adjacent lane detector performance is constrained by the accuracy of our LMT detector. Therefore, the overall adjacent lane

detection rate was 89.4% (± 2.6), but with a precision of 80.1% with 86.9% of recall. The common problems of this detector are related to occlusions of the adjacent lane markings.

Classification Task	Accuracy (%)	Precision (%)	Recall(%)
Road Signs and Crosswalks	86.7 ± 2.8	90.9	83.3
Lane Markings Type	93.1 ± 1.9	95.2	90.9
Adjacent Lanes	89.4 ± 2.6	80.1	86.9

Table 1 – Performance measurement of the classification tasks.

We also wanted to validate our system in other setups. Due to the prohibitive cost of generating the database again, we used some techniques to simulate these setups. The first one was lower frame rate (i.e. frames per second, FPS). As expected, lower FPS causes lower accuracy (Figure 30), but ELAS maintains reasonable estimation quality even on 10 FPS.

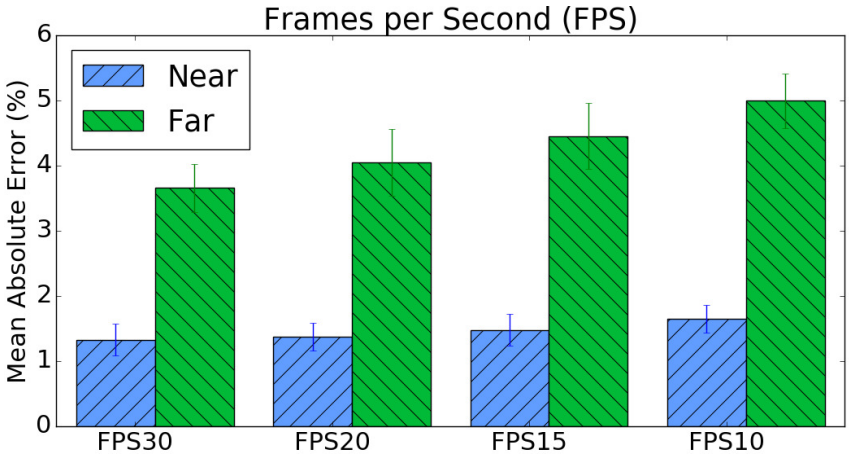


Figure 30 – Frames per Second. ELAS was tested with 30, 20, 15 and 10 frames per second (FPS30, FPS20, FPS15, FPS10, respectively). All these tests were performed using PF400. In fact, FPS30 was reported again for completeness, because it is the same as PF400.

The second one was lower image quality (Figure 31). Based on the technique described earlier, some noise was added to the image as if the image had come from a lower resolution camera or a lower quality one. As expected, lower quality causes a decrease in the overall accuracy, but again the system was capable of maintaining a good estimation quality.

Lane center deviation is accurately calculated, with an absolute mean error of 0.9% (± 0.18). Based on this lane center deviation, all lane changes are detected. Execution time varies according to the amount of elements to process in a given frame. As reported in Table 2, ELAS devotes most of its pipeline to lane estimation (Hough lines + Kalman + Particle filters). Even though, the system runs in real-time (33+FPS) considering the used experiment setup. Overall execution time can be further improved by parallel implementation, benefiting from the modular approach.

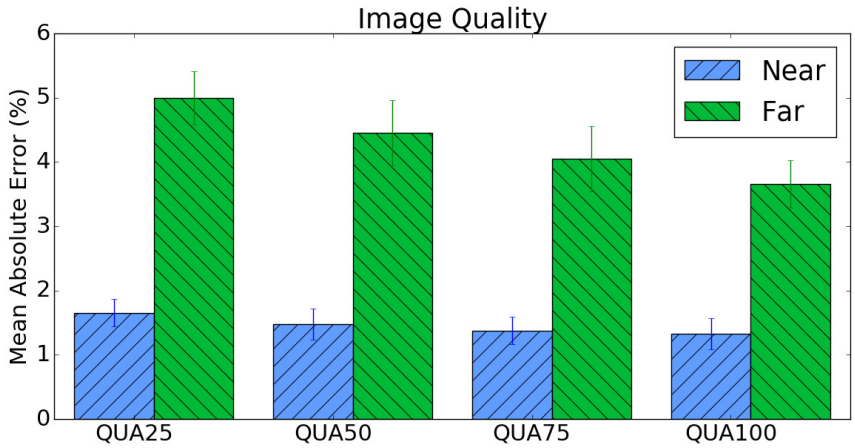


Figure 31 – Image Quality. Lower image qualities were simulated based on the image resolutions where 100% (QUA100) represents the original image and the others represent the relative resolutions: 75%, 50% and 25% (QUA75, QUA50, QUA25, respectively).

Execution Time (ms)	
Feature Maps Generation	2.64 ± 0.27
Crosswalk and Road Signs Detection and Removal	1.59 ± 0.87
Candidates Generation and Kalman	8.46 ± 0.76
Particle Filter	7.57 ± 0.55
Lane Marking Type Detection	6.91 ± 0.44
Adjacent Lane Detection	0.03 ± 0.01
Total	**27.2 ± 2.90**

Table 2 – Execution Time Performance of ELAS with 400 particles (in ms)

Finally, ELAS was compared with Berriel et al. (2015) in the newly proposed dataset. As it can be seen in Figure 32, ELAS lane estimation outperforms Berriel et al. (2015).

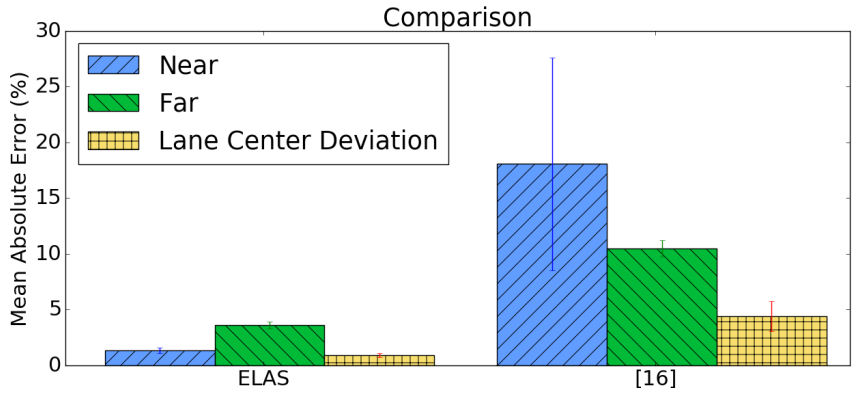


Figure 32 – Comparison between ELAS and Berriel et al. (2015)

5.3 Discussion

ELAS has proved to be ready for real-world real-time applications, but it also has some limitations. By choice, the proposed system is limited to a calibrated camera. Although many other works also have this limitation (RASMUSSEN, 2004; NIETO et al.,

2016), some of them (PAULA; JUNG; SILVEIRA, 2014) have already performed on-the-fly calibration with good results. Still, there is need for more accurate and flexible online calibration. ELAS is also limited to few specific pavement markings (arrows and stop-line). Even though most of the related works (SUCHITRA; SATZODA; SRIKANTHAN, 2012) are also limited to arrows, there is still room for more general algorithms. Also, locally darker regions on the asphalt, such as potholes or due to asphalt resurfacing, may lead to misdetection of obstacles causing sudden changes on the trustworthy area. Most of the time, these locally darker regions are classified are *unknown* by the road signs classification module, because it is not part of the known set of pavement markings. In some cases, the ego-lane differs from adjacent lanes due to asphalt resurfacing. In these cases, even when the resurfacing occludes lane markings for a while, the proposed system is able to robustly maintain the estimates based on temporal information.

(a) (b) (c)

Figure 33 – Some types of failures of the system, based on naive assumptions and side effects of particular techniques.

In the qualitative experimentation, partially shown in the videos, there are some failures cases. Three types of them are highlighted on the Figure 33. In the first case (Figure 33a), the adjacent lane detection fails. What lead to this failure is that the proposed detector naively assumes that any evidence of Hough lines at a lane width of distance from the ego-lane is enough for assuming that there is an adjacent lane there. In this particular case, the end a paved shoulder generates such evidences and it mislead the detector to a false positive. In the second case (Figure 33b), the LMT detector fails. This failure exposes another naive assumption. The LMT detector selects a region around the estimated lane and naively assumes that all the evidences in this region are related to the lane markings. In this particular case, the region is just too big that it erroneously accounts evidences generated by the bus and this leads to a misclassification. In the third case (Figure 33c), the effect of the LMT detector buffers can be seen. These buffers caused a failure during the lane markings type transitions. This effect, named as "lag" in (PAULA; JUNG, 2015), can also be seen in other works in the literature. Although this type of error may have less impact than others, they expose an undesired side effect of a commonly used technique.

Also, a few relevant metrics were chosen to report the results of the proposed work. In (SATZODA; TRIVEDI, 2014b), the authors discuss about performance evaluation

metrics and acknowledge the heterogeneous reports. There is still need for a more detailed study of discriminative metrics.

As stated in a recent survey (HILLEL et al., 2014), the absence public of benchmarks poses as a big challenge on current research. There are at least two factor that contributes to this: most of the works in this topic do not publish the datasets used in their studies (NIETO et al., 2016; MASCETTI et al., 2016; SATZODA; TRIVEDI, 2015; GAIKWAD; LOKHANDE, 2015); and most of them also do not publish their implementation (PAULA; JUNG, 2015; SHIN; TAO; KLETTE, 2015; SON et al., 2015; LIU et al., 2015). Data gathering and annotation is an expensive activity, often put aside by researchers. When a dataset is collected, most of them are kept in private, because data is an invaluable resource, even more nowadays with the increasing usage of deep learning techniques. Nevertheless, the existence of such benchmark has proved to increase progress rate in other fields such as pedestrian detection (DALAL; TRIGGS, 2005). This lack of public resources forbids researchers in this topic to perform fair comparisons between each other. That is why the dataset and algorithms developed in this work are both open to free public access.

Despite the lack of datasets for any of the tasks discussed in this work, there is a need for datasets annotated to a wider range of tasks. The dataset hereby presented, to the best of our knowledge, is the first publicly available Brazilian dataset annotated for a wide range of tasks of interest to the research community.

About the integration process, it is important to note some points: the overall image – in terms of contrast, brightness, visibility – is completely different, therefore not relying on absolute intensity values had a major impact on integrating the system. In terms of implementation, designing the system as a standalone application also made it easier to integrate into CARMEN. As CARMEN is a framework that adds lots of undesired dependencies, we wanted to keep the proposed system detached from it, so that other researchers could test the proposed system without such overhead. Besides these, the unexpected and temporary GPS antenna in front of the camera was also something that required a change. Since the system was calibrated such that it was expecting a forward-looking camera centered on the car, and it was need to use the right camera instead of the left one (most centered camera) as input sensor, the bias of the lane center deviation needed to be revised.

6 Conclusions

In this work, a system (ELAS) for vision-based ego-lane analysis is presented. This system uses a sequence of frames from a monocular forward-looking camera. Feature maps are extracted and analyzed in order to estimate a lane model. A parallel-based spline model was used to fit a wide range of road geometries. Kalman and Particle filters were used to increase the robustness of the model, at the same time of incorporating temporal information by performing lane tracking. Additionally, a novel dataset with more than 20 scenes (15,000+ frames) was generated. This dataset was captured in three different cities (Vitória, Vila Velha and Guarapari) from Brazil. It consists of several different scenarios varying the capture setup in different days. This dataset was manually annotated and made publicly available to enable other researchers to continue the development of this topic and to enable future fair comparisons, which is fairly difficult nowadays. Also, the implementation of ELAS is publicly available. Besides that, the preliminary integration process of this system into a real-world autonomous vehicle is discussed. It was discussed that ELAS, as well as many other systems in the literature, have some parameters and thresholds that may need to be tuned. The advantage of using relative thresholds instead of absolute ones proved to be useful, especially in the integration process, where only subtle changes were required despite of the high different in the image input.

Our experiments evaluated ELAS using real-world driving environments. There was also different capture setup involved, including wobbling capture, different camera angles, besides scenarios variations. Results showed this system presents high performance in lane estimation and high classification rates on the other examined tasks (i.e. road signs, crosswalks, LMT and adjacent lanes). Additionally, lane center deviation results proved the system can be used to perform lane departure warning accurately, as well as lane change detection. Moreover, experimental results validated its robustness against lower quality and lower frame rates. During the integration of ELAS to CARMEN, another dataset was captured and used to qualitatively evaluate the performance of ELAS. Ultimately, results showed that ELAS is able to robustly perform real-time vision-based ego-lane analysis.

6.1 Future Works

As future works, we want to investigate how the parameters of the system can be automatically (or semi-automatically) adjusted based on prior camera calibration. In the preprocessing and feature extraction phase, machine learning techniques, such as convolutional neural networks (CNN) (LECUN et al., 1990), will be studied. Also, we will investigate how CNNs can be used to significantly improved the results of the related

tasks, perhaps solving many of them at once. Also, in the preprocessing, we will study how to incorporate automatic calibration (PAULA; JUNG; SILVEIRA, 2014) (offline or/and online) of the camera without compromising performance. In the same perspective, we will study dynamic selection of the region of interest based on the camera parameters.

Due to the integration of the system into an autonomous car, studies will be carried out to fuse other sensors data (e.g.: vehicle speed, LiDAR data, GPS, etc.) in order to increase the overall performance and robustness of the system. Vehicle speed can be useful to adjust the lane estimator when the car is stopped (traffic jam, traffic light, etc.). LiDAR sensors, such as the Velodyne available on IARA, can be used to improve robustness on scenes with hard shadows or sudden illumination changes. Localization data can be used to allow for reuse of information on previously known places, therefore increasing robustness as well. Also, experimentation will be extended to an even more diverse set of scenes.

Publications

As a result of this work, two publications were produced, one published in 2015 and another currently (2016) under review:

- BERRIEL, R., de Aguiar, E., Filho, V. V. de S., Oliveira-Santos, T. A Particle Filter-based Lane Marker Tracking Approach using a Cubic Spline Model. *In: 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images.* IEEE, 2015. p. 149–156. ISSN 1530-1834.
- BERRIEL, R., de Aguiar, E., Filho, de Souza, A. F., Oliveira-Santos, T. Ego-Lane Analysis System (ELAS): dataset and algorithms. *Under Review on Image and Vision Computing (Special Issue: Automotive Vision).* 2016.

Bibliography

AHMETOVIC, D. et al. Zebra crossing spotter: Automatic population of spatial databases for increased safety of blind travelers. In: *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*. New York, NY, USA: ACM, 2015. (ASSETS '15), p. 251–258. ISBN 978-1-4503-3400-6. Cited on page 39.

ALY, M. Real time detection of lane markers in urban streets. In: *Intelligent Vehicles Symposium, 2008 IEEE*. Eindhoven: IEEE, 2008. p. 7–12. ISSN 1931-0587. Cited 2 times on pages 34 and 36.

BERRIEL, R. et al. A particle filter-based lane marker tracking approach using a cubic spline model. In: *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*. Salvador,Bahia: IEEE, 2015. p. 149–156. ISSN 1530-1834. Cited 4 times on pages 18, 35, 53, and 70.

BERTOZZI, M.; BROGGI, A. Gold: A parallel real-time stereo vision system for generic obstacle and lane detection. *Image Processing, IEEE Transactions on*, IEEE, v. 7, n. 1, p. 62–81, 1998. Cited on page 26.

BÓDIS-SZOMORÚ, A.; DABÓCZI, T.; FAZEKAS, Z. A far-range off-line camera calibration method for stereo lane detection systems. In: IEEE. *2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007*. Warsaw, 2007. p. 1–6. Cited on page 33.

BORKAR, A.; HAYES, M.; SMITH, M. T. Robust lane detection and tracking with ransac and kalman filter. In: *Proceedings of the 16th IEEE International Conference on Image Processing*. Piscataway, NJ, USA: IEEE Press, 2009. (ICIP'09), p. 3225–3228. ISBN 978-1-4244-5653-6. ISSN 1522-4880. Cited on page 37.

BORKAR, A.; HAYES, M.; SMITH, M. T. A novel lane detection system with efficient ground truth generation. *IEEE Transactions on Intelligent Transportation Systems*, v. 13, n. 1, p. 365–374, March 2012. ISSN 1524-9050. Cited on page 59.

BORKAR, A. et al. A layered approach to robust lane detection at night. In: *Computational Intelligence in Vehicles and Vehicular Systems, 2009. CIVVS '09. IEEE Workshop on*. Nashville, TN: IEEE, 2009. p. 51–57. Cited 2 times on pages 33 and 37.

BROGGI, A. Parallel and local feature extraction: a real-time approach to road boundary detection. *Image Processing, IEEE Transactions on*, IEEE, v. 4, n. 2, p. 217–223, 1995. Cited on page 35.

BROGGI, A.; CATTANI, S. An agent based evolutionary approach to path detection for off-road vehicle guidance. *Pattern Recognition Letters*, Elsevier, v. 27, n. 11, p. 1164–1173, 2006. Cited on page 36.

CHEN, C. et al. DeepDriving: Learning affordance for direct perception in autonomous driving. In: *The IEEE International Conference on Computer Vision (ICCV)*. Santiago: IEEE, 2015. Cited on page 35.

CHENG, H.-Y. et al. Lane detection with moving vehicles in the traffic scenes. *Intelligent Transportation Systems, IEEE Transactions on*, IEEE, v. 7, n. 4, p. 571–582, 2006. Cited on page 36.

CRISMAN, J. D.; THORPE, C. E. UNSCARF-a color vision system for the detection of unstructured roads. In: *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*. Sacramento, CA: IEEE, 1991. p. 2496–2501 vol.3. Cited on page 34.

CRISMAN, J. D.; THORPE, C. E. Scarf: A color vision system that tracks roads and intersections. *Robotics and Automation, IEEE Transactions on*, v. 9, n. 1, p. 49–58, 1993. Cited on page 34.

DAHLKAMP, H. et al. Self-supervised monocular road detection in desert terrain. In: *Proceedings of Robotics: Science and Systems*. Philadelphia, USA: MIT Press, 2006. Cited on page 37.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: IEEE. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. San Diego, CA, USA, 2005. v. 1, p. 886–893. Cited on page 72.

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY. *DARPA Grand Challenges*. 2007. Available from Internet: <http://archive.darpa.mil/grandchallenge/>. Visited on: 02/06/2016. Cited on page 37.

DENATRAN - Departamento Nacional de Trânsito. *Manual Brasileiro de Sinalização de Trânsito: Sinalização Horizontal*. Brasília, Brazil, 2007. v. 4. Cited 4 times on pages 15, 40, 47, and 57.

EUREKA. *Programme for a European Traffic System with Highest Efficiency and Unprecedented Safety (PROMETHEUS)*. 1987. Available from Internet: <http://www.eurekanetwork.org/project/id/45>. Visited on: 24/05/2016. Cited on page 25.

FERNANDEZ-MALOIGNE, C.; BONNET, W. Texture and neural network for road segmentation. In: *Intelligent Vehicles '95 Symposium., Proceedings of the*. Detroit, MI: IEEE, 1995. p. 344–349. Cited on page 35.

FOEDISCH, M.; TAKEUCHI, A. Adaptive real-time road detection using neural networks. In: *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*. Washington, DC: IEEE, 2004. p. 167–172. Cited on page 34.

GAIKWAD, V.; LOKHANDE, S. Lane departure identification for advanced driver assistance. *IEEE Transactions on Intelligent Transportation Systems*, p. 910–918, April 2015. ISSN 1524-9050. Cited 2 times on pages 34 and 72.

GAO, Q.; LUO, Q.; MOLI, S. Rough set based unstructured road detection through feature learning. In: *2007 IEEE International Conference on Automation and Logistics*. Jinan, CN: IEEE, 2007. p. 101–106. ISSN 2161-8151. Cited on page 34.

GORDON, N. J.; SALMOND, D. J.; SMITH, A. F. M. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F - Radar and Signal Processing*, v. 140, n. 2, p. 107–113, April 1993. ISSN 0956-375X. Cited on page 53.

GUO, C.; MITA, S. Drivable road region detection based on homography estimation with road appearance and driving state models. In: *Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on*. Wellington, NZ: IEEE, 2009. p. 204–209. Cited on page 34.

HAGA, T.; SASAKAWA, K.; KURODA, S. I. The detection of lane boundary markings using the modified spoke filter. In: *Intelligent Vehicles' 95 Symposium., Proceedings of the*. Detroit, MI: IEEE, 1995. p. 293–298. Cited on page 34.

HASELHOFF, A.; KUMMERT, A. On visual crosswalk detection for driver assistance systems. In: IEEE. *Intelligent Vehicles Symposium (IV), 2010 IEEE*. San Diego, CA, 2010. p. 883–888. Cited on page 39.

HILLEL, A. B. et al. Recent progress in road and lane detection: A survey Recent progress in road and lane detection : a survey. *Machine Vision and Applications*, v. 25, n. 3, p. 727–745, 2014. Cited 3 times on pages 27, 34, and 72.

HUANG, A. et al. Multi-sensor lane finding in urban road networks. In: *Robotics: Science and Systems*. Zurich, Switzerland: MIT Press, 2008. Cited on page 26.

HUANG, A. S. *Lane estimation for autonomous vehicles using vision and lidar*. Thesis (PhD) — Massachusetts Institute of Technology, 2010. Cited on page 35.

HUVAL, B. et al. An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*, 2015. Cited on page 35.

IPEA - INSTITUTO DE PESQUISA ECONÔMICA APLICADA. *Acidentes de Trânsito nas Rodovias Federais Brasileiras*: caracterização, tendências e custos para a sociedade. 2015. Available from Internet: <http://www.ipea.gov.br/portal/images/stories/PDFs-/relatoriopesquisa/150922_relatorio_acidentes_transito.pdf>. Visited on: 24/05/2016. Cited on page 25.

IVANCHENKO, V.; COUGHLAN, J.; SHEN, H. Detecting and locating crosswalks using a camera phone. In: IEEE. *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*. Anchorage, AK, 2008. p. 1–8. Cited on page 39.

JIANG, R. et al. New lane model and distance transform for lane detection and tracking. In: *Computer Analysis of Images and Patterns: 13th International Conference, CAIP 2009, Münster, Germany, September 2-4, 2009. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 1044–1052. Cited on page 37.

KALMAN, R. E. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, American Society of Mechanical Engineers, v. 82, n. 1, p. 35–45, 1960. Cited on page 51.

KIM, Z. Realtime lane tracking of curved local road. In: IEEE. *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*. Toronto, Ont., 2006. p. 1149–1155. Cited on page 36.

KIRYATI, N.; ELDAR, Y.; BRUCKSTEIN, A. M. A probabilistic hough transform. *Pattern recognition*, Elsevier, v. 24, n. 4, p. 303–316, 1991. Cited on page 48.

KREUCHER, C.; LAKSHMANAN, S.; KLUGE, K. A driver warning system based on the LOIS lane detection algorithm. In: *Proceedings of IEEE International Conference on Intelligent Vehicles*. Stuttgart, Germany: IEEE, 1998. v. 1, p. 17–22. Cited on page 36.

KUK, J. G. et al. Fast lane detection & tracking based on hough transform with reduced memory requirement. In: *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. Funchal: IEEE, 2010. p. 1344–1349. Cited 2 times on pages 35 and 36.

LECUN, Y. et al. Advances in neural information processing systems 2. In: TOURETZKY, D. S. (Ed.). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990. cap. Handwritten Digit Recognition with a Back-propagation Network, p. 396–404. ISBN 1-55860-100-7. Cited on page 73.

LEE, J. W. A machine vision system for lane-departure detection. *Computer vision and image understanding*, Elsevier, v. 86, n. 1, p. 52–78, 2002. Cited on page 38.

LEWIS, J. P. Fast template matching. In: *Vision interface*. Quebec: Canadian Image Processing and Pattern Recognition Society, 1995. p. 120–123. Cited on page 47.

LI, C. et al. Multi-lane detection based on multiple vanishing points detection. In: *Sixth International Conference on Graphic and Image Processing (ICGIP 2014)*. Beijing, China: SPIE, 2015. v. 9443, p. 94430E–94430E. Cited on page 34.

LI, Q. et al. A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios. *IEEE Transactions on Vehicular Technology*, IEEE, v. 63, n. 2, p. 540–555, 2014. Cited on page 31.

LIPSKI, C. et al. A fast and robust approach to lane marking detection and lane tracking. In: IEEE. *Image Analysis and Interpretation, 2008. SSIAI 2008. IEEE Southwest Symposium on*. Santa Fe, NM, 2008. p. 57–60. Cited on page 36.

LIU, W. et al. Multi-type road marking recognition using adaboost detection and extreme learning machine classification. In: IEEE. *2015 IEEE Intelligent Vehicles Symposium (IV)*. Seoul, 2015. p. 41–46. Cited 2 times on pages 40 and 72.

LUM, H.; REAGAN, J. A. Interactive highway safety design model: accident predictive module. *Public Roads*, v. 58, n. 3, 1995. Cited on page 25.

MA, B.; LAKSHMANAN, S.; HERO, A. O. Pavement boundary detection via circular shape models. In: IEEE. *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*. Dearborn, MI, 2000. p. 644–649. Cited on page 36.

MALLOT, H. A. et al. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological cybernetics*, Springer, v. 64, n. 3, p. 177–185, 1991. Cited on page 42.

MARAGOS, P.; SCHAFER, R. Morphological skeleton representation and coding of binary images. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 34, n. 5, p. 1228–1244, oct 1986. ISSN 0096-3518. Cited on page 48.

MARR, D.; HILDRETH, E. Theory of edge detection. *Proceedings of the Royal Society of London B: Biological Sciences*, The Royal Society, v. 207, n. 1167, p. 187–217, 1980. Cited 2 times on pages 34 and 43.

MASCETTI, S. et al. Zebrarecognizer: Pedestrian crossing recognition for people with visual impairment or blindness. *Pattern Recognition*, Elsevier, 2016. Cited 2 times on pages 39 and 72.

MCCALL, J. C.; TRIVEDI, M. M. Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *Intelligent Transportation Systems, IEEE Transactions on*, v. 7, n. 1, p. 20–37, 2006. Cited 3 times on pages 26, 34, and 37.

MONTEMERLO, M.; ROY, N.; THRUN, S. Perspectives on standardization in mobile robot programming: The carnegie mellon navigation (carmen) toolkit. In: IEEE. *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*. Las Vegas, Nevada, USA, 2003. v. 3, p. 2436–2441. Cited on page 88.

NEDEVSCHI, S. et al. 3D lane detection system based on stereovision. In: IEEE. *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*. Washington, WA, USA, 2004. p. 161–166. Cited 2 times on pages 34 and 37.

NIETO, M. et al. Real-time lane tracking using Rao-Blackwellized particle filter. *Journal of Real-Time Image Processing*, 2012. ISSN 1861-8200. Cited on page 34.

NIETO, M. et al. Real-time lane tracking using Rao-Blackwellized particle filter. *Journal of Real-Time Image Processing*, Springer, v. 11, n. 1, p. 179–191, 2016. ISSN 1861-8219. Cited 4 times on pages 37, 70, 71, and 72.

NIETO, M.; LABORDA, J. A.; SALGADO, L. Road environment modeling using robust perspective analysis and recursive bayesian segmentation. *Machine Vision and Applications*, v. 22, n. 6, p. 927–945, aug 2010. ISSN 0932-8092. Cited on page 43.

NIETO, M. et al. Robust multiple lane road modeling based on perspective analysis. In: *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. San Diego, CA: IEEE, 2008. p. 2396–2399. Cited on page 37.

PAULA, M. B. d.; JUNG, C. R.; SILVEIRA, L. da. Automatic on-the-fly extrinsic camera calibration of onboard vehicular cameras. *Expert Systems with Applications*, Elsevier, v. 41, n. 4, p. 1997–2007, 2014. Cited 2 times on pages 71 and 74.

PAULA, M. Braga de; JUNG, C. R. Automatic detection and classification of road lane markings using onboard vehicular cameras. *IEEE Transactions on Intelligent Transportation Systems*, v. 16, n. 6, p. 3160–3169, Dec 2015. ISSN 1524-9050. Cited 6 times on pages 33, 39, 59, 68, 71, and 72.

RASMUSSEN, C. Grouping dominant orientations for ill-structured road following. In: *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. Washington, DC: IEEE, 2004. v. 1, p. I–470–I–477 Vol.1. ISSN 1063-6919. Cited 5 times on pages 32, 35, 37, 70, and 71.

ROTARU, C.; GRAF, T.; ZHANG, J. Extracting road features from color images using a cognitive approach. In: *Intelligent Vehicles Symposium, 2004 IEEE*. Parma, Italy: IEEE, 2004. p. 298–303. Cited on page 37.

SATZODA, R.; TRIVEDI, M. Vision-based lane analysis: Exploration of issues and

approaches for embedded realization. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. Portland, OR: IEEE, 2013. p. 604–609. Cited on page 39.

SATZODA, R.; TRIVEDI, M. M. Selective salient feature based lane analysis. In: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. The Hague, The Netherlands: IEEE, 2013. p. 1906–1911. Cited on page 32.

SATZODA, R. K.; SUCHITRA, S.; SRIKANTHAN, T. Gradient angle histograms for efficient linear hough transform. In: *Image Processing (ICIP), 2009 16th IEEE International Conference on*. Cairo, Egypt: IEEE, 2009. p. 3273–3276. Cited on page 36.

SATZODA, R. K.; TRIVEDI, M. M. Efficient lane and vehicle detection with integrated synergies (ELVIS). In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Columbus, OH: IEEE, 2014. p. 708–713. ISSN 2160-7508. Cited on page 26.

SATZODA, R. K.; TRIVEDI, M. M. On performance evaluation metrics for lane estimation. In: *2014 22nd International Conference on Pattern Recognition (ICPR)*. Stockholm, Sweden: IEEE, 2014. p. 2625–2630. Cited 4 times on pages 17, 35, 63, and 71.

SATZODA, R. K.; TRIVEDI, M. M. On enhancing lane estimation using contextual cues. *IEEE Transactions on Circuits and Systems for Video Technology*, p. 1870–1881, Nov 2015. ISSN 1051-8215. Cited 2 times on pages 34 and 72.

SCHOENBERG, I. J. Contributions to the problem of approximation of equidistant data by analytic functions, part b: On the problem of osculatory interpolation, a second class of analytic approximation formulae. *Quaterly Applied Mathematics*, v. 4, n. 2, p. 45–99 and 112–141, 1946. Cited on page 36.

SHIN, B.-S.; TAO, J.; KLETTE, R. A superparticle filter for lane detection. *Pattern Recognition*, v. 48, n. 11, p. 3333 – 3345, 2015. ISSN 0031-3203. Cited 2 times on pages 37 and 72.

SHIN, B.-S.; XU, Z.; KLETTE, R. Visual lane analysis and higher-order tasks: a concise review. *Machine vision and applications*, Springer, v. 25, n. 6, p. 1519–1547, 2014. Cited on page 26.

SON, J. et al. Real-time illumination invariant lane detection for lane departure warning system. *Expert Systems with Applications*, Elsevier, v. 42, n. 4, p. 1816–1824, 2015. Cited 2 times on pages 38 and 72.

SOQUET, N.; AUBERT, D.; HAUTIERE, N. Road segmentation supervised by an extended v-disparity algorithm for autonomous navigation. In: *2007 IEEE Intelligent Vehicles Symposium*. Istanbul: IEEE, 2007. p. 160–165. ISSN 1931-0587. Cited on page 34.

SUCHITRA, S.; SATZODA, R. K.; SRIKANTHAN, T. Detection & classification of arrow markings on roads using signed edge signatures. In: *Intelligent Vehicles Symposium (IV), 2012 IEEE*. Alcala de Henares: IEEE, 2012. p. 796–801. ISSN 1931-0587. Cited 2 times on pages 39 and 71.

THRUN, S.; BURGARD, W.; FOX, D. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. Cambridge, MA: The MIT Press, 2005. ISBN 0262201623. Cited on page 54.

WANG, C.-C.; HUANG, S.-S.; FU, L.-C. Driver assistance system for lane detection and vehicle recognition with night vision. In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Alberta, Canada: IEEE, 2005. p. 3530–3535. ISSN 2153-0858. Cited on page 34.

WANG, S. et al. Rgb-d image-based detection of stairs, pedestrian crosswalks and traffic signs. *Journal of Visual Communication and Image Representation*, Elsevier, v. 25, n. 2, p. 263–272, 2014. Cited on page 39.

WANG, Y.; DAHNOUN, N.; ACHIM, A. A novel system for robust lane detection and tracking. *Signal Processing*, v. 92, n. 2, p. 319–334, 2012. Cited on page 34.

WANG, Y.; SHEN, D.; TEOH, E. K. Lane detection using spline model. *Pattern Recognition Letters*, Elsevier, v. 21, n. 8, p. 677–689, 2000. Cited on page 32.

WANG, Y.; TEOH, E. K.; SHEN, D. Lane detection and tracking using b-snake. *Image and Vision computing*, Elsevier, v. 22, n. 4, p. 269–280, 2004. Cited 2 times on pages 35 and 36.

WHO - World Health Organization. *WHO global status report on road safety 2013: supporting a decade of action*. Geneva, 2013. Cited on page 25.

WHO - World Health Organization. *WHO global status report on road safety 2015*. Geneva, 2015. Cited on page 25.

WIJESOMA, W. S.; KODAGODA, K. S.; BALASURIYA, A. P. Road-boundary detection and tracking using ladar sensing. *Robotics and Automation, IEEE Transactions on*, IEEE, v. 20, n. 3, p. 456–464, 2004. Cited on page 26.

WU, B.-F.; LIN, C.-T.; CHEN, Y.-L. Dynamic calibration and occlusion handling algorithms for lane tracking. *Industrial Electronics, IEEE Transactions on*, IEEE, v. 56, n. 5, p. 1757–1773, 2009. Cited on page 35.

YENIKAYA, S.; YENIKAYA, G.; DÜVEN, E. Keeping the Vehicle on the Road: A Survey on On-road Lane Detection Systems. *ACM Comput. Surv.*, v. 46, n. 1, p. 2:1—-2:43, 2013. Cited 3 times on pages 31, 34, and 35.

YU, J.; HAN, Y.; HAHN, H. An efficient extraction of on-road object and lane information using representation method. In: *Signal Image Technology and Internet Based Systems, 2008. SITIS'08. IEEE International Conference on*. Bali: IEEE, 2008. p. 327–332. Cited on page 36.

ZHANG, H.; HOU, D.; ZHOU, Z. A novel lane detection algorithm based on support vector machine. In: PIERS. *Progress In Electromagnetics Research Symposium*. Hangzhou, China, 2005. Cited on page 34.

ZUO, W.-h.; YAO, T.-z. Road model prediction based unstructured road detection. *Journal of Zhejiang University SCIENCE C*, Springer, v. 14, n. 11, p. 822–834, 2013. Cited on page 32.

Appendix

APPENDIX A – Integration into CARMEN

In the context of real-world operation, systems such as the one proposed in this work are likely to have to work alongside — in cooperation and concurrently — with other systems. In order to approximate to this scenario, the system hereby proposed was integrated into an autonomous driving framework. Although the proposed system requires only a stream of frames as input (i.e. it is independent of a vehicle, requiring only a camera), its robustness could be improved if other sensor were used, such as the vehicle speed, the vehicle pose, LiDARs and others. Even though, this integration process aims to incorporate ELAS into a real-world autonomous vehicle, working alongside of other modules, but only using a camera as input.

Currently, the High Performance Computing Laboratory (LCAD) of the Universidade Federal do Espírito Santo (UFES) is developing an autonomous car named Intelligent Autonomous Robotic Automobile (IARA). IARA, shown in Figure 34, was built using a Ford Escape Hybrid adapted with sensors, control mechanisms and a set of computers. It also serves as a mobile laboratory enabling several experiments in a variety of research topics. One of the goals of this project is the completely autonomous navigation of IARA in a 50km route between UFES and a neighbor city, Guarapari.

Figure 34 – Intelligent Autonomous Robotic Automobile — IARA

The system behind IARA is being developed using the framework CARMEN (Carnegie Mellon Robot Navigation Toolkit). CARMEN is a modular software and was originally built to lower the barrier to implementing new algorithms for robots (MON-

TEMERLO; ROY; THRUN, 2003). Therefore, the proposed system was wrapped as a module of CARMEN. The CARMEN open-source project was discontinued in 2008, but LCAD keeps its own modified version[1] to use on the IARA.

The goal of the integration process is to incorporate the proposed system in a real-world case, where it has to work with sensors and share computational resources with other systems. As reported in (MONTEMERLO; ROY; THRUN, 2003), the overhead introduced by the modularity of CARMEN is outweighed by its advantages: flexibility, network support, reliability and extensibility. The integration process mainly comprises the development of a module that could read the video stream from a camera, process each frame and report its outputs to the other modules.

IARA is equipped with three Bumblebee stereo cameras: two 3-sensor stereo cameras, one looking forward and another looking slightly to the left; and a 2-sensor stereo camera looking backwards. As the proposed system expects input frames from a forward-looking camera, the camera shown by the red arrow in Figure 35 was chosen to be used in the integration. One of the problems with this setup is that, as it can be seen in Figure 35, there is something that the system has to cope with: a GPS antenna right in front of the chosen camera, reducing its visibility. Although the position of the GPS antenna can be temporary, the system had to deal with it. No other camera could be chosen, because of the assumption of a forward-looking camera as sensor input. Besides these limitations, the proposed system has to operate in real-time, even though it has to share computation resources with many other algorithms.

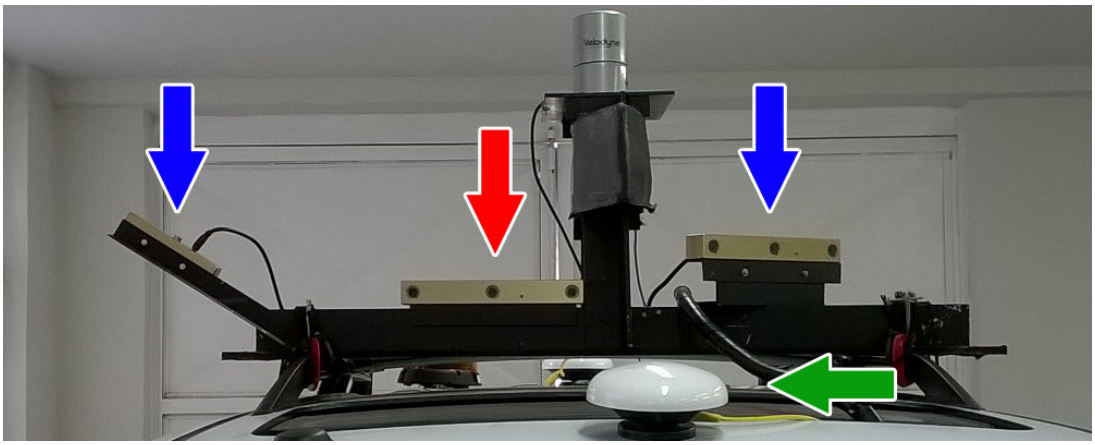


Figure 35 – Some of the sensors of IARA. Blue and red arrows represent the 3 Bumblebee cameras. The red one was the forward-looking camera used as input sensor. The green arrow points to one of the GPS antenna that is in front of the chosen camera.

To integrate the system into CARMEN, we chose to develop a wrapper for the system, so that it would not be limited to operate inside CARMEN (which would be a huge

[1] Version of the CARMEN developed by LCAD: https://github.com/LCAD-UFES/carmen_lcad

constraint), but could be a standalone application. This decision was also made taking into consideration future researchers that might want to use ELAS for future comparisons.

The module developed to integrate into CARMEN dispatch messages with the output of the Lane Analysis process. As this integration did not aim to interact with the control systems, the output of the system was rendered using a visualization module available in the framework: the Viewer3D. Viewer3D enables rendering in a 3D perspective. To render the output of ELAS into the Viewer3D, it was developed a drawer responsible to render the lane boundaries into this module, as can be seen in the Figure 36.

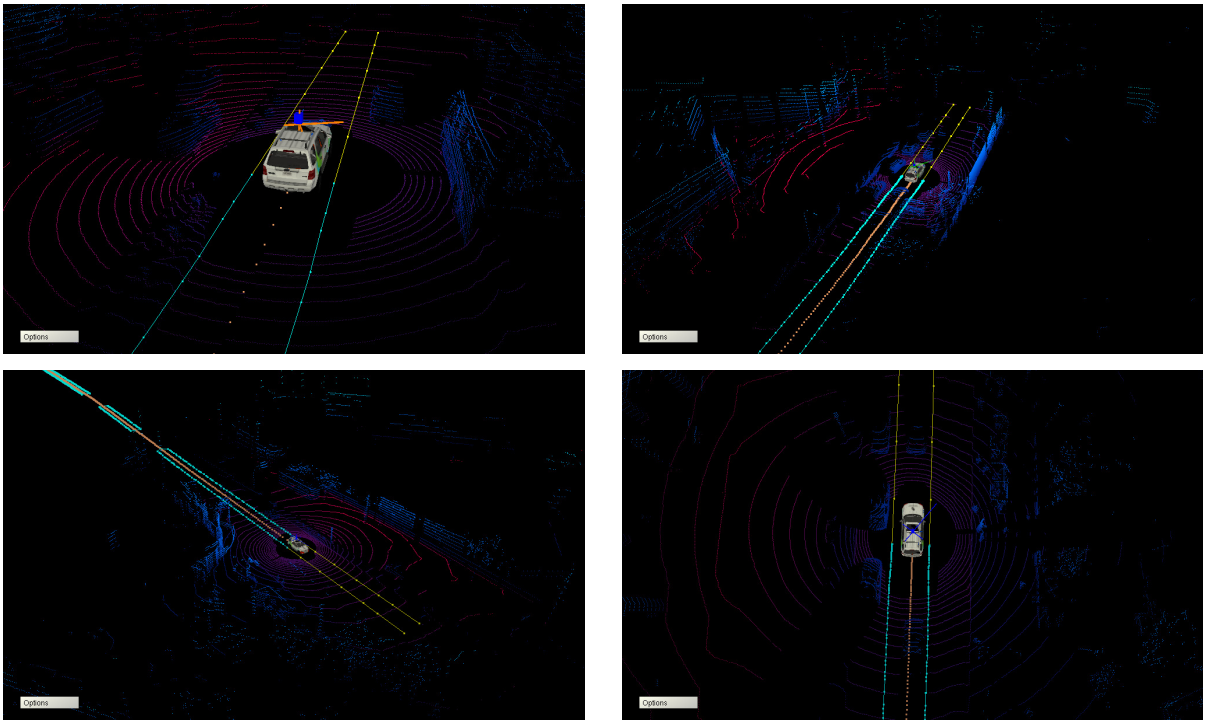


Figure 36 – Sample images of the Viewer3D module. The trail of the lane position is represented by the cyan dots and lines. The actual lane estimation is rendered by the yellow dots and lines.

APPENDIX B – Annotated Dataset

This annotated dataset contains over 15,000 frames of more than 20 scenes. Samples of all these scenes are shown below. To name a few of the features available on this dataset: different days, different weather conditions (including a small scene on a rainy day), multiple lane marking types transitions, lane changes, cars overtaking, various intersections, crosswalks, different pavement markings, different textures, shaky capture, etc.

In quantitative terms, this dataset contains 22 scenes with a total of 17,092 frames. In 98.54% of the frames, at least one of the sides has white lane markings, while 12.88% of them are yellow. In 5.18% of the frames at least one side has no lane markings. This numbers include frames containing transitions of lane markings type, i.e. images with two lane markings type simultaneously. Lane markings type transitions are present in 1,854 frames (10.85%). Lane change maneuver are being performed in 5.42% of this dataset, where 64.72% is from right to left and 35.28% in the opposite direction. There are intersections in 2.11% of the frames. In 7.28% of all images, there is at least one pavement marking. 33.92% of the pavement markings are crosswalks and 15.27% are annotated as unknown, i.e. they are of none of the classes of interest and 50.81% comprises arrows and stop lines. There is at least one adjacent lane in 50.34% and 72.14% of the frames for each side, right and left respectively. A description of each scene can be found below.

Scene ID: ROD_S01
This scene comprises 901 frames captured in the highway that connects Vila Velha to Guarapari, named Rodovia do Sol. It contains lane changes, two different lane marking types (including transitions) and there was no traffic when the scene was recorded. The scene was captured in a cloudy day.

Scene ID: ROD_S02
This scene comprises 701 frames captured in the highway named Rodovia do Sol, that connects Vila Velha to Guarapari. It was captured in a cloudy day and the driver deliberately performs lateral moves inside the lane. There is also a soft curve after a speed limit enforcement camera.

Scene ID: ROD_S03
This scene comprises 351 frames captured in the end of a highway in the city of Guarapari. It was captured in a rainy day and contains rain drops in the windshield and windshield wipers. This scene also contains a lane change maneuver and two types of lane markings (without transitions).

Scene ID: VV_S01
This scene comprises 901 frames captured in the city of Vila Velha. Although there is only two lane markings type with no transitions, there are some parts of this scene with asphalt overlay. There is also the presence of curbs.

Scene ID: VV_S02
This scene comprises 901 frames captured in the urban roads of the city of Vila Velha. It starts in a soft curve as it approaches a traffic light. Due to the red light, the driver slows down. This scene contains a lane change, pavement markings and a crosswalk.

Scene ID: VV_S03
This scene comprises 801 frames captured in the urban roads of the city of Vila Velha in a weekday with regular traffic. The driver goes through two intersections, passing by various crosswalks, pavement markings (some of them fading). The route contains different lane marking types, including transitions.

Scene ID: VV_S04
This scene comprises 901 frames captured on a weekday in the urban roads of the city of Vila Velha with regular traffic. It contains two lane marking types, including transitions. During the scene, the drive goes under a viaduct that casts shadows on the road. Also, at the end, the driver slows down the car because of traffic.

Scene ID: VIX_S01
This scene comprises 901 frames captured on a weekend on the urban roads of the city of Vitória. It contains regular traffic for the day. There are two lane markings type with no transitions. Also, there are three arrows during this scene.

Scene ID: VIX_S02
This scene comprises 901 frames captured on a weekend on the urban roads of the city of Vitória. There is almost no traffic. There are various lane markings type (including transitions) and slight change in lane width. Also, there are different pavement markings and a crosswalk.

Scene ID: VIX_S03
This scene comprises 901 frames captured on a weekend on the urban roads of the city of Vitória. It contains regular traffic, including an overtaking from a motorcycle that partially occludes the lane markings. Also, there are different lane markings type (including transitions), intersections, crosswalks and pavement markings.

Scene ID: VIX_S04
This scene comprises 901 frames captured during a weekend in one of the main boulevards of the city of Vitória. It contains different lane markings type (including transitions) and crosswalks. Also, there are pavement markings, intersections and a lane change maneuver is performed.

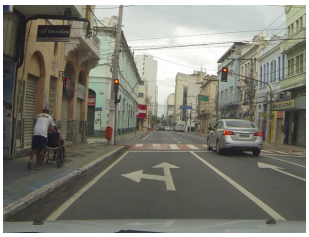
Scene ID: VIX_S05
This scene comprises 901 frames captured during a weekend in one of the main boulevards of the city of Vitória. It contains different lane markings type (including transitions), pavement markings, crosswalks and intersections. Also, a vehicle partially occludes the lane markings during a curve.

Scene ID: VIX_S06
This scene comprises 901 frames captured in the urban roads of the city of Vitória. It contains different lane markings type (including transitions), pavement markings, crosswalks and intersections. Also, there is a part of the scene where asphalt resurfacing partially occludes lane markings.

Scene ID: VIX_S07
This scene comprises 451 frames captured in the urban roads of the city of Vitória. The capture is very unstable, resulting in a shaky video sequence. Besides this, this scene contains different lane markings type, pavement markings, crosswalks, intersections, and asphalt resurfacing partially occludes lane markings.

Scene ID: VIX_S08
This scene comprises 421 frames captured on a weekend on the urban roads of the city of Vitória. It contains different lane markings (including transitions and a red marking in parallel in both sides) and crosswalk. Also, there are writings on the lane and a vehicle partially occludes the lane markings at a given time.

Scene ID: VIX_S09
This scene comprises 301 frames captured on a weekend on the urban roads of the city of Vitória. It contains different lane markings (including transitions and a red marking in parallel in both sides) and crosswalk. Also, there are writings and non-arrow symbols on the lane.



Scene ID: VIX_S10
This scene comprises 601 frames captured on a weekend on the urban roads of the city of Vitória. It contains different lane markings (including transitions and a red marking in parallel in both sides) and crosswalk. Also, there are writings and non-arrow symbols on the lane.

Scene ID: VIX_S11
This scene comprises 451 frames captured on a weekday on the bridge that connects Vitória to Vila Velha, named Terceira Ponte. The material of the road is different, therefore the road is lighter and lane markings are surrounded by black strips. Also, the traffic is regular.

Scene ID: BR_S01
This scene comprises 1201 frames captured on a weekend in a federal highway named BR-101. It contains shadows on the road being cast by other vehicles and trees. There is a straight path followed by a curve with multiple lane markings type, including various transitions.

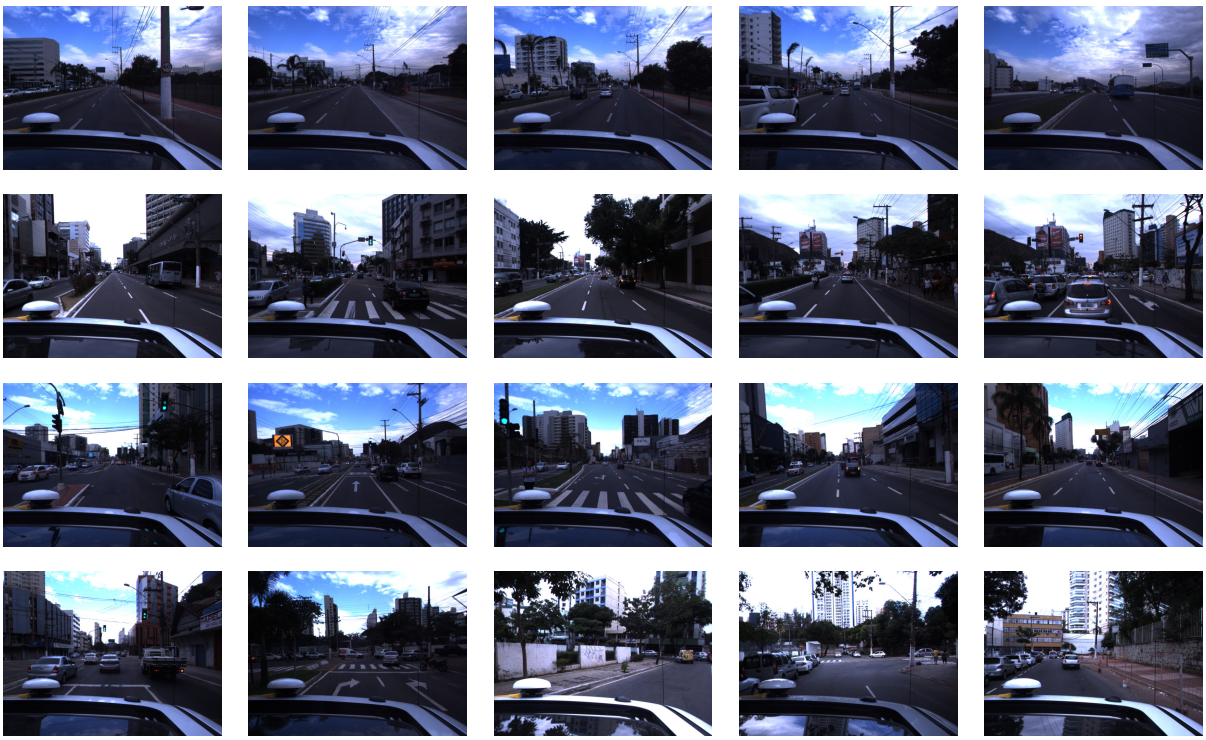
Scene ID: BR_S02
This scene comprises 1001 frames captured on a weekend in a federal highway named BR-101. It contains almost all the lane markings type (6 out of the 7 types) including multiple transitions on both sides. There is also regular traffic and a soft curve on the road.

Scene ID: GRI_S01
This scene comprises 901 frames captured in a urban road in the city of Guarapari. It starts in a soft curve with low traffic. There is also a lane change maneuver and different lane markings type (including transitions).

Scene ID: GRI_S02
This scene comprises 901 frames captured in a urban road in the city of Guarapari. It contains two types of lane markings (without transitions). A lane change maneuver is performed. The scene was recorded with typical traffic on the street.

APPENDIX C – Scenes from IARA

This dataset was recorded using a Bumblebee camera, at 13 frames per seconds, on average. It contains a usual day of traffic (weekday) on two of the main boulevards of the city of Vitoria/ES, Brazil. Also, to this recording session, it was used a custom driver integrated into CARMEN that runs on the autonomous vehicle developed by the High Performance Computing Lab, the IARA.



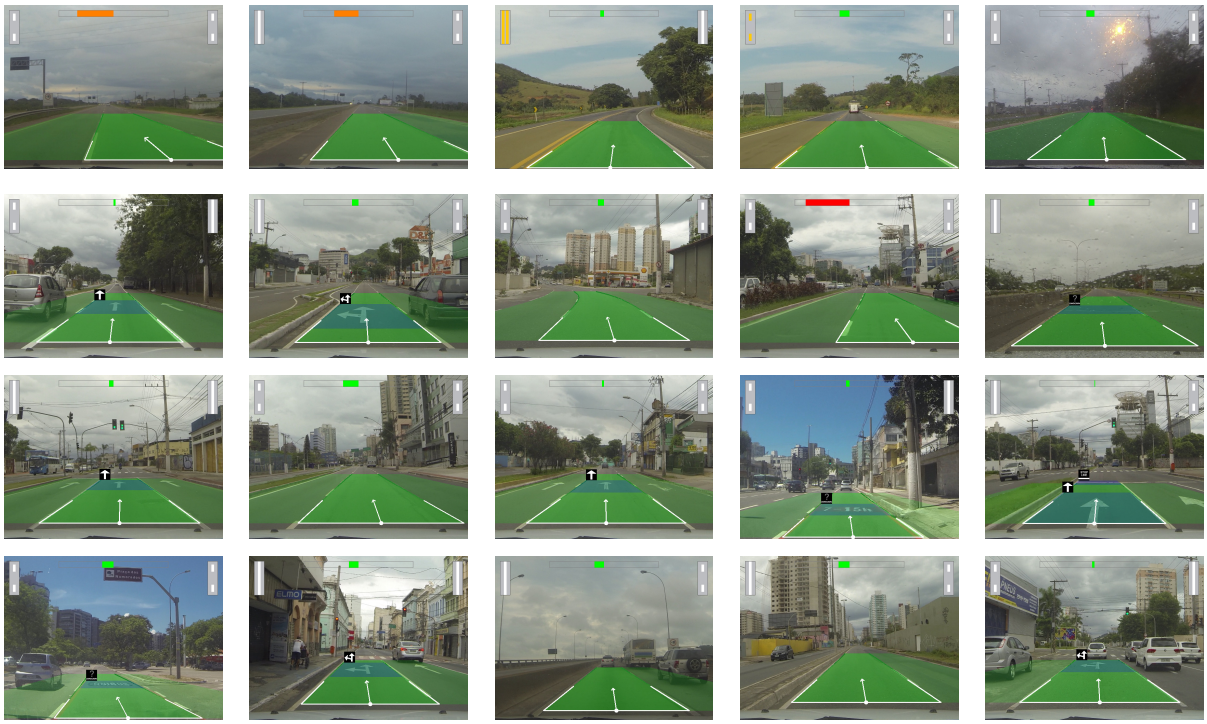
APPENDIX D – Other Scenes

Two scenes were chosen to perform qualitative evaluation. The first one consist of 3.1km of a rainy day in a highway. It comprises more than 3,600 frames recorded at 29.97 frames per second using a GoPRO HERO 3. Some samples can be seen below.

The second one consist of a long scene: 23.2km in a highway, including a passage by a toll. It comprises more than 30,000 frames recorded at 29.97 frames per second using a GoPRO HERO 3. Some samples can be seen below.

APPENDIX E – Qualitative Results

All datasets were used for qualitative results. On the images below, ELAS potential can be seen, as well as its limitations. Firstly, the annotated dataset was tested. It has images with different lane marking types, different pavement markings, during lane changes and other events of interest. Besides the qualitative results herein reported, quantitative evaluation was also performed for this dataset.



Also, ELAS was tested on the other two scenes chosen for qualitative analysis. On them, it can be seen soft rain, the classification of some drops of water as the *Unknown* class, wipers, among other events of interest.

At last, the integration into CARMEN was evaluated qualitatively on a dataset recorded using the IARA. As can be seen, this dataset has a different overall intensity, the images were captured using a camera in a different pose and it has a lower frame rate, when compared to the previous datasets. Despite of that, ELAS still works quite well.