Visual computing techniques for automated LIDAR annotation with application to intelligent transport systems

By

José Javier Barandiarán Martirena

Department of Computer Science and Artificial Intelligence

eman ta zabal zazu

Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

PhD Advisors:
Prof. Dr. Manuel Graña Romay UPV-EHU
and
Dra. Oihana Otaegui Madurga Vicomtech
Universidad del País Vasco
Euskal Herriko Unibertsitatea
Donostia - San Sebastián
2021

(c)2021 JOSE JAVIER BARANDIARAN MARTIRENA

Visual computing techniques for automated LIDAR annotation with application to intelligent transport systems

by

José Javier Barandiarán Martirena

Abstract

The concept of Intelligent Transport Systems (ITS) refers to the application of communication and information technologies to transport with the aim of making it more efficient, sustainable, and safer. Computer vision is increasingly being used for ITS applications, such as infrastructure management or advanced driver-assistance systems. The latest progress in computer vision, thanks to the Deep Learning techniques, and the race for autonomous vehicle, have created a growing requirement for annotated data in the automotive industry. The data to be annotated is composed by images captured by the cameras of the vehicles and LIDAR data in the form of point clouds. LIDAR sensors are used for tasks such as object detection and localization. The capacity of LIDAR sensors to identify objects at long distances and to provide estimations of their distance make them very appealing sensors for autonomous driving. This thesis presents a method to automate the annotation of lane markings with LIDAR data. The state of the art of lane markings detection based on LIDAR data is reviewed and a novel method is presented. The precision of the method is evaluated against manually annotated data. Its usefulness is also evaluated, measuring the reduction of the required time to annotate new data thanks to the automatically generated pre-annotations. Finally, the conclusions of this thesis and possible future research lines are presented.

Visual computing techniques for automated LIDAR annotation with application to intelligent transport systems

por

José Javier Barandiarán Martirena

Resumen

El concepto de Sistemas Inteligentes de Transporte (ITS) se refiere a la aplicación de tecnologías de la comunicación e información al transporte con el objetivo de hacerlo más eficiente, sostenible y seguro. La visión por computador es cada día más utilizada en aplicaciones ITS, como el mantenimiento de infraestructuras o sistemas avanzados de ayuda a la conducción. Los últimos avances en la visión por computador, gracias a las técnicas de aprendizaje profundo, y la carrera por el vehículo autónomo, ha provocado en la industria de la automoción una necesidad creciente de datos anotados. Los datos para anotar se componen de imágenes captadas por las cámaras de vídeo instaladas en los vehículos y datos LIDAR en forma de nubes de puntos. Los sensores LIDAR se utilizan para tareas como detección de objetos y localización. La capacidad de los LIDAR para detectar a largas distancias y proveer de estimaciones de su distancia, los hace muy provechosos para la conducción autónoma. Esta tesis presenta un método para la anotación automática de líneas de carril con datos LIDAR. Revisa el estado del arte de la detección de líneas de carril basado en LIDAR y presenta un método novedoso. La precisión del método es evaluada con datos anotados manualmente. Su utilidad también es evaluada, midiendo la reducción del tiempo requerido para anotar nuevos datos gracias a las preanotaciones generadas automáticamente. Finalmente, se presentan las conclusiones de esta tesis y posibles líneas futuras de trabajo.

Originality Statement

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at The University of the Basque Country or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at The University of the Basque Country or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

The author hereby grants to University of the Basque Country permission to reproduce and to distribute copies of this thesis document in whole or in part.

Signed:____________________________________

Acknowledgements - Agradecimientos

En primer lugar, quiero dar las gracias a mis directores de tesis Prof. Manuel Graña y Dra. Oihana Otaegui por su confianza, apoyo y dedicación. También al Prof. Alex García-Alonso con quien comencé esta pelea del doctorado.

Agradecer a todas las personas que trabajan y han trabajado anteriormente en Vicomtech por conseguir que el proyecto siga creciendo cada año más robusto, después de 20 años de vida. Espero que consideréis esta tesis como un digno fruto de vuestro trabajo.

Por supuesto, gracias a mi familia a quienes les debo todo.

Este trabajo ha recibido financiación del programa de la Unión Europea Horizon 2020 a través de varios proyectos, entre los que cabe destacar Cloud-LSVA.

Contents

1 **Resumen** **1**

2 **Introduction** **7**

- 2.1 Motivation . . . 7
  - 2.1.1 The importance of transportation . . . 7
    - 2.1.1.1 Safety . . . 7
    - 2.1.1.2 Environmental impact . . . 8
    - 2.1.1.3 Economical value . . . 8
    - 2.1.1.4 Social impact . . . 8
  - 2.1.2 Intelligent transportation systems . . . 10
  - 2.1.3 Computer Vision for ITS . . . 11
  - 2.1.4 Data annotation . . . 13
  - 2.1.5 LIDAR sensors . . . 17
  - 2.1.6 Horizontal road markings . . . 18
- 2.2 Objectives . . . 19
- 2.3 Contributions . . . 19
- 2.4 Research environment and context . . . 20
  - 2.4.1 Projects . . . 20
- 2.5 Publications . . . 22
- 2.6 Structure of the thesis . . . 24

3 **State of the art** **27**

- 3.1 Road markings detection . . . 27
  - 3.1.1 Non real-time methods . . . 27

3.1.1.1 Image Processing 28

3.1.1.2 Scan line Processing 29

3.1.2 Real-time methods 30

3.2 Road points segmentation 32

3.3 LIDAR reflectivity calibration 34

3.4 Image and point cloud annotation 36

4 Lane markings automated annotation 39

4.1 Problem statement 39

4.1.1 Manual annotation 41

4.2 Outline of the proposed approach 42

4.3 Point cloud preprocessing and preparation 44

4.3.1 Filtering scan points by distance and height 45

4.3.2 Scans accumulation 46

4.3.3 Noise reduction 46

4.3.4 Intensity calibration 48

4.3.4.1 Angle of incidence 49

4.3.5 Lane markings contrast enhancement 50

4.4 Lane markings detection 51

4.4.1 Block cropping 52

4.4.2 Candidate lane marking points detection 53

4.4.3 Tracking detected lane markings 56

4.4.4 Lane markings classification 57

4.4.5 Refinement of dashed segment extremes 58

4.4.6 Annotations output 58

5 Evaluation 61

5.1 Evaluation methodology 61

5.2 Dataset description 63

5.3 Results 66

5.3.1 Parameter sensitivity analysis 73

6 Conclusions and future work **77**

6.1 Conclusions 77

6.2 Future work 78

6.2.1 Detect other road markings 78

6.2.2 Detect road edges 78

6.2.3 Improve LIDAR extrinsic parameters 79

6.2.4 Combine LIDAR with cameras 79

Bibliography **81**

List of Figures

2.1 Total U.S.A. greenhouse gas emissions by economic sector in 2018 [1]. 9
2.2 Household expenditure by consumption purpose in 2018 in Europe [2]. 10
2.3 Levels of driving automation defined by the Society of Automotive Engineers (SAE) [3]. 12
2.4 Driver monitoring [4]. 13
2.5 Vehicle counting and classification by its length [5]. 14
2.6 Web-based annotation interface [6]. 16
2.7 Velodyne Alpha Prime 3D LIDAR and RIEGL VMX-2HA mobile mapping system 18

3.1 Road markings segmented and classified by conditional GAN [7]. . . . 29
3.2 Real-time road markings detection: above [8], below [9]. 32
3.3 Road detection [10]. 34
3.4 Before and after intensity calibration [11]. 35
3.5 3D annotation and annotations transferred to 2D segmentations [12]. 37

4.1 Detected lines back-projected to the images from the cameras of the vehicle. 41
4.2 The user interface of the web-based application used for the annotation of lane markings. 43
4.3 Annotation pipeline. 44
4.4 Point cloud preprocessing steps 45
4.5 Accumulated point cloud before (left) and after (right) filtering 47
4.6 Alignment error introduced by incorrect extrinsic calibration 47
4.7 Point cloud before and after noise reduction 48

4.8 Angle of incidence 50

4.9 Point cloud contrast enhancement 51

4.10 Computational pipeline of the method. 52

4.11 Point cloud block extraction with a cropping box 53

4.12 Point cloud vertical projection of the average intensity 55

4.13 Tracking and road definition parameters 57

4.14 Dashed segment refinement 59

4.15 Detected and classified lane markings 60

5.1 Instance of the visualization of manual annotation of lane markings used as ground truth for training and evaluation. 62

5.2 Diagram of the evaluation process. 64

5.3 Dataset examples 65

5.4 A sample frame extracted from of the video recorded along each trace ordered from left to right and top to bottom. 65

5.5 F-measure 69

5.6 Lane markings in poor condition 70

5.7 F-measure for dashed segments 71

5.8 Time required for annotation 72

5.9 F-measure ($\varepsilon = 5$ cm) varying parameter blo_l. 73

5.10 F-measure ($\varepsilon = 5$ cm) obtained varying parameters bin_d and bin_w. . . 74

5.11 F-measure ($\varepsilon = 5$ cm) obtained varying parameters bin_m and bin_g. . . 75

List of Tables

5.1 Traces characteristics 66
5.2 Parameters used for the quantitative evaluation 67
5.3 Road definition parameters 67
5.4 F-measure and traces characteristics 68
5.5 F-measure for dashed segments 72

Chapter 1

Resumen

Introducción

En la última década, la Visión por Computador ha experimentado un importante progreso en áreas como la detección objetos, reconocimiento, o segmentación gracias a los avances en redes neuronales, renombradas como redes de aprendizaje profundo (deep learning) [13]. Este renacimiento de una vieja aproximación ha sido facilitado por el creciente poder computacional de las tarjetas GPU y la abundancia de datos etiquetados. El rendimiento de los sistemas basados en deep learning depende fuertemente de la calidad y cantidad de los datos disponibles para su entrenamiento.

Con la carrera por los vehículos autónomos, muchas empresas de la industria de la automoción, de tecnología, y de alquiler de vehículos con conductor están generando cantidades masivas de datos utilizando sus flotas de vehículos equipados con cámaras, LIDARs, radares y otros sensores. Esos son necesarios para el desarrollo y validación de sus propios sistemas avanzados de ayuda a la conducción (ADAS) o vehículos autónomos. La cantidad de a anotar es tan grande, que la tarea de anotación es uno de los cuellos de botella del progreso de estos sistemas De hecho, muchas de esas compañías utilizan simuladores virtuales para generar datos sintéticos automáticamente anotados para no depender tanto de datos reales anotados. Aunque modelos entrenados solo con datos sintéticos puede tener un rendimiento muy bueno [14], los datos reales anotados siguen siendo necesarios para obtener una mejor precisión en dominios específicos.

Los datos reales registrados deben ser anotados manualmente. La anotación o etiquetado consiste en la identificación y segmentación de cada grupo de píxeles o puntos 3D correspondientes a cada uno de los actores o elementos (vehículos, peatones, señales, carretera, etc.) presentes en las imágenes o escaneos LIDAR. Esta tarea es repetitiva y muy ardua. Hay que tener en cuenta que cada cámara graba un mínimo de 30 imágenes por segundo, y cada vehículo suele estar equipado con cuatro o más cámaras. Los sensores LIDAR registran cientos de miles de puntos por segundo. Un vehículo puede estar equipado con más de un LIDAR.

Un anotador humano debe procesar cada fotograma de cada cámara, dibujando una caja alrededor de cada peatón, vehículo, señal, etc. Esta tarea puede acelerarse utilizando detecciones automatizadas y herramientas de seguimiento. Con estas herramientas el anotador se dedica únicamente a verificar y corregir las preanotaciones y completar los elementos que no hayan sido detectados.

La anotación con datos LIDAR es similar. Los actuales LIDAR mecánicos completan entre 10 y 20 revoluciones o escaneos por segundo. Cada escaneo individual de los LIDAR instalados en el vehículo pueden registrarse formando una única nube de puntos 3D. La tarea del anotador consiste en procesar cada nube de puntos, creando cajas 3D en torno a cada vehículo, peatón, señal edificio, etc., y dibujar líneas 2D en el suelo identificando elementos como bordillos, líneas de carril o señales horizontales.

Las marcas viales horizontales son un elemento visual fundamental da la infraestructura viaria porque proporcionan información para la navegación, seguridad y normas de conducción. Las líneas que delimitan los carriles y los propios límites de la carretera son particularmente importantes. La detección de estas marcas es requerida para funciones como sistemas de mantenimiento del carril, salida del carril, y la planificación automática de maniobras con posicionamiento preciso. Los sistemas ADAS y los vehículos autónomos requieren tecnologías de localización con una precisión de unos pocos centímetros. Igualmente importantes son los mapas de alta definición con información a nivel de carril.

La anotación de líneas de carril puede realizarse con las imágenes de las cámaras que rodean al vehículo, pero es difícil obtener una reconstrucción 3D de las líneas utilizando cámaras. En cambio, la anotación utilizando datos LIDAR proporciona una reconstrucción 3D directa de los puntos que forman las líneas. Los LIDAR además de información de distancia, proporcionan información de la intensidad con la que cada superficie refleja el láser, permitiendo una identificación de las líneas respecto al resto de la carretera porque están trazadas con pintura reflectante. Otra ventaja es que los escaneos individuales pueden ser acumulados en una única nube de puntos, de forma que los anotadores solo tienen que enfocarse en una fuente de datos.

Objetivos

El principal objetivo de esta tesis es investigar en el uso de sensores LIDAR y métodos de computación para la detección y anotación automática de líneas de carril. El trabajo se ha enfocado en la realización de un sistema de asistencia a la anotación para operarios humanos proporcionándoles información de detección que puede ser empleada para acelerar y aliviar su trabajo.

Los objetivos operacionales implementados para alcanzar el objetivo general son:

- Implementación de una herramienta basada en web para la anotación de datos LIDAR que ha sido empleada y validad por anotadores profesionales.
- Definición e implementación de los pasos de preprocesado de los datos para la anotación automática, incluyendo la calibración de los sensores y el filtrado de las nubes de puntos.
- Colecta de un conjunto de datos experimental desde vehículos reales que se ha utilizado para la evaluación cuantitativa del sistema propuesto.
- Definición de un proceso de evaluación con anotadores humanos.
- Realización de experimentos computacionales para la evaluación del sistema propuesto.

Contribuciones

Las contribuciones de esta tesis son las siguientes:

- Un método de anotación automática de líneas de carril utilizando datos LIDAR. Este método está dirigido a asistir a los anotadores humanos. Las anotaciones de líneas de carril se están utilizando para el desarrollo de nuevos sistemas ADAS que contribuirán a mejorar la seguridad y reducir el impacto ambiental del transporte por carretera. El método fue publicado en [15].

- La integración del método implementado como un proceso back-end permite la explotación de los recursos computacionales ofrecidos por arquitecturas en la nube.

- El desarrollo de una herramienta de anotación basada en web que permite entornos de trabajo remoto. Esta herramienta está todavía siendo desarrollada por un grupo de investigadores en Vicomtech. El autor ha colaborado en la implementación de la sincronización, procesado y visualización de vídeo y datos LIDAR. La herramienta se presentó en la conferencia Web3D 2019 [6].

- Una metodología para la evaluación de la precisión de las anotaciones automáticas de líneas de carril.

Método para la anotación automática

El método utiliza como entrada nubes de puntos 3D con valores intensidad y datos de odometría generados con un LIDAR 3D y un sistema de navegación GPS/IMU instalado en un vehículo. La odometría se utiliza para acumular los escaneos en una nube de puntos para todo el trayecto recorrido, aumentando así la densidad de puntos. Antes de la detección, los escaneos individuales y la nube de puntos acumulado sufren una serie de preprocesos para eliminar ruido, reducir el número de puntos de a procesar y mejorar el contraste de las líneas de carriles.

Un paso fundamental del preprocesado consiste en la calibración de la intensidad del LIDAR. La intensidad medida por los láseres que componen un sensor LIDAR debe ser calibrada para compensar las diferentes sensibilidades de estos. La calibración consiste en organizar los puntos en una rejilla horizontal y calcular un factor de compensación para cada láser de forma que todos observen la misma intensidad en las mismas celdas de la rejilla de calibración.

Para la detección de las líneas, la nube de puntos acumulada y filtrada se procesa mediante bloques transversales a lo largo de la trayectoria definida por la odometría. En cada bloque se obtiene un perfil de la intensidad de la carretera y se detectan las líneas de carril como máximos en dicho perfil y una estimación del ancho de la línea en ese punto. Bloque tras bloque, los nuevos puntos detectados son emparejados con las líneas detectadas anteriormente. De esta forma se van trazando una serie de polilíneas 3D. Finalmente las líneas son clasificadas como continuas o discontinuas en función del patrón formado por los puntos detectados.

Las líneas detectadas son finalmente refinadas para cumplir con los criterios de anotación exigidos a los anotadores. En el caso de las líneas continuas, son muestreadas con una distancia constante y prefijada para reducir el número de puntos. Para las líneas discontinuas solo se conservan los puntos que delimitan cada uno de los segmentos. Además, se aplica un proceso para mejorar la posición longitudinal de estos extremos.

Evaluación

Para la evaluación cuantitativa de la precisión del método se ha generado un conjunto de datos anotados manualmente. Se propone una nueva metodología de evaluación enfocada en medir la precisión del posicionamiento de las líneas de carril. El método propuesto compara polilíneas 3D muestreándolas con una distancia fija, convirtiéndolas la comparación de polilíneas 3D en una comparación de puntos 3D. De esta forma la comparación puede hacerse mediante la distancia Euclídea entre puntos y permite evaluar la precisión de las anotaciones variando un umbral de precisión requerido.

En la evaluación realizada se ha obtenido una precisión del 92% con un error lateral máximo de 5 cm. Esto significa que los anotadores que utilizan las pre-anotaciones automáticas solo tendrán que arreglar o completar el 8% del total de líneas de carril presentes en los datos. Para validar esta observación, se ha realizado un experimento con anotadores profesionales midiendo el tiempo que necesitaban para anotar varias grabaciones desde cero y utilizando las preanotaciones. El resultado obtenido indica que los anotadores necesitaron un 60% menos de tiempo para finalizar la tarea cuando anotaban utilizando las preanotaciones.

Conclusiones y futuras líneas de trabajo

Esta tesis has presentado un método para la anotación de líneas de carril utilizando datos LIDAR. La aproximación ha sido validad en un conjunto de datos real generado con un vehículo sensorizados. Como resumen del trabajo realizado:

- Hemos implementado el preprocesado y preparación de los datos LIDAR.
- Hemos implementado un sistema de anotación basado en web. El sistema se está utilizando en distintos proyectos en Vicomtech y por compañías para explotación comercial. La herramienta permite la anotación de vídeo y LIDAR directamente en un navegador web, sin requerir instalación, utilizando un PC de bajo coste. Este tipo de instalación permite la distribución remota y masiva de tareas de anotación.
- Se ha definido e implementado un método de detección de líneas de carril eficiente y preciso.
- Se han llevado a cabo experimentos para la evaluación de la precisión método y para comprobar su validez para la asistencia en tareas de anotación manual.

Chapter 2

Introduction

This chapter presents the motivation for this Thesis in section 2.1. Section 2.2 describes the objectives of this research work. A summary of the generated contributions is given in section 2.3. The research environment and context are described in section 2.4. The related publications are listed in section 2.5. Section 2.6 presents the structure of this Thesis.

2.1 Motivation

2.1.1 The importance of transportation

Transportation is an important Human activity. Its relevance stands clear by observing the number of deaths in traffic accidents, the impact on the environment, the number of jobs associated with the transportation sector or the amount of time we spent commuting every day.

2.1.1.1 Safety

In its global status report on road safety for the year 2018 [16][1], the World Health Organization (WHO) estimated that 1.35 million people die each year in road traffic accidents and between 20 and 50 million are injured. Road traffic injuries are the 8th leading cause of death, the first among children and young adults aged between

---

[1] http://www.who.int/violence_injury_prevention/road_safety_status/2018/

5 and 29 years. In Spain, 102,299 accidents occurred during 2018, with 1,806 deaths and 8,935 hospitalizations [17][2].

2.1.1.2 Environmental impact

Transportation generates about the 23% of carbon dioxide emissions and is the fastest growing contributor [18][3]. In the U.S.A., transportation represented the 28% of greenhouse gas emissions in 2018 as shown in Figure 2.1[4]. Road transportation causes around the 70% of emissions, mostly generated by light-duty vehicles, such as cars, and freight transport. Air pollution contributes to the death of 4.2 million people per year [19][5].

2.1.1.3 Economical value

The transportation sector is one of the biggest employment generators. In Europe more than 11 million people work in the this sector, accounting for 5% of total employment [20][6]. The European Commission expects that, in the period from 2010 to 2050, passenger and freight transportation will grow by about 42% and 60%. In the U.S.A. there are 3.6 million truck drivers and 7.95 million people work in jobs related to trucking activity [21][7].

2.1.1.4 Social impact

Figure 2.2 shows the household expenditure of the Europeans. We spent the 13% of total household expenditure on transport, the second largest expenditure after housing 24% [2][8].

---

[2] http://www.dgt.es/es/seguridad-vial/estadisticas-e-indicadores/publicaciones/principales-cifras-siniestralidad/

[3] https://webstore.iea.org/co2-emissions-from-fuel-combustion-2018-highlights

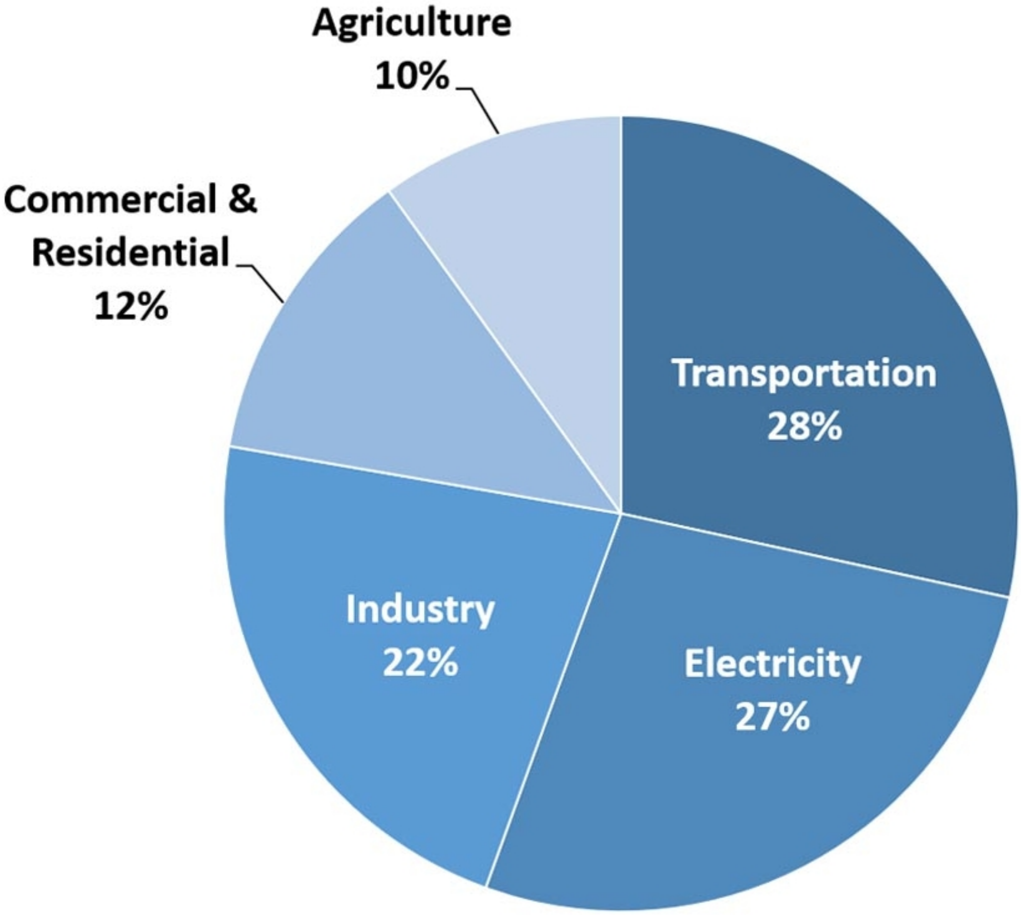
[4] https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions

[5] https://www.who.int/health-topics/air-pollution

[6] https://ec.europa.eu/transport/sites/transport/files/mobility-package-factsheet-overall.pdf

[7] https://www.trucking.org/economics-and-industry-data

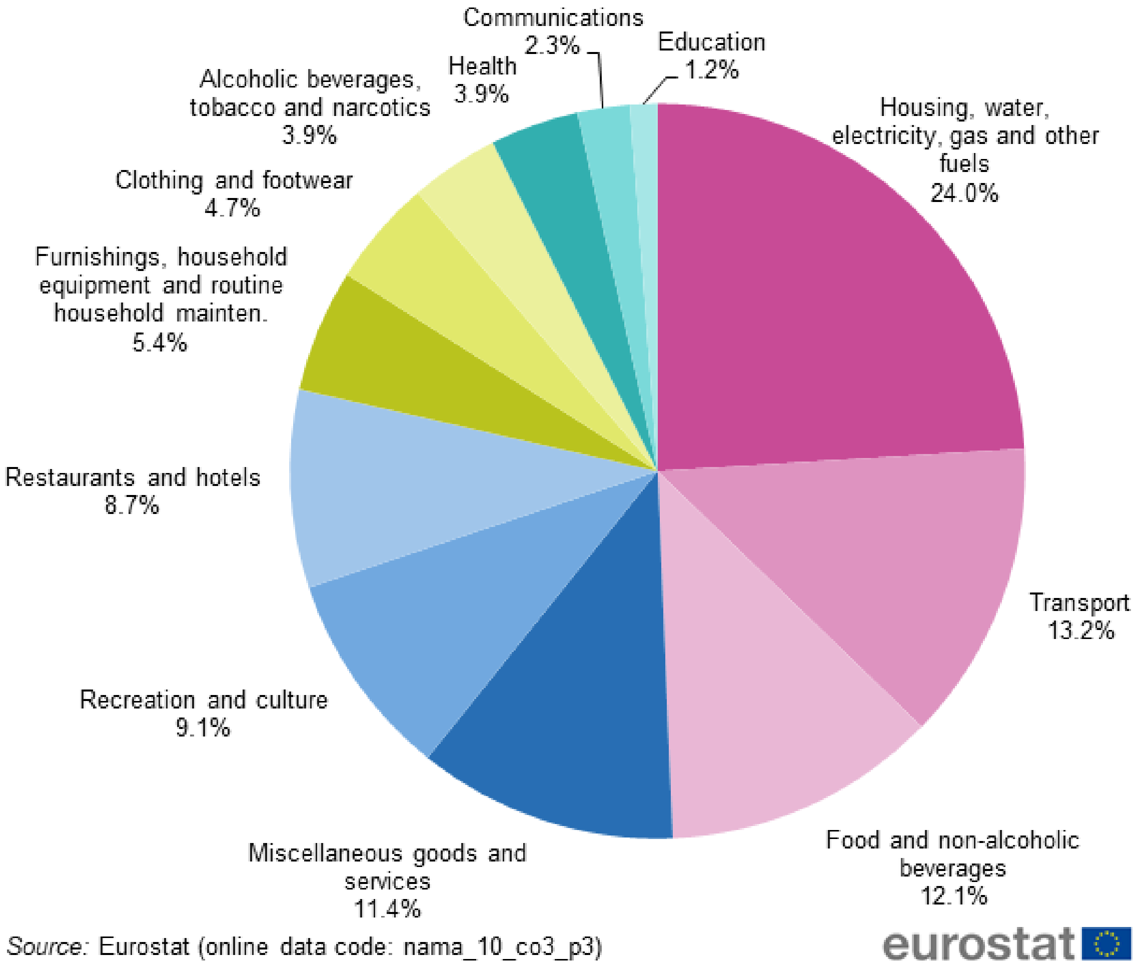
[8] https://ec.europa.eu/eurostat/statistics-explained/index.php/Household_consumption_by_purpose



U.S. Environmental Protection Agency (2020). Inventory of U.S.
Greenhouse Gas Emissions and Sinks: 1990-2018

Figure 2.1: Total U.S.A. greenhouse gas emissions by economic sector in 2018 [1].

In 2018, the average commute time (one way) was 27 minutes in the U.S.A.. There were 4.3 million workers with commutes of 90 minutes or more. On average, the Americans dedicated 225 hours to commuting that year [22][9].



Source: Eurostat (online data code: nama_10_co3_p3)

Figure 2.2: Household expenditure by consumption purpose in 2018 in Europe [2].

2.1.2 Intelligent transportation systems

The concept of Intelligent Transportation Systems (ITS) refers to the application of communication and information technologies to human and goods transportation with the aim of making it more efficient, sustainable and safer. ITS can be applied to all the elements of the transportation system: infrastructure, vehicles, drivers, and users. Some of these applications are:

---

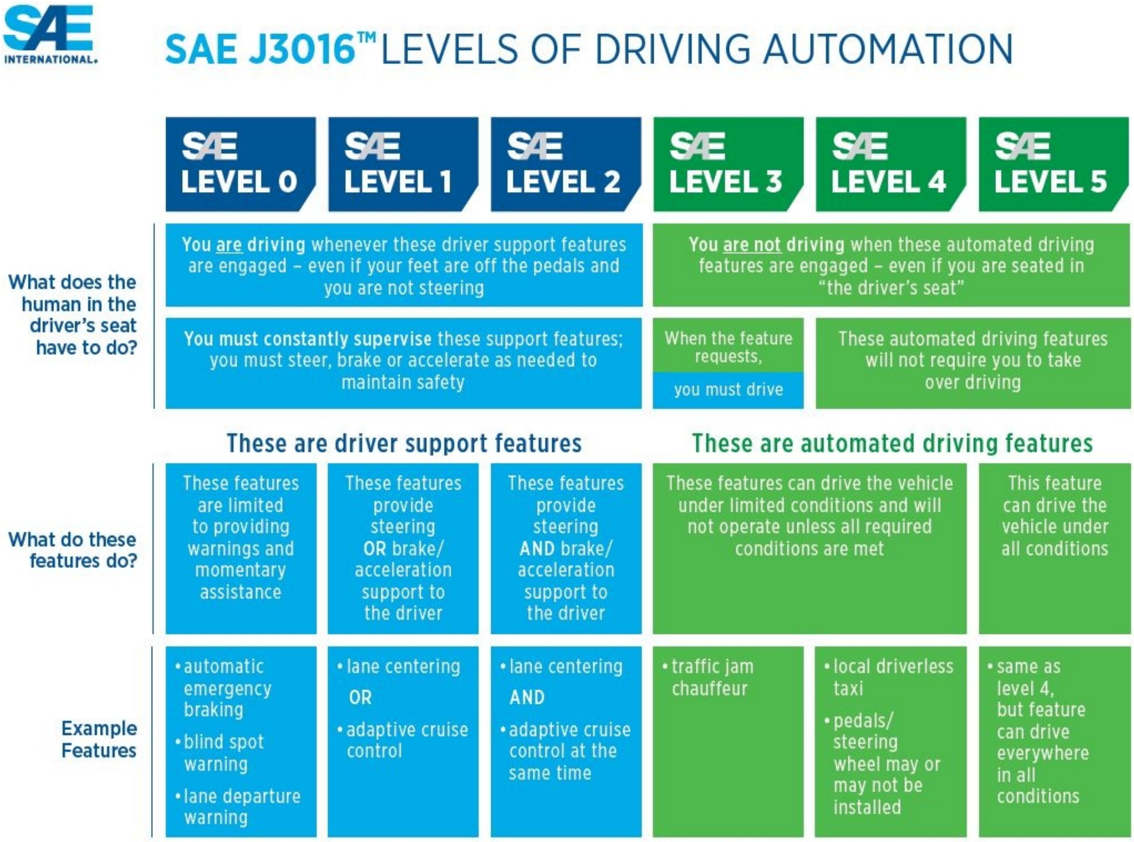
[9] https://www.washingtonpost.com/business/2019/10/07/nine-days-road-average-commute-time-reached-new-record-last-year

- Traffic monitoring, congestion and accident detection
- Vehicle counting and classification, vehicle tracking, license plate recognition
- Free flow tolling
- High occupancy lane control
- Parking management
- Information panels, adaptive signal control
- Infrastructure maintenance
- Traveler information systems, route guidance
- Driver monitoring
- Advanced driver-assistance systems (ADAS)
- Autonomous Vehicles (AV)

ITS require data acquisition, processing, communication and visualization. Data can be acquired with sensors, such as, inductive loops, rubber tubes, weather stations, light sensors, cameras, radar, LIDAR (light detection and ranging), etc. Data can be processed in-place by an embedded computer or remotely in a control center. Data can flow between vehicles, between vehicles and infrastructure, and between infrastructure and control center. The visualization of data can be done on a mobile device, on the vehicle screen, on variable-message road panel or at the control center.

2.1.3 Computer Vision for ITS

Computer Vision is now incorporated in many vehicles, as it is one of the main technologies of ADAS, such as Lane Departure Warning (LDW), Lane Keeping Assist (LKA) or Adaptive Cruise Control (ACC). This progress in automation is leading the automotive industry towards fully autonomous vehicles. Figure 2.1.3[10] identifies the diverse levels of driving automation. At the fifth level of driving automation a human driver is not needed. Below, at third and fourth automation levels, vehicles will require driver monitoring capabilities, such as gaze tracking or drowsiness detection for knowing if the driver is able to remain safely at the controls.



SAE INTERNATIONAL

SAE J3016™ LEVELS OF DRIVING AUTOMATION

	SAE LEVEL 0	SAE LEVEL 1	SAE LEVEL 2	SAE LEVEL 3	SAE LEVEL 4	SAE LEVEL 5
What does the human in the driver's seat have to do?	**You are driving** whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering			**You are not driving** when these automated driving features are engaged – even if you are seated in "the driver's seat"		
	You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety			When the feature requests, you must drive	These automated driving features will not require you to take over driving	
	These are driver support features			These are automated driving features		
What do these features do?	These features are limited to providing warnings and momentary assistance	These features provide steering **OR** brake/ acceleration support to the driver	These features provide steering **AND** brake/ acceleration support to the driver	These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met		This feature can drive the vehicle under all conditions
Example Features	• automatic emergency braking • blind spot warning • lane departure warning	• lane centering **OR** • adaptive cruise control	• lane centering **AND** • adaptive cruise control at the same time	• traffic jam chauffeur	• local driverless taxi • pedals/ steering wheel may or may not be installed	• same as level 4, but feature can drive everywhere in all conditions

Figure 2.3: Levels of driving automation defined by the Society of Automotive Engineers (SAE) [3].

Computer Vision is also being used by ITS products deployed at the road infrastructures such as:

- Free flow tolling where the vehicles are not required to slow down using Automatic Number Plate Recognition (ANPR).

- Vehicle tracking or re-identification through different video surveillance cameras.

---

[10] https://www.sae.org/news/2019/01/sae-updates-j3016-automated-driving-graphic

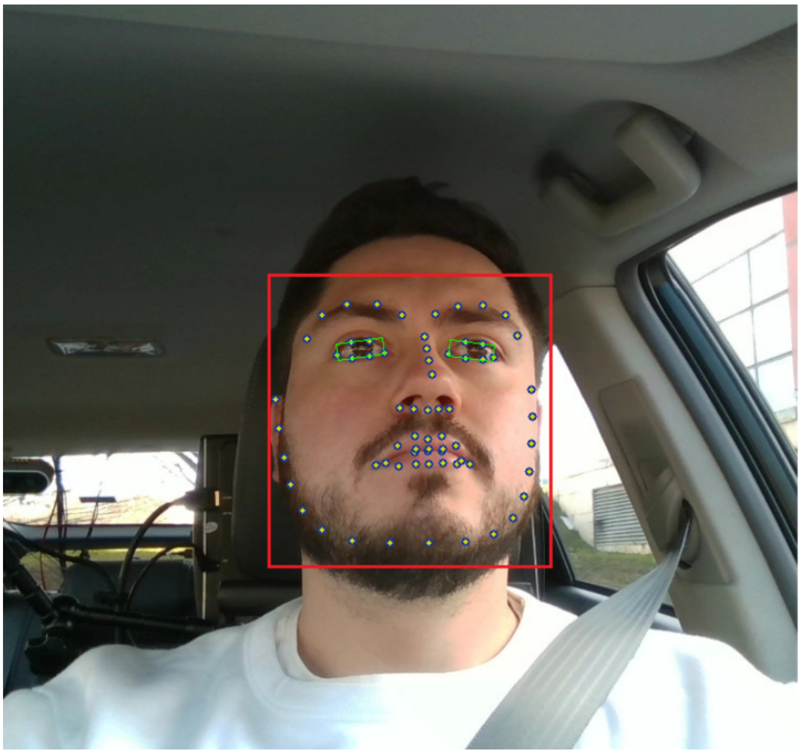
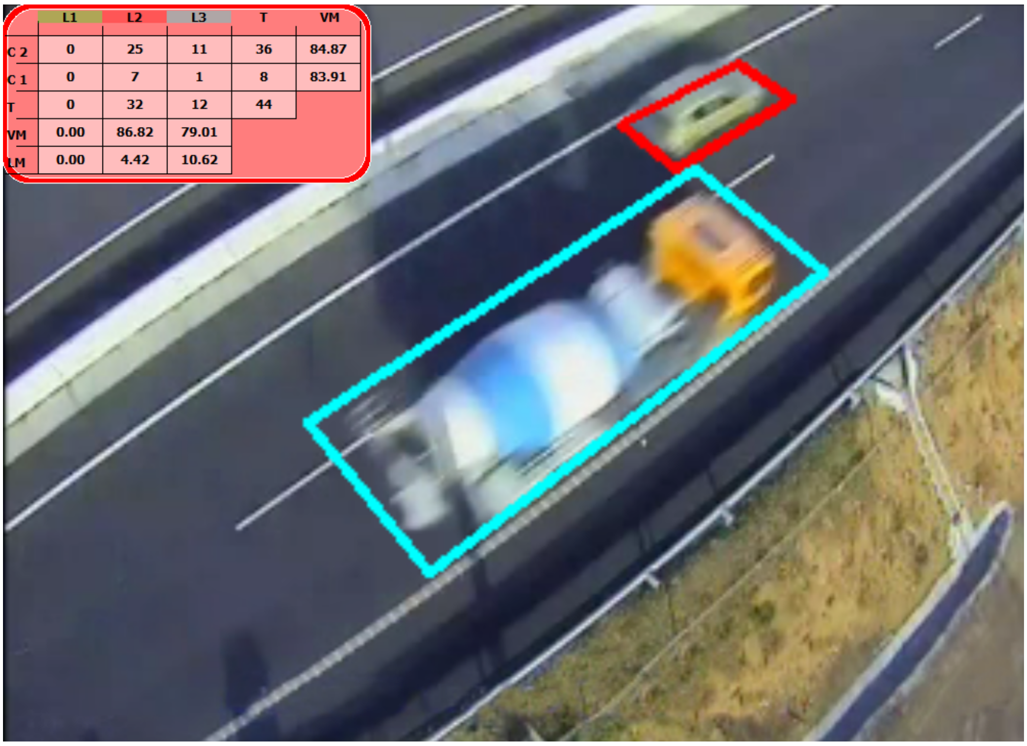


Figure 2.4: Driver monitoring [4].

- High occupancy lane control by counting the number of passengers inside the vehicles.
- Traffic monitoring, jam detections, vehicle counting and classification by speed, length, number of axles, brand, or vehicle type.
- Red light and speed cameras.
- Road maintenance decision support systems.
- Automated road asset inventory.

2.1.4 Data annotation

In the last decade, the field of Computer Vision has shown an important progress in areas such as object detection, recognition, and segmentation thanks to the advances in neural networks, revived as deep learning [13]. This renaissance of an old approach has been facilitated by the increased computational power of GPUs and the increasing availability of labeled data. The performance of systems based on



	L1	L2	L3	T	VM
C 2	0	25	11	36	84.87
C 1	0	7	1	8	83.91
T	0	32	12	44	
VM	0.00	86.82	79.01		
LM	0.00	4.42	10.62		

Figure 2.5: Vehicle counting and classification by its length [5].

deep learning strongly depends on the quality and amount of the available training data.

With the race to achieve Autonomous Vehicles, many automotive, IT, and ride-hailing companies are recording massive amounts of data using fleet of vehicles equipped with cameras, LIDARs, radars, and other sensors. That data is required for the development and validation of their own ADAS and AV. The amount of data to be annotated is so big, that the annotation task is one of the key bottlenecks in achieving improvements. In fact, most of those companies are using virtual simulators to generate automatically annotated synthetic data for training and testing. Although, models trained with synthetic data only can have a very good performance [14], real data is still required for better accuracy in specific domains.

The recorded data is manually annotated. The annotation consists in the identification and segmentation of the group of pixels or 3D points corresponding to the actors and elements (vehicles, pedestrians, signals, road, etc.) present in the images or LIDAR scans. This task is very repetitive and slow. Each camera records at least 30 images per second, usually an AV is equipped with four or more cameras, and

LIDARs records hundreds of thousands points per second.

A human annotator must process each frame of all the cameras, drawing a box around each pedestrian, vehicle, signals, etc. This task can be accelerated using automated pre-detection and tracking tools. With these tools the annotator task is reduced to verify and correct the pre-annotations and complete missing elements. Another kind of annotation is pixel-wise annotation. In this task, the annotator should identify the correspondence of each pixel, drawing a color mask over the image where the color of each pixel identifies an individual element (pedestrian number one, pedestrian number two, etc.). This automated process is known as instance segmentation.

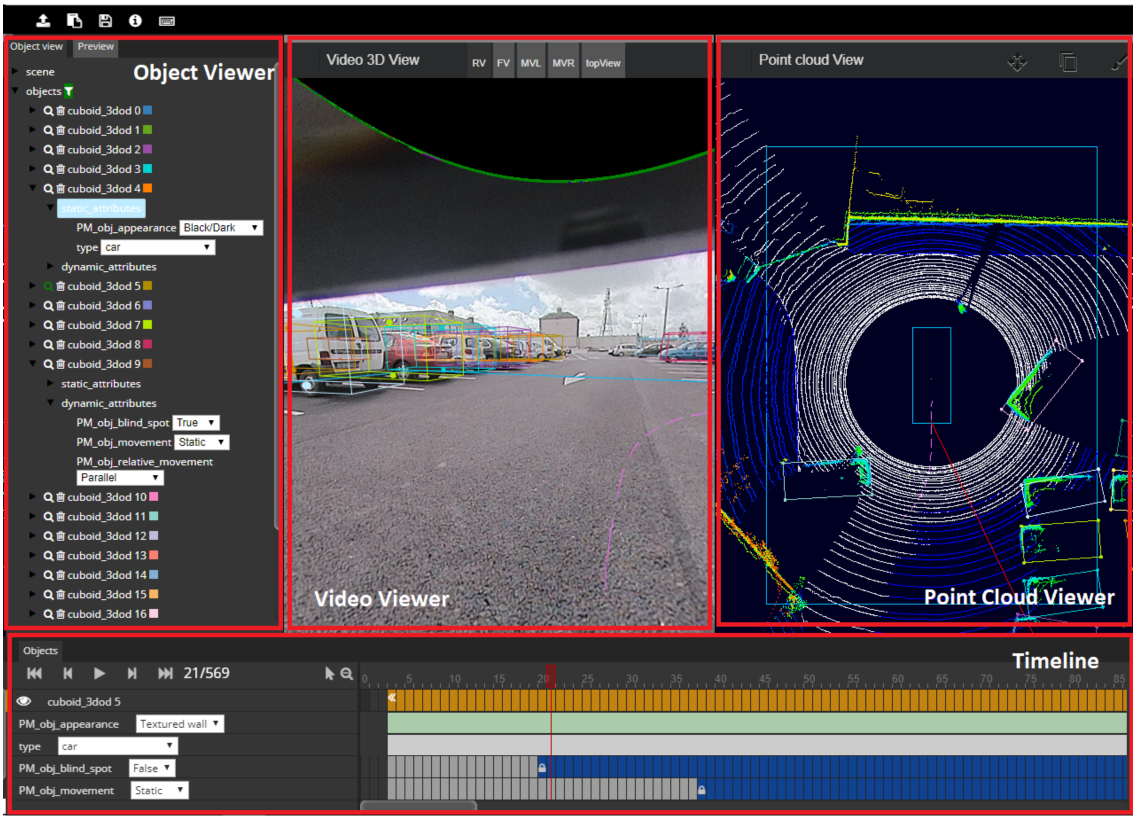
The annotation with LIDAR data is similar. Current spinning LIDARs complete around 10 and 20 revolutions (aka scans) per second. The individual scans of all the LIDARs installed on a vehicle can be registered on a single 360 degrees scan stored as 3D point cloud. The task of the annotator is to process each scan, creating 3D boxes around each vehicle, pedestrian, signal, building, etc., and drawing 2D lines on the ground identifying elements like curbs or the road markings. Fig. 2.6 shows the web-based user interface of the annotation application developed by Vicomtech.

Data annotation is nowadays a growing market employing many thousands of workers. There are many companies that offer data annotation services, for instance:

- Offering specialized software for annotation with automated tools (Scale[11]),
- human annotators (Amazon Mechanical Turk[12], Clickworker[13]), and
- their own annotation platforms and annotators (Mindy Support[14], Hive[15], Playment[16], Humans in the loop[17], Superannotate[18]).

---

[11] http://www.scale.com
[12] http://www.mturk.com
[13] http://www.clickworker.com
[14] http://www.mindy-support.com
[15] http://www.thehive.ai
[16] http://www.playment.io
[17] http://www.humansintheloop.org
[18] http://www.superannotate.com


Object view
Preview
Object Viewer
scene
objects
cuboid_3dod 0
cuboid_3dod 1
cuboid_3dod 2
cuboid_3dod 3
cuboid_3dod 4
static_attributes
dynamic_attributes
cuboid_3dod 5
cuboid_3dod 6
cuboid_3dod 7
cuboid_3dod 8
cuboid_3dod 9
static_attributes
dynamic_attributes
cuboid_3dod 10
cuboid_3dod 11
cuboid_3dod 12
cuboid_3dod 13
cuboid_3dod 14
cuboid_3dod 15
cuboid_3dod 16
Video 3D View
RV
FV
MVL
MVR
topView
Video Viewer
Point cloud View
Point Cloud Viewer
Objects
21/569
Timeline
0
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
cuboid_3dod 5

PM_obj_appearance Black/Dark
type car
PM_obj_blind_spot True
PM_obj_movement Static
PM_obj_relative_movement Parallel

PM_obj_appearance Textured wall
type car
PM_obj_blind_spot False
PM_obj_movement Static

Figure 2.6: Web-based annotation interface [6].

- There are also open-source tools for annotation (makesense.ai[19], labelme[20], coco-annotator[21], VoTT[22], CVAT[23], VGG Image Annotator[24], Semantic segmentation editor[25]).

Tools that provide support in the annotation task of large volumes of data are required. These tools should be capable of exploiting the computing resources and adaptability offered by cloud architectures to pre-annotate the data automatically.

---

[19] http://www.makesense.ai
[20] http://www.labelme.csail.mit.edu
[21] https://www.github.com/jsbroks/coco-annotator
[22] https://www.github.com/Microsoft/VoTT
[23] http://www.github.com/openvinotoolkit/cvat
[24] http://www.robots.ox.ac.uk/~vgg/software/via
[25] https://github.com/Hitachi-Automotive-And-Industry-Lab/semantic-segmentation-editor

2.1.5 LIDAR sensors

LIDAR sensors are capable of measuring distances. 2D LIDARs use a single laser beam deflected by a spinning mirror to measure distances in a single plane. 3D LIDARs combine multiple lasers with different altitude angles, being capable of generating 3D point clouds 360 ° around the sensor. The latest Velodyne[26] Alpha Prime 3D LIDAR includes 128 laser beams and can sense points at 200m distance with a frequency of 20 revolutions per second. Recently, solid state LIDARs, without mechanical parts, are being commercialized. These LIDARs have a limited field of view, they cannot sense the 360 ° environment, but they are smaller and therefore they could be better integrated in the vehicles and they promise to be cheaper.

The ability of sensing the 360 ° environment of the vehicle in 3D and at long distances, makes LIDAR a very powerful sensor for autonomous driving. LIDARs are used in autonomous driving for detection of objects and localization and mapping mainly. In addition to measuring distances, LIDARs can also measure the reflectivity of the surfaces, making them useful for detecting the horizontal road markings. Recently, many companies commercializing LIDAR sensors have emerged, focusing their products for the automotive industry. Velodyne, the pioneer company, developed their first 3D LIDAR for the DARPA Grand Challenge in 2007. Many other companies have followed: Ouster, Robosens, Hesai, Livox, Ibeo, LeddarTech, LeiShen, Luminar, Quanergy, Innoviz, Sense, Cepton, Benewake.

Mobile mapping systems, such as the RIEGL VMX-2HA[27], use high performance 2D LIDARs combined with cameras and a global positioning system for generating geo-referenced high-density 3D point clouds [23]. These expensive systems are designed to be installed on specialized vehicles and to generate dense and very precise 3D reconstructions for mapping or infrastructure management.

---

[26] https://www.velodynelidar.com

[27] http://www.riegl.com

Velodyne
VMX-2HA

Figure 2.7: Velodyne Alpha Prime 3D LIDAR and RIEGL VMX-2HA mobile mapping system

2.1.6 Horizontal road markings

Horizontal road markings are a key visual element of the road infrastructure as they provide guidance and information about navigation, safety, and enforcement. The lane markings that delimit the lanes and road boundaries are particularly important. The detection of these markings is required for functions such as lane keeping assisted system (LKAS), Lane Departure Warning (LDW), and automatic maneuver planning with precise positioning. Advanced driver-assistance systems (ADAS) and autonomous driving (AV) both require self-localization technologies accurate to a few centimeters. This precision can be achieved with a combination of multiple sensors [24] such as video cameras, Integrated Navigation Systems (INS), combining Global Navigation Satellite Systems (GNSS) and Inertial Measurement Units (IMU), and LIDAR. Moreover, High Definition (HD) maps with lane-level information are necessary for safe maneuver planning. The creation of these maps on a global scale would be an unfeasible challenge without automation.

Road markings suffer extraordinary degradation due to wearing from atmospheric and traffic conditions. Road administrations require tools to accelerate the creation of an inventory of road assets and for maintenance purposes [23,25]. Therefore, many researchers have focused on the automated detection and evaluation of markings [26–28].

The advantage of LIDAR sensors for road marking detection compared to image-based methods is that they are less sensitive to illumination or weather conditions.

On the other hand, the data density of LIDAR could be lower depending on the sensor and they are expensive. LIDAR estimates metric information directly, thereby avoiding the difficulties associated with stereo cameras or ground plane calibration; it is also easier to remove vertical objects.

2.2 Objectives

The main objective of this thesis is to research on the use of LIDAR sensors and computational methods for the automated detection and annotation of lane markings. Specifically, the work has focused on the realization of an annotation assisting system for the human operators that provides them with a detection information that can be used to speed up and ease their work.

Operational objectives implemented in the pursue of this general objective are:

- Implementation of a web-based annotation tool for LIDAR data that has been used and validated by professional annotators.
- Definition and implementation of preprocessing steps that prepare the data for automated annotation, including sensor calibration and filtering noisy points.
- Gathering an experimental dataset from real life vehicles that can be used for quantitative evaluation of the proposed system.
- Definition of an evaluation process involving human operators.
- Carrying out computational experiments for the assessment of the proposed system.

2.3 Contributions

The contributions of this thesis are:

- A method for the automatic annotation of lane markings using LIDAR data. This method is intended as a support for the human operator doing the commercial annotation of road markings. The lane markings annotations are being used as ground-truth for the development of new ADAS systems that will contribute to increase the safety and reduce the environmental impact of road transportation. The method was published in [15].

- The integration of the implemented method as a back-end allows the exploitation of the computing resources offered by cloud architectures.
- The developed web-based tool allows distributed remote working environment. This tool is still being developed by a group of researchers at Vicomtech. The author has collaborated in the synchronization, processing and visualization of video and LIDAR data. The tool was presented at the Web3D 2019 conference [6].
- A methodology for the evaluation of the precision of these automatic annotations.

2.4 Research environment and context

The research work reported in this Thesis was done during the participation of the PhD candidate in different research projects at the Intelligent Transport Systems and Engineering department of the applied research center Vicomtech, with the collaboration and guidance of Professor Manuel Graña (Computational Intelligence Group from the University of the Basque Country UPV/EHU), PhD Oihana Otaegui (Director of Intelligent Transport Systems and Engineering at Vicomtech), PhD Marcos Nieto (Principal Researcher of Intelligent Transport Systems and Engineering at Vicomtech) and PhD Julián Flórez (General Director of Vicomtech).

2.4.1 Projects

The most important related R&D projects funded under H2020:

- Headstart: Will define testing and validation procedures of Connected and Automated Driving (CAD) functions including its key enabling technologies (i.e., communications, cyber-security, positioning) by cross-linking of all test instances such as simulation, proving ground and real world field tests to validate safety and security performance according to the needs of key user groups (technology developers, consumer testing groups and type approval authorities).

  - Full title: Harmonised eruropean solutions for testing automated road transport
  - Funded under: H2020-EU.3.4.2.2., H2020-EU.3.4.1.2., H2020-EU.2.1.6.3. and H2020-EU.2.1.6.1.2..
  - Duration: 1 February 2018 – 31 January 2021
  - Project coordinator: Daimeler AG

- TransSec: The project addresses terrorism attacks with trucks in European countries. The project aims to initiate the development of security trucks that cannot be misused for other purposes such as terror attacks.
  - Full title: Autonomous emergency manoeuvring and movement monitoring for road transport security
  - Funded under: H2020-EU.3.4.
  - Duration: 1 January 2019 – 31 December 2021
  - Project coordinator: Idiada Automotive Technology
- Autopilot: The project aimed to bring IoT into the automotive world transforming connected vehicles into highly and fully automated vehicle.
  - Full title: AUTOmated driving Progressed by Internet Of Things
  - Funded under: H2020-EU.3.1.4 and H2020-EU.2.1.1.
  - Duration :1 January 2017 – 31 December 2019
  - Project coordinator: ERTICO
- Cloud-LSVA: European project aimed to build a software platform for efficient and collaborative semiautomatic labelling and exploitation of large-scale video data. The project addresses the needs of the automotive industry for tools that

can manage the extremely large volumes of data required for the development of ADAS and cartography market. The works of the Thesis are fully aligned with the objectives of this project.

  - Full title: Cloud Large Scale Video Analysis
  - Funded under: H2020-EU.2.1.1
  - Duration: 1 January 2016 – 31 December 2018
  - Project coordinator: Vicomtech

- inLane: European project with the vision to develop a low-cost, lane-level, precise turn-by-turn navigation application through the fusion of EGNSS and Computer Vision technology.
  - Full title: Low Cost GNSS and Computer Vision Fusion for Accurate Lane Level Navigation and Enhanced Automatic Map Generation
  - Funded under: H2020-EU.2.1.6.
  - Duration: 1 January 2016 – 30 June 2018
  - Project coordinator: Vicomtech
- VI-DAS: This project goals are improved road safety by development and deployment of ADAS and navigation aids in societally acceptable and personalised manner, based on a reliable combination of the overall traffic scene understanding and essential consideration of the driver's physical, mental, demographic and behavioural state.
  - Full title: Vision Inspired Driver Assistance Systems
  - Funded under: H2020-EU.3.4.
  - Duration: 1 January 2016 – 31 August 2019
  - Project coordinator: Vicomtech

2.5 Publications

Publications related with the research done for this thesis:

- **Barandiarán, J**, Nieto, M, Cortés, A., Otaegui, O., Flórez, J. and Graña, M., Automated Annotation of Lane Markings Using LIDAR and Odometry, in IEEE Transactions on Intelligent Transportation Systems, October 2020, doi: 10.1109/TITS.2020.3031921.

- Cortés, A., Rodríguez, C., Vélez, G., **Barandiarán, J.** and Nieto, M., Analysis of Classifier Training on Synthetic Data for Cross-Domain Datasets, in IEEE Transactions on Intelligent Transportation Systems, July 2020, doi: 10.1109/TITS.2020.3009186.

- Mujika, A., Dominguez, A., Tamayo, I., Senderos, O., **Barandiarán, J.**, Aranjuelo, N., Nieto, M. and Otaegui, O. (2019). Web-based Video-Assisted Point Cloud Annotation for ADAS validation. In Web3D 2019 – The 24th International ACM Conference on 3D Web Technology (pp. 1–9), doi: 10.1145 / 3329714.3338128.

Previous publications related with the application of computer vision for ITS:

- De-Maeztu, L., Elordi, U., Nieto, M., **Barandiarán, J.** and Otaegui, O., A temporally consistent grid-based visual odometry framework for multi-core architectures. Journal of Real-Time Image Processing 10, 759–769 (2015), doi: 10.1007/s11554-014-0425-y.

- Nieto, M., Ortega, J. D., Otaegui, O., Cortés, A., **Barandiarán, J.**, Unzueta, L., Computer vision: the emerging cost-effective technology for vehicles, 9th ITS European Congress, ERTICO 2013.

- Unzueta, L., Nieto. M., Cortés, A., **Barandiarán, J.**, Otaegui, O.and Sanchez, P., Adaptive Multicue Background Subtraction for Robust Vehicle Counting and Classification, in IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 2, pp. 527-540, June 2012, doi: 10.1109/TITS.2011.2174358.

- Nieto, M., Unzueta, L., **Barandiarán, J.**, Cortés, A., Otaegui, O.and Sanchez, P., Vehicle tracking and classification in challenging scenarios via slice sampling. EURASIP J. Adv. Signal Process. 2011, 95 (2011), doi: 10.1186/1687-6180-2011-95

- Nieto, M., Unzueta, L., Cortés, A., **Barandiarán, J.**, Otaegui, O.and Sanchez, P., Real-time 3D Modeling of Vehicles in Low-cost Monocamera Systems. Proceedings of the Sixth International Conference on Computer Vision Theory and Applications VISAPP 2011, pp. 459-464, March 2011.

- Muñoz, O., Gloria, P., Hernández, J., Gozálvez, J., Unzueta, L., Cortés, A., **Barandiarán, J.**, Otaegui, O.and Sanchez, P., INTELVIA-An Integral System for Safe and Intelligent Traffic Operational Management, 17th ITS World Congress 2010.

Previous publications related with other applications of computer vision:

- Congote, J., Barandiarán, I., **Barandiarán, J.**, Montserrat, T., Quelen, J., Ferran, C., Mindan, P. J, Mur, O., Tarres, F., Ruiz, O., Real-time depth map generation architecture for 3D videoconferencing, 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video 2010, pp. 1-4, doi: 10.1109/3DTV.2010.5506599.

- **Barandiarán, J**, Murguia, B. and Boto, F., Real-Time People Counting Using Multiple Lines, Ninth International Workshop on Image Analysis for Multimedia Interactive Services, Klagenfurt, 2008, pp. 159-162, doi: 10.1109/WIAMIS.2008.27.

- **Barandiarán, J.,** and Borro, D., Edge-Based Markerless 3D Tracking of Rigid Objects, 17th International Conference on Artificial Reality and Telexistence (ICAT 2007), Esbjerg, Jylland, 2007, pp. 282-283, doi: 10.1109/ICAT.2007.62.

2.6 Structure of the thesis

This thesis is structured as follows:

- Chapter 3 presents the state of the art of computer vision techniques related to lane markings detection and annotation.

- Chapter 4 presents the proposed method for the automatic annotation of lane markings. The preprocessing steps of the data are detailed, providing the fundamental justification of the proposed system.

- The evaluation of this method is presented in Chapter 5. The data used for the evaluation is introduced. The overall methodological considerations are detailed. Finally, we provide quantitative results of the evaluation process, both from the point of view of quantitative measures of recognition accuracy and human related improvement of their performance.

- Finally, Chapter 6 presents the conclusions and future lines of work.

Chapter 3

State of the art

This Chapter presents in Section 3.1 a review of research works related with the processing of LIDAR data for road marking detection, which is the main objective of this Thesis. Section 3.2 reviews the different approaches for road detection. Section 3.3 reviews LIDAR intensity calibration issues and approaches. Finally, works focused on the automated generation of annotations are reviewed in section 3.4.

3.1 Road markings detection

This section presents a review of previous research works about the detection and annotation of road marking using LIDAR data.

3.1.1 Non real-time methods

This category contains research works that focus on the creation of high definition (HD) maps, roadway feature mapping, 3D street modeling and road asset inventory, where real-time processing is not required [26–37]. Most of these works used one or more single-layer high-performance LIDAR instrument mounted on the roof of a vehicle which is oriented perpendicular to the road level to generate high-density scans of the road surface. As the data is processed offline, consecutive scans are accumulated using the odometry estimated with a GNSS/IMU system to obtain a point cloud with high density of points. For example, in [26] a density of 4,000–7,000 points/m^2 was reported. Among these methods two sub-categories exists:

3.1.1.1 Image Processing

These methods apply image processing techniques to top-view or bird's eye view raster images of the road surface generated rendering the point cloud [7, 26, 28, 29, 31, 32, 34–36].

In [31], a geo-referenced reflectance intensity image was generated for the entire point cloud using inverse distance weighted interpolation. Road markings were detected by applying a fixed threshold to the intensity value, finding connected components in the resulting binarized image, and filtering by shape moments. Finally, the progressive probabilistic Hough transform was used to extract the segments of dashed lines. The same kind of image was generated in [29], but instead of generating an image for the entire point cloud, the data was partitioned into blocks around each position of the vehicle, and a reflectance image was generated for each block. Multi-threshold segmentation dependent on the point-density was then used to extract the road markings. Finally, authors apply morphological operators to reduce detection noise. This method was improved in [26] using dynamic thresholding based on weighted neighboring difference histogram-based dynamic thresholding and multi-scale tensor voting to remove noise.

In [32] raster images were generated for sections that are 30 m wide and 10 m long around the navigation points. The intensity of the 3D points that fall within each pixel was averaged and normalized with respect to their global minimum and maximum, using a pixel size of $6cm^2$. The images were segmented by applying range dependent thresholding. In [35] a Gaussian mixture model with two components was used to detect points belonging to the road markings before creating the intensity image. This step reduced the amount of noise in the image. The image is segmented in cells with similar laser beam angle markings and Otsu thresholding was used for each cell. In [28] the holes in the image were filled using an in-painting algorithm. The markings were segmented using an Expectation-Maximization algorithm.

In [34] top-view images of the point clouds were used to train an end-to-end segmentation Convolutional Neural Network (CNN). Images with a fixed size of $6,144 \times 3,072$ pixels were generated for each patch of the point cloud with $1cm^2$ per pixel. These images were cropped to patches of 512×512 pixels to use as input for the

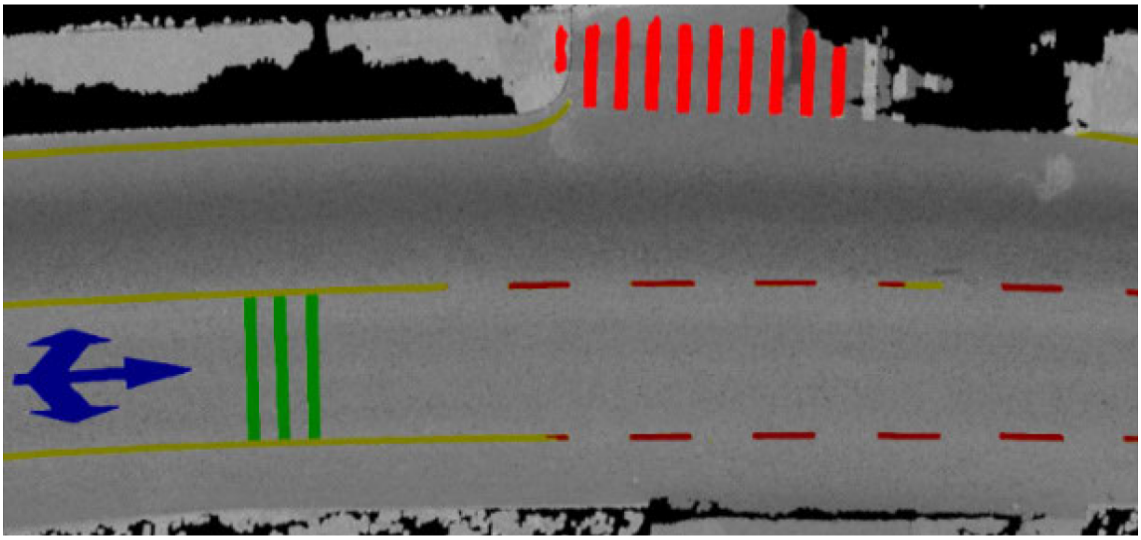


Figure 3.1: Road markings segmented and classified by conditional GAN [7].

CNN. The output of the CNN was a probability image that was smoothed to remove detection noise. Afterwards it was thresholded, so that lane markings were detected as connected components and fitted by third-degree polynomial curves. Similarly, in [36] a CNN encoder-decoder network was trained to extract road markings from interpolated top-view images.

Another deep-learning approach was proposed in [7]. A conditional generative adversarial network (cGAN) was used to directly segment and classify road markings in top-view images of the point cloud. The network generated colorized images with the segmented and classified road markings, assigning a color for each class (solid lines, dashed lines, arrows, crosswalk, etc.). Then, the road markings were clustered and vectorized using different methods depending on the class. Fig. 3.1 shows a visualization of results from this approach.

Regarding data fusion approaches, [38] proposed to segment and classify the road marking fusing image and LIDAR information. They used a DeepLabV3 network [39] to segment the top-view of the camera image and combined the segmented image with a top-view of the point cloud as input of another segmentation network.

3.1.1.2 Scan line Processing

Instead of converting the point clouds to images, some methods directly process the points analyzing the scan lines produced by the LIDAR [27, 30, 33, 37]. In

[30] intensity peaks were detected in each scan line using adaptive thresholding. The detected candidate points were projected to a 2D image and lane markings were detected using the Hough transform. After removing false positives using the trajectory of the vehicle and geometry checks, RANSAC curve fitting was used to localize each marking accurately. In [37] the intensity values of each scan line were smoothed using median filtering and peaks were detected in the intensity gradient. In [27] markings were detected as local maxima in the intensity response of each LIDAR profile. Then, points were grouped based on distance. Segments of dashed lines were grouped based on their principal direction. Finally, spline curves were fitted to each lane mark. A lane-level map generation method is presented in [33], where lane marking points were detected by applying a 1D Laplacian filter to the intensity of each laser beam. Then line-segments were detected with the RANSAC algorithm. These segments were used in a particle filter for lane estimation.

3.1.2 Real-time methods

One of the purposes of lane marking detection is the localization of the vehicle either inside the ego-lane or globally using an HD map. This application requires the data to be processed in real-time. This is a challenge, not only because the algorithms must be very fast, but because the amount of available data is significantly reduced. The following papers presented real-time methods that detected lane markings by processing each LIDAR scan independently or by using an accumulation of the past scans in a short interval [8, 9, 24, 40–43]. These studies used multi-layer and multipurpose scanners that were not specifically installed to scan the road, resulting in point clouds with a much lower density. In addition, the accumulation of past scans is limited owing to the real-time requirement.

In [40] a real-time method is presented for lane markings detection using data from a four-plane laser scanner. The points were projected in the traveling direction to create a histogram quantizing them into bins. The sensitivity of the scanner was adjusted to detect only the lane markings. Consecutive scans were accumulated in the histogram and an adaptive threshold based on the histogram gradients was used. In the last step, linear regression was applied to the detected points.

The lane markings detection approach developed by Team AnnieWAY for the DARPA Urban Challenge 2007 is described in [41]. A LIDAR with 64 layers was used. Consecutive scans were registered and accumulated using GNSS/INS information. Points with a high-intensity gradient were thresholded and lane segments were detected using the Radon transform. The detected segments were used to estimate an offset with a digital map.

In [42] an image or reflectivity map was composed by stacking the consecutive scan-lines of a one-dimensional laser scanner. This map was thresholded with dynamic thresholding and the Canny edge detection filter was applied. Lane markings of the ego-lane were detected by accumulating white pixels from the thresholded and edges of images in two detection windows at both sides of the position of the vehicle. The lanes were modeled with a clothoid curve and tracked with a Kalman filter.

In [43], lane markings were detected by processing the reflectivity signal of each scan line of a 64-layer LIDAR. After applying a FIR band-pass filter to remove noise, candidate points were found using a fixed threshold. The points were grouped and fitted to clothoid curves. In sequential frames the lanes markings were tracked using a Kalman filter.

In [9], a LIDAR sensor with 32 layers was used. A modified version of Otsu thresholding was applied to each scan line. A localization method was presented based on the extracted road markings. The same sensor was used in [8]. Road marks were extracted using the laser intensity with a fixed threshold and road lines were detected by searching for a set of parallel lines. The detected road lines were represented as 3D points and integrated to generate a lane-level digital map using a GNSS/INS system. Fig. 3.2 shows instances of the capture of point clouds in both systems with the identification of the detected road marks.

In [24], a navigation filter combining GNSS, IMU, LIDAR, and camera was presented. The LIDAR and camera were used to estimate the lateral position inside the ego-lane. Using four-layer LIDAR, the ego-lane markings were detected by fitting an ideal scan intensity profile to the actual LIDAR data.

The point clouds we used in our experimentation have a relatively low density,

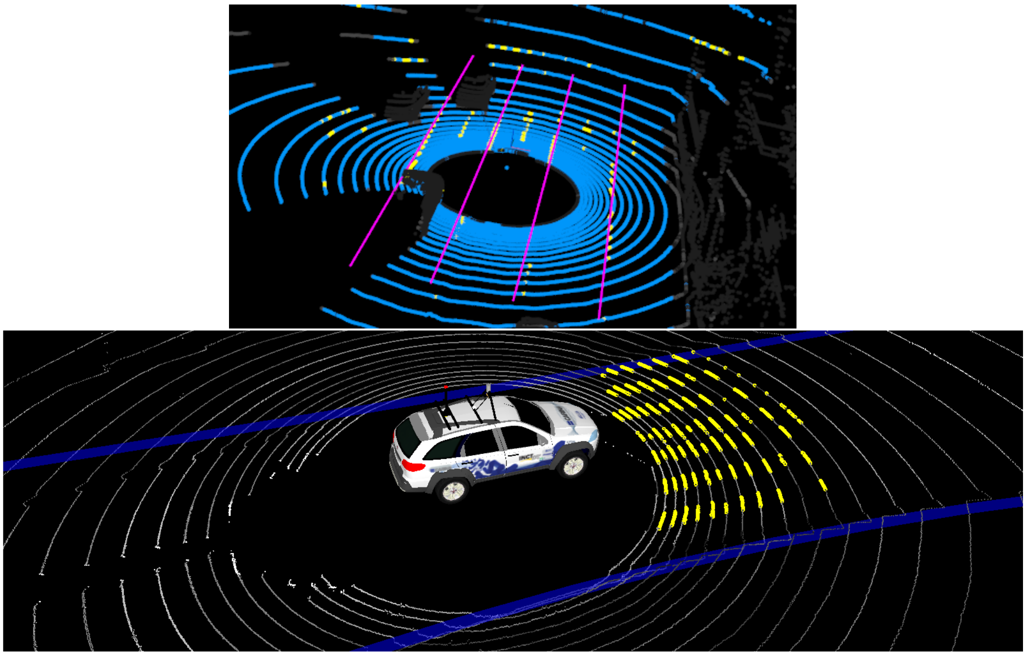


Figure 3.2: Real-time road markings detection: above [8], below [9].

between 50 and 1,000 points *per* m^2, which means that the methods based on the rasterization of top-view images are unsuitable to solve our problem. The low point-density causes the lane markings to appear blurred on the raster images, which complicates the segmentation. On the other hand, methods based on the scan-line based methods do not leverage the accumulation of previous and future scans, as is possible in offline processing. This paper presents a batch processing offline method based on the analysis of histograms, similar to the scan-line based method presented in [40], but generating virtual scan-lines using the accumulated point clouds. A similar idea was proposed in [29] for road surface extraction to detect the curbs in pseudo scan-lines.

3.2 Road points segmentation

Most of the previously reviewed methods, require a first step to detect the points of the road surface. This step, also known as road extraction, prevents problems with moving vehicles or low objects as barriers and reduces the number of points to be processed for the road markings detection. Some of the works proposed to segment the road points aim at detecting the curbs present in urban environments. In [26,29], the point cloud was processed by blocks along the trajectory of the vehicle. At each block, a vertical profile of the point cloud was generated perpendicularly to the trajectory and the curbs were detected via slope and elevation-difference thresholds. In [9], curbs were detected by thresholding the distance between consecutive rings generated by a 3D LIDAR. In [44], the road limits were detected applying a principal component analysis to the local neighborhood of each point of each scan-line.

In a different approach, [45] partitioned the point cloud using an octree and removed the non-road voxels thresholding by height. [46], knowing that the orientation of the road surface changes gradually, proposed to extract the road by fitting planes using RANdom SAmple Consensus (RANSAC) to sections of the point cloud and applying a Kalman filter to track the change of the local planes. A similar approach was used in [47], the road surface at each scan-line was approximated by a second order polynomial using RANSAC.

In [32], the road limits were detected using parametric active contour models applied to 2D top-view images generated with the elevation, intensity, and pulse width attributes of the point cloud. The pulse width helps to distinguish different types of surfaces. The images were generated by sections of 30x10x5 m with an overlap of 2 m for consecutive sections. The final road contour was calculated intersecting the snake curve estimated from consecutive sections. They obtained very good results in rural, urban, and national primary roads.

A real-time localization method was presented in [10]. Using a 3D LIDAR the road limits were detected by analyzing the scan-lines or rings data. Assuming that the road is a flat and smooth surface, the points were segmented fitting circular arcs and checking the fitting error. Lane markings were detected accumulating two consecutive scans using inertial data and applying a fixed threshold and the Hough transform to a top-view image.

A fully convolutional neural network was used in [48] for detecting the road using top-view images encoding mean elevation and density of the point cloud. They generated images of 200x400 pixels with a resolution of 10 cm per pixel. They trained and tested the algorithm with the KITTI-ROAD dataset [49] that includes road area annotations in the camera frames, LIDAR data and the camera-LIDAR

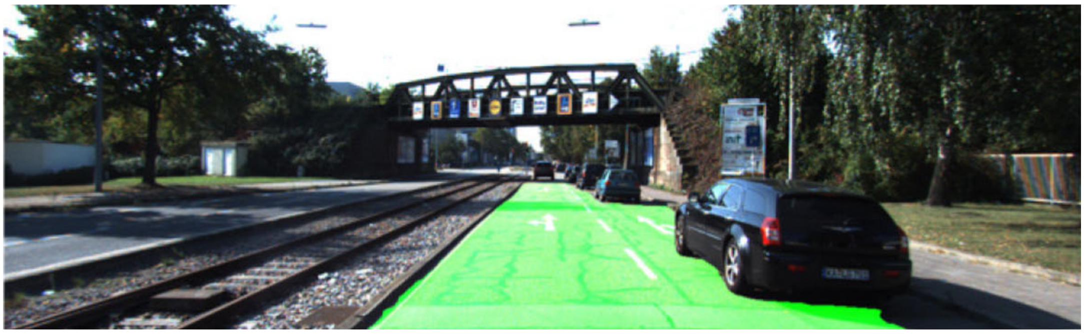


Figure 3.3: Road detection [10].

calibration. To generate the training data, the point clouds were interpolated and projected to the camera frame and each point was labeled using the annotated road mask. The segmented point clouds were then transformed to the top-view images. To augment the training data, each image was rotated around the vertical axis and mirrored.

In [10], a real-time LIDAR-Camera road detection method was presented. The LIDAR point cloud was transformed to images with the LIDAR's perspective and a road mask was generated using a height difference threshold between adjacent pixels. This sparse mask was projected to the camera image an up sampled using a Delaunay triangulation. The camera images were also segmented using a CNN. LIDAR and image segmentations were finally fused using a multi-modal conditional random field model. Fig. 3.3 shows a visualization of the road detection achieved with this approach.

3.3 LIDAR reflectivity calibration

The intensities or reflectivity values measured by the sensor cannot be directly used because they usually suffer from calibration problems. In some sensors, the intensity attenuates with the distance and angle of incidence. This attenuation must be compensated applying some functions supplied by the manufacturer or the sensor must be calibrated. In [50], the wall of a building with a constant color plane was used to estimate the attenuation of the lasers. With 3D LIDARs, the intensities measured by the different lasers that compose the sensor usually do not agree with

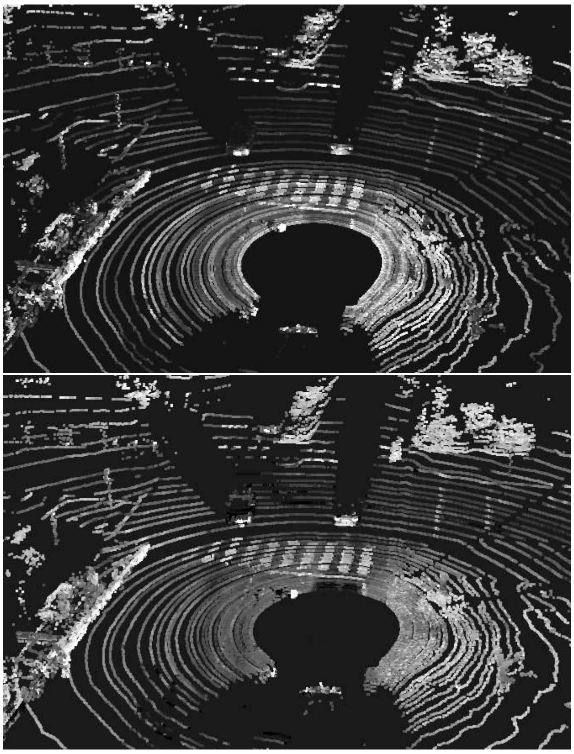


Figure 3.4: Before and after intensity calibration [11].

each other, meaning that they measure different reflectivity for the same physical point. These differences in response should be calibrated and compensated. This calibration can be performed analyzing a recorded sequence of scans where the sensor was moved back and forth in an environment that contains surfaces with different known reflectivity [9, 11, 51]. For each intensity value reported by each laser, a compensation factor should be estimated to minimize the intensity difference with the other lasers. The result of the calibration process is a look up table, where each cell represents the calibrated intensity for each measured intensity by each laser. Fig. 3.4 gives an instance of data cloud before and after the sensor calibration.

A data-driven computational mapping method of the ground reflectivity with multiple LIDARs was presented in [52]. The aim of the work was to generate ground maps that preserve the edge sharpness. Individual gradient maps were generated for each laser, which are then fused applying selection fusion and de-noising operators. The final map is reconstructed from the fused gradient map.

3.4 Image and point cloud annotation

In this Section, some works focused on automated LIDAR data annotation are reviewed. [12] proposed to reduce the time required for semantic instance labelling by annotating 3D point clouds and transferring the information to the images. They said that the annotation in 3D is more time efficient because objects are easily separated in 3D and the annotations from a single point cloud can be transferred to multiple sequential images. The point clouds were manually annotated, placing cuboids and ellipsoids around static objects, and drawing 2D polygons in bird's eye view, that are then extruded into 3D, around the road, sidewalks, grass, etc. They proposed a method for transferring the 3D annotations to image segmentations using a conditional random field model. Fig. 3.5 shows some results of this process.

A similar strategy was applied in the Apolloscape dataset [53] for annotating static objects. Additionally, dynamic objects were automatically pre-annotated using an image segmentation network. Lane markings were manually labelled drawing 2D polygons on a bird's eye view of the point cloud.

In [54], a ground-truth generation method for 3D object detection is presented. The annotators were asked to make mouse clicks on all instances of a single object class, one click per object, on a colorized LIDAR point cloud. Then a PointNet model [55] was modified to automatically segment the points belonging to each object taking the annotated clicks as input. Then, with the segmented points, a second network was used to estimate an approximate centroid of the bounding box of each object, and another network to estimate the final bounding box.

In [56], an image-based lane detection method and a framework for an automated ground truth generation are presented. They proposed to annotate the position of lane markings using time-sliced (TS) images, that are generated stacking specific rows of pixels from consecutive video frames. The annotators had to place a few points along the lane marking in two or more TS images. The annotated points are interpolated using a spline. The interpolated points are then transformed to the input frames and interpolated again to generate the ground truth for each lane marking.

A method for annotation of road and ego-lane using a single dashboard camera

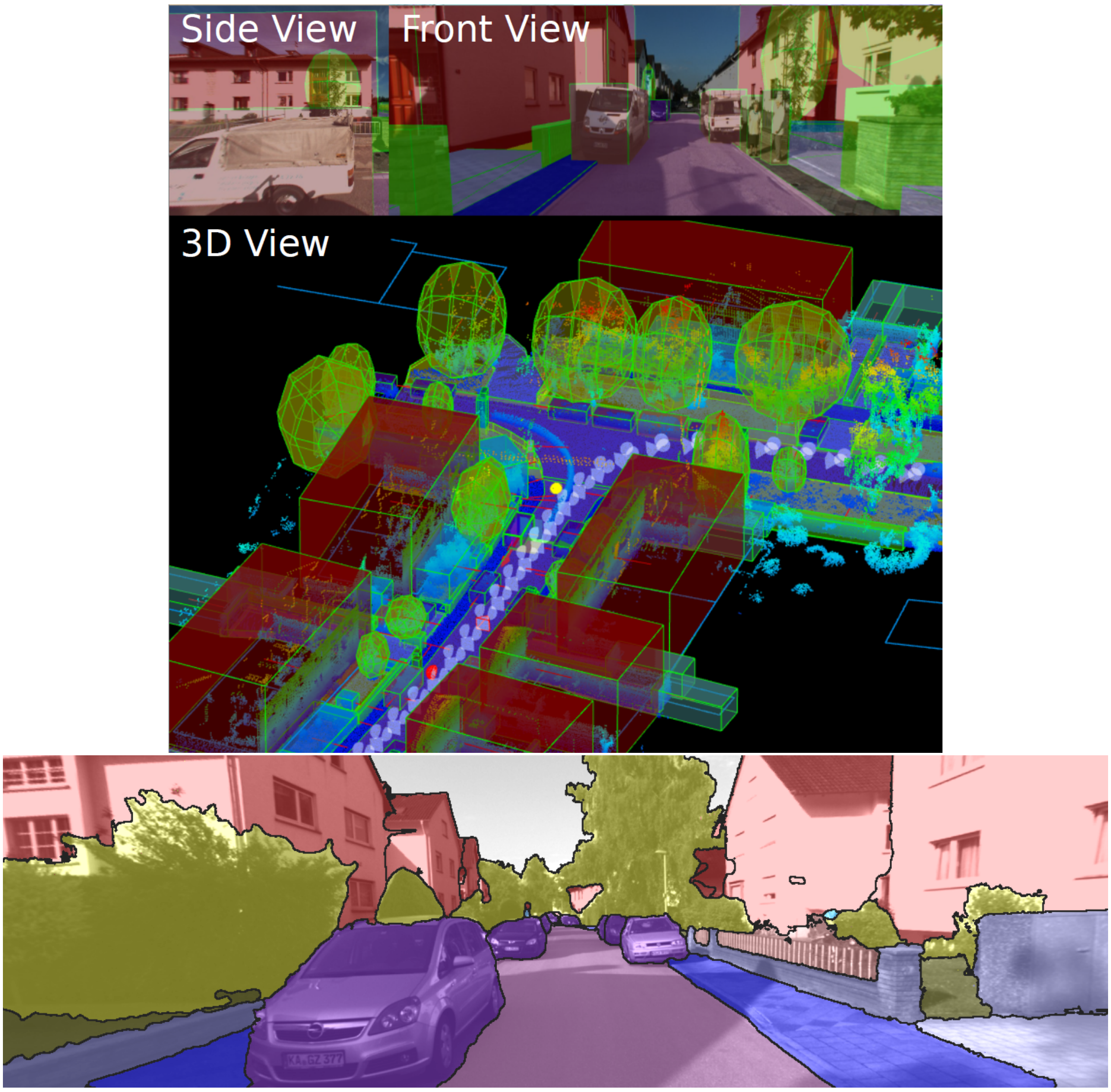


Figure 3.5: 3D annotation and annotations transferred to 2D segmentations [12].

was presented in [57]. In the first place, 3D reconstructions of 200 m sequences were computed using a structure from motion framework. Then, the road points are segmented knowing the camera position inside the vehicle and assuming that the road is flat. The limits of the ego-lane are fixed, and it is assumed that the vehicle travels parallel to and centered in the ego-lane. This automatic 3D annotations are then projected to the images and manually corrected by annotators. With the generated dataset they trained a SegNet model [58] for automatic segmentation of the road and the ego-lane.

Chapter 4

Lane markings automated annotation

This Chapter presents the problem of lane markings annotations and the outline of the proposed approach in Sections 4.1 and 4.2. Section 4.3 describes the different preprocessing steps required for the preparation of the data before applying the proposed automated detection system. Finally, Section 4.4 gives the details of the proposed approach for lane markings detection.

4.1 Problem statement

As said in Chapter 2, lane markings are a key visual element of the road infrastructure for navigation and safety. The development of AV and ADAS require large amounts of annotated lane markings data providing the ground-truth to train and validate deep learning approaches [13]. The manual annotation of lane markings is a very slow and tough task as the edge of each line must be precisely located.

The annotation can be directly done on the images from the cameras surrounding the vehicle. However, the relative positions between the cameras and the road surface plane must be known to correctly project the annotated points to the road surface. The positions between the cameras and the ground plane can be previously calibrated, but the road surface is usually not strictly planar introducing discrepancies between the cameras, where a point appears correctly located in one camera and incorrectly in another one. To fix the projection, the annotator should change the height of each point to locate them over the road surface. Another possibility is to reconstruct the 3D surface of the road, but this can be very difficult depending on the weather and illumination conditions.

Alternatively, the annotation of lane markings can be done using LIDAR data. As these sensors directly measure 3D points of the road surface, the annotations are correctly located *per se*. If the relative positions between the cameras and LIDARs were previously calibrated, the annotations done with the LIDAR data can be projected onto the images, as shown in Fig. 4.1. Another advantage of LIDAR data over optical images is that the individual scans can be accumulated to generate a single point cloud if the odometry of the vehicle is known. For the annotators it is easier to move through a continuous point cloud that jumping back and forth between scans or images. These reasons make LIDAR data rather attractive for the task of the annotation of lane markings.

Despite the advantages of LIDAR data over images for the annotation of lane markings, the manual annotation using point clouds is still a very slow process. Therefore, it is very interesting to develop tools that can automate as much as possible the annotation task. The problem is how to automatically pre-annotate LIDAR data to help the annotators to finish their task faster. The automated annotation tool should process the LIDAR data and detect the position and width of each lane marking.

The use of a multi-layer low-frequency general-purpose LIDAR mounted on a vehicle driving at normal speed for identification and segmentation of road markings imposes a dire challenge due to low contrast markings and low-density point clouds, which are in the order of hundreds of points per square meter, as opposed to point clouds with thousands of points generated by high performance scanning systems, as reported in [36]. The different sensitivity of each laser is also challenging because the combined signal becomes very noisy. Reflectivity calibration of the lasers is required for lane markings detection, but this calibration is difficult with low density of points because it requires many observations of multiple lasers for each area. The extrinsic calibration of the LIDAR is also very important, because the accumulation of the

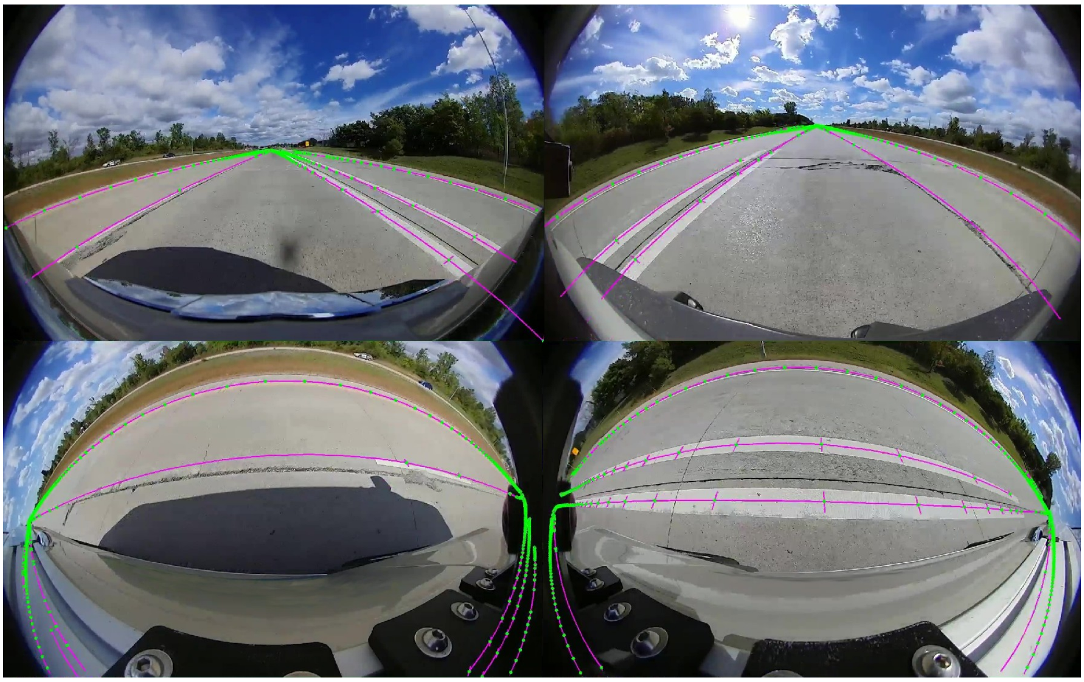


Figure 4.1: Detected lines back-projected to the images from the cameras of the vehicle.

scans using the odometry must be precise to maximize the contrast of the lane markings against the road. However, despite these challenges, such low-frequency systems are the focus of this work, since they are notoriously simpler and cheaper to set-up and can support multi-purpose recording stages, decreasing costs, and favoring the generation of larger datasets in less time.

The main objective of this research work is to efficiently generate annotations by batch processing large amounts of data daily recorded by test vehicles under natural driving conditions. The exponential growth of annotation time and cost can be controlled with automatic pre-annotation if it is efficient and the hourly cost of the required computational cluster is kept low [59].

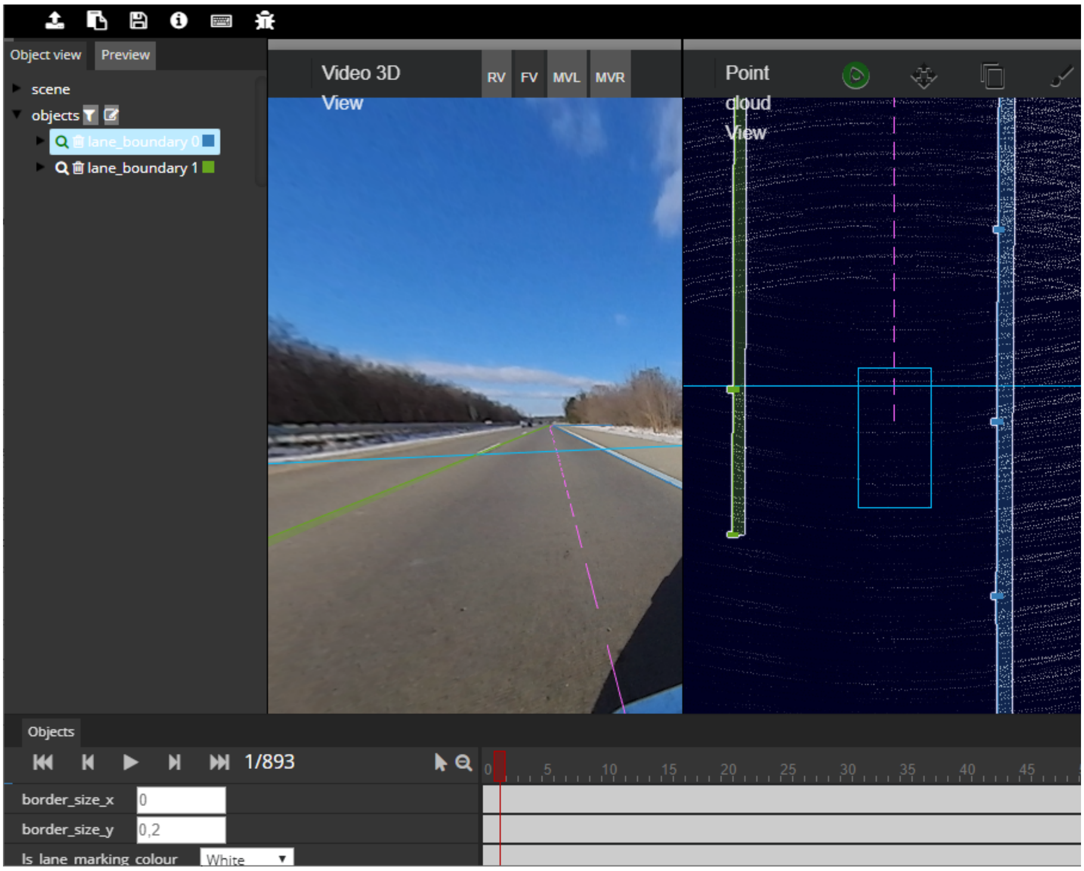
4.1.1 Manual annotation

In a manual annotation task, the human annotator is asked to draw the lane markings by placing points over one edge of each line and defining their width. If the line is solid, the points must be spaced regularly at a fixed distance. If the line is dashed, the points must be placed at the extremes of each segment. This task is performed using a specifically designed application that presents the top-view of the point cloud colored with the intensity information. The back-projection of the annotations to the images is also shown to the human annotator. The annotator creates the points with the mouse by clicking over the point cloud and can delete or correct the created points. The view of the point cloud can be moved and zoomed to increase the precision of annotations. The view can be reset to the default position over the vehicle by pressing a key. With another key, the time of the recording can be advanced to follow the vehicle as it travels the road until the end of the data section to be annotated.

The human annotator is also asked to specify additional information about the lane markings, such as type, color and condition, and information about the scene, such as the weather and the type, condition, and color of the road surface. Fig. 4.2 shows the user interface of the web-based application used for data annotation that includes the camera view as well as the point cloud visualization.

4.2 Outline of the proposed approach

This Thesis reports the work towards a principled model for the automatic annotation of lane markings on highways and non-urban roads. This model does not require pre-annotated data as opposed to supervised machine learning approaches, including the deep learning architectures that are the current fashion in many signal processing tasks. The method works on point clouds with reflectivity and odometry information generated by a mobile multi-layer LIDAR. The odometry is used to accumulate the scans in an overall point cloud defined on a system of coordinates centered at the beginning of recorded travel. Odometry information is also used to process the LIDAR data decomposing it into blocks centered at different points in the trajectory of the vehicle. At each block, candidate lane marking points are detected by generating virtual scan-lines and applying a dynamically optimized filter function to the LIDAR intensity values. The lane markings are tracked block wise, annotating the centerline as a 3D polyline and their width is estimated and classified as either solid or dashed. Fig. 4.3 presents an overview of the annotation pipeline.


Object view
Preview
scene
objects
lane_boundary 0
lane_boundary 1
Video 3D View
RV
FV
MVL
MVR
Point cloud View
Objects
1/893
0
5
10
15
20
25
30
35
40
45

border_size_x 0

border_size_y 0,2

Is lane marking colour White

Figure 4.2: The user interface of the web-based application used for the annotation of lane markings.

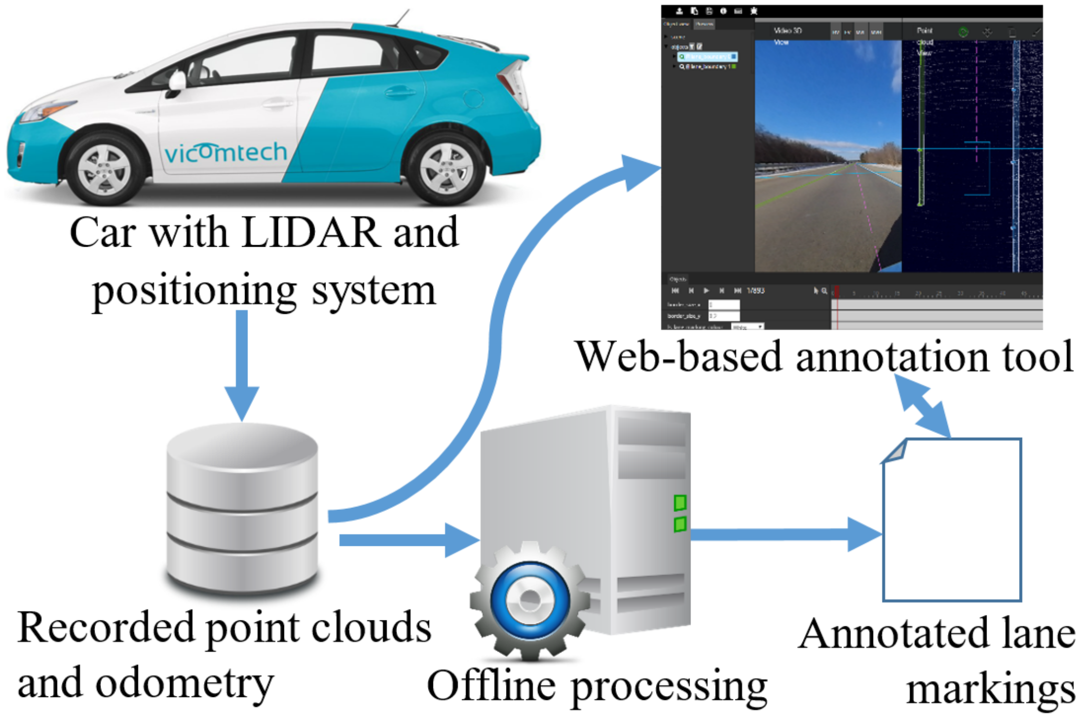

vicomtech
Car with LIDAR and positioning system
Web-based annotation tool
Recorded point clouds and odometry
Offline processing
Annotated lane markings

Figure 4.3: Annotation pipeline.

At the end of this pipeline the automatic annotations are manually corrected and validated.

The main contributions of this approach are:

(i) a novel method for the automatic annotation of lane markings. The method is efficient and robust also with low-density point clouds;

(ii) a new method for evaluating the precision of the annotations. To evaluate the benefits of including this method into an annotation pipeline, an experiment was conducted to measure the time required by professional annotators using the automatic pre-annotations.

4.3 Point cloud preprocessing and preparation

Before carrying out the lane markings detection, the LIDAR data must be prepared to leverage the odometry information, reduce the amount of noise and improve the contrast of the lines. Instead of processing each LIDAR scan point cloud individually, it is better to build an accumulated point cloud combining the scans using

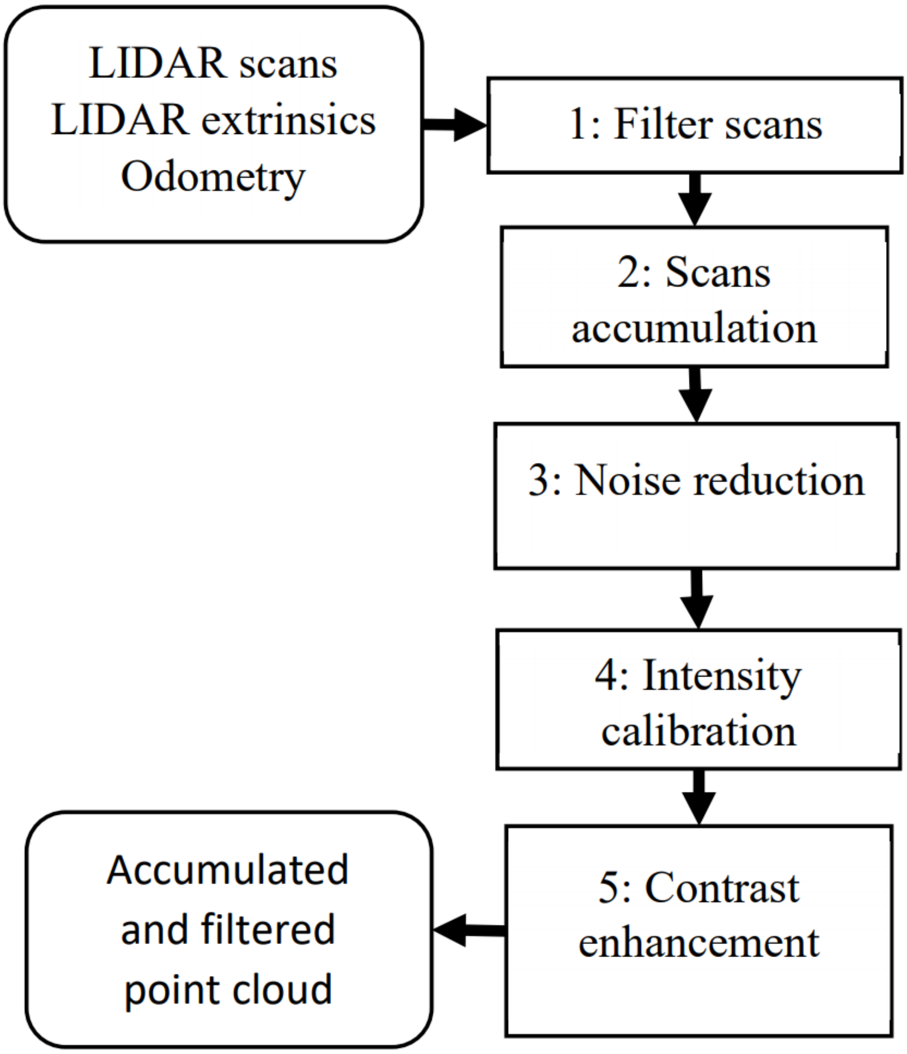

LIDAR scans
LIDAR extrinsics
Odometry
1: Filter scans
2: Scans accumulation
3: Noise reduction
4: Intensity calibration
5: Contrast enhancement
Accumulated and filtered point cloud

Figure 4.4: Point cloud preprocessing steps

the odometry. In this section, the steps required to prepare this point cloud are described. Fig. 4.4 shows the preprocessing steps that are applied to generate the accumulated and filtered point cloud.

4.3.1 Filtering scan points by distance and height

The first step consists in decreasing the number of points to be processed during the detection and reduce the amount of noise present in the point clouds. The uncertainty of the measure increases with the distance to the sensor. For each scan, distant points must be removed because they may contain too much error, so that if they are included in the accumulated point cloud, they would introduce more noise than useful information. Another motivation to remove distant point is that any small error in the calibrated extrinsic parameters of the sensor respect to the vehicle, will introduce big errors in the accumulation point cloud for distant points. Each scan is also filtered by height to remove most of the non-road points. To filter by height, first, the points must be transformed to the coordinates system of the vehicle applying the extrinsic parameters of the sensor. With this transformation, the points are correctly leveled, correctly aligned with the horizontal plane and with the zero height at the road surface.

4.3.2 Scans accumulation

Once the scans are filtered, they are accumulated using the odometry information to increase the points density. The odometry of the vehicle is provided as a list of transformation matrices, each composed of a rotation and a translation. To transform each scan to its accumulated position, the points are transformed to the coordinate system of the vehicle multiplying by the extrinsic transformation matrix of the sensor, and to the global coordinates system multiplying by the odometry matrix of the vehicle. A spinning LIDAR senses each point sequentially, meaning that each point has its own timestamp. The vehicle motion during the completion of the scan or 360 ° revolution must be considered. To estimate the correct global position of each point, the odometry of the vehicle must be interpolated at its precise timestamp. Ideally, the odometry information should have a much higher frequency than the LIDAR to reduce errors due to the timestamp interpolation. Fig. 4.5 shows the accumulated point clouds without and with filtering applied.

4.3.3 Noise reduction

Although the amount of noise is reduced with the filters applied to the individual scans, the point cloud still contains noise due to errors in the calibrated extrinsic parameters of the sensor or sensors. The data analyzed in this research, contains alignment errors in the vertical axis which are mostly as shown in Fig. 4.6. Longitudinal and transversal alignment errors are also present, but these are more difficult

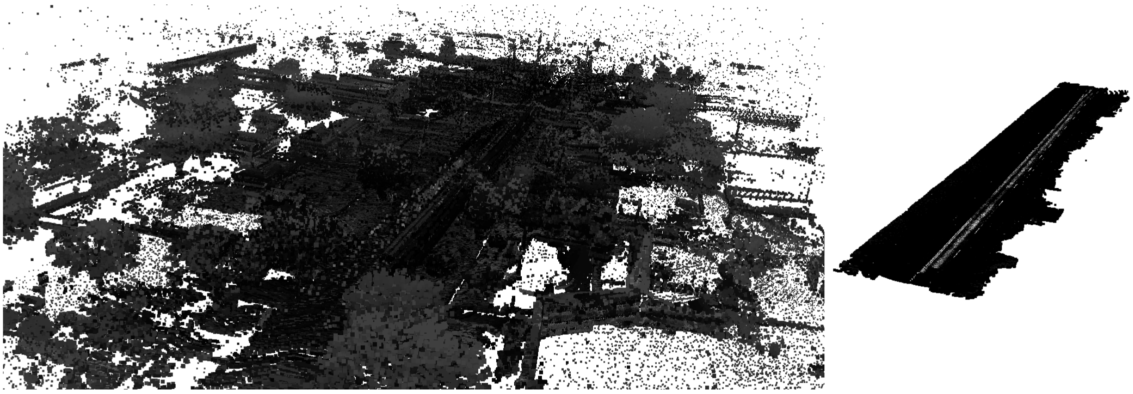


Figure 4.5: Accumulated point cloud before (left) and after (right) filtering

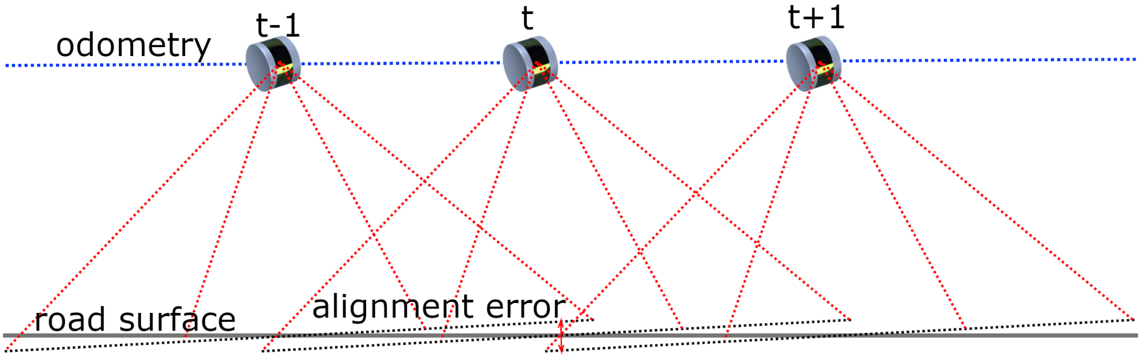


Figure 4.6: Alignment error introduced by incorrect extrinsic calibration

to fix due to the low density of the point cloud and the lack of abundant points in vertical structures, such as walls, on the non-urban environment. Section 5.2 gives details about the analyzed data. The alignment error in the vertical axis produces that the road surface points do not compose a perfect plane but a very noisy cloud.

To further reduce the noise in the vertical axis, the point cloud is processed by blocks extracted along the driving trajectory, with the same algorithm that will be explained in Section 4.4.1. To preserve the longitudinal and transversal slopes of the road surface, the accumulated point cloud is cropped into longitudinally short blocks. Each block is transformed back to the vehicle coordinates system, multiplying by the inverse transformation of the odometry, then the vertical coordinate of the points is set to zero, so that they are flattened on the road surface. Finally, the blocks are transformed to its global position. This way, the resultant point cloud is flat, but preserves the slopes of the road surface. Fig. 4.7 shows the result of the noise reduction.

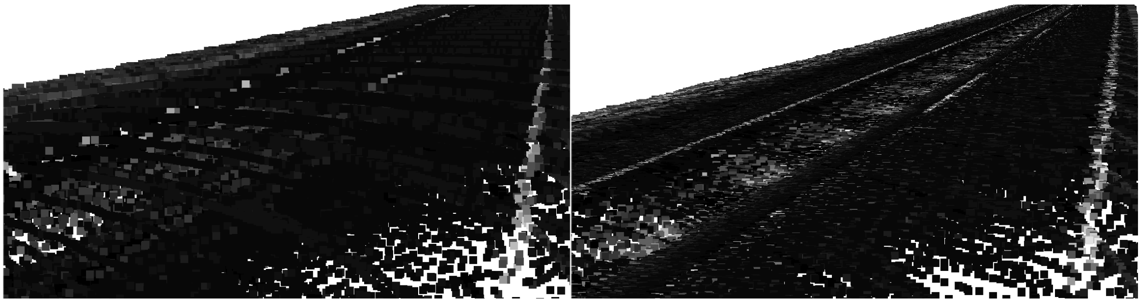


Figure 4.7: Point cloud before and after noise reduction

4.3.4 Intensity calibration

The intensity measured by the different lasers that compose the LIDAR sensor must be calibrated to compensate for the different sensitivity of the lasers. This calibration reduces the amount of noise of the histogram of intensities and improves the contrast of the lane markings. The implementation is based on the method proposed by Levinson and Thrun [11].

The calibration consists in the estimation of a compensation factor for each laser to reduce its differences with the other lasers. This compensation factor must be estimated for each intensity value measured by the lasers, because their response functions are nonlinear. The result of the calibration is a look-up table that contains the compensated intensity value for each intensity observed by each laser beam. Usually, LIDARs measure intensity as an integer value between 0 and 255, therefore for a LIDAR with 32 lasers, the calibrated look-up table has a size of $32*256$ entries. If the data was recorded with multiple LIDARs simultaneously, the calibration can be performed jointly, generating a look-up table for each LIDAR.

To estimate the compensation factor for each laser j, for each of the points measured by each laser, the neighbor points measured by the other lasers must be found. To accelerate this process, the points are projected to a 2D grid map that stores the intensity values associated with each laser identifier, eliminating the cost of finding the neighbor points. The calibrated intensity $c(j, a)$ for laser j is the conditional expectation (conditional mean) of all the intensities measured by the other lasers in the same cells where laser j observed an intensity a. The intensity values of the look-up table that are not observed can be interpolated.

The size of the grid cells is an important parameter for the result of the calibration. It must be configured considering the density of the point cloud. Ideally, this size should be very small to ensure that each cell contains points with similar real intensity only. However, if the size of the grid cells is too small, there would be a lot of empty cells or cells with a single point, while the calibration requires at least two points per cell. On the other hand, if the size is too big, the cells would contain distant points that do not belong to the same physical surface point, such as white points from a line marking and dark points from the surrounding asphalt. Therefore, the calibration would not be accurate, reducing the overall contrast of the resulting point cloud, which will directly impact the later detection of lane markings.

4.3.4.1 Angle of incidence

The intensities measured by the lasers depend on the angle of incidence with the road surface. If the LIDAR is perfectly leveled with the ground plane, the angle of incidence of each laser is constant, because they draw cones that intersect the ground as circles. In this case, there is no need to use the angle of incidence for the calibration. However, when the LIDAR is tilted, for example to concentrate more lasers on the road surface, the angle of incidence of the lasers changes with the spinning angle of the sensor. Therefore, a laser can measure different intensities for the same surface. In this case, the angle of incidence must be considered to obtain a correct calibration. Fig. 4.8 shows a representation of how the angle of incidence changes with the orientation of the LIDAR.

The angle of incidence can be calculated for each point by knowing the position of the LIDAR at the time of the pulse firing. It is calculated as the angle between the road surface normal vector and the vector defined by the measured 3D point and the position of the LIDAR. The normal vector of the surface can be approximated with the vertical axis of the vehicle defined by the known odometry, assuming a planar road and that the vehicle is leveled with the ground. Under these assumptions, the calculation of the angle of incidence can be completely avoided using the spinning angle of the sensor as a proxy, because there is a direct and constant relation between them.

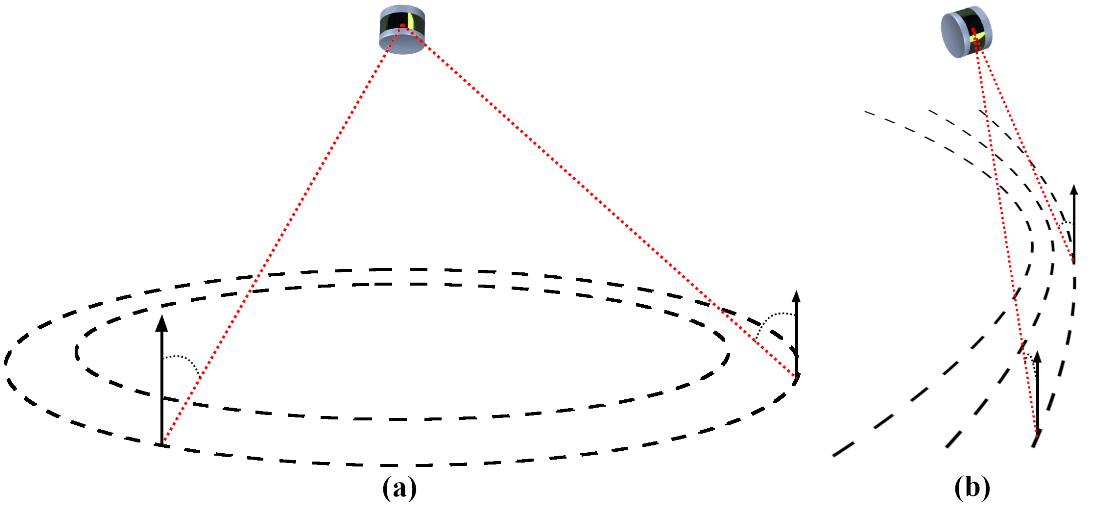


Figure 4.8: Angle of incidence with LIDAR without (a) and with tilt (b)

The angle of incidence is incorporated to the calibration as another dimension of the look-up table, by quantizing the values into bins of one degree. This way, the calibrated intensity $c(j, a, \alpha)$ for laser j is the conditional expectation of all intensities measured, by the other lasers or laser j with a different angle, where laser j measured an intensity a with an angle α. Following the previous example, for a LIDAR with 32 lasers, that measures intensity as an integer value between 0 and 255 and quantizing the angle into 360 bins, the calibrated look-up table has a size of $32 * 256 * 360$ entries.

4.3.5 Lane markings contrast enhancement

After the intensity calibration, the contrast of the lane markings can be further improved knowing that their average intensity is brighter than the rest of the road surface and the points corresponding to the lane markings are much less abundant. The histogram h of intensities and the corresponding cumulative distribution function cdf are calculated as:

$$h\left(i\right) = \frac{n_i}{n}, 0 \leq i < 256, \tag{4.1}$$

$$cdf\left(i\right) = \sum_{j=0}^{i} h\left(j\right), \tag{4.2}$$

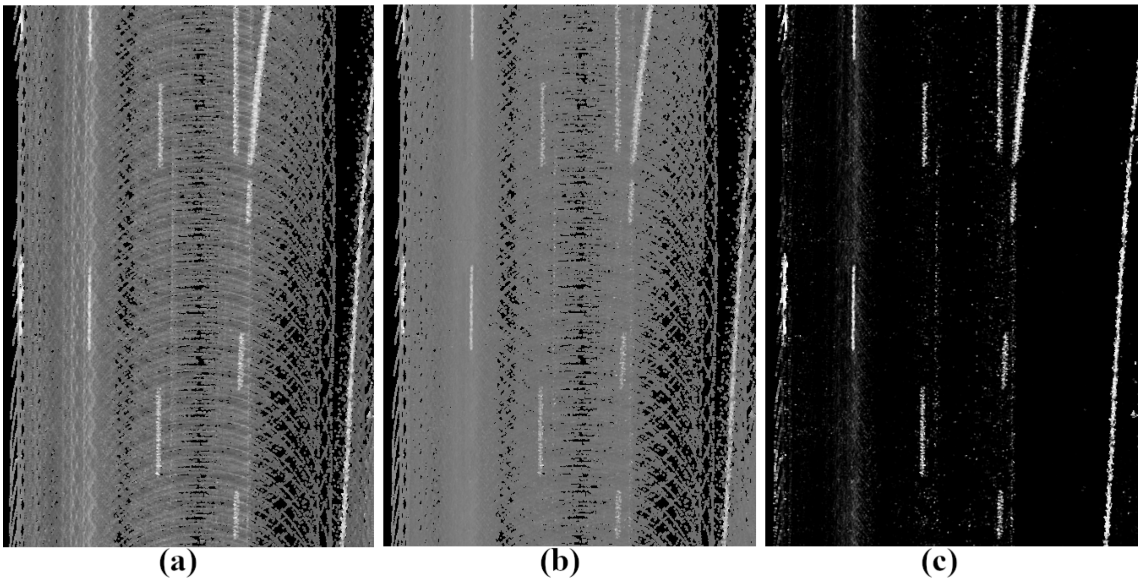


Figure 4.9: Point cloud before (a) and after (b) intensity calibration, and after (c) contrast enhancement

where n_i is the number of points with intensity i and n is the total number of points.

The intensities below a threshold are clamped to zero to remove most of the road points. This threshold is calculated as a combination of the intensity value whose cdf is greater than the 90% of the total mass of the histogram, and the intensity value at which the *cdf* has the steepest slope. The remaining part of the histogram is equalized applying equation 4.3, to increase the contrast between the lane markings and the rest of the road surface.

$$h\left(i\right) = round\left(\frac{cdf\left(i\right) - cdf_{min}}{n - cdf_{min}} * 254\right) + 1 \tag{4.3}$$

Fig. 4.9 shows the result of the intensity calibration and the contrast enhancement steps.

4.4 Lane markings detection

This Section describes the computational pipeline of the proposed method for the detection of the lane markings using the previously accumulated and preprocessed point cloud. An overview of the proposed computational pipeline is presented in Fig. 4.10. The accumulated point cloud composed of the registered scans is progressively

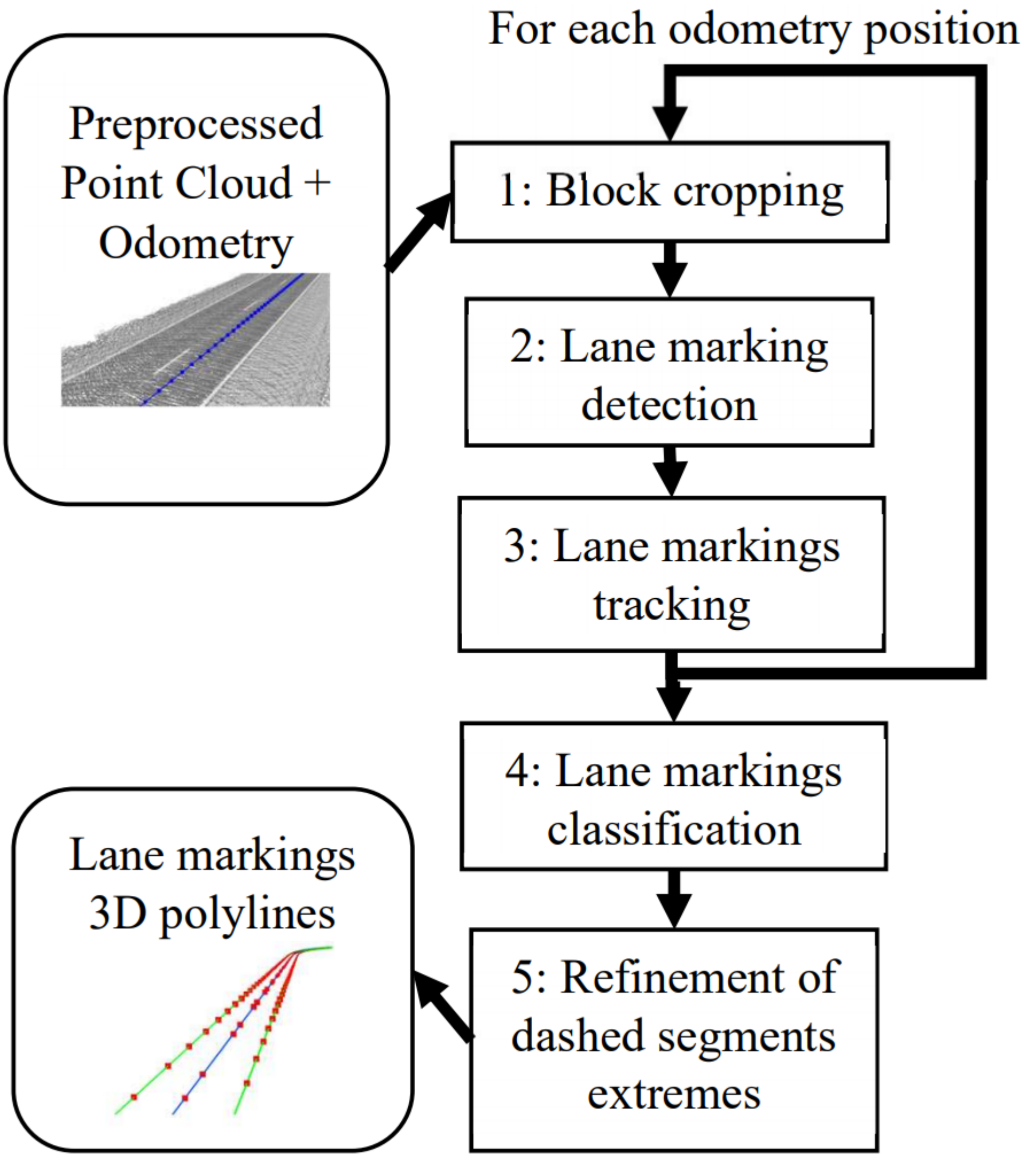

For each odometry position
Preprocessed Point Cloud + Odometry
1: Block cropping
2: Lane marking detection
3: Lane markings tracking
4: Lane markings classification
5: Refinement of dashed segments extremes
Lane markings 3D polylines

Figure 4.10: Computational pipeline of the method.

processed using a decomposition of the accumulated point cloud into blocks following the trajectory of the vehicle. At each block, candidate lane marking points are detected and matched with those found in previous blocks. Once the trajectory is completely processed, lane markings are classified as solid or dashed. In the last step, the position of the extremes of the dashed segments is refined.

4.4.1 Block cropping

First, the trajectory defined by the odometry is discretized into nodes that are separated by a fixed distance blo_d. At each node, a block of the point cloud is

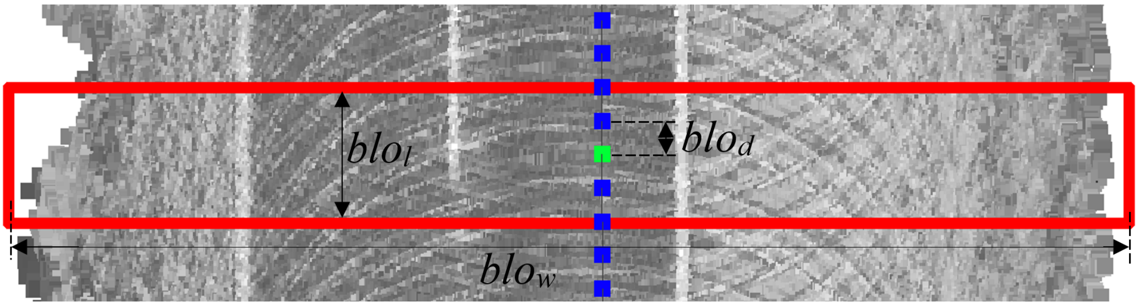


Figure 4.11: Point cloud block extraction with a cropping box (red) centered at the position of the vehicle (green) and oriented perpendicularly to its travelling direction.

extracted using a cropping box of fixed dimensions, centered at the position of the vehicle and oriented perpendicularly to the travelling direction, as shown in Fig. 4.11. The cropping box is blo_l long and blo_w wide.

The cropping process is computationally expensive if it is repeated over the entire accumulated point cloud. To make it more efficient, only a small subset of the scans is used. This subset is updated as the vehicle traverses the point cloud. The scans used are those of which the bounding box intersects the cropping box. As the scans are ordered by the scanning time, the subset can be updated very efficiently. The intersection is calculated using 2D axis-aligned bounding boxes.

The cropped points are transformed, applying the inverse transformation matrix of the sub-sampled odometry, to the vehicle coordinate system. In this reference system, the position of the vehicle is at (0,0,0), the XY plane is aligned with the road surface and the traveling direction is aligned to the Y-axis as shown in the upper part of Fig. 4.12. In this approach it is assumed that the vehicle is traveling approximately parallel to the centerline of the road, which is true most of the times, even when the vehicle is executing a lane change maneuver (lateral displacement is typically much smaller than longitudinal translation).

4.4.2 Candidate lane marking points detection

An average intensity profile is computed from the extracted point cloud block. This process can be regarded as virtual scan-line generation that leverages the accumulation of scans produced by a multi-layer LIDAR. The horizontal axis is divided into bins with a fixed width bin_w. The total number of bins is calculated as $nb = blo_w/bin_w$. The average intensity at each bin b_i is calculated using Inverse distance weighting as:

$$b_i = \frac{\sum_{j=1}^{np} w_{ij} i_j}{\sum_{j=1}^{np} w_{ij}} \tag{4.4}$$

$$w_{ij} = \begin{cases} 1 - (d_{ij}/bin_d) & d_{ij} < bin_d \\ 0 & otherwise \end{cases} \tag{4.5}$$

where i_j is the intensity of each contributing point p_j and d_{ij} is the distance between the point p_j and the center of the bin b_i. Each point contributes to the bins of which the central position is at a maximum horizontal distance bin_d. The total number of points of the block is np.

To remove the influence of variations in the ground intensity, a strongly smoothed version of the projection is subtracted. Smoothing is achieved applying a windowed median filter bin_m. Furthermore, after removal of the ground intensity, the result is smoothed with a Gaussian smooth filter bin_g for noise removal, and negative values are thresholded as illustrated in the lower part of Fig. 4.12.

Candidate lane marking points are detected as local maxima whose intensity value is greater than a dynamic threshold, which is set calculated with the mean and standard deviation of the data. A nonlinear windowed optimization mechanism is proposed, which automatically adapts the top-hat filter to the lane marking shape, including its width, accounting for variations from road to road and different types of lane markings. Compared to previous approaches, which select a fixed expected width, or compute heuristics to find the parameters [60], our approach is guaranteed to converge to the best fit using a flat-top Gaussian function:

$$f(x) = p_0 e^{-((x-p_1)/p_2)^4} \tag{4.6}$$

The fitting process uses the Levenberg-Marquardt algorithm to find the parameters that minimize the difference between the function $f(x)$ and the local environment of the maximum:

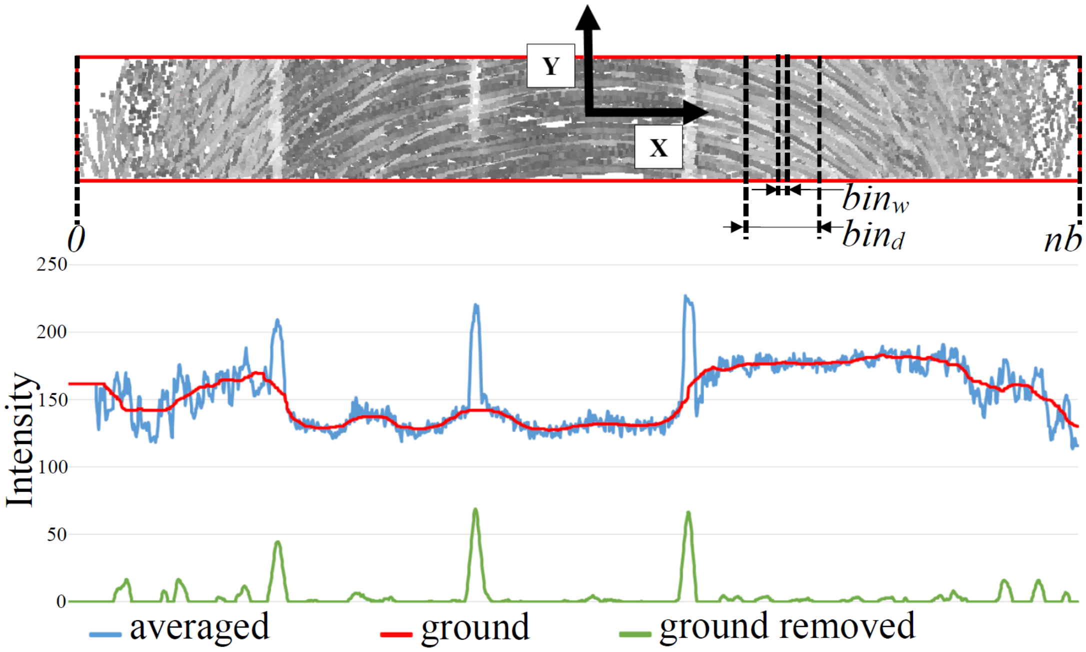


Figure 4.12: Extracted point cloud block corrected after transformation with the odometry data and vertical projection of the average intensity.

- p_0: Controls the height of the function. The resultant value should be like the average intensity value at the found maximum.

- p_1: Shifts the position of the maximum horizontally. The resultant value should not shift the position more than half of the estimated width.

- p_2: Controls the width of the function to fit narrower or wider markings. The width of the marking is calculated as $2 * p_1$ and it should be between mar_{mw} and mar_{Mw}.

The detected candidate lane markings points are converted to 3D points. The X-coordinate is equal to the position of the corresponding maximum in the vertical projection and the Y and Z coordinates are equal to zero. After applying the odometry transformation, the points are aligned perpendicularly to the travelling direction and their vertical position is defined by the translation and rotation of the odometry that contain the actual height of the road at that point and its transversal slope.

4.4.3 Tracking detected lane markings

In the previous step some candidate points have been detected. In this step, they must be connected with the previously detected lane markings or create a new one if it is detected for the first time. Associated with each previously detected line, a Kalman filter [61] is maintained to predict its X-coordinate or lateral position. Each point is connected with the closest point of a previously detected lane marking if these criteria are met:

- If the previous marking has only one point:
    - Distance between the current point and the point of the marking is smaller than tra_l.
    - The estimated line width at both points is similar.
- If the previous marking has more than one point:
    - Distance between the current point and the last point of the marking is smaller than the maximum distance between dashed segments seg_{Md}. The current point could be the first point of a new segment of a dashed line.
    - Distance between the X-coordinate of the current point and the predicted one of the line is smaller than tra_t. The prediction allows to connect dashed segments in curves.

In both cases the angle between the direction of the line that connects the current and previous points, and the travelling direction of the vehicle must be smaller than tra_a. This prevents incorrect connections assuming that the lane markings are always almost parallel to the travelling direction. Some margin is allowed to track lines of exit an entrance lanes.

Each candidate point that has not been connected is discarded as a false positive detection or creates a new marking detection if is farther than the minimum distance between markings mar_d to all the previously detected markings. Fig. 4.13 shows a graphical representation of the involved parameters.

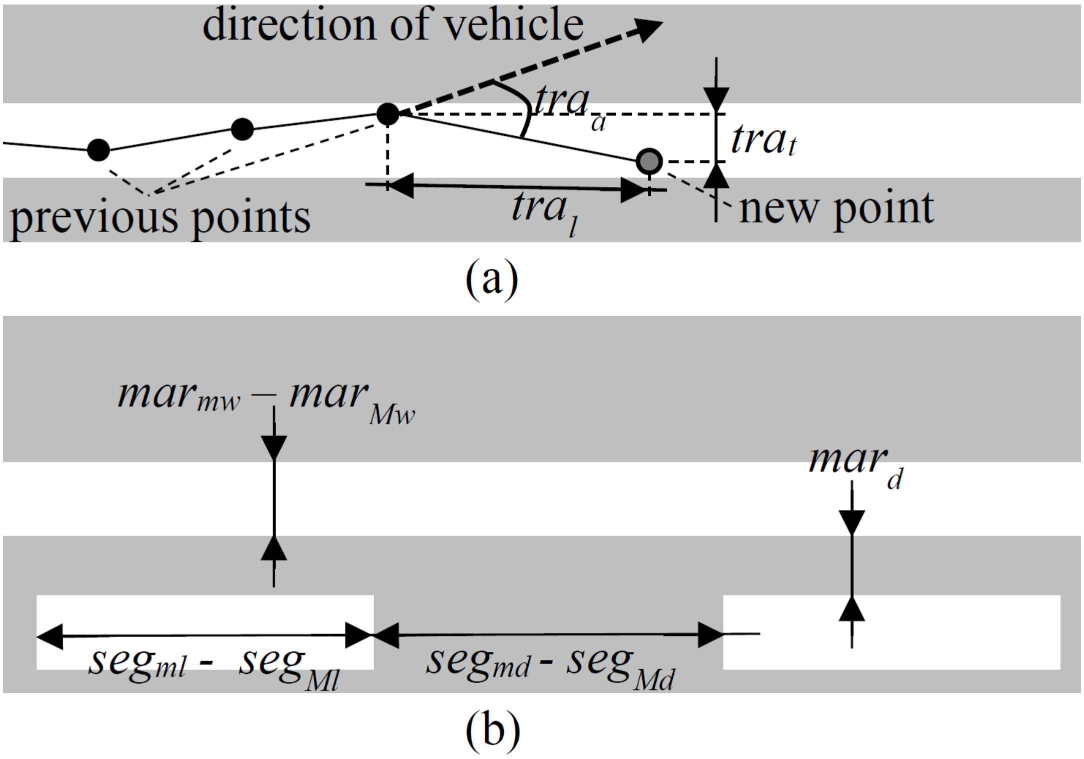


Figure 4.13: (a) Tracking and (b) road definition parameters.

At the end of this step, the Kalman filter of the lines that have received a new point is updated and it is initialized for the new lines.

4.4.4 Lane markings classification

The cropping, detection and tracking steps are repeated at each block until the last step of the odometry. Finally, each lane marking is represented as a polyline or a list of 3D points in the coordinate system of the accumulated point cloud and its estimated average width.

In this step, the detected markings are classified as either solid or dashed lines. The points of each line are grouped based on a minimum seg_{md} and maximum seg_{Md} expected distance between segments and a minimum seg_{ml} and maximum seg_{Ml} segment length. A section of a line is classified as dashed when it presents a series of segments of similar length that are approximately equally spaced and is classified as continuous otherwise. Finally, the intermediate points inside each segment of dashed lines are removed. The result is shown in Fig. 4.15.

4.4.5 Refinement of dashed segment extremes

The detected extreme points of the dashed segments present a longitudinal error because the virtual scan-line approximates the real intensity of the road above the central line of the cropped point cloud block. This error must be reduced to obtain precise annotation of the dashed segments.

For each segment, the point cloud is cropped with a cropping box whose width doubles the estimated width of the line and the length of the detected segment plus a margin seg_r before and after the extremes. The vertical projection of the cropped points is calculated and smoothed with a median filter. The projection is calculated using the same method used for the detection of lane markings, with nb being the number of bins. The precise longitudinal positions of the extremes, represented by n and m, are found in the projection as the positions that maximize equation 4.7, which combines the first derivatives of the projection and the difference between the average intensities inside and outside the extremes:

$$g(n, m) = 2(b'_n - b'_m) + d(n, m), \tag{4.7}$$

where b'_n and b'_m are the first derivatives of the projection at n and m. $d(n, m)$ is the difference between the average intensities of the projection inside and outside the extremes n and m:

$$d(n, m) = \frac{\sum_{i=n}^{m} b_i}{m - n + 1} - \left(\frac{\sum_{j=0}^{n-1} b_j + \sum_{j=m+1}^{nb} b_j}{nb - (m - n + 1)} \right), \tag{4.8}$$

where b_i and b_j are the intensity of each bin inside and outside the extremes. Fig. 4.14 shows an example projection of a dashed segment. With this step the longitudinal error of the segment extremes is significantly reduced concluding the detection of the lane markings.

4.4.6 Annotations output

The result of the detection method is a set of polylines classified as either solid or dashed, as shown in Fig. 4.15. Note that, as mentioned before, the dashed lines

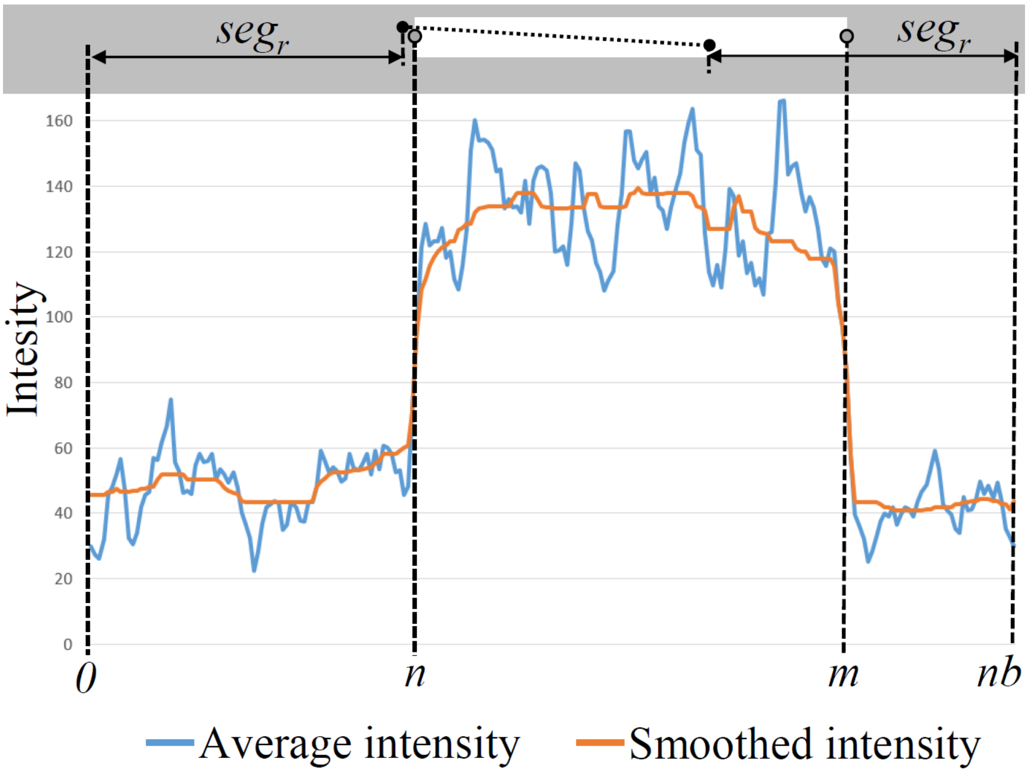


Figure 4.14: Vertical projection of a point cloud block cropped around a dashed segment.

have points at the extremes of the segments only. The detected polylines are written to a JSON text file, following the Video Content Description (VCD) annotation format [62]. VCD is being developed at Vicomtech and is presented as an open-source metadata structure for labeling 2D and 3D objects, pixel-wise segmentation, actions, events, contexts, semantic relations, odometry, and calibration.

Specifically, the lane markings are formatted as static objects, containing the list of 3D points, the estimated width, and type (solid or dashed). The generated VCD file can be loaded in the annotation tool to serve as pre-annotations to be refined by a human annotator. Once the annotator finishes the task, the annotation tool updates the VCD file that can be finally used as ground-truth information for training and testing.

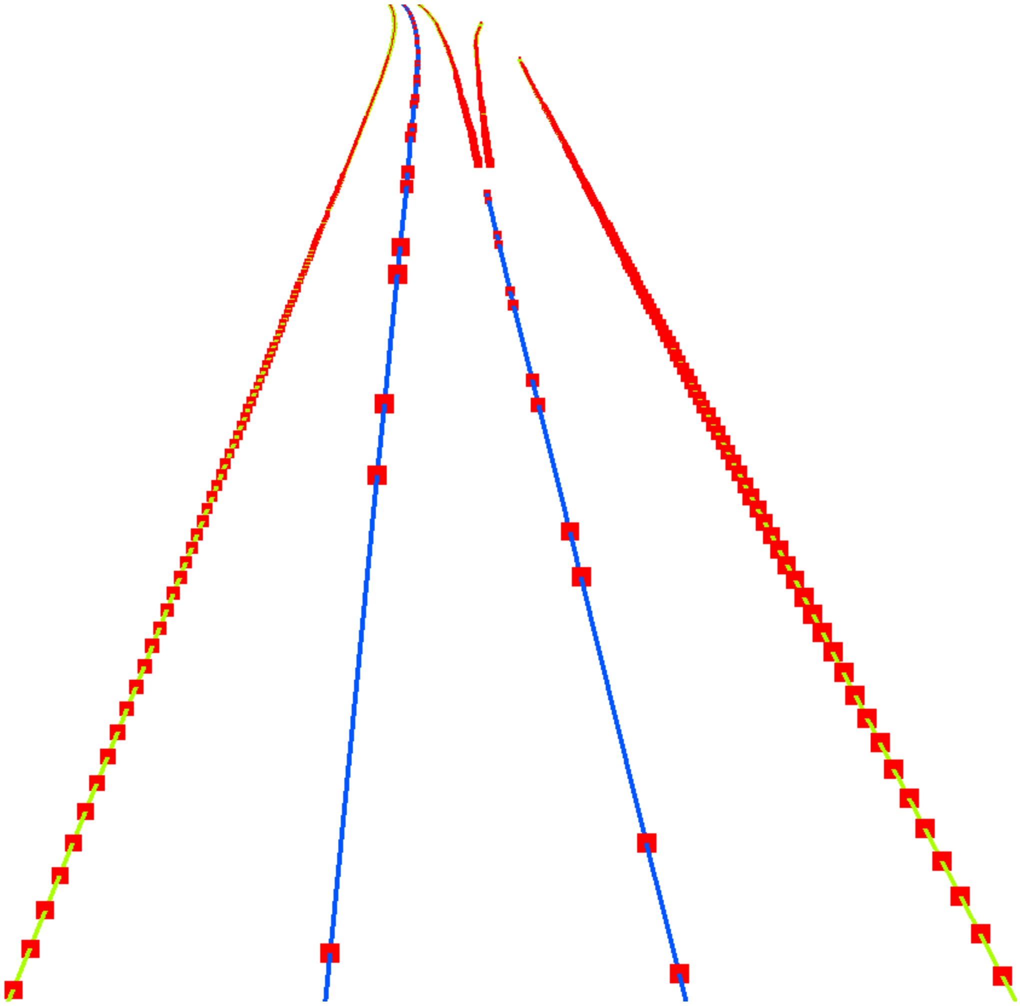


Figure 4.15: Detected lane markings points (red) and polylines classified as either solid (green) or dashed (blue).

Chapter 5

Evaluation

This chapter presents the evaluation of the lane markings detection method. Section 5.1 describes the followed system evaluation methodology. Section 5.2 describes the dataset used for the evaluation. Finally, Section 5.3 presents the results achieved by the proposed system.

5.1 Evaluation methodology

The method presented in Chapter 4 was evaluated against the ground truth generated by manual annotation of the lane markings using a web-based tool [6]. This tool allows interactive annotation of the markings in a top-view of the point cloud. The users (human annotators) are asked to annotate the extremes of the dashed segments of dashed lines and the solid lines by placing points at approximately every 2 m. The annotators also classified the markings as either dashed or solid and estimated the line width. This task can be carried out with high accuracy and efficiently owing to the graphical display of the annotation tool, which can be seen in Fig. 5.1.

One of the key difficulties associated with measuring the quality of human annotations is that the 3D polylines to be compared do not necessarily contain the same number of segments, neither is the number of points per polyline equal or even sampled with the same criteria. To overcome these difficulties, this Thesis proposes an approach that compares 3D polylines after sampling them with a certain metric

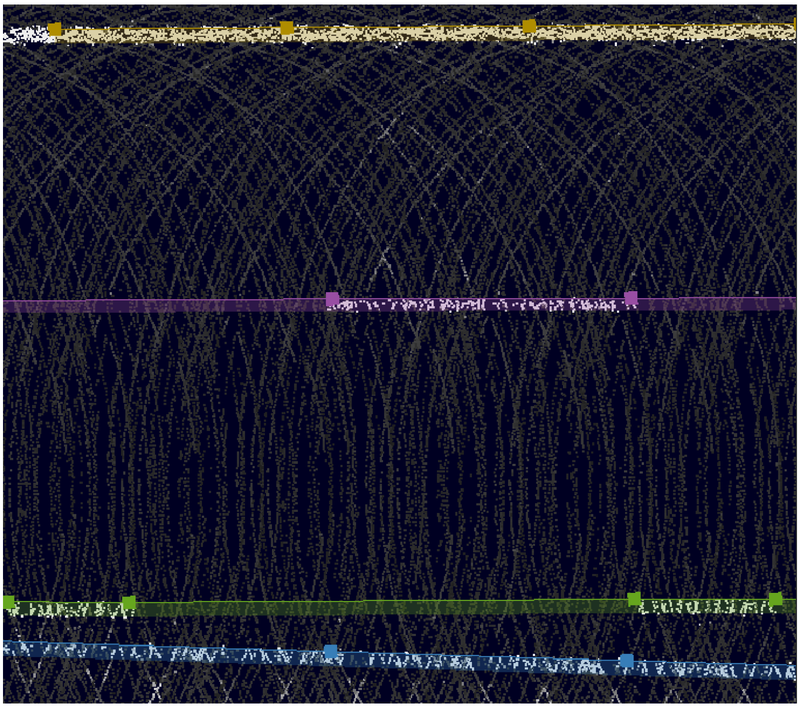


Figure 5.1: Instance of the visualization of manual annotation of lane markings used as ground truth for training and evaluation.

step sam_d, to convert sets of 3D polylines into sets of 3D points. This comparison can be achieved by using 3D Euclidean distances, ensuring that the maximum error in the distance measurement is half the discretization step.

After sampling the polylines, a set of ground truth 3D points, G, and a set of predicted 3D points under evaluation, E, are obtained. For each point $g \in G$, its Euclidean distance to all other points $e \in E$ can be calculated, keeping the minimum distance $d^*(g) = \min(d(g, E))$. This distance $d^*(g)$ approximates the transversal distance from the point g to the closest detected line.

The accuracy of the mapping of ground truth points G to predicted points E is computed as follows: a point $\boldsymbol{g}$ is considered as a True Positive TP if $d^*(\boldsymbol{g}) \leq \varepsilon$, being ε a certain tolerance value. Otherwise, it is a False Negative. Iterating over all $\boldsymbol{g} \in G$, the total number of TPs and FNs is computed. Consequently, FP can be approximated by $FP = |E| - TP$, where $|E|$ is the cardinality of E. This approximation is correct if the defined sampling step is sufficiently small compared to the actual distance between points in the 3D polyline. In our demonstration experiments a sampling step of 1 cm is enough to ensure that the approximation is correct.

Once TP, FN, and FP are computed, Recall R and Precision P curves against the selected tolerance value are calculated. Additionally, the F-measure F can be also computed to provide a single-value metric. These performance measures are especially suitable for LKA and LDW systems, as the results can be evaluated given a required maximum lateral error ε.

$$R = TP/(TP + FN); \tag{5.1}$$

$$P = TP/(TP + FP); \tag{5.2}$$

$$F = 2(RP)/(R + P) \tag{5.3}$$

These three values range from zero to one. R can be interpreted as the number of ground truth lines that are detected, P measures the number of correctly detected lines given that they are closer than the chosen tolerance to a ground truth line, and F is a global quality measure of the detection algorithm. If the R value is equal to one means that the algorithm detected all ground truth lines, i.e. the number of FNs was zero. If the P value is equal to one means that all detected lines were correct, i.e. none of them was a FP detection. The F metric is equal to one, only when both R and P are equal to one, i.e. the algorithm detected all ground truth lines and without any FP detection.

The main goal of the automated annotation tools is to reduce the time required by human annotators. Therefore, an additional evaluation with professional annotators was performed asking them to annotate the traces without and with the automatic pre-annotations. Fig. 5.2 shows a diagram description of the evaluation process encompassing the quantitative comparison.

5.2 Dataset description

The dataset used for experimentation was recorded with a 64-layer LIDAR and a GNSS/INS system. All the point clouds are previously registered using the odometry information: the motion of the sensor during the scans is therefore compensated

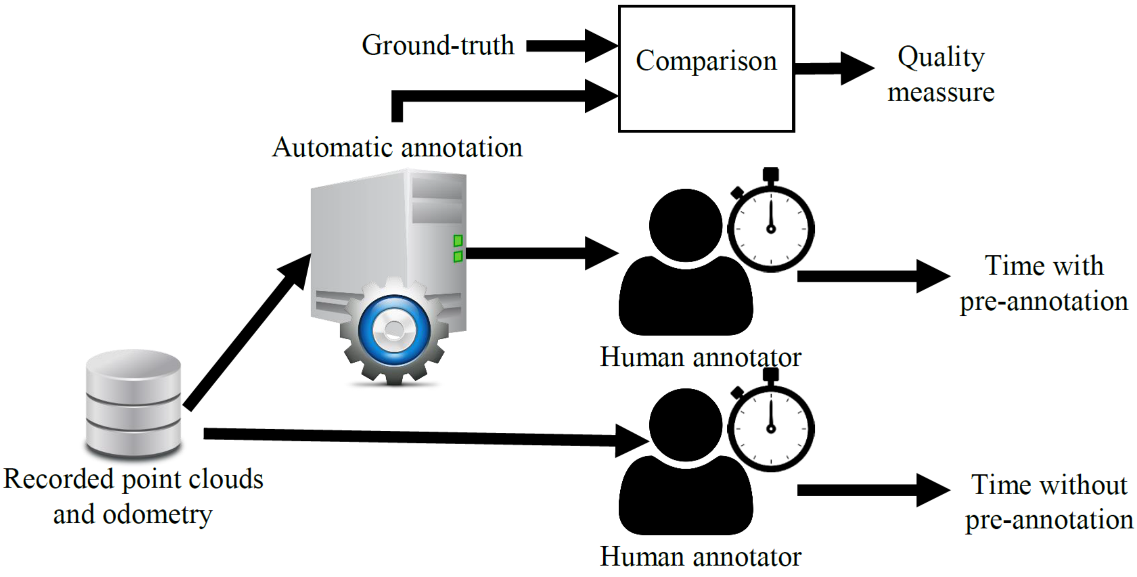

Ground-truth
Comparison
Quality meassure
Automatic annotation
Time with pre-annotation
Human annotator
Recorded point clouds and odometry
Time without pre-annotation
Human annotator

Figure 5.2: Diagram of the evaluation process.

and accumulated to have a rough estimate of the actual position of the vehicle along the path. However, the relationship between each LIDAR point and the corresponding scan ID is maintained and the points are ordered by increasing scan ID. The odometry information contains the position of the vehicle and orientation. Fig. 5.3 shows an example LIDAR trace from the dataset. The blue dots correspond to the sensor locations where the scans where captured, thus the line provides an estimation of the trajectory followed by the vehicle while recording LIDAR and video data.

A total of 10 km was recorded on 12 different days travelling along highways and non-urban roads at driving speeds between 50 and 120 km/h. The LIDAR was installed on the roof of the vehicle to serve as a general 3D sensor for autonomous driving experiments and it was not specifically oriented to concentrate the lasers on the road surface, so only a small fraction of the points of each scan were relevant for our study. Its point-density is in the range of 50 and 1,000 points per m^2 (350 in average) depending on the travelling speed, traffic, and weather conditions. The dataset includes 6 straight and 6 curved roads with up to 4 lanes, and a total of 3 exit and 3 entrance lanes appeared during the recordings. Other road markings such as arrows, triangles, or text signs (such as STOP) are not present. The characteristics of each trace are listed in Table 5.1. Fig. 5.4 shows a frame extracted from video

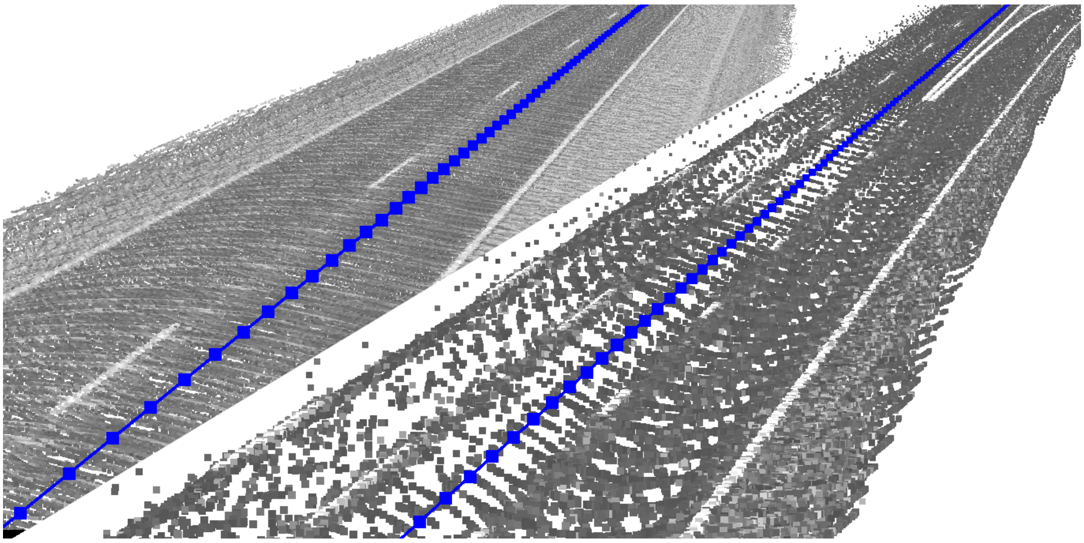


Figure 5.3: Dataset examples: accumulated LIDAR scans and trajectory of the vehicle (blue dot lines).

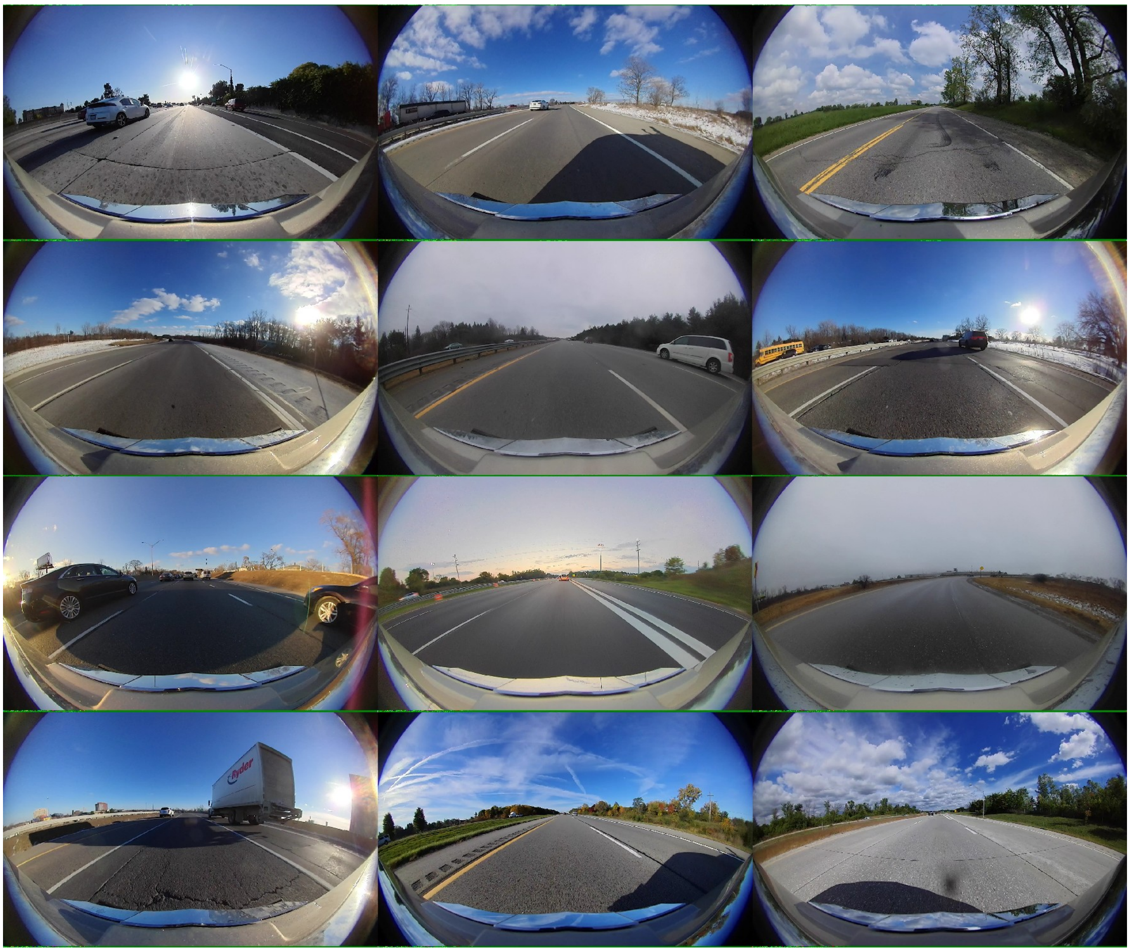


Figure 5.4: A sample frame extracted from of the video recorded along each trace ordered from left to right and top to bottom.

Table 5.1: Traces characteristics

Trace	Weather	Hour	Lanes	Type	Road
1	sunny	18	4	straight	concrete
2	sunny	15	2	slight curve	grey asphalt
3	sunny	12	2	straight	grey asphalt
4	sunny	15	3	slight curve	grey asphalt
5	foggy	11	3	straight	grey asphalt
6	sunny	16	3	slight curve	dark asphalt
7	sunny	17	3	straight	dark asphalt
8	clear evening	19	4	straight	dark asphalt
9	cloudy wet	14	1-3	tight curve	dark asphalt
10	sunny	16	3	slight curve	dark asphalt
11	sunny	9	2	straight	dark asphalt
12	sunny	11	3	slight curve	concrete

captured during each trace by the front-view camera of the vehicle.

The point cloud was previously preprocessed as describe in Section 4.3. The intensity calibration was performed for each trace without the angle of incidence because it was recorded with a vertically oriented LIDAR.

The point clouds were stored in files using the pcd format of the PCL library[1] using a custom point type. The information for each point contains six fields: the 3D coordinates, the intensity, the number of the scan (LIDAR revolution) to which it belongs, and the identifier of the LIDAR that measured it to allow recording with multiple LIDARs. The odometry was recorded in csv files containing the timestamp in microseconds, the 3D position of the vehicle and the rotation as Euler angles. Additionally, the timestamps of the LIDAR scans were recorded in another csv file to be able to interpolate the odometry readings.

5.3 Results

The parameters of the approach and their values used for the quantitative evaluation are listed in Table 5.2. Table 5.3 lists the road definition parameters.

Fig. 5.5(a) shows the global quality F measure for each trace for increasing values of tolerance ε. Fig. 5.5(b) plots the average of F over all traces. Depending

---

[1] https://pointclouds.org/

Table 5.2: Parameters of the proposed approach and their values used for the quantitative evaluation experiments.

Parameter	Value	Description
blo_d	0.5 m	distance between blocks
blo_l	1 m	block length
blo_w	20 m	block width
bin_w	1 cm	bin width
bin_d	4 cm	maximum distance to center of the bin
bin_m	35 cm	median filter width
bin_g	25 cm	gaussian filter width
tra_l	1.5 m	tracking maximum longitudinal distance
tra_t	0.5 m	tracking maximum transversal distance
tra_a	7 degrees	tracking maximum degrees
seg_r	2 m	margin for dashed segments refinement

Table 5.3: Road definition parameters

Parameter	Value	Description
mar_{Mw}	50 cm	lane marking maximum width
mar_{mw}	10 cm	lane marking minimum width
mar_d	15 cm	minimum distance between lane markings
seg_{Md}	15 m	maximum distance between dashed segments
seg_{md}	1 m	minimum distance between dashed segments
seg_{ml}	1 m	minimum length of the dashed segments
seg_{Ml}	9 m	maximum length of the dashed segments

Table 5.4: F-measure obtained with $\varepsilon = 5$ cm at each trace. Also shown their average speed and point-density per square meter.

Trace	R	P	F	Speed (km/h)	Points per m^2
1	0.97	0.96	0.96	84	303
2	1.00	1.00	1.00	109	449
3	0.88	0.89	0.88	51	768
4	0.77	0.80	0.79	105	507
5	1.00	1.00	1.00	124	376
6	0.80	0.80	0.80	113	461
7	0.94	0.94	0.95	47	1033
8	0.97	0.98	0.97	99	40
9	0.69	0.64	0.66	62	49
10	0.99	0.99	0.99	111	320
11	1.00	1.00	1.00	115	367
12	0.99	0.99	0.99	113	375

on the tolerance value, some sections of the detected lines are counted as FPs and the corresponding sections of the ground truth as FNs. With $\varepsilon = 5$ cm an average $F = 0.92$ was obtained. This value means that, on average, 92% of the lane markings were correctly detected with a precision of $\varepsilon = 5$ cm. In other words, if $\varepsilon = 5$ cm is the requested precision for the manual annotators, they will only have to correct 8% of the lane markings identified by the automated annotation system. That F measure cannot be compared with the values reported by other works such as [36], not only because different datasets were used, but because they use an intersection over union metric that does not represent the transversal distance error of the detections.

Table 5.4 lists the values of R, P, and F obtained with $\varepsilon = 5$ cm, and the speed of the vehicle and point-density of each trace. The achieved values of R and P are almost equal because the actual number of FPs and FNs are very low.

The scores obtained for traces 3, 4, 6, and 9 are exceptionally low. Trace 9 was recorded during a rainy day, consequently the point cloud presents many holes and low point-density. On top of that, the road markings were in a poor condition. In traces 3, 4, and 6 the forward and backward LIDAR points are misaligned in the transversal direction because of an inaccurate calibration of the LIDAR extrinsic parametrers that was detected lately. This error has a direct impact on the detection of markings as the average intensity in the projection is halved. This problem is

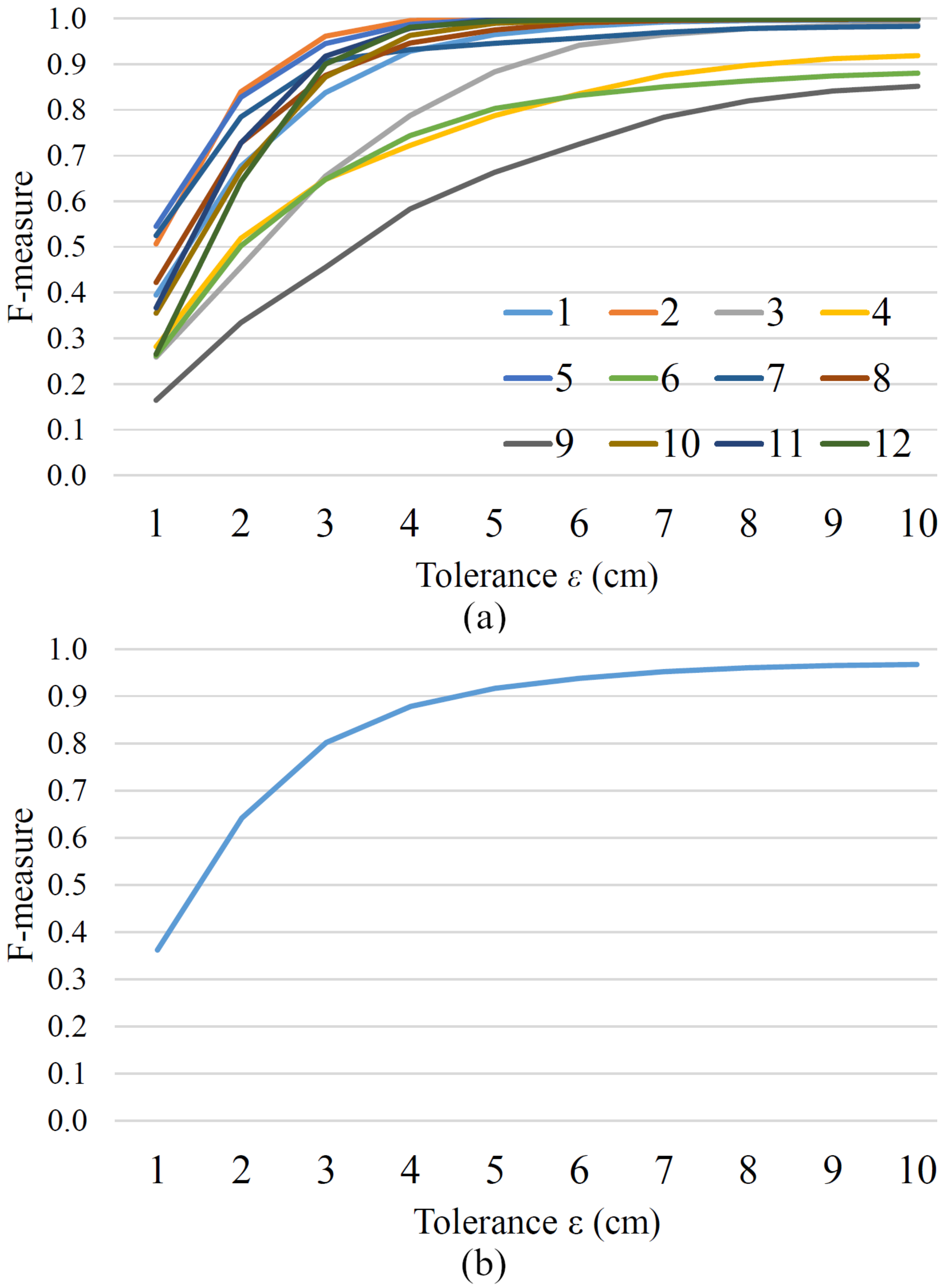


Figure 5.5: (a) F-measure obtained for each trace, and (b) average F-measure over the traces, for increasing tolerance values ε.

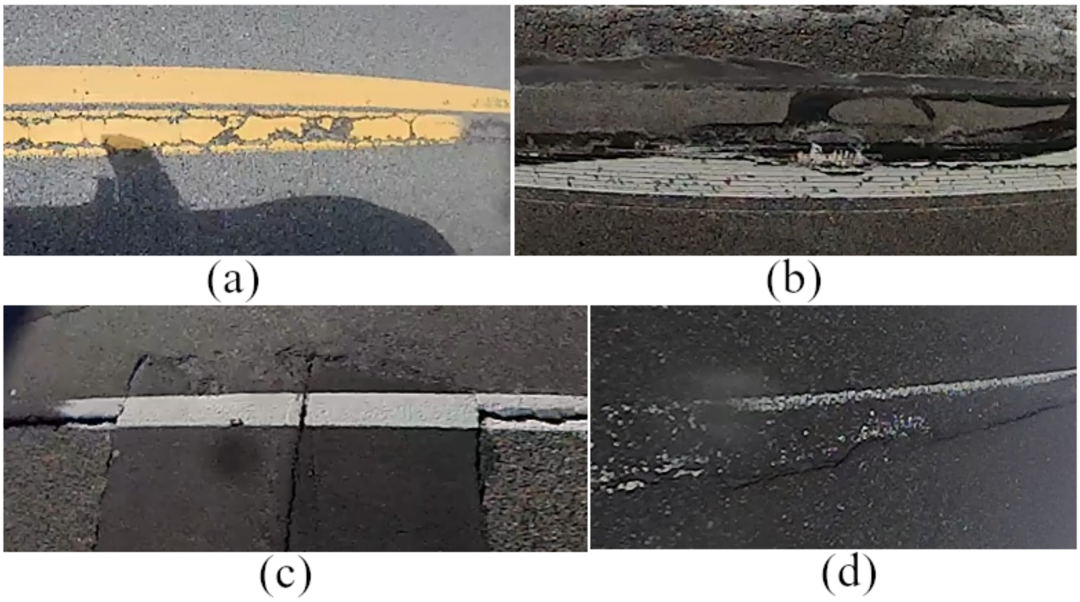


Figure 5.6: Lane markings in poor condition from traces (a) 3, (b) 4, (c) 6, and (d) 9.

even worse in trace 3 with a double solid-dashed line because the gap between the lines does not exist in the point cloud. In addition, the markings present a poor condition. Examples of these markings are shown in Fig. 5.6.

To measure the quality of the detection of extremes of the dashed lines, sampling of the polylines is unnecessary because only the extremes of the segments are detected and annotated. This means that each detected point should have one and only one matching point in the ground truth. In this case, the distance between points includes a transversal and longitudinal error because the line sampling is not applied. Fig. 5.7(a) shows the obtained values of F for dashed segments for each trace for increasing tolerance δ values. Fig. 5.7(b) shows the average F over the traces. Trace 3 is not shown because the dashed line could not be detected due to a LIDAR calibration error. Trace 10 is also not shown because it only contains solid lines. With $\delta = 20$ cm, an average $F = 0.94$ was obtained. Table 5.5 shows the obtained values of R, P, and F with $\delta = 20$ cm for each trace.

The algorithm was implemented in C++ and takes an average of 30 s to process each trace, with an average travelled distance of 800 m and 6 million points per trace, on an Intel i5-4590 3.3 GHz CPU.

Finally, an evaluation experiment with three professional annotators was con-

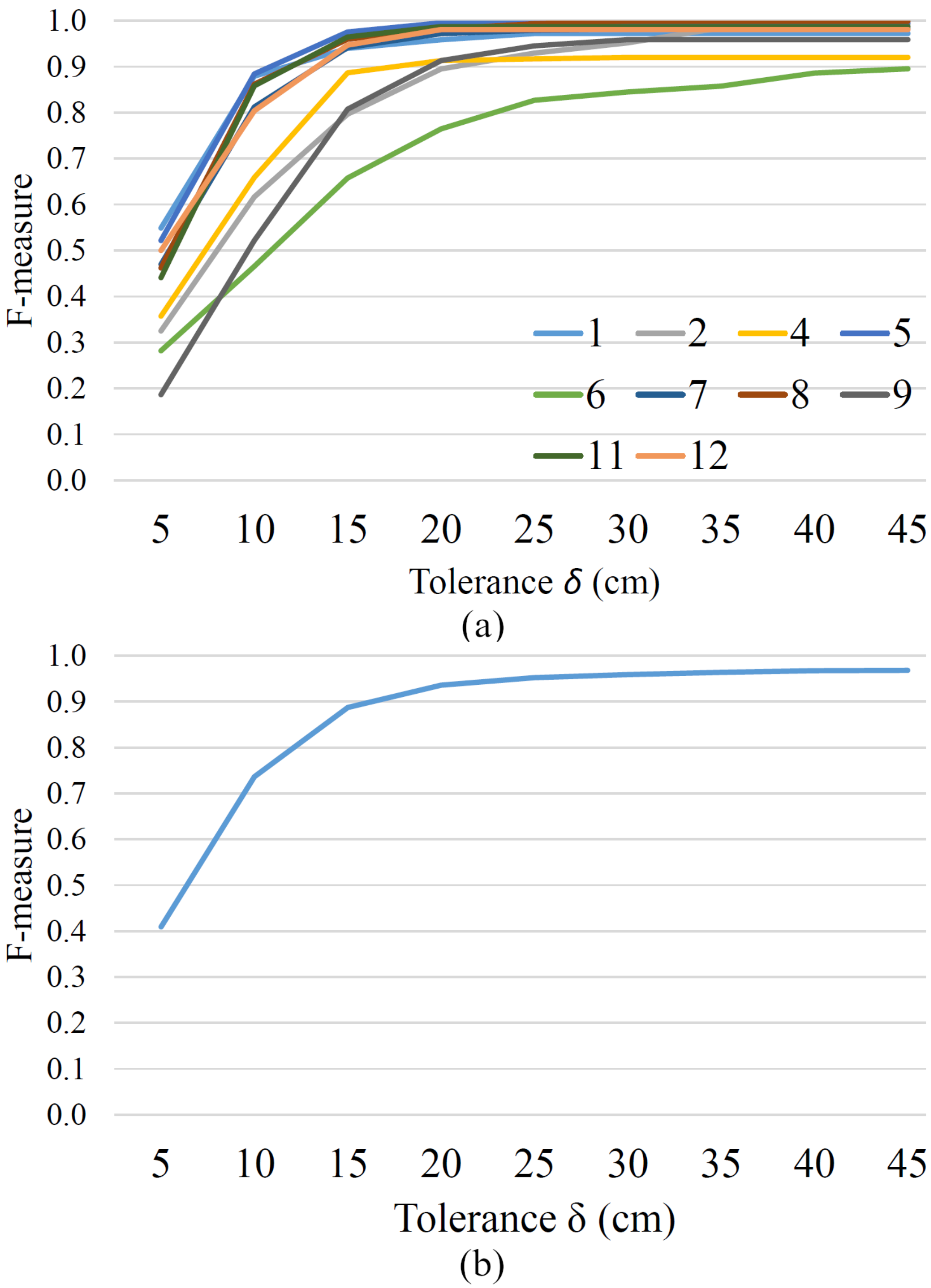


Figure 5.7: (a) F-measure obtained for each trace, and (b) average F-measure for dashed segments while increasing the tolerance δ.

Table 5.5: F-measure obtained for detection of dashed segments with tolerance $\delta = 20$ cm at each trace

Trace	R	P	F
1	0.95	0.95	0.96
2	0.90	0.90	0.90
4	0.85	0.99	0.91
5	1.00	1.00	1.00
6	0.77	0.76	0.76
7	0.97	0.97	0.97
8	0.98	0.98	0.98
9	0.92	0.92	0.91
11	1.00	0.98	0.99
12	0.98	0.98	0.98

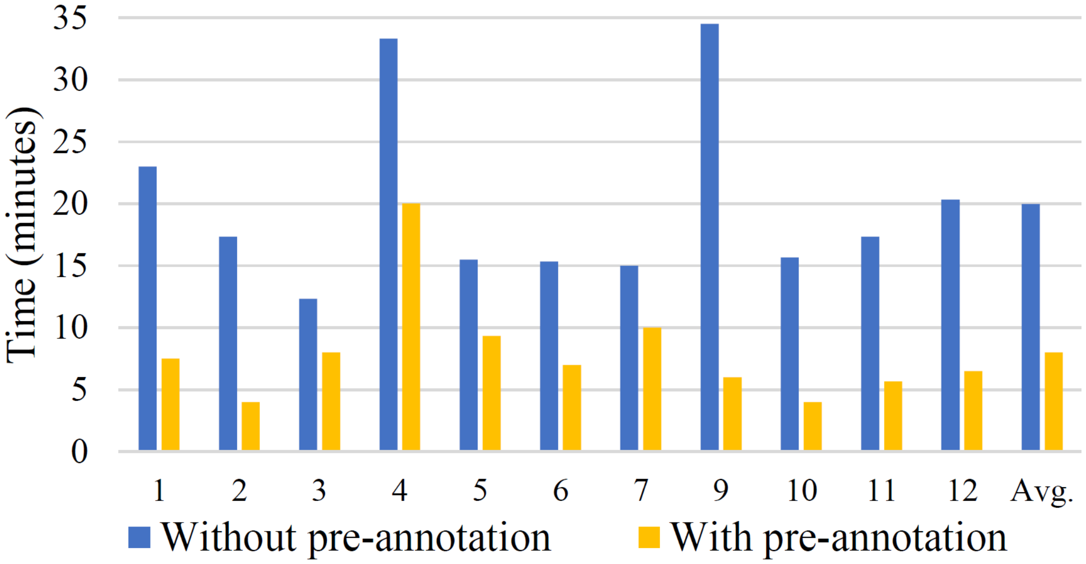


Figure 5.8: Time required to manually annotate each trace without and with automated pre-annotation.

ducted. They were asked to annotate all the traces twice, one starting from zero without pre-annotations and a second one starting with the automated pre-annotations changing the order of presentation of the traces between evaluations to avoid learning bias. Fig. 5.8 shows the average time required by the annotators to annotate each trace. On average, the use of pre-annotation saves 60% of the time required for manual annotation. The average time for annotating each trace was 20 min starting from zero and 8 min when provided with an automatically generated pre-annotation. Using the pre-annotation, the annotators required approximately around 1 min for every 100 m of road.

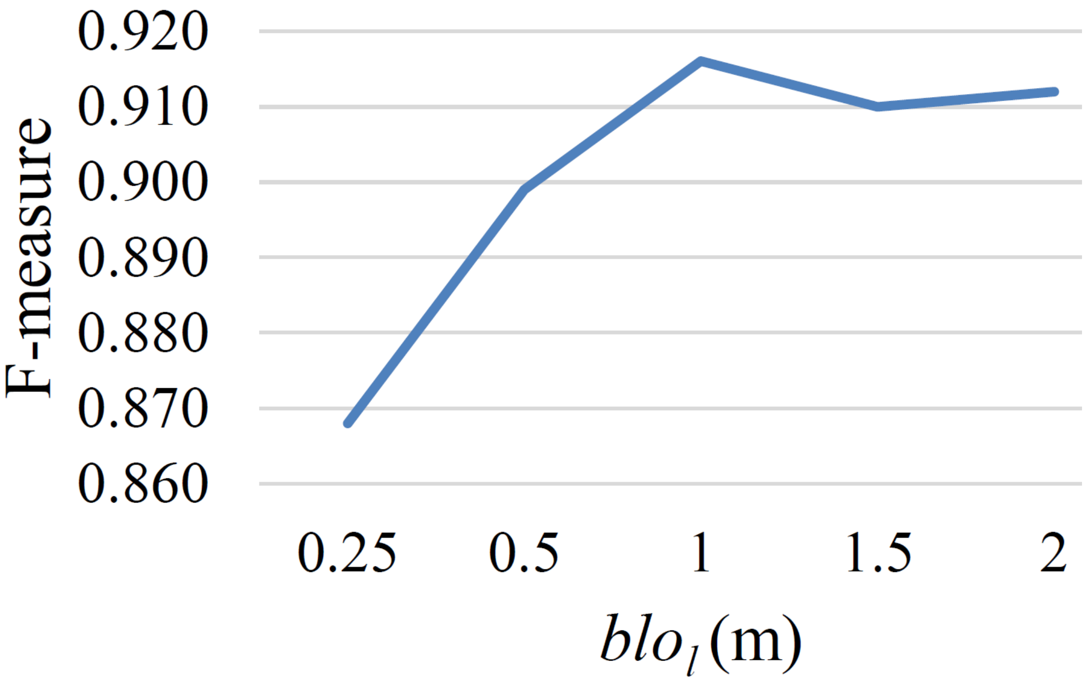


Figure 5.9: F-measure ($\varepsilon = 5$ cm) varying parameter blo_l.

5.3.1 Parameter sensitivity analysis

A sensitivity analysis was carried out to find the best values of the parameters. The following figures show the obtained F varying one of the parameters while keeping the other parameters fixed to the values shown in Table 5.2. Fig. 5.9 shows the F obtained varying the block length blo_l. With $blo_l = 0.25$ m the distance between blocks must be at least $blo_d = 0.25$ m, otherwise there would be unprocessed blocks. Values of blo_l lower than 0.25 m would generate too small blocks with very few points. Values greater than 2 m would generate too big blocks that would produce unreliable virtual scan-lines.

Fig. 5.10 shows the F obtained varying the width of the bins bin_w and the maximum distance bin_d to generate the virtual scan-lines. Fig. 5.11 shows the F obtained varying the size of the filters bin_m and bin_g.

The parameter blo_w was set to the maximum road width in the dataset. The tracking parameters (tra_l, tra_t and tra_a) and the margin for dashed segments refinement (seg_r) have less influence on the results. Changing them resulted in no significative change of the F measure.

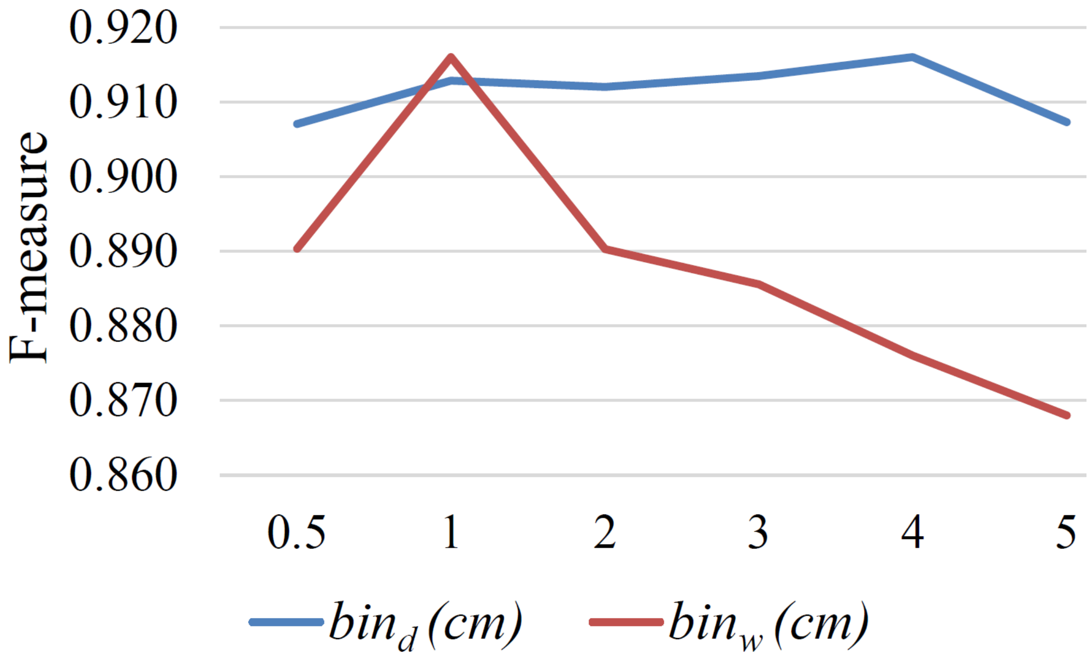


Figure 5.10: F-measure ($\varepsilon = 5$ cm) obtained varying parameters bin_d and bin_w.

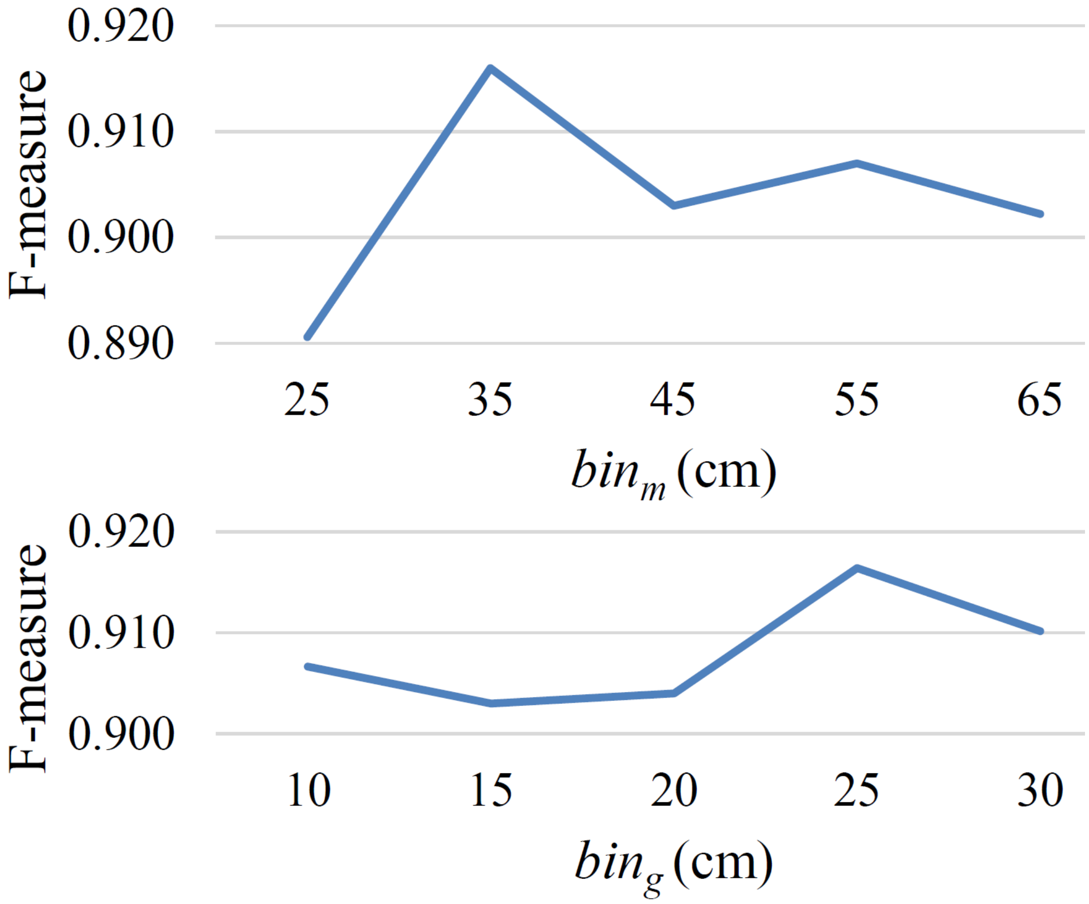


Figure 5.11: F-measure ($\varepsilon = 5$ cm) obtained varying parameters bin_m and bin_g.

Chapter 6

Conclusions and future work

6.1 Conclusions

This thesis has presented a method for the automatic annotation of lane markings using LIDAR data. The approach has been validated on a dataset of LIDAR traces collected by sensorized vehicle on several test trips.

As a summary of the work done:

- We have implemented the collection, preprocessing and preparation of the LIDAR data in order to achieve the test of the proposed solutions.

- We have implemented the web-based annotation system for the assessment of our system by human experts. This system is also being used by other projects at Vicomtech and other companies for commercial exploitation. The tool allows the annotation of video and LIDAR data within a web browser, without installation, using a low-end PC. This setup allows the remote distribution of annotation tasks.

- We have defined and implemented the computational pipeline for the detection of the line markings in the road LIDAR data. This pipeline is founded on fundamental concepts and training is reduced to fitting some of the parameters of the computational processes, which can be done very efficiently in short time for specific sensor systems.

- We have carried out the validation experiments both as automatic annotation predictor and as human annotator assistance.

The validation results showed that 92% of the lines and 94% of the dashed segments were correctly detected with a tolerance of 5 cm and 20 cm, respectively. The automatic annotations produced by the method were evaluated with the help of professional human annotators, showing a reduction of 60% in the time required for manual annotation of large datasets. The processing time of our automated approach is in the order of 4 s per 100 m of travelled road. The results showed that the automatically annotated lane markings are sufficiently precise, requiring very few interventions by manual annotators, who required only one minute to annotate 100 m of road when provided with the automatically generated pre-annotations.

6.2 Future work

The work presented in this thesis can be extended in several ways that will be commented in the following subsections.

6.2.1 Detect other road markings

The presented method is focused on the detection of the lane markings that separate the driving lanes and it could be not suitable to detect other road markings such as arrows, texts, speed signs, etc. To detect those elements has more sense to implement an image-based detector that works with rasterized top-view images of the point cloud. Current approaches based on deep learning architectures have received great attention in the recent years. It is possible that YOLO kind of detectors would be useful for demonstration of concept, though they have been reported as not robust to changing environmental conditions in some independent empirical evaluations.

6.2.2 Detect road edges

Another interesting feature of road infrastructure assessment system is the detection of the road limits. These limits could be defined by concrete barriers, guardrails, curbs, or it could be a direct transition from asphalt to grass or any other type of terrain. These could be detected using the vertical information of the LIDAR to detect guardrails, barriers, or curbs, or measuring the flatness of the surfaces combined with changes in the intensity to detect the end of the asphalt or concrete surface. Techniques of texture analysis of the LIDAR data may be rather useful for the discrimination of the different surfaces in this case. The deep learning approaches that could be more relevant are the semantic segmentation networks, already used for the processing of visual information in assistive driving systems.

6.2.3 Improve LIDAR extrinsic parameters

One of the main difficulties for the lane markings detection was that the given extrinsic calibration of the LIDAR was not perfect. This misalignment produced that the accumulated point cloud presented a big quantity of noise that directly impacts the detections. Some filters were applied to reduce the problem, but the main cause was not faced. A possible solution could be to refine the calibrated extrinsic parameters of the sensor. This could be done using global optimization method that tries to minimize the noise in the point cloud by adjusting the extrinsic parameters. The challenge is how to measure that noise. A possibility is to measure the variance in the vertical coordinate of the points along the point cloud. Another possibility could be to use a point cloud alignment algorithm, such as Iterative Closest Point (ICP), to measure the alignment error between the scans.

6.2.4 Combine LIDAR with cameras

The combination of cameras and LIDARs is a research line that is already obtaining good results for object detection. The annotation of lane markings could also be improved with this combination, benefiting from the greater data density and color information of cameras and the depth and intensity information of LIDARs. To achieve this combination, the calibration between the sensors must be very accurate, because the objects to be detected can be as small as a 5 cm thick line. Similarly to the previous point, this calibration could be improved driven by the lane markings detection, trying to maximize the alignment between camera and LIDAR data.

Bibliography

[1] United States Environmental Protection Agency. Sources of greenhouse gas emissions, 2018.

[2] European Commission. Household consumption by purpose, 2018.

[3] Society of Automotive Engineers (SAE). Levels of driving automation, 2019.

[4] J.D. Ortega, M. Nieto, L. Salgado, and O. Otaegui. User-adaptive eyelid aperture estimation for blink detection in driver monitoring systems. In Karsten Berns, Markus Helfert, and Oleg Gusikhin, editors, *Proceedings of the 6th International Conference on Vehicle Technology and Intelligent Transport Systems, VEHITS 2020, Prague, Czech Republic, May 2-4, 2020*, pages 342–352. SCITEPRESS, 2020.

[5] L. Unzueta, M. Nieto, A. Cortés, J. Barandiarán, O. Otaegui, and P. Sánchez. Adaptive multicue background subtraction for robust vehicle counting and classification. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):527–540, 2012.

[6] A. Mujika, A. Domínguez, I. Tamayo, O. Senderos, J. Barandiarán, N. Aranjuelo, M. Nieto., and O. Otaegui. Web-based video-assisted point cloud annotation for adas validation. In *Web3D The 24th International ACM Conference on 3D Web Technology*, pages 1–9, 2019.

[7] Lirong Liu, Hao Ma, Siyun Chen, Xinming Tang, Junfeng Xie, Gang Huang, and Fan Mo. Image-Translation-Based Road Marking Extraction from Mobile Laser Point Clouds. *IEEE Access*, 8:64297–64309, 2020.

[8] Jiyoung Jung and Sung-Ho Bae. Real-Time Road Lane Detection in Urban Areas Using LiDAR Data. *Electronics*, 7(11):276, oct 2018.

[9] Alberto Y. Hata, Fernando S. Osorio, and Denis F. Wolf. Robust curb detection and vehicle localization in urban environments. *IEEE Intelligent Vehicles Symposium, Proceedings*, (June):1257–1262, 2014.

[10] Shuo Gu, Yigong Zhang, Jinhui Tang, Jian Yang, Jose M. Alvarez, and Hui Kong. Integrating Dense LiDAR-Camera Road Detection Maps by a Multi-Modal CRF Model. *IEEE Transactions on Vehicular Technology*, 68(12):1–1, 2019.

[11] Jesse Levinson and Sebastian Thrun. Robust vehicle localization in urban environments using probabilistic maps. In *IEEE International Conference on Robotics and Automation*, pages 4372–4378, 2010.

[12] Jun Xie, Martin Kiefel, Ming-Ting Sun, and Andreas Geiger. Semantic Instance Annotation of Street Scenes by 3D to 2D Label Transfer. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, volume 38, pages 3688–3697. IEEE, jun 2016.

[13] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 2019.

[14] Andoni Cortés, Clemente Rodríguez, Gorka Vélez, Javier Barandiarán, and Marcos Nieto. Analysis of Classifier Training on Synthetic Data for Cross-Domain Datasets. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–10, 2020.

[15] Javier Barandiarán, Marcos Nieto, Andoni Cortés, Oihana Otaegui, Julián Flórez, and Manuel Graña. Automated Annotation of Lane Markings Using LIDAR and Odometry. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–11, 2020.

[16] World Health Organization. Global status report on road safety, 2018.

[17] Dirección General de Tráfico. Principales cifras de siniestralidad, 2018.

[18] International Energy Agency. Co2 emissions from fuel combustion 2018 highlights, 2018.

[19] World Health Organization. Air pollution, 2019.

[20] European Commission. Shaping the future of mobility, 2019.

[21] American Trucking Associations. Reports, trends & statistics, 2019.

[22] The Washington Post. Nine days on the road. average commute time reached a new record last year., 2018.

[23] Lingfei Ma, Ying Li, Jonathan Li, Cheng Wang, Ruisheng Wang, and Michael A. Chapman. Mobile laser scanned point-clouds for road object detection and extraction: A review. *Remote Sensing*, 10(10):1–33, 2018.

[24] Christopher Rose, Jordan Britt, John Allen, and David Bevly. An integrated vehicle navigation system utilizing lane-detection and lateral position estimation systems in difficult environments for GPS. *IEEE Transactions on Intelligent Transportation Systems*, 15(6):2615–2629, 2014.

[25] I. Puente, H. González-Jorge, J. Martínez-Sánchez, and P. Arias. Review of mobile mapping and surveying technologies. *Measurement: Journal of the International Measurement Confederation*, 46(7):2127–2145, 2013.

[26] Haiyan Guan, Jonathan Li, Yongtao Yu, Zheng Ji, and Cheng Wang. Using Mobile LiDAR Data for Rapidly Updating Road Markings. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2457–2466, 2015.

[27] Alberto Holgado-Barco, Belén Riveiro, Diego González-Aguilera, and Pedro Arias. Automatic Inventory of Road Cross-Sections from Mobile Laser Scanning System. *Computer-Aided Civil and Infrastructure Engineering*, 32(1):3–17, 2017.

[28] Jaehoon Jung, Erzhuo Che, Michael J. Olsen, and Christopher Parrish. Efficient and robust lane marking extraction from mobile lidar point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 147(June 2018):1–18, jan 2019.

[29] Haiyan Guan, Jonathan Li, Yongtao Yu, Cheng Wang, Michael Chapman, and Bisheng Yang. Using mobile laser scanning data for automated extraction of road markings. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87:93–107, 2014.

[30] Xin Chen, Brad Kohlmeyer, Matei Stroila, Narayanan Alwar, Ruisheng Wang, and Jeff Bach. Next generation map making: geo-referenced ground-level LIDAR point clouds for automatic retro-reflective road feature extraction. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09*, number Ladybug 3, page 488, New York, New York, USA, jun 2009. ACM Press.

[31] Bisheng Yang, Lina Fang, Qingquan Li, and Jonathan Li. Automated Extraction of Road Markings from Mobile Lidar Point Clouds. *Photogrammetric Engineering & Remote Sensing*, 78(4):331–338, 2012.

[32] Pankaj Kumar, Conor P. McElhinney, Paul Lewis, and Timothy McCarthy. Automated road markings extraction from mobile laser scanning data. *International Journal of Applied Earth Observation and Geoinformation*, 32:125–137, 2014.

[33] Avdhut Joshi and Michael R. James. Generation of accurate lane-level maps from coarse prior maps and lidar. *IEEE Intelligent Transportation Systems Magazine*, 7(1):19–29, 2015.

[34] Bei He, Rui Ai, Yang Yan, and Xianpeng Lang. Lane marking detection based on convolution neural network from point clouds. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pages 2475–2480, 2016.

[35] Mario Soilán, Belén Riveiro, Joaquín Martínez-Sánchez, and Pedro Arias. Segmentation and classification of road markings using MLS data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 123:94–103, 2017.

[36] Chenglu Wen, Xiaotian Sun, Jonathan Li, Cheng Wang, Yan Guo, and Ayman Habib. A deep learning framework for road marking extraction, classification and completion from mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 147:178–192, 2019.

[37] Li Yan, Hua Liu, Junxiang Tan, Zan Li, Hong Xie, and Changjun Chen. Scan line based road marking extraction from mobile LiDAR point clouds. *Sensors*, 16(6):1–21, 2016.

[38] Ruochen Yin, Biao Yu, Huapeng Wu, Yutao Song, and Runxin Niu. FusionLane: Multi-Sensor Fusion for Lane Marking Semantic Segmentation Using Deep Neural Networks. mar 2020.

[39] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, C V Aug, and Hartwig Adam. deeplabv3+:Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[40] K. Dietmayer, N. Kämpchen, K. Fürstenberg, J. Kibbel, W. Justus, and R. Schulz. Roadway Detection and Lane Detection using Multilayer Laserscanner. *Advanced Microsystems for Automotive Applications 2005*, pages 197–213, 2005.

[41] Sören Kammel and Benjamin Pitzer. Lidar-based lane marker detection and mapping. *IEEE Intelligent Vehicles Symposium*, pages 1137–1142, 2008.

[42] Michael Thuy and Fernando León. Lane Detection and Tracking Based on Lidar Data. *Metrology and Measurement Systems*, 17(3), jan 2010.

[43] Tan Li and Deng Zhidong. A new 3D LIDAR-based lane markings recognition approach. *2013 IEEE International Conference on Robotics and Biomimetics, ROBIO 2013*, (December):2197–2202, 2013.

[44] Belén Riveiro, H. González-Jorge, Joaquín Martínez-Sánchez, L. Díaz-Vilariño, and Pedro Arias. Automatic detection of zebra crossings from mobile LiDAR data. *Optics & Laser Technology*, 70:63–70, jul 2015.

[45] Ming Cheng, Haocheng Zhang, Cheng Wang, and Jonathan Li. Extraction and Classification of Road Markings Using Mobile Laser Scanning Point Clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(3):1182–1196, mar 2017.

[46] Joseph Lam, Kresimir Kusevis, Paul Mrstik, Robin Harrap, and Michael Greenspan. Urban scene extraction from mobile ground based LiDAR data. *Proceedings of 3DPVT*, 2010.

[47] Laurent Smadja, Maison De Technopole, and Leonard De Vinci. Road extraction and environment interpretation from LIDAR sensors. *IAPRS*, 2010.

[48] Luca Caltagirone, Samuel Scheidegger, Lennart Svensson, and Mattias Wahde. Fast LIDAR-based road detection using fully convolutional neural networks. *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 1019–1024, 2017.

[49] Jannik Fritsch, Tobias Kuhnl, and Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. In *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2013.

[50] Jinyong Jeong and Ayoung Kim. LiDAR Intensity Calibration for Road Marking Extraction. *2018 15th International Conference on Ubiquitous Robots, UR 2018*, 1:455–460, 2018.

[51] Jesse Levinson and Sebastian Thrun. Unsupervised Calibration for Multi-beam Lasers. In *Springer Tracts in Advanced Robotics*, volume 79, pages 179–193. 2014.

[52] Juan Castorena. Computational Mapping of the Ground Reflectivity With Laser Scanners. *IEEE Transactions on Image Processing*, 28(9):4288–4298, sep 2019.

[53] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The ApolloScape Open Dataset for Autonomous Driving and Its Application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2702–2719, oct 2020.

[54] Jungwook Lee, Sean Walsh, Ali Harakeh, and Steven L. Waslander. Leveraging Pre-Trained 3D Object Detection Models for Fast Ground Truth Generation. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2504–2510. IEEE, nov 2018.

[55] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:77–85, 2017.

[56] Amol Borkar, Monson Hayes, and Mark T. Smith. A novel lane detection system with efficient ground truth generation. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):365–374, 2012.

[57] Brook Roberts, Sebastian Kaltwang, Sina Samangooei, Mark Pender-Bare, Konstantinos Tertikas, and John Redford. A Dataset for Lane Instance Segmentation in Urban Environments. In *The European Conference on Computer Vision (ECCV)*, 2018.

[58] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, dec 2017.

[59] Sergio Sánchez-Carballido, Orti Senderos, Marcos Nieto, and Oihana Otaegui. Semi-Automatic Cloud-Native Video Annotation for Autonomous Driving. *Applied Sciences*, 10(12):4301, jun 2020.

[60] Marcos Nieto, Lorena Garcia, Orti Scnderos, and Oihana Otaegui. Fast Multi-Lane Detection and Modeling for Embedded Platforms. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 1032–1036. IEEE, sep 2018.

[61] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):32–46, 1960.

[62] Marcos Nieto, Orti Senderos, and Oihana Otaegui. Boosting ai applications: Labeling format for complex datasets. *SoftwareX*, 13:100653, 2021.