Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros Industriales

Reconstrucción del entorno del vehículo para guiado autónomo en escenarios complejos

Reconstruction of the vehicle environment for autonomous guidance in
complex scenarios

Tesis Doctoral

Autor

Miguel Clavijo Jiménez
Ingeniero Industrial

UNIVERSIDAD
POLITÉCNICA
DE MADRID

Junio 2020

Departamento de Ingeniería Mecánica
Escuela Técnica Superior de Ingenieros Industriales

Reconstrucción del entorno del vehículo para guiado autónomo en escenarios complejos

Reconstruction of the vehicle environment for autonomous guidance in
complex scenarios

Tesis Doctoral

Autor

Miguel Clavijo Jiménez
Ingeniero Industrial

Directores

Felipe Jiménez Alonso

Dtpo. Ingeniería Mecánica - ETS INGENIEROS INDUSTRIALES - UPM

Francisco Serradilla García

Dtpo. Inteligencia Artificial - ETS INGENIEROS INFORMÁTICOS - UPM

UNIVERSIDAD
POLITÉCNICA
DE MADRID

Junio 2020

Agradecimientos

Después de tantas páginas escritas, ensayos realizados, líneas código, horas delante del ordenador, etc. parece que se le pone el punto final a un trabajo que exige lo mejor de uno, pero al mismo tiempo produce una gran satisfacción llegar al momento de escribir estas líneas. Personalmente puedo afirmar que realizar la tesis ha supuesto una de las experiencias más enriquecedoras hasta el momento. No solo se trata de afrontar nuevos retos, desarrollar un método y llegar a unas conclusiones. Pienso que es un periodo de tiempo en el que uno desarrolla una actitud diferente frente al trabajo, reeduca la forma de pensar y adopta un nuevo sentido crítico en lo que realiza. Además, me siento muy afortunado de haberla podido llevar a cabo dentro de un gran equipo de personas donde he podido aprender continuamente. He podido realizar proyectos realmente interesantes en los que nunca hubiera imaginado que estaría involucrado. He tenido la oportunidad de realizar una estancia en Alemania y conocer a personas muy interesantes con las que compartir nuevos puntos de vista. Dar ponencias en ciudades que no estarían en mi lista de prioridades y, sin embargo, quedar sorprendido. Por todo ello, me gustaría agradecer a todas aquellas personas que me han ayudado en estos años.

En primer lugar, me gustaría agradecerle a mi director Felipe Jiménez Alonso por todo su tiempo y esfuerzo dedicado. Han sido muchas horas las que hemos pasado debatiendo y dándole forma a esa idea que “tenía que funcionar”. Lo siento por sus alumnos si alguna de esas ideas haya podido inspirar algún problema de examen. Ha sido una persona referente durante toda la tesis, de la que he podido aprender y depositar mi confianza desde el primer momento. Asimismo, me gustaría agradecerle su tiempo y dedicación a mi codirector de tesis Francisco Serradilla García. Gracias a su dedicación en el ámbito de la Inteligencia Artificial, despertó en mí un gran interés en esa línea. Sus planteamientos dieron lugar a investigar en una temática en la que, a día de hoy, seguimos trabajando y profundizando.

Igualmente, un pilar fundamental en estos años han sido mis compañeros de trabajo, sin los cuales, no hubiera sido posible realizar muchas de esas ideas que rondaban en el aire. Me gustaría agradecer especialmente a Óscar todo su tiempo, involucración y apoyo. Han sido muchos días y muchas tardes realizando ensayos, adquiriendo datos, buscando sensores (aplicando un método sofisticado de extracción), diseñando en CAD y largo etcétera de actividades distintas que hemos tenido que afrontar. Ha sido sin duda, una gran ayuda tenerle al lado durante todos estos años y del que he podido aprender (OSTLR). De igual modo, quería destacar el apoyo de Alberto, siempre dispuesto a echar una mano por muy “cargado”

que estuviese, incluso siendo capaz de cruzarse Viena corriendo para que tan solo pudiésemos comer juntos. Me gustaría también agradecer personalmente al resto del equipo, aunque prefiero no nombrarlos para no dejarme a nadie fuera. Asimismo, tener unas palabras de agradecimiento también para el resto de personal del INSIA, en especial a los compañeros de la nave que siempre han estado dispuestos a ayudar y aportar su punto de vista.

De igual modo, agradecer al INSIA por facilitar el uso de sus instalaciones y materiales para llevar a cabo muchos de los ensayos de esta tesis. Agradecer a la Universidad Politécnica de Madrid y en concreto al Programa Propio de I+D+i por promover la Investigación, Desarrollo e Innovación de calidad y excelencia a través de sus becas para realizar el doctorado y ayudas para la realización de estancias en el extranjero, con las cuales ha sido posible completar esta tesis con mención internacional.

En lo personal, me gustaría destacar el incondicional apoyo de toda mi familia, especialmente el de mis padres desde el primer momento. Siempre han demostrado ser los primeros en animarme a afrontar nuevos retos y gracias a ellos se concluye ahora otra etapa. Siempre han servido de inspiración para dar la mejor versión de uno mismo.

Como no podía ser de otra manera, agradecer a Nela toda su ayuda, sobre todo en estas últimas semanas cuando parecía que quedaba todo por hacer. Ha conseguido que vea las cosas con perspectiva y con su “poco a poco” hemos conseguido cerrar otro capítulo más.

Para finalizar, me gustaría agradecer a todos mis amigos que han sabido entender por qué no me apuntaba a “ese plan” o a “aquel otro”. Muchas gracias por todos los ánimos que he recibido por vuestra parte.

Se cierra esta etapa, pero solo porque hay que poner un punto final como se decía al principio. Sin embargo, se dibuja un panorama ideal para seguir creciendo, desarrollando esas ideas y ver hasta dónde nos lleva el avance tan acelerado que la ciencia y la tecnología están protagonizando en los últimos años.

A mis abuelos.

*Science is more than a body of knowledge.
It's a way of thinking.¹*

Carl Sagan.

¹1996, última entrevista de Carl Sagan

Resumen

La tecnología LiDAR 3D cada vez está teniendo más repercusión en un campo emergente de aplicación, el vehículo autónomo. Fabricantes, desarrolladores e investigadores integran este tipo de sensores para aplicaciones de reconocimiento del entorno debido a las ventajas que aporta. Además, los sensores láser escáner han ganado especial protagonismo en la conducción autónoma a partir de la evolución de los láseres 2D a 3D, debido a que estos últimos suplen satisfactoriamente los problemas de observación parcial de los objetos que tenían los sensores 2D. La información espacial que proporcionan los LiDAR 3D es significativamente más completa, lo que otorga de versatilidad para ser utilizados para otros fines.

Esta versatilidad se hace patente en la variedad de algoritmos basados en tecnología LiDAR que se encuentran en la literatura. Si bien, su aplicación principal es el reconocimiento de obstáculos, para garantizar una navegación autónoma terrestre segura, es necesario ampliar el campo de aplicación. En este sentido, esta Tesis Doctoral plantea el estudio de las capacidades de los sensores LiDAR y el desarrollo de algoritmos centrados en dos de las líneas principales de investigación del vehículo autónomo: la caracterización del entorno y el posicionamiento y mapeado. Ambas líneas constituyen las bases de la percepción del vehículo que, a su vez, es la encargada de generar un modelo del entendimiento del entorno. Este entendimiento extraído del entorno a partir de los sensores tiene especial relevancia en la seguridad durante la navegación cuando es utilizado en capas posteriores, como la *Planificación de trayectorias* o la *Toma de decisiones*, dentro de la estructura del vehículo autónomo.

Por ello, la Tesis aborda por un lado la problemática de la caracterización del entorno, aportando soluciones respecto a la identificación de obstáculos, no solo centrándose en entornos urbanos, sino también en entornos *off-road*, donde la orografía es irregular y el entorno desestructurado. Asimismo, debido a la versatilidad de estos sensores, se consigue llevar a cabo desarrollos para la caracterización horizontal de la vía, estableciendo los límites de ésta (ya sean escenarios urbanos, interurbanos o con terreno no preparado) o extrayendo las líneas viales, obteniendo información relevante acerca del ancho de carril y el error lateral en él o el número total de carriles disponibles.

Sobre la misma caracterización del entorno, se pueden establecer ciertos criterios que apoyen y mejoren el posicionamiento durante la navegación autónoma. De este modo, se han implementado métodos analíticos que, a través de las herramientas de caracterización desarrolladas, puedan apoyar una navegación autónoma o reconstruir el mapa por donde se circula.

Uno de los aspectos más relevantes de esta Tesis es la aplicación de algoritmos de *Machine Learning* a problemas concretos de la conducción autónoma. A partir del desarrollo de estas técnicas, concretamente en el campo del *Deep Learning*, junto a la gran cantidad de información que proporcionan los sensores LiDAR 3D, es posible diseñar modelos con gran rendimiento y precisión. En este sentido, ya han demostrado su validez posicionándose en el estado del arte respecto a problemas de segmentación semántica, pero aún existe un gran desarrollo por delante ante problemas de regresión como en el caso de lograr un posicionamiento preciso. Por ello, se ha planteado y desarrollado algoritmos basados en redes de convolución que inferen la odometría del vehículo. A partir de estos modelos es posible obtener los valores de velocidad y variación del ángulo de guiñada entre dos observaciones consecutivas, lo que proporciona una consistencia local del posicionamiento.

Además, esta estimación de la odometría puede servir para asistir a métodos convencionales cuando estos encuentran dificultades a la hora de converger a un resultado esperado. Teniendo en cuenta que el problema de posicionamiento global está condicionado analíticamente, las soluciones basadas en métodos geométricos pueden alcanzar resultados precisos. Sin embargo, en función de la información disponible del entorno, se puede originar un peor rendimiento o precisión. Los modelos basados en técnicas de *Deep Learning*, al abordar el problema del posicionamiento con otro enfoque, no son tan sensibles a variaciones en la información del entorno y, de esta manera, se pueden emplear como asistencia en métodos estrictamente geométricos y conseguir una consistencia global del posicionamiento.

Los resultados obtenidos en cada desarrollo son contrastados y evaluados en diferentes situaciones reales a través de vehículos autónomos o instrumentados. Asimismo, la adquisición de datos y creación de *datasets* para la implementación de los algoritmos, se ha realizado utilizando diferentes sensores como el Velodyne VLP-16 y el Ouster OS-1 64, en ensayos en entornos controlados, urbanos o todoterreno.

Palabras clave: LiDAR, percepción del entorno, odometría, posicionamiento, mapeado, *Deep learning*.

Abstract

LiDAR technology is increasingly having an impact in an emerging field of application: the autonomous vehicle. Manufacturers, developers and researchers integrate such sensors in applications used for recognition of the environment due to their numerous advantages. Furthermore, the laser scanner sensors have gained special prominence in autonomous driving due to the evolution of lasers from 2D to 3D, since the latter have fixed the problems of partial objects observation that 2D sensors had. In fact, the spatial information provided by LiDAR 3D is significantly more complete, which gives them a versatile nature and enables them to be used for other purposes.

This versatility is evident from the variety of algorithms based on technology LiDAR that can be found in the literature. Although its main field of application is the obstacles detection, in order to guarantee safe autonomous ground navigation, it is of utmost importance to expand such field of application. In this regard, the present Doctoral Thesis focuses the study of the capabilities of LiDAR sensors and the development of algorithms focused on the two main areas of research of the autonomous vehicle: the characterization of the environment and positioning and mapping. Both areas constitute the basis of the perception of the vehicle which is also capable of generating a model for the understanding of the surrounding environment. Such understanding of the environment developed from the sensors has special relevance for a safe navigation when it is used in subsequent layers within the structure of the autonomous vehicle, such as "Path planning" or "Decision making".

On the one hand, this Thesis deals with the problem of the characterization of the environment in order to provide solutions for obstacles identification, focusing not only on urban environments but also on off-roads areas, where the ground is uneven and the environment unstructured. On the other hand, given the versatile of these sensors, the Thesis achieves developments for the horizontal characterization of the road, estimating its limits (either in urban scenarios, interurban or not-prepared ground) or extracting road lane lines, obtaining relevant information about the lane width and the lateral error in it or the total number of lanes available.

Based on the same characterization of the environment, it is possible to establish certain criteria to support and enhance the positioning for an autonomous navigation. In this regard, analytical methods have been implemented based on the characterization tools already developed to support autonomous navigation or reconstruct the map where the vehicle drives.

One of the most relevant aspects of this Thesis is the application of Machine Learning algorithms to specific problems of the autonomous driving. Due to the development of these techniques, specifically in the field of Deep Learning, together with the large amount of information provided by the 3D LiDAR sensors, it is possible to design models with great performance and precision. In this respect, they have already proven their performance by positioning themselves in the state of the art regarding semantic segmentation problems; however, there is still room for improvement in terms of regression problems, such as the case of achieving a precise positioning. For such purpose, it is proposed and developed algorithms based on convolutional neural networks that infer the odometry of the vehicle. Based on these models, it is possible to obtain the speed value and yaw angle rate between two consecutive observations, providing a local consistency positioning.

Furthermore, this odometry estimation can be used to assist traditional methods when they encounter difficulties in converging to an expected result. Considering that the global positioning problem is analytically conditioned, the solutions based on geometric methods can achieve precise results. Nevertheless, depending on the information available from the environment, a worse performance or precision may originate. Given the fact that models based on techniques Deep Learning address the problem of positioning with another approach, they are not excessively sensitive to variations of the environment information and, therefore, can be used as assistance in methods strictly geometric, in order to achieve a global consistency positioning.

The results derived from each development are contrasted and evaluated in different real situations through autonomous or instrumented vehicles. Likewise, the data acquisition and creation of datasets for the implementation of the algorithms, have been carried out using different sensors, such as the Velodyne VLP-16 and the Ouster OS-1 64 in controlled urban or off-road environments.

Key words: LiDAR, environment perception, odometry, positioning, mapping, Deep learning.

Índice general

1. Introducción	1
1.1. Marco general de la tesis	1
1.2. Percepción del entorno basada en sensores LiDAR 3D	2
1.2.1. Ventajas e inconvenientes de los sensores LiDAR 3D frente a otros sistemas de percepción	3
1.2.2. Versatilidad de los sensores LiDAR para la percepción del entorno . .	4
1.3. Limitaciones del estado actual	5
1.4. Objetivos	7
1.5. Estructura de la Tesis	8
2. Herramientas para la caracterización del entorno	11
2.1. Calibración extrínseca de sensores LiDAR	11
2.1.1. Alternativas para calibración de sistemas de percepción	12
2.1.2. Método de calibración extrínseca automática para sensores LiDAR . .	14
2.1.2.1. Consideraciones previas	14
2.1.2.2. Descripción del algoritmo	14
2.1.3. Análisis de los límites del algoritmo de calibración	24
2.1.4. Conclusiones finales	33
2.2. Detección de obstáculos	34
2.2.1. Detección de obstáculos positivos basada en LiDAR 3D	38
2.2.1.1. Descripción del método utilizado en la detección de obstáculos positivos	39
2.2.1.2. Ejemplos de detección de obstáculos positivos en entornos reales	42
2.2.2. Detección de obstáculos negativos basada en LiDAR 3D	44
2.2.2.1. <i>Set-up</i> del sensor LiDAR para mejorar definición de puntos en el suelo	45
2.2.2.2. Descripción del método utilizado en la detección de obstáculos negativos	48
2.2.2.3. Resultados de la detección de obstáculos negativos en entornos reales	50
2.2.2.4. Discusión de los resultados del algoritmo de detección para obstáculos negativos	55
2.2.3. Conclusiones finales	56
2.3. Caracterización de la infraestructura	56
2.3.1. Trabajos previos	58

2.3.2.	Arquitectura general del algoritmo de caracterización transversal . . .	59
2.3.3.	Función para la detección de límites de la calzada.	61
2.3.4.	Función para la detección de líneas de carril y cálculo de número de carriles	63
2.3.5.	Resultados obtenidos para entornos estructurados	66
2.3.5.1.	Análisis de los resultados de los métodos de detección de límites	67
2.3.5.2.	Análisis de los resultados del método de detección de línea de carril	68
2.3.5.3.	Discusión de los resultados en entornos estructurados	69
2.3.6.	Aplicación del método de cálculo de los límites de la vía para entornos <i>off-road</i>	70
2.3.7.	Conclusiones finales	73
3.	Posicionamiento y mapeado	75
3.1.	Caracterización de elementos de referencia para apoyar la navegación autónoma	76
3.1.1.	Trabajos previos	76
3.1.2.	Metodología para caracterización y seguimiento de elemento de referencia	77
3.1.3.	Influencia del campo de visión horizontal del sensor	79
3.1.4.	Extrapolación simple del elemento de referencia utilizando modelo matemático de vehículo	81
3.1.4.1.	Estructura del algoritmo para reconstrucción de la trayectoria	82
3.1.4.2.	Influencia del peralte de la carretera	83
3.1.4.3.	Resultados con extrapolación mediante modelo dinámica vehicular	84
3.1.5.	Ponderación de puntos para relevancia homogénea del elemento de referencia	87
3.1.6.	Conclusiones finales	89
3.2.	Mapeado y localización simultánea para entornos con características específicas	90
3.2.1.	Trabajos previos	91
3.2.2.	Desarrollo de la metodología del algoritmo clásico de Visual SLAM . .	94
3.2.2.1.	Extracción de elementos característicos	94
3.2.2.2.	Asociación de datos	97
3.2.2.3.	Estimación y corrección de la posición	98
3.2.2.4.	Reconstrucción del mapa	100
3.2.2.5.	Propuesta de mejoras del algoritmo	100
3.2.3.	Ensayos preliminares y configuración final del algoritmo	102
3.2.4.	Pruebas y resultados	106
3.2.4.1.	Escenarios seleccionados	106
3.2.4.2.	Resultados en entornos reales	107
3.2.5.	Conclusiones finales	110
3.3.	Aplicación de modelos basados en redes de convolución para estimación de la odometría	110
3.3.1.	Introducción	111
3.3.1.1.	Importancia del modelo en técnicas basadas en <i>Machine Learning</i>	112

3.3.1.2.	Trabajos previos sobre Odometría Visual	113
3.3.1.3.	Organización del apartado	114
3.3.2.	Selección y tratamiento de datos	115
3.3.2.1.	Datos de entrada y salida	115
3.3.2.2.	Representación de datos	116
3.3.2.3.	Aumentación de datos (<i>Data Augmentation</i>)	120
3.3.2.4.	Descripción de <i>datasets</i>	122
3.3.3.	Ajuste del modelo	125
3.3.3.1.	Condicionamiento del problema	125
3.3.3.2.	Ajuste de hiper-parámetros	126
3.3.3.3.	Alternativas valoradas para las arquitecturas	127
3.3.4.	Selección del modelo final	135
3.3.4.1.	Ensayos realizados para obtener modelo CNN	135
3.3.4.2.	Resultados obtenidos con el modelo CNN seleccionado	140
3.3.5.	Conclusiones finales	144
3.4.	<i>Matching</i> robusto de nubes de puntos asistido por modelo de regresión de <i>Deep Learning</i> para métodos de Visual SLAM	145
3.4.1.	Justificación y objetivo del método de SLAM asistido por modelos de Deep Learning	145
3.4.2.	Trabajos previos	148
3.4.3.	Metodología propuesta para etapa de asociación asistida por modelo CNN	150
3.4.4.	Modelo de <i>Deep Learning</i> utilizado para la asistencia a la asociación de datos	152
3.4.5.	Valoración de la dependencia de la inicialización para métodos de asociación local	156
3.4.5.1.	Ensayos preliminares sobre la dependencia del tipo de inicialización en algoritmo ICP	157
3.4.5.2.	Ensayos preliminares sobre la dependencia del tipo de inicialización en algoritmo NDT	161
3.4.5.3.	Discusión de los resultados sobre la dependencia de la realización	163
3.4.6.	Verificación del funcionamiento del método ante perturbaciones de información de entorno	164
3.4.6.1.	Comportamiento del modelo de CNN en los entornos simulados con perturbaciones en la información del entorno	166
3.4.6.2.	Comportamiento de los algoritmos de asociación en los entornos simulados con perturbaciones en la información del entorno	168
3.4.6.3.	Aplicación de la metodología para etapa de asociación asistida por modelo CNN	171
3.4.7.	Conclusiones finales	175
4.	Aportaciones, conclusiones y futuros desarrollos	177
4.1.	Aportaciones y conclusiones	177
4.2.	Futuros desarrollos	181

4.3. Publicaciones	183
Referencias	195
A. Adaptación del algoritmo de clustering DBSCAN para sensores láser rotativos	205
A.1. Introducción	205
A.2. Principales algoritmos de clustering empleados en la conducción autónoma . .	206
A.3. Descripción de la adaptación del método de clustering DBSCAN	209
A.4. Resultados	211
A.4.1. Verificación en entorno ideal	211
A.4.2. Resultados obtenidos en entornos reales	213
A.5. Conclusiones finales	215

Índice de figuras

1.1. Estructura general del vehículo autónomo.	2
1.2. Organización de la tesis.	8
2.1. Patrón de calibración usado en calibración intrínseca de cámaras (Fuente: Mathworks)	12
2.2. Diagrama de flujo con las fases del algoritmo.	15
2.3. Resultados parciales de las fases principales del algoritmo.	16
2.4. Representación de los vectores normales sobre la nube de puntos obtenida. . .	17
2.5. Estudio de las direcciones principales del entorno. Representación de los vectores normales sobre superficie esférica de radio unitario.	17
2.6. Identificación de las direcciones predominantes en un mismo entorno.	18
2.7. Emparejamiento de planos equivalentes entre nubes de puntos.	22
2.8. Ajuste final de las nubes de puntos. a) Primera rotación. b) Segunda rotación, con la que se obtienen superficies paralelas. c) Ajuste de la translación.	23
2.9. Posición de los sensores LiDAR en el vehículo de prueba. a) Sensor cuyo sistema de referencia se toma como fijo. b) Sensor cuyo sistema de referencia será móvil.	25
2.10. Situación inicial nubes de puntos: en azul, asociada al sistema fijo; en rojo, asociada al sistema móvil.	27
2.11. Planos identificados en cada dirección predominante en Ensayo 1.	27
2.12. Situación final nubes de puntos Ensayo 1.	28
2.13. Situación inicial nubes de puntos Ensayo 3.	28
2.14. Planos identificados en cada dirección predominante en Ensayo 3.	29
2.15. Situación final nubes de puntos Ensayo 3.	29
2.16. Situación inicial nubes de puntos Ensayo 6.	30
2.17. Planos identificados en cada dirección predominante en Ensayo 6.	30
2.18. Situación final nubes de puntos Ensayo 6.	31
2.19. Situación final nubes de puntos Ensayo 9.	32
2.20. Planos identificados en cada dirección predominante en Ensayo 9.	32
2.21. Situación final nubes de puntos Ensayo 9.	33
2.22. Etapas genéricas para detección de obstáculos basada en LiDAR 3D. a) Nube de puntos del entorno; b) aplicación del ROI; c) identificación y segmentación de obstáculos; d) caracterización de obstáculos.	37
2.23. Delimitación del ROI y triangulación de Delaunay.	40
2.24. Cálculo del vector normal a cada triángulo.	40
2.25. Filtrado de triángulos según orientación con el eje vertical.	41

2.26. Ejemplo de funcionamiento del algoritmo de detección de obstáculos positivos. A la izquierda, se identifican obstáculos estáticos. A la derecha, se identifica, además, un peatón (obstáculo dinámico).	43
2.27. Ejemplo de funcionamiento ante dispersión de puntos del algoritmo de detección de obstáculos positivos.	44
2.28. Curvas cónicas generadas por el corte de las capas láser con el plano del suelo.	46
2.29. Valoración de ubicaciones alternativas del LiDAR sobre el vehículo.	47
2.30. Respuesta de los parámetros de extracción de obstáculos negativos.	49
2.31. Vehículo de pruebas. A la derecha, detalle de la ubicación del LiDAR 3D. . .	50
2.32. Zona de obstáculos normalizados.	51
2.33. Detecciones correctas en los obstáculos de las zonas 1 y 2.	54
2.34. Detecciones correctas en los obstáculos de la zona 3.	54
2.35. Ejemplo de situación de detección de un falso positivo.	55
2.36. Descripción del algoritmo de caracterización transversal de la carretera. Recuadradas con línea discontinua, las dos funciones principales del algoritmo. .	60
2.37. Ejemplo de patrón geométrico identificativo de bordillos de la calzada.	62
2.38. Variaciones de distancia entre dos capas consecutivas. Ejemplo patrón identificativo de bordillo.	63
2.39. Segmentación del suelo. a) Ajuste de curvas cónicas cada capa LiDAR que interseca el suelo, b) puntos filtrados pertenecientes al suelo (rojo).	64
2.40. Filtro de puntos pertenecientes a líneas viales del suelo.	64
2.41. Definición de las regiones de interés de cada tramo de línea para la asociación de datos.	65
2.42. Definición de ROI individuales de los tramos de líneas: a) Situación con información suficiente; b) situación con oclusiones donde se estima la orientación de los ROI.	65
2.43. Identificación zona transitable mediante Método II: a) Entorno real; b) nube de puntos original; c) resultado de la identificación. En verde, zona transitable.	67
2.44. Distintos ejemplos de situaciones con singularidades o detecciones erróneas: a) Detección incorrecta de arcén; b) Oclusiones debido a vehículos circundantes; c) Falta de información de alguna línea; d) No detección de carril de incorporación.	69
2.45. Resultado del algoritmo propuesto para la caracterización del entorno. Se observa la segmentación del suelo (azul oscuro) y las líneas de carril.	70
2.46. Detección de la zona transitable en entornos <i>off-road</i> mediante K-means. . . .	71
2.47. Zona transitable en entorno <i>off-road</i> mediante Método II: a) Entorno real; b) nube de puntos original; c) obtención del resultado de la identificación: en verde, los puntos dentro del ROI que son transitables.	72
3.1. Correspondencia del entorno real con datos LiDAR 3D. 1) Líneas viales centrales, 2) Líneas viales separación de arcén, 3) y 4) Barreras tipo New Jersey.	77
3.2. Ejemplo de aplicación del método sobre caracterización de barreras.	79
3.3. Respuesta del algoritmo: a) Ubicación del láser en el frontal, b) Ubicación del láser en el techo.	80
3.4. Diagrama de flujo del método para reconstrucción de la trayectoria.	82

3.5. Modelo vehicular de dos ruedas.	83
3.6. Influencia del peralte en la reconstrucción de la trayectoria en grandes distancias.	85
3.7. Comparativa de comportamiento del algoritmo con distintas configuraciones de colocación del láser.	86
3.8. Densidad de los puntos detectados (diagrama de barras), ajuste de la densidad medida (rojo) y función de ponderación (azul) respecto a la distancia al sensor.	88
3.9. Comportamiento de las variables de salida del algoritmo utilizadas para el control de la navegación autónoma: a) sin usar la función de ponderación; b) con la función de ponderación integrada.	88
3.10. Ejemplo de reconstrucción del entorno a partir de la fusión de cuatro LiDAR VLP-16, utilizando posiciones GPS-INS.	91
3.11. Etapas generales en algoritmos de Odometría Visual y Visual SLAM.	93
3.12. Ejemplo de identificación de superficies planas mediante Método 1.	95
3.13. Ejemplo de identificación de superficies planas mediante Método 2.	96
3.14. Identificación de los <i>landmarks</i> elegidos (Fuente: Im et al. (2016).)	97
3.15. Cálculo de las variables del movimiento relativo según el Método B.	99
3.16. Ejemplo de reconstrucción del mapa en entorno interior.	100
3.17. Identificación de orientaciones predominantes de planos verticales. La zona delimitada por la línea roja identifica la disposición de los vectores normales de los planos verticales.	102
3.18. Comparativa Método 1 (izquierda) y Método 2 (derecha), tanto en escenario ideal (arriba) como en entorno real (abajo). En rojo aparecen los puntos no identificados como planos.	103
3.19. Comparación del Método A (izquierda) y Método B (derecha) para la estimación de la trayectoria en entorno ideal.	104
3.20. Estimación de la trayectoria en Nave del INSIA-UPM.	105
3.21. Escenarios de ensayo. a) Nave industrial en las instalaciones del INSIA-UPM, b) parking interior, c) entorno urbano.	107
3.22. Nave INSIA-UPM (Trayectoria rectilínea)	108
3.23. Nave INSIA-UPM (Trayectoria en forma de L)	108
3.24. Parking interior.	109
3.25. Zona urbana.	109
3.26. Proyección de los puntos LiDAR en matriz de datos.	116
3.27. : Comparación de los rangos de visión horizontal entre la primera (a) y la segunda configuración (b) consideradas.	117
3.28. Ejemplos Mapas de Profundidad. Configuración con resolución de las imágenes-dato 8 x 300 x 2.	118
3.29. Ejemplos Mapas de Profundidad. Configuración con resolución de las imágenes-dato 28 x 720 x 2.	118
3.30. Ejemplo de distribución de valores de la imagen-dato antes de aplicar logaritmo (rojo) y después de ello (verde).	119
3.31. Ejemplos Mapas de Coordenadas. Configuración con resolución de las imágenes-dato 28 x 720 x 6.	119
3.32. Ejemplo de secuencia durante 6 observaciones seguidas representadas mediante <i>Grid-occupancy Maps</i>	120

3.33. Ejemplo de aplicación de la técnica <i>Mirroring</i> . a) Representación original del entorno. b) Representación espejo equivalente.	121
3.34. Ejemplo de aplicación de la técnica <i>Shaking</i> . a) Representación original del entorno. b) Representación con <i>Shaking</i> aplicado.	121
3.35. Ejemplo del entorno urbano incluido en el <i>dataset</i> Campus Sur – UPM. . . .	123
3.36. Rutas incluidas en <i>dataset</i> Campus Sur - UPM. a) Conjunto de entrenamiento (varias vueltas). b) Conjunto de evaluación.	123
3.37. Ejemplo del entorno urbano incluido en el <i>dataset</i> KITTI.	124
3.38. Capa <i>fully-connected</i> . Las flechas de la izquierda representan las entradas a la red, mientras que aquellas situadas a la derecha, las salidas. Cada conexión entre neuronas es un parámetro de peso. (Fuente: Ramsundar y Zadeh (2018)).	128
3.39. Ejemplo de estructura de las capas de procesamiento en modelos CNN.(Fuente: Geron (2019)).	128
3.40. a) Red neuronal artificial con función de activación ReLU; b) Topología genérica de Perceptrón Multicapa.	129
3.41. Topología de la red neuronal de convolución utilizada.	130
3.42. Resultados de la estimación de la topología CNN-5. En rojo, la salida esperada (<i>Ground Truth</i>) de cada una de las variables estimadas y en azul, la estimación de la red para cada una de ellas.	133
3.43. a) RMSE Velocidad; b) RMSE Variación de guiñada. En azul, error en entrenamiento y en naranja, error en validación.	134
3.44. Comparación de resolución en Mapas de Profundidad con distintos LiDAR. Rodeados en rojo, identificación de vehículos en ambas imágenes.	138
3.45. Tasa de aprendizaje en función del epoch utilizada durante los ensayos. . . .	138
3.46. Evolución del error en la estimación de la velocidad durante entrenamiento y validación.	141
3.47. Evolución del error en la estimación de la variación de guiñada durante entrenamiento y validación.	141
3.48. Estimación del modelo CNN-KITTI para la secuencia de evaluación. En azul, la señal de referencia para cada variable estimada, en rojo la salida de la red. . . .	142
3.49. Errores de translación y rotación en función de tramos de longitud y velocidades.	143
3.50. Ejemplo de etapa de asociación de nubes de puntos. a) Antes de <i>matching</i> ; b) después de <i>matching</i>	146
3.51. Estructura general del algoritmo de SLAM junto a la integración de la asistencia del modelo CNN (en rojo).	148
3.52. Diagrama de flujo de la metodología propuesta de asistencia a la etapa de asociación.	151
3.53. Ejemplo de mapa de profundidad del dataset del parking de La Gavia.	152
3.54. Trayectoria seleccionada para evaluación.	153
3.55. Estimación del modelo para la secuencia de evaluación. En azul, la señal de referencia para cada variable estimada, en rojo la salida de la red.	154
3.56. Reconstrucción de la trayectoria de la secuencia utilizada como evaluación. En azul, la trayectoria de referencia y en rojo, la reconstrucción de la trayectoria a partir de la estimación de la red.	155
3.57. Errores de translación y rotación en función de tramos de longitud.	155

3.58. Precisión obtenida con ICP en las tres situaciones en estudio.	158
3.59. Representación de rendimiento del algoritmo ICP. De naranja a azul, se representa la mayor o menor repetición del valor de número de iteraciones, respectivamente.	159
3.60. Representación del RMSE de asociación del algoritmo ICP. De naranja a azul, se representa la mayor o menor repetición del valor de error de asociación obtenido, respectivamente.	160
3.61. Precisión obtenida con NDT en las tres situaciones en estudio.	162
3.62. Representación de rendimiento del algoritmo NDT. De naranja a azul, se representa la mayor o menor repetición del valor de número de iteraciones, respectivamente.	162
3.63. Representación del RMSE de asociación del algoritmo NDT. De naranja a azul, se representa la mayor o menor repetición del valor de error de asociación obtenido, respectivamente.	163
3.64. Ejemplos de los entornos simulados. a) Limitación de alcance; b) <i>downsampling</i> aleatorio.	165
3.65. Ejemplo de entorno simulando situaciones con oclusiones.	165
3.66. Comportamiento del modelo de CNN en la situación de limitación de alcance de 10 m. Las líneas discontinuas indican el tramo de limitación simulada. . .	167
3.67. Comportamiento del modelo de CNN en la situación de limitación de oclusión fija. Las líneas discontinuas indican el tramo de limitación simulada.	168
3.68. Comportamiento algoritmo ICP ante limitación de la información del entorno a 10 metros. A la izquierda, el número de iteraciones y el error de asociación frente a la velocidad; a la derecha, frente a la variación de guiñada.	169
3.69. Error de asociación en función del tiempo en la situación con una oclusión variable. Las líneas discontinuas indican el tramo de limitación simulada. . .	170
3.70. Reconstrucción de la trayectoria en la simulación de la limitación de alcance a 15 m con ICP. En azul sin aplicación del método propuesto. En naranja, aplicando el método.	172
3.71. Reconstrucción de la trayectoria en la simulación de la limitación de alcance a 15 m con NDT. En azul sin aplicación del método propuesto. En naranja, aplicando el método.	174
3.72. Ejemplo de reconstrucción del mapa sobre la trayectoria de evaluación.	174
A.1. Ejemplos del funcionamiento de algunos métodos de clustering (Fuente: scikit-learn).	207
A.2. Esquema de funcionamiento algoritmos DBSCAN (Fuente: De Chire - Trabajo propio, CC BY-SA 3.0)	209
A.3. Definición de los parámetros $\Delta\theta$, $\Delta\theta$ y d . a) Vista en planta; b) vista de perfil.	209
A.4. Simulación en entorno ideal.	211
A.5. Variación de la divergencia en función de la distancia obtenida en simulación de entorno ideal.	212
A.6. Variación de la densidad en función de la distancia obtenida en simulación de entorno ideal.	212
A.7. Comparativa de funcionamiento del algoritmo DBSCAN con VLP-16.	213

A.8. Comparativa de funcionamiento del algoritmo DBSCAN con OS-1 64.	214
A.9. Aportación de la compensación de densidad de puntos. A la izquierda, solo con compensación de la divergencia. A la derecha, el método con compensación de divergencia y densidad.	215

Índice de tablas

2.1. Tipos de escenarios de prueba para calibración.	24
2.2. Resumen de ensayos para calibración.	25
2.3. Resumen resultados de rotación y translación por cada ensayo.	26
2.4. Número de emparejamientos de planos en cada dirección principal.	26
2.5. Zonas de obstáculos normalizados para entornos <i>off-road</i>	51
2.6. Matriz de confusión de los resultados del algoritmo de detección de obstáculos negativos.	53
2.7. Resultados de funciones de determinación de calzada (análisis sobre 875 frames)	67
2.8. Resultado individual del algoritmo de determinación del número de carriles. .	68
3.1. Desviaciones de los parámetros de control según la colocación del láser. . . .	86
3.2. Resultados de los métodos de mejora propuestos	106
3.3. Resultados en entornos reales.	108
3.4. Hiper-parámetros a tener en cuenta durante el ajuste del modelo.	127
3.5. Resultados obtenidos durante entrenamiento y validación (test) de las topologías CNN.	132
3.6. Comparación de resultados en entrenamiento según configuración de datos de entrada. Incluyendo 3 (t, t-1, t-2) y 2 observaciones consecutivas (t, t-1). . . .	135
3.7. Comparación de resultados en evaluación según configuración de datos de entrada. Incluyendo 3 (t, t-1, t-2) y 2 observaciones consecutivas (t, t-1). . . .	136
3.8. RMSE variación de guiñada frente al valor de dropout.	140
3.9. Rendimiento y precisión en algoritmo ICP en las tres situaciones planteadas.	158
3.10. Rendimiento y precisión en algoritmo NDT en las tres situaciones planteadas.	161
3.11. Designación de las simulaciones propuestas.	166
3.12. Resultados del modelo CNN pre-entrenado para las situaciones simuladas. . .	167
3.13. Resultados del algoritmo ICP para las situaciones simuladas.	169
3.14. Resultados del algoritmo NDT para las situaciones simuladas.	170
3.15. Comparativa de rendimiento y error de asociación del ICP para las situaciones simuladas.	171
3.16. Comparativa de error de translación y rotación del ICP para las situaciones simuladas.	171
3.17. Comparativa de rendimiento y error de asociación del algoritmo NDT para las situaciones simuladas.	173

3.18. Comparativa de error de translación y rotación del algoritmo NDT para las situaciones simuladas.	173
---	-----

Lista de acrónimos

APE	Absolute Pose Error.
BUS-CAN	Bus Controller Area Network.
CNN	Convolutional Neural Network.
DBSCAN	Density-based spatial clustering of applications with noise.
DNN	Deep Neural Network.
GNSS	Global Navigation Satellite System.
GPS	Global Positioning System.
ICP	Iterative Closest Point.
IMU	Inertial Measurement Unit.
LiDAR	Light Detection and Ranging.
MAE	Mean Absolute Error.
MHT	Multiple Hypothesis Tracking.
MSAC	M-estimator Sample Consensus.
NDT	Normal Distributions Transform.
RANSAC	Random Sample Consensus.
RMSE	Root-Mean-Square Error.
RNN	Recurrent Neural Network.
ROI	Region of Interest.
RPE	Relative Pose Error.
SIFT	Scale-invariant feature transform.
SLAM	Simultaneous Localization And Mapping.
V2X	Vehicle-to-everything communication.

Capítulo 1.

Introducción

1.1. Marco general de la tesis

La conducción autónoma ha sido de gran interés para fabricantes, desarrolladores e investigadores en los últimos años. En concreto, la creciente densidad de vehículos y las reglas que rigen el tráfico en cada país, suponen un gran reto de cara a afrontar una navegación autónoma en un futuro próximo. Esta línea de investigación conlleva una gran complejidad implícita y, por tanto, invita a encontrar soluciones a cualquier casuística que deba afrontar el vehículo autónomo.

Los retos a los que se enfrenta la navegación autónoma terrestre, además de ser muy diversos, dependerá del grado de automatización del vehículo (SAE, 2016) ya que cuanto mayor sea este, mayor cantidad de escenarios o situaciones deberá ser capaz de resolver. Independientemente del grado de automatización, atendiendo al esquema general del vehículo autónomo que se presenta en Pendleton et al. (2017), las principales competencias del desarrollo de algoritmos se distinguen en *Percepción*, *Toma de decisiones* y *Control de bajo nivel*. Cada uno de estos sistemas abarca diversas líneas de desarrollo que, a su vez, emplearán algoritmos específicos para llevar a cabo cada funcionalidad del vehículo. Esta estructura también contempla las ventajas que pueden aportar los sistemas de Conducción Cooperativa mediante comunicaciones V2X. Se crea así un sistema complejo de interrelaciones, flujo de información y algoritmos específicos que deben funcionar de forma coordinada y garantizando un tiempo de ejecución crítico para lograr la conducción totalmente autónoma y segura.

La presente Tesis Doctoral se enmarca concretamente en los sistemas de *Percepción* (figura 1.1). Ésta se puede definir como la capacidad que tiene un sistema autónomo de adquirir información y extraer conocimiento del entorno (Pendleton et al., 2017). Este sistema engloba las capacidades relacionadas con la *Caracterización del entorno* y con el *Posicionamiento y mapeado*, ambas desarrolladas ampliamente en los capítulos 2 y 3, respectivamente.

Respecto a la *Caracterización del entorno* se entiende como el propio desarrollo de un entendimiento contextual del entorno. Es decir, la localización de obstáculos, detección de marcas viales y/o señales de tráfico, caracterización de los datos, etc. Mientras que el *Posicionamiento y mapeado* determina la localización del vehículo respecto al entorno, ya sea mediante un posicionamiento a nivel global o local referido a un mapa digital del entorno reconocido.

Figura 1.1: Estructura general del vehículo autónomo.

Para completar esta visión de conjunto, cabe señalar que, a pesar de las relaciones entre los distintos sistemas mostrados en la figura 1.1, la implementación es secuencial en algunos casos. Así, por ejemplo, el modelo creado con la información del entorno, resultado de la etapa de *Percepción*, servirá de entrada para las funcionalidades de la Toma de decisiones. Por ello, se destaca la importancia de un correcto funcionamiento en los desarrollos que abarca la *Percepción*, al tratarse de la primera etapa de desarrollo y crearse estas dependencias entre etapas sucesivas.

1.2. Percepción del entorno basada en sensores LiDAR 3D

Las capacidades del vehículo autónomo se ven ampliadas continuamente debido en gran medida por el desarrollo de algoritmos basados en nuevas tecnologías y métodos. En concreto, los algoritmos para la percepción del entorno utilizan principalmente la información proporcionada por sensores como cámaras, stereo-vision, radar, LiDAR, GPS, IMU, módulos de comunicación DSRC, etc. Todos estos sensores dotan al vehículo de un alto grado de información del entorno y lo que le rodea. Sin embargo, esta información debe ser tratada y modelizada con el fin de generar un entendimiento de la escena por parte del vehículo.

La dificultad para crear un modelo que aporte una información completa y correcta del entorno suele dar lugar a fusionar estos sensores, ya que, mediante la redundancia de información, se logra mitigar las carencias de cada sistema por separado. El objetivo principal de llevar a cabo una fusión sensorial es cumplir con las exigencias de seguridad de la navegación autónoma o los sistemas de asistencia al conductor. Por ello, respecto a sensores de percepción, se han llevado a cabo soluciones basadas en la fusión de visión artificial, radar y LiDAR. Incluso, algunas implementaciones recurren al uso de mapas electrónicos de la carretera, complementando a los sensores de percepción mencionados.

De estas tecnologías utilizadas en la supervisión del entorno, la tecnología LiDAR es una de las que mayor relevancia está teniendo recientemente. Estos sensores láser pueden considerarse que ofrecen buenas prestaciones globales, aunque la visión artificial está ampliamente establecida en esta línea y proporciona una gran información del entorno. Sin embargo, en este caso, la falta de fiabilidad en ciertas situaciones restringe su utilización en aplicaciones donde la seguridad es un factor crítico y, en tal caso, se recurre a la fusión antes mencionada.

1.2.1. Ventajas e inconvenientes de los sensores LiDAR 3D frente a otros sistemas de percepción

Se han llevado a cabo diversos estudios comparando las ventajas y limitaciones que ofrecen los diferentes sistemas de percepción (X. Zhang et al., 2015; Tan et al., 2014; Widmann et al., 2000). En ellos, se destaca el potencial de la visión artificial para proporcionar una buena información sobre el entorno, sin embargo, también indican ciertos problemas que pueden ocurrir debido a cambios de iluminación, condiciones meteorológicas adversas y, en general, pueden suponer una alta carga computacional.

Por otro lado, los sensores radar y láser, pueden aportar ventajas alternativas a los anteriores, incluso mostrando prestaciones similares en algunas situaciones (Widmann et al., 2000). Ambos pueden obtener un alcance de hasta 200 metros, aunque a efectos prácticos, estas distancias no aportan información relevante. El principal inconveniente de los sensores radar es su limitado campo de visión lateral, mientras que con los láseres rotativos se puede obtener el rango completo de visión horizontal. No obstante, en situaciones con lluvia, niebla o nieve, el radar no obtiene, en general, dificultades durante la identificación, mientras que el LiDAR, sí se ve afectado. En estas situaciones, las partículas en el aire pueden provocar reflexión o dispersión del haz láser atenuando la señal, reduciendo el rango de alcance o incluso, adquiriendo falsos puntos del entorno (Widmann et al., 2000; Spies y Spies, 2006). Estas limitaciones han sido reducidas con la introducción de más capas láser, obteniendo mayor resolución del entorno, o añadiendo algoritmos de filtrado tras la adquisición de datos que permitan eliminar falsos puntos detectados.

Sin embargo, los sensores LiDAR sufren principalmente de dos desventajas: la dispersión entre puntos y las oclusiones. En el primer caso, la dispersión entre puntos aumenta cuanto mayor es la distancia al láser. Esta divergencia producida entre puntos adyacentes dentro de una misma capa y, a su vez, verticalmente entre capas, da lugar a una disminución en la definición de los elementos que se encuentran en el entorno. Por ello, los fabricantes de estos sensores desarrollan modelos con los que se obtiene mayor densidad de puntos o recurren a cambiar la distribución entre capas para que se encuentren más cercanas entre sí en las zonas críticas para la aplicación.

Por otro lado, la otra desventaja que puede ocurrir durante la adquisición de datos del entorno son las oclusiones. Estas pueden referirse a elementos del entorno que se encuentren más cercanos al sensor, ocultando a otros elementos que se sitúen detrás. En tal caso, es un problema que no solo sufren los sensores láser, sino cualquier otro sensor de percepción, ya que es dependiente del entorno. Sin embargo, durante la conducción, el sensor láser queda expuesto en el vehículo ante pequeñas partículas que aparezcan como, por ejemplo, polvo, gotas de lluvia, nieve, suciedad o incluso, insectos, pudiendo producir pequeñas oclusiones.

En tal caso, al igual que ocurre en las cámaras, una mayor apertura óptica, se traduce en un mayor diámetro de haz, en el caso del láser, lo que permite que la potencia de la señal sea atenuada solo parcialmente en lugar de quedar completamente bloqueada y, por tanto, la nube de puntos muestra un mínimo impacto.

La información espacial que los sensores LiDAR proporcionan en forma de nubes de puntos, ha propiciado el aumento de desarrollos basados en esta tecnología. Por otra parte, la mayoría de los modelos LiDAR adquieren también la información de la reflectividad del objeto donde impacta el haz. Esta característica, puede aportar ventajas para identificar marcas viales o señales de tráfico reflectantes, enriqueciendo la información del entorno al proporcionar una visión más cercana a la visión del conductor o las cámaras.

1.2.2. Versatilidad de los sensores LiDAR para la percepción del entorno

Las principales ventajas de los sensores LiDAR están en su fiabilidad en la detección y su adaptabilidad para ser implementados en diferentes aplicaciones, más aún con el desarrollo de estos sensores. En los últimos años, los sensores láser rotativos han evolucionado de tener solo una capa de adquisición (LiDAR 2D), a disponer de modelos comerciales de hasta 128 capas, con una distribución vertical variable y configurable, y tasas de adquisición entre 5 Hz y 20 Hz. Así, se pueden encontrar modelos LiDAR 3D específicos para el desarrollo de algoritmos para la conducción autónoma, donde la distribución entre capas láser proporciona una mayor densidad de puntos en la zona media, donde se espera encontrar mayor cantidad de información relevante. Todo ello ha ocasionado que este tipo de sensores haya cobrado mayor importancia y que, fabricantes y desarrolladores de prototipos de vehículo autónomo, lo tengan en cuenta como sensor principal de percepción.

Esta versatilidad se hace notable en la variedad de aplicaciones que tiene el LiDAR como base de su funcionamiento. Los datos en forma de nubes de puntos ofrecen una información tridimensional que es aprovechada para diseñar algoritmos más robustos o completar la información de otros sensores.

En este sentido, la primera aplicación directa y, hoy en día, establecida como la función principal de estos sensores, es la detección de obstáculos. A partir de la información de las nubes de puntos, se ha posibilitado identificar objetos localizados a distintas distancias y, sobre todo, mejorar la segmentación tras la identificación. El tratamiento de datos con nubes de puntos ha permitido aplicar algoritmos de *clustering* que ofrecen un gran rendimiento cuando los datos se disponen en el espacio como DBSCAN, OPTICS, K-means, K-medoids o K-prototype.

De esta manera, el potencial demostrado respecto a la identificación de obstáculos ha hecho evolucionar los métodos de detección y aplicarlos a objetivos más específicos como herramienta de caracterización del entorno. Así, por ejemplo, el LiDAR 3D se puede encontrar aplicado a la extracción de bordillos en entornos urbanos como en Fernández et al. (2014), la evaluación de grietas en el asfalto para llevar a cabo labores de mantenimiento (Jo y Ryu, 2015) o el cálculo de los límites transitables de la vía (W. Zhang, 2010), entre otros.

Además, no solo se ha evolucionado en el desarrollo de algoritmos para la caracterización, sino que también, se ha establecido como sensor de apoyo al posicionamiento. En este caso,

existen diversas soluciones de posicionamiento y reconstrucción del mapa del entorno como J. Zhang y Singh (2016); Shan y Englott (2018). En estos casos, al igual que la visión artificial, se hace uso de la capacidad de extracción de elementos del entorno para estimar el posicionamiento del vehículo. En este tipo de algoritmos, tienen un especial interés para obtener el posicionamiento en entornos interiores, donde otros sensores de posicionamiento (receptores GNSS) encuentran dificultades.

Más recientemente, el creciente desarrollo e interés por las técnicas de *Machine Learning* y en especial de *Deep Learning*, han encontrado un campo de aplicación extenso en el vehículo autónomo. Esto se debe, en parte, a que muchas de las capacidades de la conducción autónoma tratan de simular un comportamiento humano basado en la experiencia. En este caso, la visión artificial, junto con las redes neuronales de convolución, han conseguido posicionarse con métodos de referencia para la segmentación del entorno. El LiDAR 3D, por su parte, está consiguiendo establecerse igualmente, adaptando estas técnicas para aplicaciones que abarcan desde la identificación de objetos, por ejemplo Wirges et al. (2019), el posicionamiento del vehículo (Nicolai et al., 2016) o la optimización de la reconstrucción de un mapa (Ramezani et al., 2020). El interés por utilizar estas técnicas se debe principalmente a los requisitos de rendimiento demandados por el vehículo. En este sentido, el tiempo de ejecución sobre los modelos pre-entrenados es mucho menor, por lo que el tiempo crítico de cómputo del algoritmo quedaría reducido al tratamiento de datos de entrada y de salida.

1.3. Limitaciones del estado actual

Actualmente existen modelos de vehículos con grado de automatización hasta nivel 3 y no se vislumbra en un futuro cercano la conducción totalmente autónoma. Hasta el momento, han tenido lugar numerosos avances en el vehículo autónomo, dotándolo de altas prestaciones. Sin embargo, las dificultades para seguir avanzando en este sentido radican en la diversidad de escenarios y situaciones que se pueden dar, los requisitos de rendimiento o incluso el coste económico de su integración. Es por ello por lo que se precisa profundizar en estas limitaciones:

Limitaciones sobre la utilización de sensores LiDAR en algoritmos de *Percepción*:

- La información incompleta o poco fiable que se produzca durante la *Percepción* del entorno repercutirá al sistema de *Toma de decisiones*. Por ello, obliga a que las funcionalidades de este último adopten medidas conservadoras y, en el peor de los casos, erróneas.
- El coste de los sensores láser, a pesar de su continua evolución y abaratamiento, siguen suponiendo un coste sustancial respecto al resto de sensores y, a su vez, sobre el coste total del vehículo. Por tanto, a pesar de las ventajas en la percepción del entorno que puede suponer la utilización de LiDAR, algunos fabricantes e integradores no contemplan utilizarlos como solución exclusiva de identificación de obstáculos.

Limitaciones de los algoritmos de fusión sensorial:

- A partir de la fusión sensorial se puede obtener una información más completa del entorno siempre que la conjunción de sensores garantice una representación fiable del entorno y, en la medida de lo posible, no se obtenga falsos positivos o negativo.
- La fusión sensorial requiere, entre otros factores, establecer el mejor posicionamiento de los sensores sobre el vehículo. Sin embargo, llevar a cabo la tarea de calibración entre ellos mediante elementos externos, mediciones, etc. no resulta eficiente para valorar posibles configuraciones (sobre todo en las primeras etapas de experimentación).

Limitaciones en la caracterización de elementos y el posicionamiento del vehículo:

- Los algoritmos de detección de obstáculos basados en tecnología láser obtienen un buen funcionamiento general. Sin embargo, suelen estar orientados hacia entornos urbanizados y con terreno preparado, por lo que su aplicabilidad en entornos todoterreno presenta mayor incertidumbre.
- Los mapas electrónicos para navegación pueden apoyar el posicionamiento del vehículo, sin embargo, no disponen del detalle y precisión que se requiere para funcionalidades de seguridad.
- Algunas aplicaciones de posicionamiento requieren alta precisión, incluso a nivel de carril, sin embargo, la tecnología actual de posicionamiento por satélite no ofrece los niveles de precisión adecuados. En ciertos entornos complejos, puede producirse pérdidas de señal o calidad de ésta.
- Las soluciones desarrolladas hasta el momento para lograr el posicionamiento global sobre un mapa reconstruido pueden obtener una buena precisión. Los métodos analíticos en tal caso suelen condicionar el problema de forma correcta, sin embargo, encuentran dificultades cuando la información disponible del entorno deja de ser relevante para el correcto funcionamiento de estas técnicas convencionales.

Limitaciones sobre la flexibilidad y rendimiento de los algoritmos:

- Si bien el razonamiento humano, basándose en la experiencia, es capaz de resolver situaciones complejas o desconocidas a priori, los sistemas automáticos cuentan con menos flexibilidad para abordar estas situaciones.
- Uno de los factores cruciales para que la navegación autónoma sea una realidad, es garantizar el tiempo de ejecución en sus algoritmos. Sin embargo, muchas de las aplicaciones requieren una gran cantidad de procesamiento y adolecen de no poder ejecutarse en tiempo real. Por ejemplo, para un posicionamiento preciso del vehículo, algunas soluciones recurren a una reconstrucción del mapa del entorno, la cual precisa de la secuenciación de una serie de algoritmos orientados a resolver tareas específicas.

1.4. Objetivos

En la presente Tesis Doctoral se propone estudiar las capacidades y versatilidad de los sensores LiDAR 3D para una navegación autónoma a través del desarrollo de técnicas que posibiliten la reconstrucción del entorno. Para ello se abarcan dos de las líneas fundamentales dentro de la percepción del vehículo autónomo: la caracterización de elementos y las técnicas de posicionamiento y mapeado. Debido a la gran variabilidad de situaciones donde el vehículo deberá desenvolverse, los algoritmos propuestos son implementados y evaluados sobre zonas reales urbanas, interurbanas o con terreno no preparado.

Así, el objetivo principal del trabajo de investigación realizado puede enunciarse como:

Desarrollar técnicas avanzadas de percepción del entorno basadas en LiDAR 3D que permitan una navegación autónoma terrestre segura y la mejora del entendimiento del escenario, basándose en la caracterización de este y proporcionando un posicionamiento preciso bajo cualquier circunstancia que deba resolver el vehículo autónomo.

Como objetivos parciales se plantean los siguientes:

1. Implementar algoritmos con la premisa de mantener una carga computacional contenida para poder llevar a cabo funcionalidades tanto de caracterización y como de posicionamiento de forma eficiente. Dotando de mayores capacidades a los algoritmos, se logrará proporcionar una información más completa del entorno, que se traslada directamente al sistema de Toma de decisiones. Se pretende encontrar un equilibrio entre rendimiento y precisión. Utilizar el mismo sensor para varios propósitos, resultará en una optimización de costes y eficiencia.
2. Desarrollar una metodología de calibración que facilite la integración entre dos o más sensores láser para una fusión sensorial. Priorizar la capacidad para adaptarse a distintas configuraciones de los sensores sobre el vehículo, garantizando al menos una fusión de alto nivel (donde se fusionan las entidades resultantes de cada sistema de percepción por separado).
3. Implementar una solución de detección de obstáculos orientada a entornos complejos, donde no solo sea operativa cuando el escenario presente cierto grado de infraestructura, sino también en terrenos irregulares *off-road*.
4. Desarrollar una metodología viable para la caracterización horizontal de la vía, donde a partir de la información extraída con el láser, se pueda completar la información disponible de mapas electrónicos y cumplir así con los requisitos de seguridad durante la navegación.
5. Aplicar los métodos de caracterización del entorno para apoyar un posicionamiento a nivel de carril, mejorando la eficiencia de los recursos del vehículo. Evaluar asimismo el desempeño de los sistemas de posicionamiento basado en láser en zonas complejas, donde los sistemas de posicionamiento por satélite no consiguen la precisión adecuada.
6. Aplicar modelos basados en *Deep Learning* para proporcionar una estimación del posicionamiento. Las ventajas de estos modelos permitirán reducir la carga computacional

en los sistemas de reconstrucción del mapa para lograr una ejecución en tiempo real.

7. Desarrollar una metodología generalizable para métodos analíticos de reconstrucción del mapa del entorno basada en modelos de *Deep Learning*, aportando una solución flexible para afrontar situaciones adversas. Se reducen, además, problemas derivados de la dependencia geométrica del entorno en los métodos analíticos.

1.5. Estructura de la Tesis

La presente Tesis se divide en cuatro capítulos. El primero de estos sirve como introducción para situar el campo de desarrollo de los métodos propuestos en los capítulos siguientes. Además, se resaltan las carencias generales observadas que pueden ser abordadas desde los sistemas de *Percepción*, y que guiarán los objetivos perseguidos en el trabajo de investigación. Los capítulos 2 y 3 atienden a la clasificación de los métodos utilizados dentro de la percepción del vehículo autónomo: *Caracterización del entorno* y *Posicionamiento y Mapeado*. Si bien estas dos líneas se presentan diferenciadas en la estructura general del vehículo autónomo, ambas conforman el sistema de *Percepción* y se caracterizan por tener aspectos y objetivos comunes para completar la información del entorno. Así, aunque los capítulos de la Tesis se han dividido siguiendo esta clasificación, la delimitación de cada método desarrollado no está totalmente definida (figura 1.2). Algunos desarrollos podrán ser continuación o consecuencia del anterior y tener aspectos comunes. El capítulo 4 recogerá los aspectos más relevantes de cada resultado y se destacarán las aportaciones principales de esta Tesis.

Figura 1.2: Organización de la tesis.

A continuación se detallan los aspectos más relevantes que se abarcarán en los capítulos 2 y 3.

Capítulo 2: Herramientas para la caracterización del entorno

Se desarrollan herramientas centradas en la identificación de elementos del entorno. En el apartado 2.1, se aborda el problema de la calibración extrínseca entre sensores láser, ya que, para mitigar los problemas de divergencia entre puntos u oclusiones, se suele recurrir a una fusión sensorial. Este apartado, presenta un primer acercamiento al tratamiento de las nubes de puntos, utilizando diversas funciones de caracterización y emparejamiento de características extraídas del entorno, que servirán como base para algoritmos desarrollados en los apartados sucesivos.

En el apartado 2.2 se aborda la identificación de obstáculos, al ser la principal aplicación de los sensores láser, servirá para poner en contexto la capacidad de éstos para llevar a cabo una identificación robusta. Se destaca el desarrollo orientado a entornos complejos *off-road*, ya que precisa de una identificación específica y una estrategia distinta para la identificación de obstáculos. La validación se realiza sobre ensayos en entornos reales con una implementación en tiempo real en un vehículo autónomo, utilizando el modelo VLP-16 de Velodyne.

Para completar la caracterización del entorno no es suficiente con identificar los obstáculos. Por ello, el apartado 2.3 aporta diversos métodos que sirven para complementar al modelo de reconstrucción del entorno. De esta manera, se llevan a cabo ensayos tanto en zonas urbanas, interurbanas y todoterreno, que validen los algoritmos planteados sobre la extracción del número y ancho de carriles o los límites de la zona transitable. Todo ello, permitirá, por un lado, completar la información del mapa, y por otro, aportar información relevante de cara a un posicionamiento.

Capítulo 3: Posicionamiento y mapeado

Se aborda, en este capítulo, la problemática del posicionamiento y el mapeado. A partir de los desarrollos del apartado 2.3, se observa las capacidades de obtener un posicionamiento basado en una caracterización previa del entorno. Por ello, en el apartado 3.1 se presenta un sistema de apoyo al guiado autónomo a través de la identificación de elementos característicos. Se realizará igualmente una valoración sobre la influencia de situar el sensor en la parte superior del vehículo, o bien, en el frontal, donde se restringe la adquisición por la oclusión con el propio vehículo. Para ello, se plantea reconstruir la trayectoria a partir de la implementación de un modelo de dinámica vehicular junto la influencia del peralte. Se empleará un LiDAR VLP-16 para caracterizar el entorno, un giróscopo RMS FES 33 de alta resolución para la medición del peralte, un receptor GPS Trimble R4 para un posicionamiento de referencia y la adquisición de información por BUS-CAN para obtener los datos para el modelo de dinámica vehicular.

El apartado 3.2 profundiza en la reconstrucción de la trayectoria y, por tanto, en la del mapa por donde circula el vehículo. Esta vez eliminando los elementos externos utilizados en el apartado anterior (giróscopo, señal de BUS-CAN, etc.), para basarse exclusivamente en

la información LiDAR. Una de las limitaciones que se observan en cuanto a los métodos de posicionamiento y reconstrucción del mapa simultáneos (SLAM) es su alta carga computacional para garantizar su ejecución en tiempo real. Se implementa una metodología de SLAM orientada a entornos con una infraestructura característica, por lo que permite simplificar la complejidad de estos algoritmos manteniendo la precisión.

El apartado 3.3 ahonda en la estimación del posicionamiento basado exclusivamente la información láser, pero, esta vez, no específicamente para escenarios en concreto como en el apartado anterior. Para seguir manteniendo la carga computacional baja, se recurre a estudiar la viabilidad de técnicas de *Deep Learning*. Se desarrollará una topología de redes convolucionales que proporcionen la estimación del movimiento relativo del vehículo a partir de los datos láser. A partir de esta estimación del movimiento, se valorará la precisión para garantizar el posicionamiento preciso.

El apartado 3.4 es una continuación del trabajo realizado en el apartado 3.3 en el que se pretende encontrar una metodología que asista a la etapa de asociación de características en los sistemas de SLAM convencionales. El rendimiento y precisión de los métodos de SLAM, en general, son dependientes de la etapa de *matching* de características entre dos instantes de tiempo consecutivos. Esta asociación está numéricamente definida al tratarse de un problema geométrico. Sin embargo, variaciones en la información del entorno o problemas de identificación de características, pueden dar lugar a un mal funcionamiento de los métodos estrictamente analíticos. Por ello, se plantea realizar una primera aproximación basada en la estimación de un modelo de *Deep Learning*, que ayude a mitigar estas limitaciones. Será necesario, por tanto, realizar diversas validaciones de la idoneidad de este modelo para conseguir asistir a los métodos convencionales. En este apartado, a diferencia de los anteriores, se utiliza el LiDAR Ouster OS-1 64, con el que se consigue una mayor resolución del entorno y, por tanto, una mejora en la extracción de características.

Entornos y lenguajes de programación

Los lenguajes de programación utilizados en la Tesis dependen del entorno de trabajo requerido en cada caso. Así, para el prototipado de algoritmos, pruebas o tratamiento de datos, se ha empleado MATLAB por su facilidad de programación, recursos disponibles y representación visual de los resultados. Sin embargo, los métodos que requieren adquisición y sincronización de datos u optimización del tiempo de ejecución han sido desarrollados en el *framework* ROS Kinetic y programados en Python 2.7. En el caso del desarrollo de los modelos de *Deep Learning*, se han implementado utilizando los *frameworks* TensorFlow 1.15.2 y Keras 2.3.1 y programado en Python 3.6.

Capítulo 2.

Herramientas para la caracterización del entorno

2.1. Calibración extrínseca de sensores LiDAR

Uno de los objetivos de esta tesis es explorar las capacidades de los sensores láser escáner 3D aplicados a la conducción autónoma. Sin embargo, debido a la información dispersa y a veces incompleta del entorno que proporcionan, se suele recurrir a una fusión sensorial. De esta manera, se consigue una captura del entorno mucho más densa y completa. El primer paso para lograr una fusión sensorial correcta es realizar una calibración extrínseca entre sensores y obtener un sistema de referencia común.

La dificultad en la fusión de datos entre sensores LiDAR radica precisamente en la propia naturaleza de la información proporcionada. Sin saber a priori la posición y orientación relativa entre sensores, todos ellos estarían capturando la información de un mismo entorno desde perspectivas distintas, es decir, desde sistemas de referencias independientes, cada uno con origen en el sensor. Por tanto, en este apartado, se aborda el desarrollo de una solución para la calibración extrínseca entre LiDAR.

El procedimiento clásico para llevar a cabo una calibración extrínseca entre sensores láser deriva de la metodología de calibración extrínseca habitual entre sensores de visión. En esta última, se utilizan patrones de calibración mediante los cuales se extraen puntos clave que permiten calcular la posición relativa entre sensores. En el caso de utilizar sensores láser, diversas soluciones han implementado estos patrones de calibración para, de igual modo, extraer *keypoints* que sirvan para calcular el posicionamiento relativo entre ellos. Debido a la problemática descrita anteriormente sobre la divergencia entre puntos y la información no densa del espacio, este proceso requiere una intervención manual. Por tanto, se propone un método de calibración extrínseca automático entre varios LiDAR sin necesidad explícita de patrones de calibración. Se logra, mediante la extracción y posterior emparejamiento de puntos característicos del entorno, calcular las posiciones entre sensores. A través de este método, se facilita integrar y realizar pruebas de distintas configuraciones de sensores sobre el vehículo de forma rápida, automática y versátil.

Se estudia, además, las capacidades del algoritmo de calibración propuesto desde un punto de vista geométrico y en función de la información disponible del entorno. Al tratarse de un método automático que no requiere de patrones de calibración, los puntos característicos son

extraídos del entorno y, por tanto, éste influirá en el resultado final del algoritmo. Además, la solución propuesta es escalable en el número de sensores, si bien se realizará el análisis del funcionamiento utilizando dos LiDAR.

En definitiva, el método propuesto permite fusionar sensores LiDAR en cualquier escenario sin intervención manual y sin necesidad de elementos adicionales. Ello puede permitir modificar las configuraciones de los sensores sobre el vehículo para estudiar cuál es la opción más idónea (práctica muy frecuente en el campo de la investigación) o realizar calibraciones periódicas.

2.1.1. Alternativas para calibración de sistemas de percepción

Asegurar que la percepción del entorno del vehículo sea óptima no es trivial y depende en gran medida de la integración y fusión de información que se haga de los sensores sobre el vehículo. Por tanto, una tarea recurrente en las primeras fases de investigación es probar diferentes ubicaciones de sensores de tal manera que proporcionen la mejor representación del entorno. La primera dificultad que ello conlleva es configurar de nuevo todos los sensores para que la nueva integración en el vehículo sea coherente con los datos proporcionados. Es necesario definir un sistema de referencia común en el vehículo y establecer las correspondencias entre sensores.

Cabe mencionar que, para llevar a cabo una calibración intrínseca en sensores de visión, la metodología convencional utiliza paneles de calibración (figura 2.1). Esta solución ha sido adoptada para realizar la calibración extrínseca entre sensores del mismo género, obteniendo resultados precisos. La dificultad de extrapolar este procedimiento a la calibración entre sensores láser radica de la propia tipología de datos.

Figura 2.1: Patrón de calibración usado en calibración intrínseca de cámaras (Fuente: Mathworks)

Actualmente, existen diferentes métodos para dar solución a este problema. Sin embargo, todos ellos requieren de la interacción con elementos externos como son:

- **Patrones de calibración**, a partir de los cuales determinar la posición relativa entre láseres.
- **Generar un escenario de calibración** en el que se deba posicionar el vehículo para llevar a cabo la fusión sensorial.

- **Complementar la información** capturada del entorno con otro tipo de sensores, por ejemplo, con visión artificial.

Además, los cálculos se suelen realizar introduciendo parámetros de manera manual, con el coste en tiempo y dificultad añadido. Por otro lado, suelen estar adecuados para un escenario en concreto, perdiendo la versatilidad que una calibración automática proporciona, facilitando en tal caso, estudiar cómo afectan las condiciones del entorno. En particular, en Shang et al. (2015) se emplean dos superficies planas perpendiculares entre sí. A través de esta característica se consigue la calibración, sin embargo, la región de ambos planos deber ser introducida manualmente.

En otros casos, como en Rowekamper et al. (2015), para llevar a cabo la calibración es necesario que el elemento externo de referencia sea dinámico, por lo que supone cierta limitación y complica la puesta en marcha del algoritmo. Cabe indicar que este método tiene ciertas restricciones en la orientación de los sensores. Estos pueden ser posicionados dentro de un rango de inclinaciones, lo que puede reducir la versatilidad del método de calibración. Esta característica será objeto de estudio en este apartado sobre el algoritmo propuesto.

Otros métodos estudiados en la literatura confieren un nivel de automatización al proceso, sin embargo, no llega a ser totalmente autónomo. En concreto, en Pereira et al. (2016) el proceso se consigue realizar de forma automática, pero para lograr el resultado, es necesario introducir una esfera de al menos cinco metros de diámetro. Más recientemente en Kummerle et al. (2018), se utiliza de nuevo una esfera como elemento de calibración, esta vez, con un diámetro de 50 cm. En este desarrollo, consiguen obtener una alta precisión de calibración entre cámaras y sensores LiDAR a través del movimiento de este elemento. En Heide et al. (2018), se propone un procedimiento semiautomático al ser necesario introducir una primera aproximación por parte del usuario, al mismo tiempo, este algoritmo depende de varios elementos del entorno que no facilitan su utilización.

El objetivo principal del estudio realizado es generar un sistema de referencia común para dos sensores LiDAR posicionados en diversas orientaciones sobre el vehículo con el propósito de disponer de un único conjunto de datos del entorno más detallado.

Para ello, se ha desarrollado un algoritmo que corrige las limitaciones de los métodos anteriores y cumple con las condiciones de automatización del proceso y versatilidad. Ha sido probado en distintos escenarios exteriores e interiores, con mayor y menor grado de infraestructura alrededor y con distintas posiciones relativas entre sensores. De esta manera, se realiza un extenso análisis de resultados donde se pueden extraer conclusiones sobre su precisión y limitaciones.

Aunque en algunas situaciones, los resultados obtenidos carecen de la precisión requerida, el método supone un avance novedoso en el que la única limitación es la de ser utilizado en un entorno donde haya superficies planas, como suele ser cualquier escenario interior o exterior urbanizado. Asimismo, el algoritmo proporciona como salida una valoración del resultado. Este cálculo del error y su retroalimentación ofrece la posibilidad de mejora de la precisión en un proceso iterativo.

2.1.2. Método de calibración extrínseca automática para sensores LiDAR

A continuación, se presenta un método para obtener todas las transformaciones geométricas necesarias para conseguir la fusión sensorial entre varios sensores LiDAR y, por tanto, un sistema de coordenadas común para todos ellos. Este método es aplicable a cualquier configuración y número de sensores láser sobre el vehículo. Sin embargo, para el seguimiento de la descripción del algoritmo y las figuras que representan resultados intermedios, los cálculos se basan en la calibración entre dos sensores, donde uno de ellos mantiene su sistema de coordenadas como el absoluto del sistema, situado en el techo del vehículo, mientras que, para el otro sensor, en el frontal del vehículo, se calculan para conseguir la calibración extrínseca.

2.1.2.1. Consideraciones previas

Los cálculos deben restringir los seis grados de libertad del sistema de coordenadas móvil para que la nube de puntos se solape con aquella del otro láser. La restricción de los seis grados de libertad es posible lograrla mediante la identificación de al menos tres puntos característicos del entorno en ambos LiDAR. Sin embargo, debido a las capacidades tecnológicas de estos sensores, no es posible llevarlo a cabo directamente. Estos sensores no proporcionan una información densa o completa del entorno, mediante la cual sea posible definir con precisión la posición de un mismo punto en el espacio visto desde los dos láseres. Además de ofrecer una información dispersa en el espacio, sufren la divergencia entre puntos respecto a la distancia al sistema de referencia y pueden ocasionarse oclusiones por algún objeto. Por ello, la adquisición de los puntos no se produciría en el mismo lugar entre láseres que estén posicionados en distintos lugares (incluso en algunos modelos LiDAR no existe repetibilidad de las coordenadas del mismo punto de una revolución a la siguiente).

Este hecho se hace palpable en la comparación con la calibración extrínseca mediante sensores de visión artificial cuando se utiliza un patrón de calibración. En tal caso, la información densa cuando se trabaja con imagen, permite definir con precisión los puntos clave dentro de un patrón de calibración, mientras que cuando se utilizan sensores láser, no es posible. En el caso de que se utilizara un patrón de este tipo, la nube de puntos obtenida por el LiDAR determinaría con precisión solamente las coordenadas de los puntos que impacten sobre el propio elemento de calibración. Todo ello conlleva a que la información del entorno es ligeramente distinta de un láser a otro y, por tanto, la estrategia a seguir para conseguir una calibración correcta entre ellos presenta dificultades.

2.1.2.2. Descripción del algoritmo

En el flujograma de la figura 2.2, se pueden observar las fases que componen la metodología seguida, así como las interacciones entre ellas. El algoritmo consta siempre de unas fases establecidas, sin embargo, dependiendo de los resultados intermedios (figura 2.3, resultados parciales), las interacciones entre fases variarán.

El punto de partida es la información proporcionada por los sensores. Ambas nubes de puntos contienen las coordenadas cartesianas de los puntos del entorno referenciados al sistema de coordenadas de cada láser. El resultado obtenido por este algoritmo de calibración auto-

mática será una matriz de rotación y translación que, al ser aplicada sobre la nube de puntos del láser situado en la parte frontal del vehículo, coincidirá con el sistema de coordenadas global del sistema.

Figura 2.2: Diagrama de flujo con las fases del algoritmo.

Figura 2.3: Resultados parciales de las fases principales del algoritmo.

a) Identificación de las tres direcciones predominantes en el entorno

El objetivo de la primera fase del algoritmo será encontrar las tres superficies predominantes del entorno comunes en las dos nubes de puntos consideradas. Una vez calculadas y relacionadas entre sí, será posible realizar los cálculos necesarios para un emparejamiento preciso entre puntos característicos.

El primer paso es obtener los vectores normales en cada punto. Apoyándose en n_{puntos} vecinos, es posible tener una superficie sobre la cual calcular el vector normal asociado a cada punto. Los vectores obtenidos son unitarios y perpendiculares a la superficie que describen con ambos sentidos de orientación, tal como se observa en la figura 2.4. Por tanto, si se representan todos estos vectores sobre una esfera de radio unitario, se puede realizar el estudio de las direcciones normales predominantes en el entorno (figura 2.5).

Figura 2.4: Representación de los vectores normales sobre la nube de puntos obtenida.

Figura 2.5: Estudio de las direcciones principales del entorno. Representación de los vectores normales sobre superficie esférica de radio unitario.

La superficie de esta esfera unitaria describe la distribución de direcciones normales del entorno. Una consideración a tener en cuenta, para el estudio de la densidad, es que los extremos de los vectores diametralmente opuestos describen superficies paralelas. Además, para restringir correctamente los seis grados de libertad, será necesario que las tres direcciones predominantes obtenidas sean sustancialmente distintas entre sí. De esta manera, se garantiza una restricción geométrica robusta que, de no ser así, el emparejamiento entre planos se vería comprometido. El resultado de este paso es un conjunto de tres vectores por cada láser. Al tratarse del mismo entorno, el resultado esperado es que las direcciones predominantes sean similares y, por tanto, también sus orientaciones relativas. En la figura 2.6, se observa como a pesar de posicionar los dos sensores LiDAR con orientaciones distintas, se identifican correctamente las direcciones predominantes en el entorno.

Figura 2.6: Identificación de las direcciones predominantes en un mismo entorno.

Debido a oclusiones o un posicionamiento desfavorable del láser, la representación del mismo entorno puede ser considerablemente distinta de un sensor al otro. Este hecho podría dar lugar a determinar otras direcciones predominantes. Esta comprobación se realiza analizando las orientaciones relativas de estas direcciones principales. Comparando los ángulos que forman los vectores entre las nubes de puntos dos a dos, es posible descartar aquella dirección que no se corresponda y realizar de nuevo el cálculo.

b) Análisis de la geometría de las dos ternas de vectores

Identificar y relacionar correctamente las direcciones predominantes del entorno es un paso fundamental en este algoritmo de calibración. Cometer un error en la identificación de las mismas direcciones desde los dos sensores, repercutirá en los resultados posteriores

del método.

El punto de inicio en esta fase será una terna de vectores por cada LiDAR implicado. Cada una de estas ternas está referenciada a su sistema de coordenadas relativo, situado en el centro del sensor. Como se ha comentado previamente, para el seguimiento de los cálculos, se emplean dos sensores láser. El primero de ellos, situado en el techo del vehículo, mantendrá su sistema de referencia como el absoluto del sistema y sólo se calcularán las transformaciones geométricas necesarias para trasladar y rotar el sistema de referencia del LiDAR posicionado en la parte frontal. Por ello, una vez identificadas las direcciones principales en cada nube de puntos, se pasa a comprobar el emparejamiento correcto entre ternas.

En primer lugar, cada una de las direcciones predominantes del sistema referencia móvil se rota hasta hacerlo coincidir con cada una de las direcciones predominantes del sistema fijo. Esto da lugar a una combinación de nueve emparejamientos. La comparación geométrica se realiza dos a dos, de manera que se pueda cuantificar la diferencia angular entre los dos sistemas de referencia. Por tanto, es necesario definir una métrica que pondere el error cometido en esta relación, para ello se obtiene el coeficiente de $error_{emparejamiento}$ por cada emparejamiento:

$$error_{emparejamiento} = \sum_{i=1}^3 ||\vec{a}_i| - |\vec{b}_i|| \quad (2.1)$$

Donde $\vec{a}_i$ y $\vec{b}_i$ son los vectores comparados de las direcciones predominantes del sistema de referencia móvil y fijo, respectivamente. Se añade el valor absoluto para comparar la dirección de los vectores independientemente del sentido de estos. Finalmente, se tomará el emparejamiento de ternas de vectores que minimice este coeficiente.

Una vez obtenida la relación entre ternas de vectores, se procede al cálculo de ángulos entre los vectores de cada sistema de referencia y posteriormente se comparan entre sí. En este cálculo, si se considera que una dirección no es lo suficientemente similar a la de la otra terna, se descarta y se selecciona la siguiente dirección predominante obtenida previamente mediante el método de la esfera. Este proceso es iterativo hasta encontrar una similitud entre ángulos suficiente.

En la práctica, las direcciones predominantes del entorno suelen ser perpendiculares entre sí, ya que estas suelen estar definidas por suelo, fachadas de edificios alrededor, vehículos o elementos verticales del entorno. Por ello, se ha optimizado este algoritmo para beneficiar aquellos ángulos que formen 90° entre sí mediante la ponderación con los siguientes coeficientes:

$$coeficiente_1 = 1 - \left| \frac{\arccos\left(\frac{\vec{v}_1 \cdot \vec{v}_2}{\|\vec{v}_1\| \cdot \|\vec{v}_2\|}\right)}{\frac{\pi}{2}} - 1 \right| \quad (2.2)$$

$$\text{coeficiente } 2 = 1 - \left| \frac{\arccos\left(\frac{\vec{v1} \cdot \vec{v3}}{\|\vec{v1}\| \cdot \|\vec{v3}\|}\right)}{\frac{\pi}{2}} - 1 \right| \quad (2.3)$$

$$\text{coeficiente } 3 = 1 - \left| \frac{\arccos\left(\frac{\vec{v2} \cdot \vec{v3}}{\|\vec{v2}\| \cdot \|\vec{v3}\|}\right)}{\frac{\pi}{2}} - 1 \right| \quad (2.4)$$

Donde $\vec{v1}$, $\vec{v2}$ y $\vec{v3}$ son los vectores de las direcciones predominantes asociados a un sistema de referencia. Además, se define un coeficiente como métrica asociada a la geometría implícita en la terna como la multiplicación de ellos tres:

$$\text{coeficiente } 1 = \text{coeficiente } 1 \cdot \text{coeficiente } 2 \cdot \text{coeficiente } 3 \quad (2.5)$$

Por último, se compara este coeficiente entre los dos sistemas de referencia, de tal manera que, si uno de ellos difiere más de un 30% del otro, se estudian los coeficientes individuales y se descartaría la dirección que obtiene un valor atípico.

Este cálculo no se considera suficientemente robusto como para ser la solución definitiva del cálculo de orientación relativa entre nubes de puntos del algoritmo. Aun así, sirve como primera aproximación de las siguientes fases del algoritmo y, además, se utilizará como resultado si en etapas posteriores existe falta de información.

c) Cálculo de las intersecciones entre planos como invariante del entorno

Anteriormente se ha mencionado la problemática de la información proporcionada por los sensores LiDAR para definir correctamente el contorno o puntos específicos de los elementos del entorno, lo que afecta desfavorablemente el cálculo de una calibración. Esta problemática conlleva buscar una invariable del entorno fiable para poder obtener una calibración correcta.

Esta fase del algoritmo parte de haber identificado las tres direcciones principales de cada nube de puntos y una primera aproximación de cómo se disponen una respecto a la otra. Cada dirección predominante tiene asociada un conjunto de superficies cuyo vector normal coincide. En primer lugar, se realiza un filtrado de aquellas superficies que puedan introducir ruido para el desarrollo del algoritmo (poca densidad de puntos, áreas poco relevantes, etc.).

A continuación, conociendo los vectores normales de las superficies y, por tanto, las ecuaciones que definen sus planos, se calculan las intersecciones entre ellos. Estas intersecciones se consideran el elemento invariable del entorno ya que corresponderán a puntos virtuales que no dependen de la información dispersa proporcionada por el láser. Esta información crea una huella del entorno, a partir de la cual se puede realizar una calibración precisa ya que se elimina la influencia del posicionamiento de los sensores.

Partiendo de la primera aproximación de la fase anterior, se pueden orientar ambos

conjuntos de intersecciones y establecer similitudes geométricas entre ellos para, así, definir la correspondencia entre nubes de puntos.

Estas similitudes geométricas están relacionadas con la posición relativa entre, al menos tres, intersecciones de cada nube de puntos. Por ello, el método perderá precisión si el número de intersecciones es menor, en cuyo caso, se utilizará la información de la aproximación de la fase anterior. Tras un primer filtrado de aquellas intersecciones que por su distancia al láser no proporcionen suficiente precisión, se establecen dos coeficientes que ponderan el posicionamiento relativo entre intersecciones de cada nube: un primer coeficiente según la longitud de los segmentos que une cada intersección a las dos más cercanas y un segundo coeficiente que tiene en cuenta la distancia de cada intersección a su láser correspondiente. Mediante estos dos coeficientes es posible crear las correspondencias de similitud. En el caso de la mejor correspondencia obtenida, se hace coincidir uno de sus puntos y se realizan dos rotaciones para tener finalmente el solapamiento entre conjunto de intersecciones.

En este paso, es posible comprobar la coherencia del resultado imponiendo dos restricciones: el valor de la translación total, obtenida al hacer coincidir las intersecciones con mejor correspondencia, no debe ser mayor a cuatro metros, ya que, al estar los láseres posicionados en el vehículo, esta distancia no debe ser mayor que las propias dimensiones de este. Por otro lado, la rotación total (aquella de la primera aproximación de la fase anterior y la calculada en el solapamiento de intersecciones) no debe ser mayor a 90° , ya que la información en tal caso sería la misma que se obtiene en una simetría central. En caso de no cumplir con alguna de estas restricciones, se vuelven a realizar los cálculos para la siguiente mejor correspondencia obtenida.

d) Emparejamiento de superficies reales obtenidas por los sensores

A partir del emparejamiento de los puntos virtuales entre las nubes de puntos de cada láser, es posible calcular las dos rotaciones necesarias para que ambas direcciones predominantes sean paralelas. A partir de este ajuste, es posible establecer el *matching* entre superficies reales equivalentes de cada nube.

Mediante la comparación de características geométricas se seleccionan las parejas de planos que mejor correspondan entre sí por cada dirección predominante. Para ello, se comparan dos a dos los planos en direcciones equivalentes teniendo en cuenta el área de las superficies, la ecuación que define su plano, el número de puntos pertenecientes a estos y la posición del centroide.

Finalizadas todas las asignaciones entre parejas de planos, se seleccionan aquellas que mejor afinidad hayan tenido por cada dirección, de tal manera que se obtienen tres parejas de planos con mayor probabilidad de haber sido asignadas correctamente como se muestra en el ejemplo de la figura 2.7.

Figura 2.7: Emparejamiento de planos equivalentes entre nubes de puntos.

Estas tres parejas se utilizan para el cálculo del ajuste final de translación y rotación que se debe aplicar a la nube de puntos con el sistema de referencia móvil (figura 2.8). Se realizan dos rotaciones que acoten 3 gdl, a partir del vector normal de los planos emparejados (figura 2.8.a y figura 2.8.b). Una vez se encuentran paralelos entre sí, se calcula la intersección entre los tres planos de cada uno de los láseres para, posteriormente, obtener el valor final de translación como la diferencia entre las posiciones de dichas intersecciones. Tras este cálculo, los láseres comparten el sistema de referencia (figura 2.8.c).

Figura 2.8: Ajuste final de las nubes de puntos. a) Primera rotación. b) Segunda rotación, con la que se obtienen superficies paralelas. c) Ajuste de la translación.

e) Cálculo del error

El error se calcula en base será la distancia paralela entre superficies. Para ello, tras la fusión de las nubes de puntos se calcula de nuevo la ecuación del plano correspondiente a las superficies semejantes. Como se hizo en anteriores fases del algoritmo, se calcula la rotación necesaria para que ambos planos sean paralelos. Una vez dispuestos en la nueva situación, es posible calcular la distancia paralela entre planos como el error cometido por el algoritmo.

El cálculo del error servirá como valoración del resultado obtenido y conocer si los resultados son fiables. Según el valor obtenido de error, este se realimenta al algoritmo (figura 2.8.d) para comenzar un proceso iterativo. Este proceso iterativo minimiza la distancia de translación y trata de hacer coincidir los planos los planos emparejados. El algoritmo concluye con este último resultado cuando el error está por debajo de un umbral preestablecido y se estima la precisión de la calibración.

2.1.3. Análisis de los límites del algoritmo de calibración

La metodología para la calibración extrínseca descrita anteriormente ha sido implementada y comprobada en una serie de ensayos utilizando datos reales con dos sensores LiDAR de similares características. Como se mencionaba anteriormente, esta solución sería extrapolable a sensores de diferentes características y con un *set-up* que implicara más de dos láseres.

Los ensayos realizados se dividen en dos categorías principales: entornos interiores y entornos exteriores, tal como se observa en la tabla 2.1. El objetivo es analizar el rendimiento del algoritmo en distintos escenarios que estén comprometidos por la complejidad del entorno y la cantidad de información útil disponible. Por un lado, la complejidad del entorno será mayor en entornos interiores, donde la mayor cantidad de elementos dará lugar a información difusa (ruido) y mayor dificultad para el *matching* entre elementos característicos. Por otro lado, la disponibilidad de información útil será más escasa en lugares exteriores debido a la menor cantidad de elementos geométricos planos, por tanto, en este caso, se verá afectada la etapa de identificación.

Tabla 2.1: Tipos de escenarios de prueba para calibración.

	Escenarios interiores	Escenarios exteriores
A
B

En ambos tipos de escenarios, se ha realizado la adquisición de datos situando un LiDAR en el techo, que será tomado como el sensor de referencia fijo, y otro LiDAR situado en la parte frontal del vehículo, que será el sensor cuyo sistema de coordenadas se trasladará (figura 2.9).

Figura 2.9: Posición de los sensores LiDAR en el vehículo de prueba. a) Sensor cuyo sistema de referencia se toma como fijo. b) Sensor cuyo sistema de referencia será movable.

Cada ensayo incluirá valores distintos de rotación en torno a los ejes del sistema de coordenadas del sensor fijo, siendo X el eje transversal al vehículo, Y el eje longitudinal y Z el eje vertical. El resumen de los ensayos realizados se muestra en la tabla 2.2.

Tabla 2.2: Resumen de ensayos para calibración.

Ensayo	Escenario	Rotación ideal (°)		
		X	Y	Z
1	Interior A	0	0	0
2	Interior B	0	0	0
3	Exterior A	0	0	0
4	Exterior B	0	0	0
5	Interior A	-15	0	0
6	Interior A	0	-30	0
7	Interior B	0	-30	0
8	Interior A	0	-90	0
9	Exterior A	0	-90	0

Los ensayos propuestos, pretenden incrementar gradualmente la complejidad y poner en compromiso la robustez del método. Los primeros ensayos se realizan sin ninguna rotación para cada uno de los cuatro escenarios, donde se analiza cómo afecta el nivel de información útil del entorno para extraer un valor coherente. Por otro lado, en los ensayos siguientes se busca el límite de funcionamiento correcto del algoritmo con valores de rotación cada vez mayores. Esto resulta interesante a la hora de diseñar un nuevo *set-up* de los sensores sobre el vehículo. Conociendo los límites del método, se podrá seleccionar un mayor rango de posibilidades.

A continuación, se presentan en la tabla 2.3 los resultados para cada uno de los ensayos propuestos, indicando en rojo aquellos valores calculados que divergen del valor esperado:

Tabla 2.3: Resumen resultados de rotación y translación por cada ensayo.

Ensayo	Escenario	Rotación ideal (°)			Rotación calculada (°)			Translación calculada (m)		
		X	Y	Z	X	Y	Z	X	Y	Z
1	Interior A	0	0	0	-3,3	-1,5	-0,9	0,436	2,490	-1,233
2	Interior B	0	0	0	-2,9	-1,2	-1,8	0,250	2,445	-1,101
3	Exterior A	0	0	0	-2,6	-1,3	-1,9	0,418	2,287	-1,311
4	Exterior B	0	0	0	-3,1	-1,6	-1,3	0,628	1,985	-1,148
5	Interior A	-15	0	0	-15,5	-0,5	-3,5	0,151	2,520	-1,054
6	Interior A	0	-30	0	-5,4	-60,8	-12,7	4,854	1,355	7,057
7	Interior B	0	-30	0	-11,0	-61,4	-0,9	8,304	0,967	14,015
8	Interior A	0	-90	0	157,0	-4,0	-23,2	6,466	9,499	-0,416
9	Exterior A	0	-90	0	-17,5	-86,9	-29,8	0,368	2,404	-1,295

Cabe destacar que el algoritmo funciona correctamente tanto entornos interiores como exteriores cuando los ángulos de rotación son inferiores a 30°. En un análisis pormenorizado de cada ensayo, se ha observado que existe una correlación entre los resultados obtenidos por este método y el número de planos emparejados correctamente. En este sentido, se puede observar en la tabla 2.4, el número de emparejamientos de planos identificados en cada escenario, así como el porcentaje correctos de estos.

Tabla 2.4: Número de emparejamientos de planos en cada dirección principal.

Ensayo	Escenario	Planos emparejados			De los cuales correctos
		Dirección I	Dirección II	Dirección III	
1	Interior A	4	2	1	86%
2	Interior B	5	4	1	70%
3	Exterior A	1	1	1	100%
4	Exterior B	1	1	1	33%
5	Interior A	2	5	1	75%
6	Interior A	2	3	3	0%
7	Interior B	2	4	1	0%
8	Interior A	1	1	1	33%
9	Exterior A	1	1	1	67%

El número de planos identificados es significativamente mayor en entornos interiores, como era de esperar. En entornos exteriores, se produce una compensación entre el número de planos identificados y la información útil extraída. De tal forma que, aunque sean tres los planos identificados, el resultado de la calibración es correcto. Por el contrario, en escenarios interiores, al existir una infraestructura con mayor número de superficies planas, se identifican mayor número de ellas, sin embargo, el emparejamiento correcto es más complejo, siendo nulo en los ensayos donde el ángulo de rotación es de 30°.

Para una mayor comprensión de los errores producidos, se describen en profundidad algunos de los ensayos realizados:

Ensayo 1 - Interior A:

En azul, se representa la nube de puntos obtenida por el sensor fijo, situado en el techo del vehículo. En rojo, se observan los puntos recogidos por el láser posicionado en el frontal del vehículo. Este ensayo consta de únicamente una translación, no existe una rotación significativa en torno a los ejes del sistema de referencias del sensor fijo (en azul).

La situación inicial con las dos nubes de puntos descentradas, cada una con su sistema de coordenadas local, se muestra en la figura 2.10:

Figura 2.10: Situación inicial nubes de puntos: en azul, asociada al sistema fijo; en rojo, asociada al sistema móvil.

Los planos identificados y relacionados correctamente en este ensayo han sido los que se pueden observar en la figura 2.11:

Figura 2.11: Planos identificados en cada dirección predominante en Ensayo 1.

Se observa como en la dirección II, la identificación de pequeños fragmentos de planos en dichas direcciones introduce un ruido que afectará al posterior *matching*. Sin embargo, en la dirección III, la información del único plano identificado es más fiable debido a la uniformidad de la información.

Por tanto, el ajuste de los parámetros de calibración hace que las dos nubes de puntos se superpongan, tal y como se muestra en la figura 2.12:

Figura 2.12: Situación final nubes de puntos Ensayo 1.

Ensayo 3 - Exterior A:

El ensayo 3 es interesante al obtener un buen resultado pese a identificarse solamente tres planos, uno en cada dirección principal. Las nubes de puntos originales en este caso son las representadas en el figura 2.13:

Figura 2.13: Situación inicial nubes de puntos Ensayo 3.

La poca infraestructura del entorno hace que la etapa de identificación del algoritmo sea más complicada. Sin embargo, se consigue identificar correctamente un plano por cada dirección

y emparejarlo con el correspondiente del otro sensor, por lo que no es necesario mayor nivel de información (figura 2.14).

Figura 2.14: Planos identificados en cada dirección predominante en Ensayo 3.

Tras la obtención de una información consistente, las dos nubes de puntos quedan superpuestas de forma correcta (figura 2.15).

Figura 2.15: Situación final nubes de puntos Ensayo 3.

Ensayo 6 - Interior A:

En este ensayo se prueba una rotación de 30° alrededor del eje Y, paralelo al eje longitudinal del vehículo. Esta rotación dará lugar a que el sensor rotado, capte parte del techo del escenario interior. La situación inicial es la que se presenta en la figura 2.16.

Figura 2.16: Situación inicial nubes de puntos Ensayo 6.

Como consecuencia de esta orientación, la complejidad para calcular las direcciones predominantes es mayor, aunque la cantidad de superficies planas es elevada. En concreto, este ensayo resulta interesante ya que se realizan 8 emparejamientos de planos en total, pero ninguno de ellos correctamente, tal como se observa en la figura 2.17.

Figura 2.17: Planos identificados en cada dirección predominante en Ensayo 6.

A la vista de los emparejamientos realizados incorrectamente en este ensayo, se puede extraer que la causa es el error cometido en las relaciones entre direcciones predominantes de cada láser. Es decir, tras una mala asociación de esas direcciones, el posterior *matching* será incorrecto ya que se restringe a los planos obtenidos en la supuesta misma dirección.

Además, la asociación errónea entre las direcciones da lugar a unos resultados de translación incoherentes. Las translaciones calculadas son mayores que las propias dimensiones del vehículo. Por tanto, conociendo estas dimensiones máximas se podría forzar a seguir calculando el emparejamiento correcto entre direcciones, siendo esta la auténtica razón por la

que se ha obtenido una solución incorrecta. El resultado final se puede observar en la figura 2.18, donde se aprecia que la asociación entre direcciones predominantes es el origen del error cometido por el algoritmo.

Figura 2.18: Situación final nubes de puntos Ensayo 6.

A la vista de los resultados, en el sistema de coordenadas del sensor fijo, la dirección III corresponde con la normal al suelo, mientras que, en el sensor rotado, se ha identificado la dirección normal al suelo como la dirección I.

Ensayo 9 - Exterior A:

La rotación de 90° en torno a un eje corresponde con la situación más desfavorable para realizar el emparejamiento entre planos característicos (figura 2.19). Una rotación por encima de este ángulo recibe la misma información del entorno que la posición simétrica central del láser.

Figura 2.19: Situación final nubes de puntos Ensayo 9.

De las nubes de puntos anteriores, se pudo relacionar tres planos, uno por cada dirección principal. Sin embargo, la poca información útil de uno de esos planos causa un error en el cálculo de las rotaciones. En concreto, la dirección I, corresponde a la extraída del plano correspondiente al suelo (figura 2.20). En este caso, la información del láser rotado es tan escasa que dificulta el ajuste con el plano correspondiente en la nube de puntos del sensor fijo.

Figura 2.20: Planos identificados en cada dirección predominante en Ensayo 9.

A pesar de ello, al representarlos, se observa que la relación entre las direcciones predominantes ha sido correcta. Sin embargo, los valores de rotación en los ejes X, Y, Z, no son los esperados. Mientras que, desde la perspectiva del láser fijo, existe gran cantidad de puntos pertenecientes al suelo; en el láser rotado, dicha información se encuentra concentrada en un área muy pequeña. Este hecho da lugar a que el ajuste al plano correspondiente al suelo sea

robusto en el láser fijo al tener una información más distribuida en el espacio. Por otro lado, el ajuste al plano del suelo con un área tan concentrada da lugar a una mayor sensibilidad a ruido, lo que repercute en la solución calculada (2.21).

Figura 2.21: Situación final nubes de puntos Ensayo 9.

Por otro lado, los valores de translación obtenidos son coherentes y dentro del orden de magnitud esperado. Sin embargo, aunque las relaciones entre direcciones predominantes hayan sido correctas y el posterior emparejamiento de planos también, la precisión de los cálculos no es suficiente debido, principalmente, a la causa descrita anteriormente.

2.1.4. Conclusiones finales

Uno de los primeros pasos en la automatización de un vehículo y de vital importancia para una correcta fusión sensorial es la calibración extrínseca entre sensores. En este contexto, se ha desarrollado un nuevo algoritmo para esta calibración de forma automática de sensores LiDAR. El método propuesto no está limitado en el número de láseres utilizados ni las características de estos, aunque para la descripción de la metodología y análisis de resultados se haya aplicado para la calibración entre dos sensores. Además de la polivalencia del método, cabe destacar que se trata de un método automático, que no necesita una calibración manual previa, ni elementos utilizados para este fin como patrones de calibrado. El algoritmo realiza el reconocimiento de elementos característicos, concretamente superficies planas, y, posteriormente, su emparejamiento, minimizando el error.

Este procedimiento ha sido evaluado en diferentes escenarios y posiciones. Los ensayos se han realizado en entornos interiores, donde el algoritmo identificará un mayor número de elementos característicos que en entornos exteriores, donde se dispone de menor grado de infraestructura alrededor. Sin embargo, los resultados arrojan un buen rendimiento tanto

en un escenario como en otro ya que, aunque en el exterior, la información de geometrías planas es más escasa, esta información es útil para un correcto emparejamiento entre los sensores. Durante los ensayos, se han comprobado los límites operativos del algoritmo a través de diferentes posiciones desfavorables de uno de los LiDAR. A partir de los resultados obtenidos, se puede concluir que se logra una zona operativa de funcionamiento correcto de hasta 30° de rotación.

El método realiza una fusión de bajo nivel, donde las dos nubes de puntos quedan referenciadas al mismo sistema de referencia. Las desviaciones obtenidas en los resultados, sin embargo, no son significativas para buena fusión de alto nivel de los obstáculos que puedan presentar un riesgo para la conducción.

La configuración final del algoritmo es consecuencia de las evoluciones durante el desarrollo. Por un lado, el método para buscar las relaciones correctas entre direcciones predominantes de los dos LiDAR, se basaba en primera instancia en la comparación de propiedades geométricas de los planos. Esta comparación de las propiedades de los planos identificados por cada láser se realizaba dos a dos, lo que conllevaba una gran carga computacional. Además, esta forma de proceder no era consistente ya que, debido a oclusiones, un mismo plano visto desde dos perspectivas diferentes, puede tener características geométricas distintas. Para mejorar esa estabilidad en los cálculos se optó por cambiar de estrategia, basando el nuevo método en el cálculo de intersecciones de planos de ambas nubes de puntos. Con ello, se pretende poder extraer una invariante del entorno que pueda ser relacionada desde distintos sistemas de coordenadas. Este nuevo procedimiento, además de determinar la correspondencia entre direcciones predominantes de ambos láseres, obtiene la primera aproximación para el solapamiento de las nubes de puntos.

El algoritmo presentado en esta sección está orientado a la calibración entre sensores LiDAR. Sin embargo, ciertas etapas dentro de la metodología seguida, se puede aprovechar para otros enfoques. De esta forma, las etapas de identificación, segmentación, *matching*, servirán como base para posteriores algoritmos desarrollados para caracterización de infraestructura, detección de obstáculos, técnicas de posicionamiento y localización del vehículo, que se exponen más adelante en este documento.

2.2. Detección de obstáculos

Unos de los primeros temas que se tratan en navegación autónoma es la identificación de obstáculos y zonas transitables. En vehículos terrestres se requiere un conocimiento muy preciso de estos factores para garantizar una conducción segura. Generalmente se pueden distinguir dos tipos de entornos. Por un lado, zonas urbanas e interurbanas, caracterizadas por un alto grado de infraestructura: edificios, señales verticales, límites de la calzada, bordillos, etc. Por otro lado, zonas *off-road*, por lo general ofrecen una orografía irregular y carecen de elementos de referencia, por lo que suponen entornos complejos para extraer el conocimiento preciso que requiere el vehículo autónomo.

En este apartado se aborda el desarrollo de sistemas de detección de obstáculos y se analiza cuáles son los límites operativos de algoritmos basados en LiDAR 3D que garanticen un comportamiento robusto y preciso. Además, la fusión sensorial desarrollada en el apartado

2.1, tiene aplicación directa para poder integrar los desarrollos individuales aquí presentados, mejorando la fiabilidad de los resultados.

Partiendo de desarrollos anteriores con un láser escáner 2D (Jiménez y Naranjo, 2011; García et al., 2013), algunos conceptos han sido extrapolados o adaptados en los algoritmos propuestos. Por otra parte, debe tenerse en cuenta que, mientras que la mayoría de las soluciones aportadas en la literatura están centradas en entornos urbanos, el vehículo autónomo podrá circular por zonas menos estructuradas enfrentándose a la problemática implícita en estos terrenos. En este sentido, los algoritmos para entornos *off-road* poseen ciertas características diferenciadoras y peculiaridades respecto a la detección en entornos urbanos o de carretera, aunque puedan ser adaptados en algunas ocasiones. Se plantean, por tanto, algoritmos basados en sensores láser 3D para ambos entornos, que permitan ampliar las capacidades de los vehículos autónomos. De la misma forma, en apartados posteriores, se incluyen métodos de caracterización del entorno que amplían la comprensión del entorno y cuya implementación es complementaria a la desarrollada en este apartado.

Una metodología general característica de los algoritmos de detección de obstáculos mediante LiDAR atiende a la siguiente estructura:

- Definición de la región de interés (Region Of Interest - ROI): Cada aplicación requiere de una región de análisis y cálculo. Los sensores LiDAR suelen proporcionar una nube de puntos del entorno con amplios rangos de visión, por lo que esa gran cantidad de datos puede suponer un problema de carga computacional. Para aligerar esa tasa de información y centrar los cálculos en la zona más relevante, debe definirse un ROI. El ROI seleccionado dependerá de la aplicación para la que esté centrado y dónde se encontrará la información relevante para cálculos posteriores, de este modo, puede definirse con límites constantes, cuando la información siempre se encuentra en una misma localización; o bien, definiendo esta región en función de alguna variable, por ejemplo, la distancia de anticipación conociendo la velocidad del vehículo (Broggi et al., 1999).
- Identificación de características: Esta etapa suele configurarse como una serie de filtros secuenciales que extraen las características o patrones identificativos del objeto en cuestión. Se suelen utilizar funciones específicas de cálculo con nubes de puntos (Point Cloud Library, PtCloud para Matlab, etc.). Este tipo de librerías ofrecen funciones específicas para el tratamiento de datos en forma de nubes de puntos desordenadas y facilitan la integración en el desarrollo. Por otro lado, también el uso de variables estadísticas, regresiones o ajustes paramétricos son herramientas comúnmente utilizadas para la etapa de identificación.

Como es de esperar, esta etapa suele presentar facilidades cuanto mayor sea el grado de infraestructura en el entorno, así en entornos urbanos, las formas geométricas de los elementos facilitarán la extracción de características. Opuestamente, en terrenos *off-road*, la estrategia deberá ser distinta por la carencia de estas características y se podrán basar en el estudio entre de puntos cercanos o incluso por cada capa láser.

- Segmentación de datos: Tras un análisis y extracción de las características, los objetos del entorno quedan identificados, sin embargo, para poder realizar posteriormente un seguimiento del estado de estos, es necesario clasificarlos previamente. Por un lado, se recurre a algoritmos de *clustering*, los cuales agruparán, según un criterio específico, aquellos puntos que pertenezcan a un mismo obstáculo. Si, por ejemplo, se conoce a priori el número de obstáculos u elementos que se deben identificar, se podrán utilizar métodos de *clustering* no jerárquicos como K-means (Jain y Dubes, 1988) o K-medoids (Hastie et al., 2009). Por el contrario, si el número de objetos es variable y no se puede estimar con antelación, se pueden utilizar algoritmos como DBSCAN (Density-based spatial clustering of applications with noise), basado en la densidad y distancia euclídea entre puntos distribuidos en un espacio (Ester et al., 1996), una generalización del este llamado OPTICS (Ankerst et al., 1999) o algoritmos como Mean-Shift (Geron, 2019), aunque en este último suele dividir un mismo *clusters* en distintos agrupamientos.

Seguidamente del *clustering* de datos, puede integrarse una sub-etapa de asociación de datos. Debido a oclusiones producidas en la nube de puntos, un elemento próximo puede hacer sombra sobre uno situado más lejano, dividiéndolo en dos o más *clusters*. Para identificar cada elemento como la parte integral de los *clusters* que lo conforman y dar consistencia a la etapa posterior de *tracking*, se puede emplear algoritmos de seguimiento de hipótesis múltiples (Multiple Hypothesis Tracking, MHT (Blackman, 2004; Cox y Hingorani, 1996)) o el denominado *virtual-scan* desarrollado en Petrovskaya y Thrun (2009).

- Tracking de obstáculos: El punto de partida para esta etapa son los obstáculos identificados y clasificados. En esta etapa, se realiza un análisis del comportamiento de esos obstáculos a lo largo del tiempo para, en primer lugar, relacionar los *clusters* actuales con los observados en el *frame* anterior, y analizar si suponen un peligro potencial para la conducción o no. Además, el *tracking* permite discernir entre obstáculos estáticos y dinámicos y si estos, a su vez, se desplazan en el mismo sentido que el vehículo en estudio. Por lo general, se suelen realizar una caracterización geométrica de cada uno de los obstáculos anteriores para conocer parámetros identificativos que sirvan para realizar el *matching* entre diferentes instantes de tiempo.
- Resultado: Por último, dependiendo de la finalidad del algoritmo de detección, se definirá una salida del sistema. Esta salida podrá ser utilizada como punto de partida para un algoritmo de toma de decisiones, o bien realizar un *path planning*, calcular una trayectoria alternativa, o bien, emitir una señal de alarma a un posible operador o supervisor de la conducción.

La figura 2.22 ilustra cada una de las etapas descritas anteriormente en un algoritmo genérico de detección de obstáculos con LiDAR 3D. Puede apreciarse que tras una selección del ROI, la identificación y la etapa de segmentación de los obstáculos, quedan caracterizados, dentro de un prisma rectangular, varios peatones y un conjunto de paredes que se encuentran en el entorno.

Figura 2.22: Etapas genéricas para detección de obstáculos basada en LiDAR 3D. a) Nube de puntos del entorno; b) aplicación del ROI; c) identificación y segmentación de obstáculos; d) caracterización de obstáculos.

La metodología anteriormente descrita corresponde con un proceso estándar, según sea la aplicación o el entorno, pueden variar y/o adaptarse para conseguir la precisión adecuada.

En términos geométricos, se puede afirmar que existen dos categorías de obstáculos: positivos y negativos, según sea su altura respecto al nivel del suelo. En entornos urbanos e interurbanos, los obstáculos que aparecen con mayor frecuencia son positivos, como pueden ser peatones, otros vehículos, ciclistas, señales verticales, etc. Si bien pueden aparecer también obstáculos negativos, como baches o grietas en la calzada, estos últimos serán críticos en *off-road*. En zonas todoterreno, también podrán aparecer obstáculos positivos como árboles, arbustos, rocas, etc. Sin embargo, respecto a los obstáculos negativos, tal y como se indicaba, serán críticos en este tipo de escenarios debido a la irregularidad geométrica y tamaño que pueden presentar. Este tipo de obstáculos se dará con mayor frecuencia y podrán ser de tipo zanja, bache o surcos profundos.

Cabe destacar que, mientras para la detección de obstáculos positivos existen diversos desarrollos realizados utilizando algoritmos estándar, la detección de obstáculos negativos requiere de un tratamiento específico, más aún cuando se utiliza tecnología LiDAR debido al propio funcionamiento del láser. Estos obstáculos pueden presentar occlusiones o falta de información acerca del aspecto y dimensiones del obstáculo. Esta dificultad añadida implica elaborar metodologías con un enfoque diferente para incrementar la fiabilidad puesto que las utilizadas normalmente para obstáculos positivos, son difícilmente adaptables.

En este apartado se proponen soluciones particulares para ambos tipos de escenario. Para

entornos urbanos o interurbano, la detección de obstáculos positivos será más frecuente y se plantea una metodología versátil que, además de realizar correctamente la identificación de obstáculos, podrá servir con otro fin distinto, como se desarrollará en el apartado 3.1, donde partiendo de la caracterización de este algoritmo se describe un método para el apoyo al guiado autónomo del vehículo.

Por otro lado, se plantea una solución específica para la detección de obstáculos en entornos *off-road* donde, como se resaltaba, será crítico detectar los obstáculos negativos. Al tratarse de una solución particular donde se centra en el ámbito del terreno todoterreno, deberá de seguir garantizando su aplicabilidad en la identificación de obstáculos positivos.

2.2.1. Detección de obstáculos positivos basada en LiDAR 3D

Como se ha mencionado anteriormente, la primera y más relevante aplicación de un sensor LiDAR 3D es la detección de obstáculos, en concreto, aquellos definidos como positivos. En cualquiera de los grados de automatización que se encuentre el vehículo, es evidente que la detección de obstáculos es de vital importancia para una conducción automatizada segura. Este hecho justifica la principal línea de investigación consolidada para los láseres escáner.

La metodología general para abordar este tipo de desarrollos puede basarse en una identificación mediante la extracción de características como en Gidel et al. (2010). Este enfoque es apropiado cuando existen formas geométricas patentes en el entorno como, por ejemplo, la superficie plana del suelo, paredes de edificios cercanos, barreras de seguridad delimitadoras, biondas, etc. En la mayoría de estos casos, el emparejamiento de características del entorno con formas geométricas predefinidas se basa en variantes del algoritmo iterativo RANSAC (Torr y Zisserman, 2000). Si, por el contrario, el entorno se caracteriza por formas complejas, irregulares o sin una apariencia acorde a un patrón, es necesario aplicar un enfoque distinto. De este modo, se pueden encontrar en la literatura métodos que utilizan mapas de ocupación (*grid-occupancy maps*) como el que se presentó en Montemerlo et al. (2009), que utilizaron en su vehículo autónomo para la competición DARPA Urban Challenge (Buehler et al., 2009). Esta solución les permitió descartar objetos situados por encima del vehículo que no suponían un peligro. En Quehl et al. (2019), recurren igualmente a estos *grid maps* para estimar el aspecto y el movimiento de objetos 3D, estudiando la variación de la dispersión entre puntos láser. Con un enfoque basado en *grid maps* multicapa, en Wirges et al. (2019), desarrollan un algoritmo de detección de obstáculos utilizando técnicas de *Deep Learning*. A pesar de conseguir buenos resultados en general, la incertidumbre de la detección aumenta cuando se trata de objetos de menor tamaño, como peatones o ciclistas. Otros desarrollos, beneficiándose de la evolución de los LiDAR 3D, realizan una extracción de características mediante *voxels* (píxeles 3D) como en Asvadi et al. (2016), logrando discernir entre obstáculos estáticos o dinámicos mediante el estudio de la variación de densidad de puntos en cada *voxels*. De igual forma, existen otros métodos más sofisticados de extracción de características como es el caso de Han et al. (2012), donde se calculan segmentos rectilíneos en coordenadas polares para hacer posible la detección de obstáculos de cualquier aspecto.

A pesar de que los algoritmos anteriormente citados suelen conseguir un buen rendimiento general, suelen requerir utilizar modelos LiDAR con altas prestaciones. Estos modelos suelen tener mejor resolución vertical, que se traduce en un mayor número de capas láser. Esto

puede suponer dos limitaciones principalmente. Desde el punto de vista económico, estos sensores suelen ser más costosos a medida que se incrementa el número de capas, incluso en algunos casos duplicar el coste de un LiDAR con menores prestaciones. Por otro lado, el uso de un solo sensor de estas características da lugar a una configuración rígida del *set-up* sensorial sobre el vehículo, y que, a veces, esta configuración no sea la más conveniente para la aplicación desarrollada. Por tanto, para reducir estas limitaciones puede resultar ventajoso incluso recurrir a la fusión sensorial entre dos LiDAR 3D con menores prestaciones al ser más económico que un solo LiDAR de alta resolución y, además, permitir adaptar mejor el *set-up* sobre el vehículo a la aplicación.

Por otro lado, estos algoritmos suelen estar orientados a una aplicación en concreto. Por ello, en este apartado se propone un método que, además de ser operativo con un láser de 16 capas (Velodyne VLP-16), consigue unos resultados robustos. Además, no se enfoca exclusivamente a la detección de obstáculos positivos, sino que, además, esta metodología será reutilizada para posibilitar la caracterización de elementos del entorno en el apartado 3.1, permitiendo mejorar el posicionamiento del vehículo.

2.2.1.1. Descripción del método utilizado en la detección de obstáculos positivos

El algoritmo planteado para la detección de obstáculos positivos se basa en el pipeline general descrito previamente. Además, para mantener una baja carga computacional, una vez que la altura de los objetos identificados no es relevante para los cálculos, la nube de puntos es proyectada sobre el plano XY, ya que el problema se puede abordar a partir de una reconstrucción 2D (Petrovskaya y Thrun, 2009).

En primer lugar, la definición del ROI es definida en función de la distancia de antelación (*look-ahead distance*), tal y como se menciona en Broggi et al. (1999). Esta depende de la velocidad del vehículo y se cuantifica según la ecuación 2.6

$$L(v) = \begin{cases} L_{min} & v < v_{min} \\ \frac{L_{max}-L_{min}}{v_{max}-v_{min}}v + \frac{v_{max}L_{min}-v_{min}L_{max}}{v_{max}-v_{min}} & [v_{min}, v_{max}] \\ L_{max} & v > v_{max} \end{cases} \quad (2.6)$$

Donde L es la distancia de antelación y v la velocidad actual del vehículo. La distancia mínima (L_{min}) ha sido establecida en 5 metros y la máxima (L_{max}) en 50 metros para unas velocidades entre 30 km/h (v_{min}) y 70 km/h (v_{max}). Estos coeficientes han sido establecidos para garantizar un funcionamiento seguro para la detección y teniendo en cuenta la dispersión de puntos respecto a la distancia. Con esta delimitación de la zona de interés, se descartan los puntos no relevantes para cálculos y se reduce la carga computacional.

A continuación, la etapa de identificación de características del entorno suele ser la más representativa de cada algoritmo, ya que será la que define en mayor medida el funcionamiento global del sistema. En este caso, en lugar de realizar un ajuste geométrico a los elementos que pueden aparecer en el entorno, se optado por realizar una triangulación de Delaunay (Anderson et al., 2012) (figura 2.23). Esta triangulación, permite obtener una representación

característica del entorno resaltando las geometrías principales de los elementos que se encuentran en el rango de visión. Con el conjunto de superficies obtenidas es posible realizar un análisis pormenorizado de las características que se quieren extraer.

Figura 2.23: Delimitación del ROI y triangulación de Delaunay.

A partir del mallado anterior, se calcula la ecuación del vector normal a cada triángulo ($\vec{u}_t$) considerando dos de sus lados ($\vec{v}_{t1}$ $\vec{v}_{t2}$) según la ecuación 2.7, obteniendo por tanto el conjunto de vectores normales que define la triangulación (figura 2.24).

$$\vec{u}_t = \frac{\vec{v}_{t1} \times \vec{v}_{t2}}{\|\vec{v}_{t1}\| \times \|\vec{v}_{t2}\|} \quad (2.7)$$

Figura 2.24: Cálculo del vector normal a cada triángulo.

Posteriormente, se calcula la orientación de cada superficie a partir de los cosenos directores para los tres ejes $(\vec{i}, \vec{j}, \vec{k})$. Se obtiene, así, tres ángulos que definen dicha orientación, de los cuáles γ , servirá para comparar el ángulo que forma la superficie con la dirección del eje vertical.

$$\begin{aligned}\alpha_t &= a \cos(\vec{u}_t \cdot \vec{i}) \\ \beta_t &= a \cos(\vec{u}_t \cdot \vec{j}) \\ \gamma_t &= a \cos(\vec{u}_t \cdot \vec{k})\end{aligned}\tag{2.8}$$

Con la información de la orientación de cada superficie, es posible establecer unos umbrales máximo y mínimo que contengan los triángulos (y, por tanto, los puntos que lo definen) en función de la geometría que se quiera extraer. En concreto, para considerar obstáculos positivos, se filtrarán aquellos triángulos cuya orientación con el eje vertical esté comprendida entre 60° y 90° . Por otro lado, debido a que, según la distribución de los puntos, pueden obtenerse triángulos de dimensiones o muy grandes o pequeñas, se establece también un umbral que controle el tamaño de los mismos, con el valor mínimo y máximo que deben tener los triángulos.

Finalmente, a partir del filtro de orientación y de tamaño, los puntos que se obtienen serán aquellos candidatos a ser obstáculos en el entorno (figura 2.25). Para verificar que efectivamente se trata de obstáculos, se debe aplicar la etapa de *clustering* y *tracking*.

Figura 2.25: Filtrado de triángulos según orientación con el eje vertical.

En este punto, la nube de puntos resultante corresponde con aquellos que cumplen con los filtros anteriores. Como no es posible conocer a priori el número de *clusters* que aparecen en el entorno, se hace uso del método de *clustering* DBSCAN que garantiza una buena eficiencia y baja carga computacional. La nube de puntos a segmentar por este método contiene solo las coordenadas X, Y de cada punto, ya que una vez se han identificados los puntos que pueden pertenecer a obstáculos, no se requiere un *clustering* en tres dimensiones.

Tras realizarse la segmentación de los datos, se obtienen los diferentes *clusters* del entorno que representan un volumen positivo en el espacio. Estos quedan caracterizados tras ser incluidos en el prisma rectangular de menor volumen posible (*minimum-volume bounding box*).

La última etapa de este algoritmo corresponde con el *tracking* de los objetos identificados y caracterizados según su *bounding box*. Mediante el seguimiento de los obstáculos durante distintos *frames*, es posible determinar tanto si el obstáculo es estático o dinámico y en tal caso, estimar si su trayectoria puede colisionar con la del vehículo. Para este estudio del comportamiento de los objetos, se han considerado las siguientes variables:

- Cercanía: Este parámetro valora si el centroide de un obstáculo se encuentra en una zona preestablecida para realizar el *tracking*.
- Desplazamiento: Debido al propio funcionamiento del láser, la cantidad de puntos que impactan sobre un objeto no es constante de un *frame* al siguiente incluso cuando es estático. Este hecho da lugar a que el cálculo del centroide no se sitúe siempre en el mismo lugar dentro del *cluster*. Para mitigar esta “vibración” y poder diferenciarla de cuando se trata realmente de un desplazamiento del obstáculo relativo a la posición del láser, se ha definido esta variable. Se establece un valor mínimo de desplazamiento, superado este, se considera el obstáculo para continuar con el *tracking*.
- Acercamiento: Además de considerar que el objeto identificado tiene un movimiento relativo respecto a la posición del láser, se evalúa si se está desplazando en el mismo sentido o en sentido contrario al vehículo.
- Acercamiento en dirección al vehículo: Si el obstáculo identificado cumple todas las anteriores condiciones, se estudia si su aproximación impactase con el vehículo. Para ello, se reduce el volumen del obstáculo (a partir de su *bounding box*) a un punto que coincide con el centroide de este. Este volumen del objeto se considera para aumentar virtualmente el volumen del vehículo, tal y como se realiza en Sezer y Gokasan (2012). Finalmente, se calcula el vector desplazamiento del centroide y su punto de intersección con el nuevo volumen del vehículo. Si se estima que existe intersección entre ambos, se calcula el tiempo para colisión a partir de la observación de dicho obstáculo en *frames* consecutivos.

Tras la etapa de *tracking*, se obtiene una lista de obstáculos caracterizados que se encuentran dentro del ROI. De cada uno de ellos se conoce la posición de su centroide, volumen y el estado de las variables definidas en esta etapa. En caso de que se haya estimado que el objeto pueda colisionar con el vehículo, se añade el valor del tiempo para colisión.

2.2.1.2. Ejemplos de detección de obstáculos positivos en entornos reales

Se han llevado a cabo diferentes ensayos en entornos urbanos utilizando un láser VLP-16, acorde con los requisitos descritos previamente de no tratarse de un láser de altas prestaciones y que, mediante la metodología descrita, se garantiza una baja carga computacional. El láser ha sido ubicado en el techo del vehículo y orientado horizontalmente. En estos escenarios se

puede diferenciar la calzada delimitada por vegetación, coches aparcados o vayas.

En la figura 2.26, se observa el comportamiento general del algoritmo en dos situaciones distintas. A la izquierda, el entorno solo se caracteriza con elementos estáticos, mientras que en el escenario que se muestra a la derecha, aparece un peatón (obstáculo dinámico). En rojo, se destacan aquellos triángulos que cumplen con los umbrales establecidos respecto a la orientación y tamaño. Se observa cómo el método identifica correctamente obstáculos con altura positiva, que suelen aparecer en entornos urbanos, ya sean estáticos o dinámicos. El resto de los triángulos son descartados ya que representan una superficie plana que corresponde con la calzada.

Figura 2.26: Ejemplo de funcionamiento del algoritmo de detección de obstáculos positivos. A la izquierda, se identifican obstáculos estáticos. A la derecha, se identifica, además, un peatón (obstáculo dinámico).

La figura 2.27 muestra otro ejemplo, donde el entorno describe una curva a izquierdas delimitada por una bionda y vegetación. El algoritmo es capaz de discernir los obstáculos de forma robusta a lo largo de toda la región de interés a pesar de la divergencia entre puntos, debido principalmente por la triangulación realizada durante la etapa de identificación.

Figura 2.27: Ejemplo de funcionamiento ante dispersión de puntos del algoritmo de detección de obstáculos positivos.

2.2.2. Detección de obstáculos negativos basada en LiDAR 3D

En este apartado se plantea un algoritmo específico para la detección de obstáculos negativos. Como se destacaba, la literatura acerca de soluciones para obstáculos negativos basados en tecnología láser es más escasa que en caso de obstáculos positivos. Este hecho se fundamenta principalmente en que estos obstáculos son críticos sobre todo en entornos *off-road*, mientras que, para entornos urbanos e interurbanos, estos obstáculos suelen referirse a grietas en el asfalto o baches, que pueden no suponer un peligro crítico para la conducción. Los desarrollos en este sentido están orientados a realizar mantenimiento de calles y carreteras en entornos con terreno pavimentado (Laurent et al., 2009; Q. Li et al., 2009; W. Wang et al., 2011).

Además, los pasos generales planteados para un algoritmo convencional de detección de obstáculos con LiDAR deben ser reconsiderados para una detección de obstáculos negativos. Será más frecuente encontrar este tipo de obstáculos en un entorno *off-road*, donde la cantidad de irregularidades en el terreno requiere adaptarse a estas condiciones. Este tipo de obstáculos entraña una dificultad adicional durante su identificación ya que los sensores, en general, tendrán dificultades para poder inferir la geometría (tamaño, profundidad, etc.) con suficiente antelación para estimar si es potencialmente peligroso. La mayor dificultad que conlleva analizar una zona *off-road* la información limitada que proporcionan los sensores, debido al poco grado de infraestructura que suelen caracterizar estos escenarios. Por un lado, los posibles obstáculos pueden tener cualquier geometría, por lo que no es fácil predecir qué es realmente un peligro y lo que no, por ejemplo, discernir a través de un sensor láser entre una roca o un matorral puede no ser sencillo y que no se consiga una correcta clasificación, como ocurriría con obstáculos más estructurados. Por otro lado, en la mayoría de los casos, existe una baja definición de los límites de las zonas transitables por las que se puede circular, en muchos casos delimitados por vegetación solamente.

Es posible encontrar varios desarrollos que abordan este problema teniendo en cuenta las dificultades mencionadas. Por ejemplo, en Caraffi et al. (2007) se han centrado en detectar obstáculos en zonas *off-road* o con terreno no preparado recurriendo a la fusión entre distintos

sensores de visión artificial. Por otro lado, en Larson y Trivedi (2011) utilizan la tecnología LiDAR para ofrecer varias soluciones a la detección de obstáculos negativos en función de la región de interés definida. Haciendo uso de un LiDAR de altas capacidades, analizan las distancias y ángulos de los puntos LiDAR sobre los obstáculos y se comparan dichos datos reales con los esperados en una superficie plana. A pesar de que la velocidad del propio vehículo es un factor crucial para este método, se obtiene una buena efectividad en los resultados según la distancia vehículo-obstáculo (100%: 6-8 m, 89%: 16-20 m, 78%: 30 m y 31%: 50 m). Por otro lado, en Shang et al. (2016) basan su método en la información de un receptor GPS, una fusión sensorial entre dos LiDAR 3D y un *set-up* sobre el vehículo que permite una mejor definición del terreno. Los resultados que presentan en este trabajo se obtienen a partir de un análisis estadístico de la comparación entre la señal que se obtiene de obstáculos ideales y la que obtienen en el entorno real.

De los trabajos mencionados anteriormente, se puede extraer que para la detección de obstáculos negativos y zonas no transitables en entornos *off-road* se requiere una fusión sensorial o bien sensores de altas prestaciones para obtener una mejor definición del terreno. Sin embargo, en esta tesis se estudiará cuáles son los límites tecnológicos de un *set-up* basado en un solo sensor, de nuevo el modelo VLP-16, acotando la carga computacional. El desarrollo del algoritmo planteado se centra en entornos todoterreno, aunque las soluciones adoptadas son extrapolables a obstáculos negativos en entornos urbanos y, también, seguiría operativo para la identificación de obstáculos positivos. El algoritmo ha sido implementado en un vehículo todoterreno y probado en situaciones reales *off-road* utilizando una configuración del láser sobre el vehículo distinta a la orientación horizontal con el objetivo de mejorar la definición del terreno.

2.2.2.1. *Set-up* del sensor LiDAR para mejorar definición de puntos en el suelo

Debido a la geometría específica de los obstáculos negativos que se deben caracterizar, se replantea la ubicación del sensor láser sobre el vehículo y su orientación. Además, aunque el método propuesto aporta una solución específica para la identificación de obstáculos negativos, deberá ser apropiada para la detección de obstáculos positivos también.

Teniendo en cuenta que la información proporcionada por el LiDAR es dispersa en el espacio, hay varios factores clave a tener presentes para decidir la ubicación. La divergencia entre puntos es proporcional a la distancia según la ecuación 2.9, por lo que este factor influirá en la altura a la que se sitúe el sensor.

$$s = 2 \cdot \sin \frac{\Delta\theta}{2} \cdot d \quad (2.9)$$

Siendo s la divergencia entre puntos adyacentes calculada como la distancia en línea recta entre ellos, d la distancia de los puntos contiguos al sensor y $\Delta\theta$ la resolución angular horizontal del sensor (para el sensor VLP-16: 0,1°– 0,4°, para 5 Hz – 20 Hz, respectivamente).

Otro inconveniente notable de este tipo de sensores son las oclusiones. Un sensor situado en la parte superior del vehículo generalmente no sufre de falta de información por estas oclusiones, obteniendo un rango de visión horizontal completo. Sin embargo, hay que considerar

los puntos que rebotan sobre el propio vehículo, de tal manera que un sensor situado en el techo del vehículo obtendrá puntos que reboten sobre el propio techo o el capó y producirá una zona ciega en las inmediaciones del vehículo. Por otro lado, si el sensor es ubicado en el frontal del vehículo, el campo de visión disponible queda restringido a la zona delantera.

Teniendo en cuenta estos inconvenientes y con el objetivo de detectar la geometría específica de obstáculos negativos, se persigue obtener la mejor resolución de la orografía valorando dos aspectos sobre la ubicación del láser: la orientación y su posición en el vehículo.

En primer lugar, se valoran las alternativas de orientación del LiDAR. Las pruebas para este estudio se han realizado simulando el modelo LiDAR VLP-16 de Velodyne, con 16 capas dispuestas verticalmente cada 2 grados, mientras que la resolución horizontal es mucho mayor, tal y como se ha indicado anteriormente. Por cada vuelta del láser, los puntos se van disparando según una superficie cónica de revolución. La intersección de estas superficies de revolución con una superficie ideal plana generará curvas cónicas, como las representadas en la figura 2.28.

Figura 2.28: Curvas cónicas generadas por el corte de las capas láser con el plano del suelo.

La *Configuración A* de la figura 2.28 corresponden a la orientación habitual de funcionamiento del sensor, situado en horizontal, donde a pesar de poder obtener un rango más amplio de visión horizontal, la baja resolución vertical, da lugar a una distancia entre capas

láser mayor. La *Configuración B*, por su parte, corresponde con un giro del láser en torno el eje X, orientándolo hacia el suelo. Esta configuración reduce la distancia entre capas solamente en aquellas más cercanas al láser. Mientras que la *Configuración C*, corresponde con una orientación perpendicular al suelo, es decir, girando el láser 90° en torno al eje Y. Con esta orientación se consigue la mayor resolución del suelo, pero se reduce el campo de visión a 30° de apertura. Sin embargo, esta situación no es crítica para los obstáculos negativos, al ser generalmente estáticos, no se prevén que entren en el ROI de forma lateral, sino que únicamente presenten un peligro si se localizan en la trayectoria del vehículo, aunque en el caso de obstáculos positivos, como peatones, habría que tener en presente este aspecto. En este sentido, para evitar una entrada lateral en el ROI de un obstáculo positivo y dinámico se adoptaría el método desarrollado en el subapartado 2.2.1.

Por otro lado, la figura 2.29 muestra las diferentes alternativas que se han valorado para la ubicación del LiDAR 3D sobre el vehículo.

Figura 2.29: Valoración de ubicaciones alternativas del LiDAR sobre el vehículo.

La posición 1 correspondería con la ubicación habitual, es decir, en el techo y orientado horizontalmente. Esta ubicación a pesar de ser la más ventajosa en la mayoría de los casos por obtener un campo de visión horizontal de 360° , a veces no es posible integrar el láser o da lugar a una pérdida de definición por su mayor distancia al suelo. En el caso del desarrollo de este algoritmo, se descarta debido principalmente a esta distancia, ya que se precisa mejorar la resolución de la orografía, esta ubicación no sería favorable.

En la posición 2, sobre el capó del vehículo, se mejora la resolución del terreno al disminuir la distancia al suelo. Sin embargo, si se mantiene una orientación horizontal, la distancia entre capas puede impedir una correcta identificación de los obstáculos negativos debido a la baja resolución vertical de estos sensores. Por lo que se debería orientar ya sea en torno al eje X o Y.

En la posición 3, persiguiendo tener la mejor definición de la irregularidad del terreno, se opta por bajar aún más la posición del láser mejorando la densidad de puntos sobre el suelo. Además, para tratar de reducir las zonas ciegas entre capas debido al problema de la baja resolución vertical del láser, se decide girar 90° en torno al eje Y, tal y como se ha discutido

anteriormente. Aunque el campo de visión con la nueva orientación del láser esté acotado (30° en el caso del láser utilizado en esta aplicación), no entraña peligro para la aplicación en cuestión, ya que cómo se indicaba, los obstáculos negativos no se espera que entren en el ROI de forma lateral.

Por todo ello, tras establecer los principales factores que afectan a la resolución de la información obtenida por el LiDAR, se decide ubicar el láser en la posición 3 (figura 2.29) y girado 90° ($\text{roll} = 90^\circ$). Así, no solo se minimiza la divergencia entre puntos consecutivos de una misma capa, sino que, además, las distintas capas también quedan concentradas en la parte inmediata delantera del vehículo, justo donde este tipo de obstáculos pueden acarrear una situación peligrosa. Con este nuevo *set-up*, se realizará un análisis de las capacidades del algoritmo.

Nótese que la resolución entre capas para el sensor VLP-16 es de 2° , por lo que puede afectar de forma negativa al detectar objetos con poca sección transversal, ya que con esta orientación ($\text{roll} = 90^\circ$), la nueva divergencia horizontal es mayor. Sin embargo, este inconveniente tiene influencia sobre en el tiempo de antelación cuando se produce una detección. A su vez, este tiempo está relacionado con la velocidad de circulación, por lo que, en situaciones con orografía irregular, la velocidad máxima de circulación que se alcance no afectará demasiado a este inconveniente.

2.2.2.2. Descripción del método utilizado en la detección de obstáculos negativos

El objetivo principal del algoritmo desarrollado en esta sección es la de analizar el entorno, identificar obstáculos negativos y realizar su etapa de *tracking* para verificar su peligrosidad. Como objetivo secundario, deberá ser operativo también para obstáculos positivos que se encuentren en la trayectoria del vehículo. En términos de rendimiento, se persigue un algoritmo robusto a la vez que liviano para su implementación en tiempo real en un vehículo todoterreno.

Se parte de la información del entorno proporcionada por el sensor láser escáner 3D, colocado según la configuración anteriormente explicada, y siguiendo una estructura secuencial, se aplican las diferentes etapas para obtener la información requerida de los obstáculos del entorno.

En primer lugar, siguiendo la misma estrategia que en el subapartado 2.2.1, se designa la región de interés donde se centrarán los cálculos del algoritmo. Debido a las bajas velocidades que se llevarán en terreno *off-road*, no es necesario definir el ROI en función de la velocidad, sin embargo, sí será preciso garantizar la maniobrabilidad de este.

La fase de identificación es la más crítica debido a que debe ser la más restrictiva para extraer los posibles obstáculos sin que estos estén caracterizados por una geometría en concreto. Por tanto, el algoritmo se centra en el análisis de subconjuntos de puntos adyacentes en cada capa para extraer las irregularidades de la orografía (se recuerda que en esta configuración las capas estarán mucho más concentradas unas a las otras). Este análisis se hace bajo dos criterios para extraer los puntos del entorno que pertenezcan a un obstáculo. El primer criterio analizado es la varianza de la coordenada Z de cada subconjunto de puntos. De esta forma,

aquellos puntos que difieren sustancialmente de la monotonía del resto de puntos estudiados son identificados como un posible obstáculo. El segundo criterio adoptado es el estudio de la suavidad de estos subconjuntos de puntos consecutivos. Este parámetro, es una adaptación del filtro utilizado en J. Zhang y Singh (2016) para extraer puntos característicos (*landmarks*) del entorno y que viene dado por la ecuación 2.10.

$$suavidad\ capa = \frac{\|\sum(Q_{centro} - Q_i)\|}{S \cdot \|Q_{centro}\|} \quad (2.10)$$

Donde Q_i son las coordenadas de cada punto del conjunto de puntos adyacentes S de una misma capa. Q_{centro} es el punto pivote o punto central de cada *set* de puntos evaluados.

Con estos dos parámetros, se consigue cumplir con los objetivos de obtener una identificación robusta y de baja carga computacional. Cabe destacar que los dos parámetros utilizados para la extracción de características han sido elegidos por la eficiencia y fiabilidad para el caso del parámetro que mide la suavidad, y por la eficiencia computacional de la varianza al compararse con el cálculo de la desviación típica. Conceptualmente el parámetro de la desviación típica indica con mayor claridad si algún valor difiere del resto dentro de un conjunto, sin embargo, computacionalmente, la varianza reporta la misma solución, pero requiere menos carga computacional al evitar el coste de computar iterativamente la raíz cuadrada. La figura 2.30 ilustra la respuesta de estos dos criterios utilizados, donde el valor en ambos aumenta drásticamente cuando son referidos al obstáculo.

Figura 2.30: Respuesta de los parámetros de extracción de obstáculos negativos.

A la vista del comportamiento de cada parámetro, se han fijado unos valores umbrales mínimo para cada uno. Cuando se supera este valor, los puntos pasan a ser considerados como candidatos a un obstáculo. Si los puntos no superan ninguno de los dos parámetros, se descartan para las etapas siguientes.

La tercera etapa corresponde a la segmentación. Para ello se hace uso del algoritmo de *clustering* DBSCAN, utilizado en el método del subapartado 2.2.1. Tras haber analizado en la etapa anterior una identificación por capas, se pasa a un análisis de la nube de puntos filtrada. Es importante destacar este rasgo de cara a una implementación del algoritmo con otro sensor LiDAR 3D, ya que, en el análisis de la etapa de identificación, al realizar el cálculo por capas láser, se trabaja con una información de nube de puntos ordenada, de la cual es necesario conocer distintos parámetros (o calcularlos). Por otro lado, en la etapa de *clustering*, no es necesaria dicha información y se pasa a trabajar con nube de puntos desordenada, lo cual permite otras estrategias de optimización de cálculos iterativos.

Se obtiene un conjunto de *clusters* que describen los distintos obstáculos del entorno. Estos se caracterizan mediante el cálculo de su centroide y *bounding box*. La última etapa del algoritmo corresponde con el *tracking* de obstáculos, que se realiza como se ha descrito en el subapartado 2.2.1.

El resultado del algoritmo tras el *tracking* será un registro de obstáculos identificados y caracterizados dentro del ROI. De estos se conoce su posición, volumen, estado de las variables de *tracking* y, en caso de posible colisión, el tiempo estimado para que se produzca esta.

2.2.2.3. Resultados de la detección de obstáculos negativos en entornos reales

El algoritmo ha sido implementado en un vehículo todoterreno utilizado como plataforma militar (figura 2.31) donde se ha integrado el sensor VLP-16 con la configuración descrita en el subapartado 2.2.2.1. El método ha sido validado en entornos reales *off-road* donde había obstáculos tanto negativos como positivos de distintos tamaños y características. Se garantiza además un tiempo de ejecución del código de 10 Hz.

Figura 2.31: Vehículo de pruebas. A la derecha, detalle de la ubicación del LiDAR 3D.

En concreto, los ensayos se llevan a cabo en las instalaciones del INTA-ITM (Instituto Tecnológico La Marañosa), donde se dispone de una pista de obstáculos normalizados. La figura 2.32 muestra las tres zonas de obstáculos diferenciadas.

Figura 2.32: Zona de obstáculos normalizados.

En la tabla 2.5 se describe el tipo de obstáculo que se encuentra en cada zona designada en la figura 2.32. En concreto, los obstáculos en la zona 1 son de tipo zanja que, debido a su geometría, se consideran como negativos, siendo la dimensión crítica para el algoritmo el largo de estos (l_y). En la zona 2, se encuentran los obstáculos de tipo resaltes. Estos son considerados como positivos y cuya dimensión crítica es la altura (l_z). Sin embargo, en la zona 3, donde los obstáculos son de tipo escalón, pueden ser considerados como negativos, al aproximarse a ellos en orden descendiente o como obstáculos positivos, al hacerlo en sentido contrario. En este caso, la dimensión crítica será, de nuevo, la altura (l_z). La anchura (l_x) de cada obstáculo tendrá importancia solo en cuestiones de maniobrabilidad del vehículo, lo que sería tratado en una posterior etapa de *path planning*.

Tabla 2.5: Zonas de obstáculos normalizados para entornos *off-road*.

	Tipo obstáculo	Descripción	Tamaño $[l_x, l_y, l_z]$ (cm)	Obstáculo positivo	Obstáculo negativo
Zona 1	Zanjas		320 x 30 x 70 320 x 50 x 90 320 x 90 x 90 320 x 150 x 130		X
Zona 2	Resaltes		320 x 60 x 60 320 x 50 x 50 320 x 40 x 40 320 x 30 x 30	X	
Zona 3	Escalones* (En sentido ascendente y descendente)		350 x 90 350 x 60 350 x 50 350 x 40 350 x 30	X	X

* Nótese que en los obstáculos de tipo escalón solo se indica la anchura (l_x) y la altura de estos (l_z). En negrita se destaca la dimensión crítica de cada tipo de obstáculo.

El análisis de resultados se ha realizado para evaluar la robustez y exactitud del algoritmo. Para ello, se ha contrastado *frame a frame* el registro de obstáculos identificados frente a los obstáculos que realmente se encuentran dentro del ROI durante el ensayo, ya sea completa o parcialmente.

El ensayo consiste en un realizar un recorrido con el vehículo por todos los obstáculos, para lo que se ha empleado unos 10 minutos. La tasa de adquisición de datos del LiDAR es de 10 Hz, por lo que se analizarán 5.775 *frames* en total. A través de la evaluación *frame a frame* del entorno real, se ha establecido que en 3.299 *frames* existía (completa o parcialmente) al menos un obstáculo dentro del ROI. En el resto del ensayo, se ha establecido que no había ningún obstáculo (2.476 *frames*), donde se incluyen paradas intermedias.

Análisis de los *frames* según registro de detección de obstáculos:

Para la validación de los resultados obtenidos, se tendrán en cuenta las siguientes métricas de clasificación binaria atendiendo a si un obstáculo ha sido identificado dentro del ROI:

- Verdaderos positivos (*VP*): Existe realmente un obstáculo y lo ha registrado el algoritmo.
- Falsos positivos (*FP*): En caso de no existir un obstáculo en el entorno real, pero el algoritmo ha registrado algún obstáculo.
- Verdaderos negativos (*VN*): Si no existe obstáculo en el entorno real y el algoritmo no tiene registro de obstáculos.
- Falso negativos (*FN*): En caso de existir algún obstáculo en el entorno real, pero el algoritmo no ha registrado ningún obstáculo.

Conociendo los parámetros anteriores, se podrá realizar una medida estadística en función de la clasificación del resultado del algoritmo. En este sentido, las matrices de confusión son utilizadas frecuentemente para definir las métricas de exactitud, precisión y exhaustividad (*recall*). Se entiende como exactitud, la proporción que se obtiene entre resultados verdaderos ($VP + VN$) y el número total de *frames* evaluados (ecuación 2.11). Obtener un 100% de exactitud equivale a que el algoritmo ha identificado correctamente todos los *frames*, ya sea cuando existía algún obstáculo como cuando no. Por otro lado, la precisión, tiene en cuenta la proporción entre los verdaderos positivos y el resto de las estimaciones positivas (ecuación 2.12). Finalmente, el concepto exhaustividad se relaciona con la proporción de positivos que se identificaron de forma correcta (ecuación 2.13).

$$Exactitud = \frac{VP + VN}{VP + VN + FP + FN} \quad (2.11)$$

$$Precisión = \frac{VP}{VP + FP} \quad (2.12)$$

$$Exhaustividad = \frac{VP}{VP + FN} \quad (2.13)$$

Respecto a los 3.299 *frames* donde se ha establecido que existía algún obstáculo en el ROI, el algoritmo proporciona un registro de salida con alguna detección en 2.343 *frames* (*VP*) y sin detección de obstáculos en 956 (*FN*). Por otro lado, en cuanto a los 2.476 *frames* donde se establece que no exista ningún obstáculo, el algoritmo ha estimado correctamente 2.352 *frames* (*VN*), y solo ha cometido algún error en 124 *frames* (*FP*).

La tabla 2.6 resume los valores anteriormente indicados como matriz de confusión, con la cantidad de *frames* en cada caso.

Tabla 2.6: Matriz de confusión de los resultados del algoritmo de detección de obstáculos negativos.

		Entorno real	
		Con obstáculos	No existen obstáculos
Registro de salida del algoritmo	Detección de obstáculos	2.343 (<i>VP</i>)	124 (<i>FP</i>)
	Sin detección de obstáculos	956 (<i>FN</i>)	2.352 (<i>VN</i>)

A partir de los valores anteriores, se calculan las métricas de exactitud, precisión y exhaustividad obteniendo los siguientes valores:

$$Exactitud = 81,30\%$$

$$Precisión = 94,97\%$$

$$Exhaustividad = 71,02\%$$

De estas métricas, la más relevante para conocer el rendimiento del algoritmo es la exactitud. Sin embargo, las otras dos métricas pueden complementar la evaluación sobre los resultados obtenidos. Cabe destacar que estos valores han sido obtenidos bajo un criterio conservador, ya que *frame* se considera con obstáculo aunque este sea identificado de forma parcial. Además, esta situación se suele dar en los límites del ROI, donde debido a la divergencia entre puntos, se dificulta la identificación, mientras que cuando el obstáculo se encuentra cercano al láser, el valor de falsos negativos disminuye significativamente y, por tanto, el valor de exactitud aumentaría en la zona más cercana al vehículo.

Análisis de la proporción de detecciones correctas según el tipo de obstáculo:

En la zona 1, existen cuatro tipos de zanjas con diferentes profundidades y longitud, siendo la primera de ellas la menor de todas, tal y como se indicaba en la tabla 2.5. Cabe destacar que, para las zanjas 1 y 2, se obtiene un porcentaje menor de detección (corresponde con el cálculo de la exhaustividad, según las métricas utilizadas anteriormente) (figura 2.33), pero, a su vez, no suponen peligro para la conducción debido a sus dimensiones.

En cuanto a la zona 2, existen cuatro obstáculos positivos de tipo resalte, donde varía la altura de estos, siendo el resalte 1, el mayor de todos. De nuevo, se aprecia que la dificultad de detección aumenta conforme disminuye el tamaño del obstáculo (figura 2.33).

Figura 2.33: Detecciones correctas en los obstáculos de las zonas 1 y 2.

En el caso de la zona 3, se encuentran los obstáculos tipo escalón. Estos obstáculos se considerarán positivos o negativos, según el avance del vehículo hacia ellos, tal y como se ha indicado. En la figura 2.34, se puede observar cómo existe una mayor dificultad a la hora de detectar este tipo de obstáculos cuando se afrontan en sentido negativo. En este caso, el tamaño del escalón no implica una variación significativa en la detección de los escalones.

Figura 2.34: Detecciones correctas en los obstáculos de la zona 3.

Respecto al valor obtenido de falsos positivos, la figura 2.35 ilustra el tipo de situaciones donde se obtiene una identificación errónea, destacando en rojo los puntos identificados como obstáculo por el algoritmo, cuando realmente no se debería de considerar como tal. Esto se debe principalmente a que la irregularidad del terreno *off-road* puede presentar cambios de pendientes más o menos acusados. En función del ajuste de los parámetros de identificación, es decir, los umbrales para el filtrado por varianza de la cota Z o el parámetro de suavidad por capa, pueden dar lugar a que el algoritmo sea más sensible a las irregularidades. Sin embargo, la consistencia de este tipo de obstáculos es baja y, por tanto, durante la etapa de *tracking* son descartados al no observarse durante varios *frames* consecutivos. Por otro lado, se hace patente la necesidad de encontrar un equilibrio entre la seguridad y la precisión del algoritmo.

Figura 2.35: Ejemplo de situación de detección de un falso positivo.

2.2.2.4. Discusión de los resultados del algoritmo de detección para obstáculos negativos

En este subapartado se ha desarrollado un algoritmo de detección de obstáculos optimizado para identificar aquellos denominados como negativos. Además, utilizando un solo láser 3D, se realiza la identificación y seguimiento de obstáculos tanto negativos como positivos, lo que permite una gran versatilidad para operar individualmente o complementar, mediante una fusión sensorial, otro sistema optimizado para la detección de objetos. Entre los objetivos secundarios, se perseguía obtener una salida del sistema robusta y precisa, lo cual se ha conseguido mediante la implementación de distintos filtros de modo secuencial. Por otro lado, su integración en un vehículo todoterreno real ha permitido comprobar que cumple con el objetivo de mantener una baja carga computacional, asegurando el rendimiento para 10 Hz.

A partir de la información de un LiDAR 3D embarcado en el vehículo con un *set-up* diferente al habitual, se consigue adquirir la información detallada de la orografía. Tras la etapa de identificación y seguimiento de obstáculos, la salida del sistema es un registro de los obstáculos identificados dentro del ROI, de los cuales se proporciona la información acerca de la posición de su centroide y su *bounding box*. Además, asociado a cada obstáculo se tiene el estado de la etapa de *tracking* y, en caso de que se estime una posible intersección de las trayectorias, se añade el tiempo previsto antes de colisión. Esta salida puede ser utilizada como input para otros módulos que se integren en un vehículo con alto grado de automatización, como el cálculo de trayectorias (*path planning*) o el sistema de toma de decisiones.

En cuanto a los resultados obtenidos, al haber sido implementado en condiciones reales, se ha podido realizar una valoración estadística contrastada con información del entorno real. Los ensayos se han llevado a cabo en entorno *off-road*, donde el algoritmo es capaz de

identificar con una exactitud del 81,30%, lo que se considera un resultado resaltante teniendo en cuenta el enfoque conservador aplicado en la obtención de esta métrica. Por otro lado, a raíz de la evaluación de los resultados, la posible complicación que puede aparecer durante la identificación de obstáculos dinámicos es que estos aparezcan dentro del ROI de forma lateral, ya que la configuración utilizada (girado 90° en torno al eje Y), proporciona un campo de visión estrecho. Para solventar este problema, la solución inmediata es realizar la fusión sensorial y aumentar el rango de visión y, a su vez, la definición de la orografía.

2.2.3. Conclusiones finales

En este apartado, se han propuesto dos soluciones para la detección de obstáculos basadas en LiDAR 3D bajo la premisa de mantener ambos algoritmos con una carga computacional baja, que garantice su ejecución en tiempo real. Además, estos algoritmos pretenden explorar, manteniendo una estructura sencilla, las capacidades de un sensor LiDAR 3D sin altas prestaciones (se ha utilizado un modelo con 16 capas) para obtener resultados precisos y que aseguren un buen funcionamiento.

Así, el algoritmo propuesto para la detección de obstáculos positivos, además de emplear un pipeline secuencial básico, obtiene un resultado robusto en la identificación. Esto permite que dicho método sea versátil y pueda ser utilizado para otros objetivos. Por ejemplo, será utilizado de nuevo en el apartado 3.1 como primera etapa para la caracterización de un elemento de referencia que sirva para el apoyo de una navegación autónoma.

Por su parte, el segundo algoritmo propuesto en este apartado ofrece una solución específica para la detección de obstáculos negativos en terrenos no preparados. Sin embargo, el enfoque utilizado permite que la solución planteada garantice también su funcionamiento para obstáculos de tipo positivo.

Finalmente, cabe destacar que los algoritmos planteados no utilizan el apoyo de otros sensores o información disponible, por ejemplo, del BUS-CAN del vehículo. En este sentido, se podrían mejorar las prestaciones de los algoritmos o aumentar la fiabilidad de los resultados si se utilizan *inputs* adicionales. Por ejemplo, una realimentación de la odometría del vehículo permitiría mejorar la etapa de *tracking* de obstáculos, ya que ayudaría a discernir con mayor solidez el movimiento relativo de los obstáculos. A pesar de ello, se obtiene un buen funcionamiento general de ambos algoritmos.

2.3. Caracterización de la infraestructura

La principal aplicación para la que se han utilizado los sensores LiDAR 3D en el ámbito del vehículo autónomo ha sido la detección de obstáculos, tal y como se indicaba en el apartado 2.2. Además, el auge de estos láseres escáner se debe a la ampliación de capacidades y prestaciones del vehículo autónomo, sin olvidar que, parte de su potencial se debe a los algoritmos desarrollados.

Además de identificar obstáculos que pueden dar lugar a situaciones peligrosas para la conducción, es necesario complementar la información del entorno con un mayor entendimiento

de este. Así, es igualmente relevante conocer los límites de la calzada, el posicionamiento a nivel de carril o identificar si existen o no carriles disponibles para circular. Estos son algunos ejemplos que se pueden lograr mediante el desarrollo de algoritmos de *Percepción* del entorno y que, en su conjunto, crean un flujo de información que puede ser aprovechado por otros sistemas dentro de la arquitectura de un vehículo autónomo. Por ejemplo, etapas posteriores como el sistema de *Toma de decisiones* del vehículo autónomo, requiere de una comprensión completa del entorno para hacer posible una conducción totalmente autónoma, o incluso para el cálculo de las posibles trayectorias en un sistema de *path planning*.

Por tanto, en este apartado se plantean una serie de métodos avanzados para la comprensión de la infraestructura o entorno viario a partir del procesamiento de datos de un sensor LiDAR 3D. En concreto, se exponen metodologías para lograr los siguientes objetivos en una caracterización transversal en entornos urbanos o interurbanos:

- Detección de la zona disponible de circulación:
 - Detección de bordillos como límite de la calzada.
 - Estimación de la anchura de la calzada.
- Detección de las líneas viales en entornos urbanos:
 - Posicionamiento transversal en carril.
 - Cálculo de número de carriles disponibles en entornos de carretera.

El objetivo principal de estas aplicaciones es caracterizar distintos elementos de la infraestructura para mejorar las capacidades de vehículos autónomos o servir de apoyo para una fusión sensorial con otro sensor principal. Además, estos algoritmos no están limitados a entornos urbanos o de carretera, sino que alguno de ellos también tiene aplicación en entornos todo-terreno o sin alto grado de infraestructura. Estos algoritmos basados en LiDAR 3D pueden atender, de forma genérica, a una estructura similar a la utilizada en el apartado 2.2 para la detección de obstáculos, pero con ciertas particularidades.

Las principales diferencias que se pueden encontrar en este caso se pueden identificar en varias etapas. Por ejemplo, en cuanto a la selección de la región de interés (ROI), esta debe ser acorde a la cada aplicación. Así, para una segmentación del suelo, no bastaría con filtrar por una cota Z igual a cero, sino que habría que tener en cuenta peraltes y pendientes. De igual modo, si se quieren extraer los límites del camino, no es suficiente definir un paralelepípedo frente al vehículo, un mejor enfoque sería definir esta región en función de la trayectoria seguida, evitando que ante una curva cerrada las características en estudio (bordillos, biondas, etc.) queden fuera. O, por ejemplo, durante la etapa de identificación de características, en lugar de utilizar variables únicamente geométricas, se puede hacer uso del valor de reflectividad (proporcionado en la mayoría de los sensores LiDAR). Este factor tendrá especial interés en este apartado para la extracción de líneas de la carretera y estimar el posicionamiento del vehículo respecto a ellas. El *tracking* de elementos característicos podría quedar simplificado ya que algunos de ellos se caracterizan por ser continuos a lo largo del ROI, como ocurre en la identificación las aceras en entornos urbanos o las líneas viales en interurbanos. Este hecho permite que la estimación de su posición en distintos instantes de tiempo sea más sencilla y, de igual modo ocurre con su monitorización.

2.3.1. Trabajos previos

La integración de cada vez más capacidades y sistemas avanzados de asistencia al conductor requiere de un entendimiento del entorno completo y semánticamente ordenado. Cuando a su vez, los nuevos desarrollos se desean aplicar a vehículos autónomos o con alto grado de automatización, a estos requisitos se les suma la importancia de poder ser ejecutados en tiempo real. De ahí la gran complejidad en el diseño y desarrollo de algoritmos de percepción y su continuo estudio de mejoras.

Estos sistemas de asistencia al conductor y algoritmos específicos enfocados hacia la conducción autónoma necesitan de sensores sofisticados para poder operar bajo cualquier tipo de situación durante la conducción. En concreto, los sensores LiDAR 3D han demostrado su desempeño en la detección de obstáculos, tal y como se ha abordado el apartado 2.2, pero también tienen un gran potencial para la caracterización del entorno (Castro et al., 2016; Holgado-Barco et al., 2017) y, como se abordará en el Capítulo 3, para el posicionamiento del vehículo.

Durante la conducción, no solo es importante evitar colisionar con obstáculos que se encuentren en la trayectoria, además, es primordial conocer cuál es la zona transitable por donde se circula. Por ello, una de las tareas que se abordaron desde las primeras investigaciones en el campo del vehículo autónomo, y empleado durante las competiciones DARPA, fue reconocer los límites del camino por donde debía guiarse tanto en off-road (Ozguner et al., 2007) como en entornos urbanos (Buehler et al., 2009).

Las líneas viales son uno de los elementos físicos que aportan más información para la caracterización transversal de la calzada. Así en trabajos como Yu et al. (2017); Wu et al. (2019), realizando una fusión entre visión artificial y la información de la carretera proporcionada por un mapa electrónico, consiguen una buena eficiencia. Sin embargo, la precisión alcanzada es baja respecto al posicionamiento en curvas. Otros métodos como Atia et al. (2017); Raymond et al. (2012) utilizan herramientas de caracterización, como variantes del algoritmo de RANSAC o el ajuste por mínimos cuadrados, para ajustar la curvatura de las líneas. En Yanwen et al. (2018); L. Li et al. (2018); Toledo-Moreo et al. (2009), recurren a una fusión con LiDAR y radar para identificar las líneas de carril y estimar el error lateral. Estos métodos, a pesar de su precisión, no permiten una implementación flexible al depender en gran medida de la configuración de sensores utilizada.

A partir de estas marcas, se puede realizar un posicionamiento centimétrico a nivel de carril y recopilar información relevante para una navegación autónoma (Clavijo et al. (2017)). Sin embargo, la detección de líneas presenta una dificultad añadida cuando se trata de situaciones complejas como cruces, incorporación de carriles o cuando existe tráfico denso que pueda producir oclusiones en algunos tramos. Además, al estar localizadas sobre la calzada, se requiere normalmente un paso previo donde se realice una segmentación del suelo. En entornos urbanos, realizar esta segmentación de los puntos del suelo a partir del ajuste a una geometría plana, no suele aportar una solución definitiva debido a las posibles pendientes o irregularidades que ofrece el terreno. Por ello en Oniga y Nedevschi (2010) tienen en cuenta estas características que inducen a error durante el *matching* de características.

Otro elemento clave para la caracterización horizontal de la calzada son los bordillos. La identificación de estos como límites de la calzada ha sido una tarea recurrente en visión

artificial como en Alvarez y Lopez (2010) o Guo et al. (2012), donde proponen un método con resultados con alta precisión incluyendo entornos complejos. También se pueden encontrar trabajos previos incluso al comienzo de la integración del LiDAR como sensor principal de percepción (Montemerlo et al., 2009), donde a partir de la comparación entre la distancia entre capas del sensor LiDAR 3D y el valor esperado, se extrae el patrón característico que identifica los bordillos de la calzada. Otros enfoques, proponen una solución parcial del problema como en A. Y. Hata et al. (2014); A. Hata y Wolf (2014), donde utilizan un sensor LiDAR para estimar la curvatura que describe la carretera y un método de regresión para tratar las situaciones donde se producen oclusiones. La detección de bordillos también es usada como input en algoritmos de estimación del posicionamiento por satélite. Con este propósito y basado en tecnología láser, se puede encontrar el desarrollo de Zhao y Yuan (2012). Del mismo modo en W. Zhang (2010), se propone un método para detectar los límites de la carretera utilizando sensores LiDAR 3D. Además, se puede encontrar en la literatura, enfoques en los que la solución adoptada se realiza mediante una fusión sensorial entre visión estéreo y LiDAR, como en Fernández et al. (2015) donde el algoritmo evalúa la información 3D ofrecida por estos sensores y, mediante el uso del algoritmo ICP, formar una nube de puntos con mayor densidad de donde poder extraer las características principales.

Por todo ello, se ha desarrollado un algoritmo que engloba distintos métodos específicos para la detección de los límites de la calzada mediante elementos físicos como bordillos e identificar el carril por el que se circula a través de las líneas viales, proporcionando, por tanto, un posicionamiento transversal. El algoritmo expuesto a continuación busca obtener un resultado robusto en cualquier situación, incluyendo aquellas que sean más exigentes desde el punto de vista de la disponibilidad de puntos en el entorno. Aunque algunos de los algoritmos planteados pueden ser aplicados en entornos *off-road*, se han planteado en primera instancia para conducción por terreno preparado. Sin embargo, se realiza una valoración de la adaptación de estos métodos específicos para entornos todoterreno.

Las implementaciones y ensayos presentados en este apartado han sido realizados con un sensor LiDAR VLP-16 ubicado sobre el vehículo con una orientación horizontal.

2.3.2. Arquitectura general del algoritmo de caracterización transversal

El algoritmo para la caracterización del entorno integra a su vez dos funciones principales donde se comprueba de forma iterativa los parámetros calculados, la función de detección de los límites y, por otro lado, la función de detección de línea de carril. En la figura 2.36, el flujograma describe la arquitectura general del algoritmo.

Figura 2.36: Descripción del algoritmo de caracterización transversal de la carretera. Recuadradas con línea discontinua, las dos funciones principales del algoritmo.

Para ambas funciones se utiliza la misma nube de puntos sin tratamiento previo. En la función “Detección de límites”, el algoritmo específico determina como salida el ancho de la vía. En esta funcionalidad se integran dos métodos que se consideran en paralelo. Por otro lado, la función de “Detección de línea de carril”, identifica las líneas viales que describen el carril por donde circula el vehículo, ya que es donde habrá mayor densidad de información útil, y a partir de esta identificación, se estima el ajuste al resto de líneas. La salida de este algoritmo será el número y ancho de los carriles que componen la carretera. Posteriormente, se compara el ancho calculado de cada carril con unos límites predefinidos. De esta manera es posible discernir entre carriles o arcén y, si no hay suficientes puntos que identifiquen correctamente todos los tramos de línea, se genera una nueva línea a partir de las ya calculadas. La salida de ambas funciones es evaluada y, en caso de que no haya suficiente certeza en la estimación, se utiliza la información del *frame* anterior.

Cabe destacar que, aunque no sea requisito para el funcionamiento de este algoritmo, si se dispone de un mapa digital de la zona por donde circula el vehículo, se puede utilizar la información de este. De esta manera, se puede contrastar la salida del algoritmo con la información disponible del mapa para identificar los carriles y clasificarlos, diferenciándolos entre arcenes, carriles adicionales, convencionales o de incorporación / salida.

2.3.3. Función para la detección de límites de la calzada.

Uno de los objetivos principales que tiene la función *Detección de límites* es determinar la zona transitable en aquellas vías en las que no existen líneas viales que delimitan los carriles (generalmente, calles en zonas urbanas), además también se podría utilizar para delimitar correctamente el ROI para otros algoritmos (por ejemplo, para la detección de obstáculos). Para ello, se presentan a continuación dos métodos que extraen los límites de la calzada apoyándose en patrones geométricos.

Método I: Detección de límites mediante las variaciones en cada capa del LiDAR 3D:

La rotación del láser escáner mientras dispara pulsos de luz genera un cono de revolución. Al cortar con un plano, se generan las distintas curvas cónicas (circunferencias, elipses, parábolas e hipérbolas) según el grado de inclinación. El Método I genera el conjunto de puntos que separa la zona transitable de la calzada de la acera a través de la transición representativa que describen las curvas cónicas mencionadas.

La propia elevación de las aceras limitantes de la zona transitable provoca que en la zona de los bordillos exista una variación sustancial entre los puntos adyacentes dentro de cada capa láser (figura 2.37).

Figura 2.37: Ejemplo de patrón geométrico identificativo de bordillos de la calzada.

De esta forma, es fácilmente identificable el patrón que siguen los gradientes de las coordenadas en cada eje. Tras ensayos realizados en entornos reales y simulaciones de entornos ideales, se fijan los umbrales para identificar esta variación en las coordenadas. A partir de la extracción de los puntos láser durante la etapa de identificación, se utiliza en primer lugar, el algoritmo de *clustering* DBSCAN para caracterizar cada patrón geométrico extraído y, a continuación, asociar estos puntos de las diferentes capas láser. Esta etapa, permite también excluir posibles falsos positivos, como pueden ser elementos verticales, señales, semáforos, etc. que se encuentren situados cerca de los bordillos.

Método II: Detección de límites mediante variación de distancias entre capas consecutivas:

El método que se describe a continuación se basa en el desarrollado por A. Y. Hata et al. (2014) mencionado en la literatura, donde se compara la distancia esperada entre las curvas cónicas de intersección con el suelo (si este ofreciera una superficie ideal plana) con la distancia real obtenida. Conociendo los parámetros de la configuración del montaje del láser, es posible calcular las distancias entre capas y a través de la comparación de los datos reales, extraer las características geométricas que representan los bordillos.

En la figura 2.38 se muestra cómo el patrón que rige la transición entre la calzada y la acera es fácilmente detectable mediante la diferencia de distancias Δr entre las capas del láser. Sin embargo, es preciso hacer notar que, aunque la detección de los bordillos se realiza correctamente, se ha comprobado que este método genera falsos positivos al introducir, como posibles elementos limítrofes, cualquier obstáculo que se encuentre en la zona próxima a los bordillos. Por ello, la solución propuesta trata de optimizar la identificación evaluando diferencias de Δr constantes con incertidumbre acotada. De esta manera se evita el problema anteriormente mencionado.

Figura 2.38: Variaciones de distancia entre dos capas consecutivas. Ejemplo patrón identificativo de bordillo.

Por otro lado, es posible que en el conjunto de puntos que describe este patrón característico se incluyan aquellos que no pertenezcan realmente a la calzada. Por ello, se ha implementado, además, un filtro de variación de alturas. Evaluando el valor de cota Z de los puntos pertenecientes a una misma capa del láser, se filtran aquellos que tengan una varianza significativamente distinta a la del resto.

Tras la identificación de los límites de la calzada ya sea mediante el Método I o II, se determina como salida del algoritmo el ancho de la calzada.

2.3.4. Función para la detección de líneas de carril y cálculo de número de carriles

Una caracterización transversal completa de la vía se consigue no solo conociendo los límites de esta, sino además, aportando toda información relevante que pueda ser extraída a medida que se circula por ella. Una vez realizada la detección de los límites de la calzada, se integra la siguiente funcionalidad en la que se incluye la determinación del número de carriles, el carril por el que se circula en cada momento y el posicionamiento lateral dentro de este.

Para ello, se plantea un método para la extracción de las líneas viales de la calzada. El mayor problema no reside en detectar este tipo de elementos, sino que, al estar dibujadas en el suelo, es necesario extraer previamente los puntos que estén contenidos en este. Teniendo en cuenta esta premisa, se describe a continuación la metodología seguida para la extracción de las líneas viales.

Segmentación del suelo: Considerando el tipo de curvas cónicas mencionadas anteriormente que se producen en la intersección de las capas del láser con el suelo, es posible realizar un ajuste paramétrico a cada una de ellas. Conociendo la orientación del LiDAR ubicado en el vehículo, se realiza el ajuste a las curvas y aquellos puntos que difieran demasiado de este,

no se consideran pertenecientes al suelo y se descartan (figura 2.39). Si los puntos no se encuentran sobre la curva, el rebote habrá tenido lugar sobre un obstáculo o elemento sobre la calzada.

Figura 2.39: Segmentación del suelo. a) Ajuste de curvas cónicas cada capa LiDAR que interseca el suelo, b) puntos filtrados pertenecientes al suelo (rojo).

Filtro de reflectividad: A partir de datos experimentales se ha observado el alto contraste de la intensidad de rebote entre los puntos pertenecientes al asfalto y aquellos de las líneas de la carretera. Por tanto, se hace uso del parámetro de reflectividad disponible en la nube de puntos como primer filtrado.

Consistencia de los datos: Una vez segmentados los puntos sobre el suelo y filtrados por reflectividad, se acota una distancia máxima entre puntos detectados para eliminar falsos positivos (figura 2.40). Aunque este paso no suele ser necesario, ya que con los pasos anteriores se elimina la mayoría de los falsos positivos, pueden aparecer puntos sobre el asfalto de otras marcas viales o irregularidades, por ejemplo, debido a reparaciones, que no corresponden con las líneas delimitadoras de carril.

Figura 2.40: Filtro de puntos pertenecientes a líneas viales del suelo.

Asociación de datos: Se realiza a continuación la asociación entre los puntos que pertenecen a una misma línea divisoria. Para ello, conociendo la orientación relativa entre la calzada y el vehículo, se establecen unas pequeñas regiones de interés que incluyen aquellos puntos que, por cercanía, se consideran del mismo trazo de línea. Esta definición de los ROI individuales es el factor más relevante de este método. En la figura 2.41 se ilustra cómo estas regiones mantienen su dirección longitudinal acorde a la identificación de puntos y, además, tienen una longitud que aseguran la intersección con el siguiente tramo discontinuo de la misma delimitación, de forma que puedan ir concatenándose en un mismo *cluster* los tramos que pertenezcan a la misma línea divisoria.

Figura 2.41: Definición de las regiones de interés de cada tramo de línea para la asociación de datos.

Estimación del ROI cuando hay oclusiones: A priori, el carril por donde se circula será aquel que tenga mayor cantidad de información disponible para realizar el cálculo de las líneas delimitadoras. Sin embargo, no tiene por qué ser así necesariamente debido a oclusiones. En estos casos, cuando por falta de puntos representativos de líneas no es posible calcular la región de interés descrita en el paso anterior, se procede a establecer la orientación promedio de las áreas donde sí que se realizó el cálculo (figura 2.42). Asimismo, conociendo el ancho estimado de los carriles, es fácilmente descartable aquellos puntos aislados que no pertenecen a la línea.

Figura 2.42: Definición de ROI individuales de los tramos de líneas: a) Situación con información suficiente; b) situación con oclusiones donde se estima la orientación de los ROI.

Ajuste cuadrático a líneas viales: El paso anterior termina con los conjuntos de puntos que pertenecen a líneas de delimitación de carriles. A continuación, se realiza un ajuste cuadrático, ya que este ajuste es similar a la realidad (Jiménez et al., 2016). Al mismo tiempo, se tiene presente el cumplimiento de paralelismo entre líneas. En este sentido, se impone la igualdad en los términos lineal y cuadrático del polinomio de ajuste. Así, para una línea de carril i , quedará definido por la ecuación 2.14.

$$p_i(y) = a_i + by + cy^2 \quad (2.14)$$

Donde y es el eje paralelo a la dirección longitudinal del vehículo y $p_i(y)$, es el ajuste cuadrático de cada línea de carril identificada.

Se realiza entonces un ajuste por mínimos cuadrados para obtener $n+2$ coeficientes, siendo n el número total de líneas. El error cuadrático cometido en el ajuste a la línea i , viene dado por la ecuación 2.15, donde para el ajuste a una línea se han identificado N_i puntos cuyas coordenadas son (x_{ij}, y_{ij}) .

$$E_i = \sum_{j=1}^N (a_i + by_{ij} + cy_{ij}^2 - x_{ij})^2 \quad (2.15)$$

Cálculo del error total: Para optimizar la robustez del sistema (uno de los objetivos perseguidos en esta sección), se ha ponderado el ajuste de las curvas en función de la cantidad de puntos detectados por cada línea. Se ha comprobado que esta ponderación, que prioriza el ajuste según el número de puntos, mejora la precisión global del algoritmo. Por tanto, la ecuación que se debe minimizar quedaría según la expresión 2.16.

$$E_T = \sum_{i=1}^n N_i E_i \quad (2.16)$$

Donde n es el número de líneas identificadas, E_i es el error cuadrático del ajuste de cada línea y N_i los puntos detectados para esa línea en concreto. Finalmente se obtienen los coeficientes a , b y c , a partir de los cuales es posible calcular el posicionamiento horizontal en el carril, la anchura de los carriles y el número de estos.

2.3.5. Resultados obtenidos para entornos estructurados

El algoritmo ha sido implementado y evaluado en entornos reales utilizando un LiDAR (VLP-16) situado horizontalmente. Además, la trayectoria seguida ha sido registrada mediante un receptor GPS de Trimble R4 con corrección diferencial.

Durante la descripción del método, se ha comentado que cabe la posibilidad de utilizar el *frame* anterior si el actual no dispone de suficiente información útil. También se indicaba que incluir un mapa digital posibilita la realimentación de información incluida en este mapa para corroborar la salida de los algoritmos. En primer lugar, se evalúa la precisión y robustez de las funciones específicas del algoritmo sin utilizar estas dos posibilidades operativas. Finalmente,

con el algoritmo completo, se analiza si los errores cometidos en las funciones individuales son resueltos con el proceso iterativo propuesto en la figura 2.36.

2.3.5.1. Análisis de los resultados de los métodos de detección de límites

La tabla 2.7 muestra la comparativa entre los dos métodos propuestos para esta función específica del algoritmo. Esta valoración distingue dos zonas, una primera zona inferior a 10 metros de detección y la segunda, desde 10 metros en adelante. Esta diferenciación se selecciona atendiendo a la zona donde las capas del LiDAR tienen una mayor resolución por contener mayor densidad de puntos debido a la distribución vertical de los haces del láser. Esta resolución por cada capa es un factor clave en el funcionamiento de los métodos. En concreto, el Método I depende en gran medida de las variaciones de las coordenadas de los puntos en cada capa, y tan solo el 28% de los *frames* analizados se ha conseguido realizar correctamente con una antelación superior a 10 metros. Sin embargo, consigue realizar correctamente la identificación en un 67%, pero con una antelación inferior, lo que limita sustancialmente su aplicación debido al poco margen de maniobra. Por el contrario, con el Método II se consigue realizar una correcta extracción de los límites con una antelación superior a 10 metros en el 94% de los *frames* analizados. El desempeño de este método puede observarse en la figura 2.43.

Tabla 2.7: Resultados de funciones de determinación de calzada (análisis sobre 875 frames)

	Método I	Método II
Detección correcta: Alcance > 10 metros	28 %	94 %
Detección correcta: Alcance ≤ 10 metros	67 %	2 %
Detección incorrecta	5 %	4 %

Figura 2.43: Identificación zona transitable mediante Método II: a) Entorno real; b) nube de puntos original; c) resultado de la identificación. En verde, zona transitable.

Se puede concluir que el Método I es sensible a variaciones que puedan aparecer en una capa, al basar sus cálculos en la información de solo una de ellas. Es decir, un obstáculo en las proximidades de un bordillo podría ocasionar detecciones incorrectas al identificarlo como límite de la calzada. Por otro lado, el Método II basa el cálculo en dos capas consecutivas del LiDAR 3D, por lo que la identificación de un obstáculo como límite de la calzada es más inmune si este último es detectado solo por una de las capas.

Cabe destacar que los resultados han sido obtenidos en escenarios urbanos con condiciones reales de tráfico. Este hecho hace que existan oclusiones de los bordillos en ciertas ocasiones. Sin embargo, el Método II ofrece un alto grado de precisión. Además, su robustez queda comprobada ya que no se obtiene un número elevado de *frames* incorrectos. En caso contrario, esto limitaría su implementación.

2.3.5.2. Análisis de los resultados del método de detección de línea de carril

Cabe distinguir dos situaciones dentro de los posibles escenarios ya que, durante el ensayo, puede haber tramos donde existan carriles de incorporación o salida, cambio del número de carriles, etc. En tal caso, se tratará como “Escenario con singularidades”, mientras que en caso contrario “Escenario sin singularidades”.

La tabla 2.8 refleja la precisión particular de esta funcionalidad de forma aislada. En situaciones sin singularidades, se llega a una fiabilidad de 78%. Por otro lado, en situaciones con singularidades, esta fiabilidad alcanza un 47% de acierto. Cabe destacar que este valor se debe considerar como cota inferior ya que, por ejemplo, en situaciones singulares como la aparición de un carril adicional (incorporación o salida) no se han considerado como correctas a pesar de que el algoritmo detecte un carril más, debido a que no es posible identificar la naturaleza de este carril. Con el empleo de un mapa digital, dicho carril estaría definido con información complementaria y, por tanto, se ha optado por un enfoque conservador al no contabilizarlo como detección correcta. Además, cabe destacar que esta función específica, al igual que en la detección de los límites de la calzada, no registra una cantidad notable de *frames* consecutivos con identificación errónea, y en caso de que se produzcan, están motivados principalmente por oclusiones por el tráfico.

Tabla 2.8: Resultado individual del algoritmo de determinación del número de carriles.

	Escenario sin singularidades	Escenario con singularidades
Detección correcta	78 %	47 %*
Cálculo no realizado	8 %	18 %
Detección incorrecta	14 %	35 %*

* Situaciones con carril de incorporación o salida, no se consideran como detecciones correctas al no ser posible definirlos como tales, el algoritmo los identifica como carriles adicionales a la vía principal.

El porcentaje de “Cálculo no realizado” y de “Detecciones incorrectas” que se puede observar en la tabla 2.8, para ambos tipos de escenarios, se debe principalmente a:

- Detección del arcén como carril adicional (figura 2.44.a). Aunque se identifique de ma-

nera errónea, la primera etapa del algoritmo donde se detectan los límites de la carretera solventa este problema al ser posible la comparación de los anchos de carril.

- Oclusión de las líneas de carril (figura 2.44.b y 2.44.c). La validación ha sido llevada a cabo en situaciones reales de tráfico, por lo que, otros vehículos alrededor pueden producir oclusiones parciales de las líneas delimitadoras y en algunos casos, de la pérdida total de información de la línea de carril. Sin embargo:
 - La situación más crítica sería perder la información de las líneas de carril por donde circula el propio vehículo ya que por la ponderación del número de puntos (descrita anteriormente), las estimaciones se basan principalmente en estas.
 - En caso de perder información de los carriles intermedios, suele resolverse mediante la aproximación de la anchura estándar de carril.
 - Si la pérdida de información se produce en los carriles más alejados, el problema es menos relevante ya que solo afectaría al cálculo del valor del número de carriles.
- Aunque la incorporación de carriles se detecta con precisión por el algoritmo, sin la disponibilidad de más información (por ejemplo, a partir del mapa digital), no es posible distinguirlos de los carriles principales (figura 2.44.d).

Figura 2.44: Distintos ejemplos de situaciones con singularidades o detecciones erróneas: a) Detección incorrecta de arcén; b) Oclusiones debido a vehículos circundantes; c) Falta de información de alguna línea; d) No detección de carril de incorporación.

2.3.5.3. Discusión de los resultados en entornos estructurados

Nótese que los resultados y conclusiones aquí presentados, corresponden solo a las funcionalidades de modo aislado. El rendimiento del algoritmo teniendo en cuenta las interacciones

entre ellos o haciendo uso de la información de *frames* anteriores, no se han considerado para el porcentaje de precisión de las tablas 2.7 2.8. Las situaciones más comprometidas para obtener una correcta identificación corresponden a situaciones que presentan singularidades como incorporaciones a la vía principal, carriles de salida, etc. Incluso en estas situaciones, no limitan la operabilidad del algoritmo ya que, a efectos prácticos, la estimación del posicionamiento a nivel de carril o la identificación de los cambios de carril han sido correctamente identificadas. Además, utilizando *frames* previos del LiDAR, sería posible realizar la estimación de la caracterización transversal de la vía a lo largo de todo el ensayo.

Finalmente, se puede afirmar que el rendimiento global del algoritmo propuesto para entornos urbanos e interurbanos estima correctamente la zona transitable, las líneas de carril, el número de carriles y la anchura de estos. La figura 2.45 muestra un ejemplo del resultado del algoritmo donde queda segmentado correctamente el suelo y se identifican 3 carriles en zona interurbana.

Figura 2.45: Resultado del algoritmo propuesto para la caracterización del entorno. Se observa la segmentación del suelo (azul oscuro) y las líneas de carril.

Por último, cabe mencionar que, al incorporar la información de un mapa digital, se ha verificado que tanto el número de carriles como las líneas de carril, se han detectado correctamente. Por otro lado, esta información procedente de los mapas da lugar a un proceso iterativo en el algoritmo, en caso de una estimación incorrecta puede ocasionar problemas de convergencia. Sin embargo, durante el ensayo realizado, no han surgido este tipo de errores y por tanto se podría considerar un 100% de acierto.

2.3.6. Aplicación del método de cálculo de los límites de la vía para entornos *off-road*

Se ha abordado de forma satisfactoria la detección de los límites de la zona transitable en un entorno estructurado, donde la delimitación de dicha zona está generalmente definida por elementos físicos (ya sean bordillos o marcas viales). En esta situación, se ha obtenido resultados fiables en la mayoría de los escenarios, sin embargo, en un área todoterreno, estos

límites no están tan definidos como en el caso anterior. De esta manera, los únicos elementos de referencia que pueden dar soporte a la estimación de las delimitaciones pueden ser árboles, arbustos, etc. Además, el terreno ofrece normalmente más irregularidades que un suelo asfaltado. Por ello, en este subapartado, una vez se ha propuesto una solución para entornos con terreno preparado, se pretende encontrar qué métodos se adaptan mejor a entornos *off-road* y valorar su fiabilidad.

En primer lugar, se plantea utilizar un método básico para extraer el camino transitable en un entorno todoterreno donde se pueden distinguir tres zonas diferenciadas: la zona central, por donde transita el vehículo, y las dos zonas laterales no transitables. Por tanto, esta metodología básica pretende extraer un patrón geométrico que permita estimar cada una de estas zonas. El primer paso, como en otros algoritmos descritos en la tesis, es delimitar el ROI que corresponda con la trayectoria que sigue el vehículo. A continuación, utilizando el parámetro de la reflectividad de cada punto, es posible diferenciar las zonas de camino transitable de aquellas que no (plantas, arbustos, etc.). Por tanto, se realiza este primer filtrado, aunque no es suficiente para identificar el camino con precisión. Por ello, se procede con el estudio pormenorizado de cada una de las capas que intersecan con el suelo. Se observa que en cada una de estas capas que impactan en el suelo, se consigue identificar unos patrones geométricos debido a las rodadas de vehículos y la delimitación a ambos lados por la vegetación. Por tanto, es posible, a priori, extraer un patrón que se repite en cada capa. Se definen por tanto tres agrupaciones que, mediante el algoritmo K-means, quedan identificadas en cada capa. A continuación, tras el cálculo de los centroides de cada *cluster*, se evalúa la consistencia de estos en dos *frames* consecutivos. Finalmente, se selecciona el centroide que mejor se adapte al eje longitudinal del vehículo y se unen los centroides seleccionados de cada capa, identificando la trayectoria que sigue el vehículo.

Se han llevado a cabo varias pruebas de este método básico en entorno *off-road*, donde el vehículo todoterreno circula por caminos y senderos con vegetación a ambos lados (figura 2.46).

Figura 2.46: Detección de la zona transitable en entornos *off-road* mediante K-means.

En dicha figura se puede observar el comportamiento del algoritmo en la parte izquierda, donde se identifica la zona no transitable en rojo y el camino por donde circula el vehículo en verde. Las cruces negras indican el centroide del *cluster* identificativo del camino e indica la trayectoria por donde debe ser guiado el vehículo.

El algoritmo es capaz de identificar el camino transitable en este entorno con un alcance de 10 a 14 metros, suficiente para la distancia de detención a las velocidades que se suele circular en estas zonas. Las ventajas que aporta este método es la sencillez de implementación y la baja carga computacional. Sin embargo, su aplicación en entornos más genéricos está muy limitada. En este sentido, la fiabilidad del algoritmo disminuye en áreas que presenten cruces de caminos, bifurcaciones, etc. En las pruebas realizadas, se evalúa una muestra de 500 *frames* donde al menos 5 capas del LiDAR intersecan con el suelo. En el 54% de los *frames*, la identificación de la zona transitable es correcta en todas las capas, mientras que en el 31%, una de las capas comete algún error en dicha detección y en un 15%, el error de la detección afecta a más de una capa. Estos resultados, ponen de manifiesto la necesidad de algoritmos con mayor versatilidad.

Se plantea utilizar el algoritmo descrito anteriormente como “Método II” para la detección de los límites de la carretera, esta vez en entornos todoterreno. En la figura 2.47, se aprecia como se diferencia correctamente la zona transitable (camino: verde) de la zona no transitable (sotobosque: rojo).

Figura 2.47: Zona transitable en entorno *off-road* mediante Método II: a) Entorno real; b) nube de puntos original; c) obtención del resultado de la identificación: en verde, los puntos dentro del ROI que son transitables.

Además, el ejemplo de la figura 2.47, corresponde con una situación de bifurcación de caminos, ante esta singularidad, el algoritmo se comporta de manera correcta identificando como zona transitable los dos caminos posibles. La detección de los límites del camino alcanza una alta fiabilidad tal y como se observó en el subapartado 2.3.5.1. Sin embargo, en este caso, la falta de elementos viales no proporciona información para completar la caracterización

transversal. De esta manera, para aplicar este método en un guiado autónomo, bastaría con estimar la línea central de la zona transitable, ajustándose a las delimitaciones calculadas por el algoritmo.

Tras esta verificación, aunque el Método II estaba orientado a la detección de un elemento físico con geometría característica como son los bordillos, se observa la polivalencia y robustez del algoritmo al ser aplicado en otro tipo de entorno cuyos límites del camino están definidos de forma irregular.

2.3.7. Conclusiones finales

Se ha presentado un algoritmo capaz de caracterizar transversalmente entornos urbanos e interurbanos basado en tecnología LiDAR 3D, proporcionando información sobre los límites de la carretera, la anchura del carril, el número de carriles disponibles, permitiendo obtener un posicionamiento lateral del vehículo. Además, la integración de distintas funcionalidades específicas en el algoritmo global y las interrelaciones entre ellas da lugar a una mejora en la fiabilidad de los resultados. Se mejora, por tanto, a partir de la caracterización del entorno, el posicionamiento que se podría obtener a partir de receptores GNSS cuando la señal de satélite es débil o nula.

Los ensayos llevados a cabo en escenarios reales muestran la robustez del método tanto para la funcionalidad del cálculo de los límites, como en la extracción de las líneas de carril. En este último caso, existen dificultades en la estimación cuando el escenario presenta alguna singularidad como carriles de incorporación o salida, cruces, etc. En estos casos, la precisión en la detección del número de carriles disminuye, sin embargo, si se incluye la información de un mapa digital al algoritmo, los resultados finales podrían mejorar sustancialmente, ya que el algoritmo puede contrastar dicha información con la obtenida durante la identificación mediante láser. Aún así, los resultados presentados muestran el rendimiento de cada función específica de modo aislado.

Para conseguir una mejora en la precisión de la detección se puede recurrir a una fusión con visión artificial, debido a que esta tecnología ofrece información acerca del color o contraste y sus limitaciones podrían ser reducidas con la fusión con sensores LiDAR.

Cabe destacar que la funcionalidad del cálculo de los límites de la carretera consigue buena precisión y fiabilidad en entornos *off-road* a pesar de no estar orientada para aplicarse en estos entornos. Esta función demuestra, por tanto, su versatilidad para extraer los límites de la vía tenga o no bordillos, por lo que no se limita su aplicación a entornos urbanos que contengan estos elementos.

En la práctica, este enfoque puede ser usado para mejorar el posicionamiento si el sensor es utilizado para otros propósitos en el vehículo autónomo como puede ser para la detección de obstáculos. De esta manera, se consigue una mejora en la eficiencia de los recursos del vehículo.

Capítulo 3.

Posicionamiento y mapeado

Como ya ha quedado patente, el *Posicionamiento* es otro gran campo de desarrollo junto a la *Percepción* dentro de la investigación del vehículo autónomo. Ya que a lo largo del Capítulo 2 de esta Tesis, se ha analizado en profundidad cuáles eran los retos de la percepción del entorno, en este capítulo se abordarán los desarrollos y aplicaciones relacionados con el cálculo del posicionamiento del vehículo.

Un posicionamiento preciso del vehículo es fundamental para que la navegación autónoma sea una realidad. Hoy en día, el posicionamiento usado en vehículos para navegación no es válido para la circulación autónoma y por ello, existe un continuo desarrollo de algoritmos sofisticados que pretenden obtener este posicionamiento bajo cualquier situación por compleja que pueda llegar a ser. En este sentido, la gran variabilidad de los posibles escenarios, ya sea por el tráfico, las normas de circulación o los imprevistos que puedan aparecer, requiere que los sistemas de posicionamiento sean cada vez más robustos y adaptables a cualquier entorno y situación por donde pueda circular el vehículo autónomo. Por ello, en la mayoría de los casos, aquellos vehículos con un alto nivel de automatización fusionan la información de la estimación de la trayectoria seguida, la información extraída del entendimiento del entorno (Capítulo 2), un mapa electrónico y, a veces, un mapa creado simultáneamente.

Por todo lo mencionado anteriormente, en este capítulo se aborda el tema del posicionamiento preciso del vehículo, no solo como cálculo de las coordenadas de posición del vehículo, sino también aportando otros enfoques como el que se expone en el primer apartado, donde sirviéndose de la tecnología láser, se consigue apoyar una navegación autónoma y realimentar el lazo de control transversal del vehículo, basándose en la identificación de elementos de referencia. Por otro lado, como se ha adelantado, la fusión entre los datos del posicionamiento y un mapa durante la conducción pone de manifiesto el concepto SLAM, que más adelante se ampliará tanto con técnicas basadas en la extracción de características del entorno (apartado 3.2), como aplicando modelos de *Machine Learning* (apartado 3.3). Finalmente se abordará en el apartado 3.4, una metodología que integra los dos enfoques anteriores, fusionando un sistema analítico como los desarrollados en el apartado 3.2 y otros basados en la estimación de un modelo de *Machine Learning*, de tal manera que el sistema global se beneficie de las ventajas de cada uno.

3.1. Caracterización de elementos de referencia para apoyar la navegación autónoma

Para alcanzar un correcto seguimiento de una ruta (preestablecida o calculada) no es suficiente apoyarse en receptores GNSS. Aunque éstos han evolucionado en los últimos años, no garantizan una precisión por debajo de 1 centímetro aun utilizando corrección diferencial (Naranjo et al., 2009). Además, bajo ciertas circunstancias, como el paso por túneles o zonas con baja o nula cobertura de satélites, este hecho se agrava y, para compensarlo, es necesario comprobar el posicionamiento del vehículo en todo momento. Por ello, en esta sección se presenta un sistema alternativo para respaldar el posicionamiento del vehículo y apoyar la navegación autónoma, sin añadir ni modificar la arquitectura estándar de sensores de un vehículo autónomo. Utilizando un sensor LiDAR 3D, el método caracteriza un elemento de referencia y lo detecta de forma continua durante la trayectoria. Asimismo, se exploran alternativas que mejoren la robustez de los resultados, con el fin de que ese elemento represente una referencia fiable de navegación en todo momento. El método ha sido implementado y probado en situaciones reales, circulando por entornos delimitados por barreras en ambos sentidos.

3.1.1. Trabajos previos

Se puede encontrar en la literatura específica distintas metodologías para implementar la navegación autónoma. En Chu et al. (2012) realizan un seguimiento de *waypoints* mediante sistemas GNSS, mientras que, en otros casos, el seguimiento de la trayectoria se basa en líneas o elementos de la infraestructura (Han et al., 2012) o incluso apoyándose en el vehículo que le precede (Bergenheim et al., 2012; Broggi et al., 2000). En cualquiera de los casos, asegurar el posicionamiento a nivel de carril es crítico para un correcto seguimiento de la ruta. Los receptores GNSS, usando corrección diferencial, pueden asegurar un posicionamiento centimétrico en el mejor de los casos. Sin embargo, muchas situaciones cotidianas de conducción ofrecen una cobertura pobre o nula (entornos urbanos angostos, túneles, etc.). Aun así, también es común encontrar receptores GNSS a los que se les incluye una unidad inercial (IMU), dando lugar a los conocidos como GPS-INS. En este caso, pueden mantener un posicionamiento correcto durante un pequeño periodo de tiempo (*dead reckoning*) (Toledo-Moreo et al., 2018).

En este apartado, se analiza la posibilidad de utilizar un elemento de referencia del entorno para complementar y mejorar el guiado autónomo de un vehículo. Tras la identificación del elemento que se utilice como referencia, mediante alguno de los métodos descritos en el Capítulo 2, este desarrollo se centra en analizar la robustez y fiabilidad para calcular los parámetros de control que apoyen un algoritmo de seguimiento de ruta implementado en el vehículo. Para ello, se utiliza un LiDAR 3D con el que es posible realizar la identificación del elemento y utilizarla para calcular un error lateral y angular, que serán los parámetros utilizados por el sistema de control. Cabe destacar, que en este apartado se utilizará a modo de ejemplo la identificación de barreras delimitadoras de carril como elemento de referencia. Los resultados obtenidos son extrapolables a la utilización de cualquier otro elemento como líneas de carril, paredes de un túnel, biondas, etc.

Asimismo, teniendo en cuenta las limitaciones al integrar este tipo de sensores en el vehículo,

se analiza la influencia de la colocación del mismo. En trabajos como los presentados por Shang et al. (2014); Han et al. (2012) estudian un nuevo *set-up* del sensor sobre el vehículo para mejorar la detección de elementos del entorno, pero mantienen la ubicación del láser en el techo. En este desarrollo, sin embargo, se valoran las ventajas al posicionar el sensor LiDAR en la parte frontal del vehículo. Esta nueva ubicación puede permitir una mayor facilidad de integración, pero, por otro lado, conlleva la pérdida de información hacia la parte de atrás y, por ello, sugiere estudiar nuevas estrategias para compensar dicha falta de datos.

3.1.2. Metodología para caracterización y seguimiento de elemento de referencia

El algoritmo planteado en esta sección trata, en primer lugar, de caracterizar un elemento de la infraestructura que pueda ser utilizado como referencia y apoyo al guiado autónomo. Para ello se puede emplear cualquier elemento que se encuentre situado en la dirección de circulación como por ejemplo las líneas de carril o barreras delimitadoras de carril (figura 3.1), entre otros. En el apartado 2.3 se proponía una metodología para identificar y extraer las líneas de carril usando sensores LiDAR 3D, por ello, en este caso, se propone otra opción más de percepción para determinar un elemento de referencia, en concreto, barreras delimitadoras de carril de tipo New Jersey. A pesar de ello, el método de seguimiento del elemento de referencia y, también, las opciones de mejora propuestas más adelante son extrapolables al uso de líneas viales, así como cualquier otro elemento de referencia.

Figura 3.1: Correspondencia del entorno real con datos LiDAR 3D. 1) Líneas viales centrales, 2) Líneas viales separación de arcén, 3) y 4) Barreras tipo New Jersey.

El método seguido para esta caracterización tiene como punto de partida el mismo algoritmo ya descrito en el apartado 2.2.1 para la detección de obstáculos positivos. De esta manera, se distinguen dos etapas: en la primera, tal y como se describía en el apartado al que se hace referencia, se parte de los datos sin tratar del láser y se realizan secuencialmente las etapas de definición del ROI, triangulación 2D de la nube de puntos, filtrado según orientación de

los vectores normales y segmentación, para así, obtener los diferentes *clusters* de los objetos positivos del entorno. A partir de este punto, el algoritmo diverge según el objetivo buscado, por un lado, hacia una detección y seguimiento de obstáculos (apartado 2.2) y, por otro lado, hacia una caracterización del elemento que apoya la navegación autónoma. Se aborda a continuación esta segunda etapa del algoritmo.

Por tanto, a partir de la nube de puntos resultante de la segmentación, se debe discernir aquellos *clusters* que son un obstáculo en la trayectoria, de aquellos que pueden servir como elemento de referencia. Este último estará localizado lateral y longitudinalmente respecto al vehículo de una forma determinada. Cabe destacar que, al igual que en el trabajo de Petrovskaya y Thrun (2009), se puede construir un modelo 2D a partir de la información 3D cuando para la caracterización del elemento, una de sus dimensiones es irrelevante para el propósito perseguido. En el caso que concierne, se propone el ejemplo de caracterización de las barreras delimitadoras. Tras la selección del *cluster* que identifica esta barrera, su altura es prescindible y, por tanto, se proyectan los puntos sobre el plano XY. Sobre estos puntos, se calcula la regresión cuadrática de la siguiente forma:

$$x = f(y) = Ay^2 + By + C \quad (3.1)$$

Donde el sistema de coordenadas referenciado al vehículo en cada momento queda designado con x, y para los ejes transversal y longitudinal, respectivamente.

A continuación, es posible calcular las dos variables de control para el guiado autónomo, el error lateral y el error angular. Para calcular estos parámetros es necesario definir previamente un punto contenido en la regresión calculada, designado como $P(x_0, y_0)$, que servirá como base de los cálculos posteriores. Este punto, además de pertenecer a la regresión calculada, su vector normal debe pasar por el origen de coordenadas, localizado en este caso en el sensor. A partir de la ecuación de la recta tangente por un punto P , se deduce la ecuación de la recta perpendicular a esta:

$$x = \frac{1}{f'(y_0)(y - y_0)} + f(y_0) \quad (3.2)$$

Se deben cumplir dos condiciones para poder calcular $P(x_0, y_0)$. Por un lado, la ecuación 3.2 debe pasar por el origen de coordenadas y por otro lado, el punto P debe verificar la ecuación 3.1. Así, es posible obtener los parámetros de control según las ecuaciones 3.2 y 3.4:

$$Error\ lateral = \sqrt{(x_0^2 + y_0^2)} \quad (3.3)$$

$$Error\ angular = \arcsin\left(\frac{y_0}{Error\ lateral}\right) \quad (3.4)$$

En la figura 3.2 se puede observar un ejemplo de la metodología descrita aplicada a la caracterización de barreras. A la izquierda, se muestran las diferentes etapas para la identificación del elemento: a) Definición del ROI y triangulación, b) cálculo de vectores normales, c) filtrado según la orientación de los vectores. A la derecha, se ilustra el cálculo del punto

de referencia P y los parámetros de control (error lateral y angular).

Figura 3.2: Ejemplo de aplicación del método sobre caracterización de barreras.

3.1.3. Influencia del campo de visión horizontal del sensor

Resulta relevante analizar la influencia de la colocación del sensor LiDAR en los resultados. La posición utilizada normalmente es en el techo, donde la principal ventaja es obtener un rango de visión horizontal completo. Sin embargo, la integración en algunos casos puede resultar más sencilla en el frontal del vehículo. Por ello, a continuación, se realizan una serie de ensayos con el objetivo de poder realizar una comparativa válida entre la opción de ubicar el sensor en la parte superior del vehículo y la opción de ubicarlo en el frontal. En esta segunda opción se perderán todos aquellos puntos dirigidos hacia la parte posterior del vehículo. Para llevar a cabo los ensayos comparativos, se ha colocado el LiDAR en el techo y posteriormente, se han restringido los datos a los puntos obtenidos en la parte delantera del vehículo, tal y como ocurriría al ubicar el sensor en esta parte.

Tras ser implementado, el algoritmo ha sido probado en entornos reales utilizando barreras delimitadoras de tipo New Jersey como elemento de referencia. En concreto, la zona de ensayos se encuentra en el tramo BUS-VAO de la autovía A-6 en Madrid. Dicho tramo tiene una longitud de 12 km y 2 carriles disponibles a excepción de ciertas partes que tiene un

solo carril. Los carriles están delimitados en ambos lados por este tipo de barreras de forma continua, solo en uno de los lados se interrumpe debido a incorporaciones y salidas. Por ello, representa un entorno óptimo para utilizar un elemento del entorno como referencia y validar el algoritmo. La velocidad de conducción a lo largo del ensayo se encuentra entre 45 y 60 km/h con una frecuencia de adquisición de datos de 20 Hz. De nuevo, los resultados obtenidos en estos ensayos son extrapolables a la utilización de otro tipo de elemento de referencia.

En la figura 3.3 se puede observar la diferencia en el resultado del algoritmo cuando el láser está ubicado en la parte frontal (figura 3.3.a) y cuando se ubica en el techo del vehículo (figura 3.3.b). Las gráficas inferiores representan el valor de los parámetros de control anteriormente descritos (error lateral y error angular) en función del tiempo. La respuesta obtenida en la situación donde solo se obtienen puntos de la parte frontal del vehículo, se caracteriza por una gran componente de ruido en comparación con la obtenida en el caso de la ubicación del sensor en el techo.

Figura 3.3: Respuesta del algoritmo: a) Ubicación del láser en el frontal, b) Ubicación del láser en el techo.

A la vista de los resultados, se justifica que la ubicación favorable para una respuesta más estable sería en la parte superior del vehículo cuando se detectan elementos de referencia como las barreras laterales. Este comportamiento se debe a que la identificación del elemento en ambas direcciones da lugar a una mejor definición de la regresión cuadrática que representa dicho elemento. En el caso de que se utilizaran las marcas viales como elemento de referencia, este efecto se podría amortiguar ya que, si el sensor está en el frontal y, por tanto, más cerca del asfalto, las líneas se detectarían con mayor número de puntos.

A pesar de que la ubicación en el techo del vehículo proporciona una salida más estable del algoritmo, si por problemas en la integración sensorial, no es posible que el sensor láser sea fijado en esta ubicación, sería necesario utilizar herramientas que consigan una respuesta similar y eliminar la componente de ruido. Una de las posibles soluciones sería la utilización de técnicas de SLAM, las cuales se abarcan en el apartado 3.2. Sin embargo, se propone una solución específica y sencilla de implementar para mejorar la respuesta sin pretender realizar un SLAM completo. En este sentido, se plantea una solución utilizando la información de *frames* anteriores y extrapolando el elemento de referencia hacia la zona donde no se obtienen datos.

3.1.4. Extrapolación simple del elemento de referencia utilizando modelo matemático de vehículo

Recordando que una de las mayores desventajas de los sensores LiDAR era la información dispersa en el espacio, la reconstrucción del entorno suele ser una herramienta recurrente para remediar la falta de información densa. La idea de reconstruir incrementalmente el entorno introduce los primeros fundamentos del concepto de SLAM que se desarrollarán más adelante en esta tesis.

Como se indicaba anteriormente, sin llegar a desarrollar un algoritmo de SLAM propiamente dicho, reconstruir el elemento de referencia identificado puede solventar la problemática de ubicar el sensor láser en la parte frontal del vehículo. Por tanto, reconstruyendo la trayectoria seguida es posible extrapolar el elemento según haya sido el movimiento relativo del vehículo. En este sentido, se ha analizado y comparado la respuesta del sistema ante distintas configuraciones: con el sensor láser colocado en la parte frontal, con el sensor láser ubicado en la parte superior, obteniendo el rango completo de visión horizontal, y finalmente, con el láser en la parte delantera reconstruyendo el elemento característico hacia la zona sin datos.

El algoritmo propuesto opta por utilizar los datos disponibles del vehículo para realizar una reconstrucción de la trayectoria de 1,5 segundos (equivalente a una extrapolación del elemento entre 18 y 25 metros según las velocidades durante el ensayo). Además, se plantea utilizar un modelo matemático de dinámica vehicular para dicha extrapolación de los puntos detectados del sensor láser hacia la parte trasera del vehículo donde por oclusión del propio vehículo, se carece de información.

3.1.4.1. Estructura del algoritmo para reconstrucción de la trayectoria

El método que se describe a continuación para la extrapolación del elemento de referencia a través del modelo matemático del vehículo pretende ser una implementación sencilla que proporcione una solución específica al problema planteado. La estructura del método queda plasmada en la figura 3.4, donde los datos de entrada serán la velocidad del vehículo, el giro de volante y el peralte de la calzada. Los dos primeros son datos disponibles en el vehículo, mientras que el tercero es necesario adquirirlo externamente. En este caso, el ángulo de peralte se puede obtener aproximadamente de la lectura de un giróscopo. Aunque el propio sensor no podría distinguir entre la medición del peralte y el ángulo de balanceo del vehículo, en Jiménez (2011) proponen un método iterativo para encontrar una solución a esta cuestión, sin embargo, no es posible implementarlo en tiempo real. Además, en condiciones normales de conducción, se puede considerar que el ángulo de balanceo es despreciable.

Figura 3.4: Diagrama de flujo del método para reconstrucción de la trayectoria.

Para la estimación de la trayectoria a través de dinámica vehicular se pueden utilizar modelos con diversos grados de dificultad (Mechanical Simulation Corporation, 2001; Nalecz, 1992; Vera y Félez, 1994; Day, 1995; Mera, 2001). Uno de los más ampliamente utilizado es el de 14 grados de libertad (gdl): 6 gdl de movimientos y rotaciones de la masa suspendida, 4 gdl de movimientos de masas no suspendidas y 4 gdl de rotación de ruedas. Este modelo está implementado en el software (Aparicio et al., 2009). Integrar este modelo matemático no supone un gran incremento de carga computacional, aun así, para garantizar el tiempo de ejecución y no utilizar modelos dinámicos simplificados, los cuales no tienen en cuenta parámetros relevantes, se ha adoptado por crear una “tabla de consulta” (*lookup table*). La tabla es generada *offline* utilizando el modelo matemático completo, donde se toman como variables de entrada la velocidad instantánea del vehículo v y la rotación del volante (ambos datos disponibles en el vehículo), y se obtiene como salida la velocidad de guiñada, $\dot{\varphi}$. A partir de estos datos, y conociendo la frecuencia de adquisición de datos Δt , es posible reconstruir la trayectoria mediante las ecuaciones 3.5 y 3.6:

$$\Delta x = v \cdot \Delta t \cdot \cos(\dot{\varphi} \cdot \Delta t) \quad (3.5)$$

$$\Delta y = v \cdot \Delta t \cdot \sin(\dot{\varphi} \cdot \Delta t) \quad (3.6)$$

3.1.4.2. Influencia del peralte de la carretera

El modelo matemático implementado no tiene en cuenta la influencia del peralte de la calzada, que puede llegar a ser significativa para estimaciones de largas distancias. Se trata, por tanto, de encontrar la relación entre la compensación de giro de volante y el peralte para así, corregir la tabla *offline* mencionada anteriormente y no incrementar, de esta manera, la complejidad del sistema.

Tal y como se muestra en la figura 3.5 sobre el modelo vehicular de dos ruedas, la componente de la fuerza gravitacional F_y está actuando transversalmente al vehículo debido al peralte, sin estar actuando ninguna otra fuerza en esta dirección.

$$F_y = P \cdot \sin \xi \quad (3.7)$$

Donde el peso del vehículo viene designado por P y el peralte por ξ .

Esta fuerza genera unos esfuerzos en los neumáticos para alcanzar el equilibrio de fuerzas del sistema: $F_{y\alpha f}$ y $F_{y\alpha r}$, para los neumáticos delantero y traseros respectivamente. Se puede asumir que existe una relación lineal entre estos esfuerzos en los neumáticos y su ángulo de deriva según la ecuación 3.8.

$$F_{y\alpha} = K_{y\alpha} \cdot \alpha \quad (3.8)$$

Donde la constante de proporcionalidad, $K_{y\alpha}$, es la rigidez transversal del neumático y, por tanto, estos ángulos pueden ser calculados.

Figura 3.5: Modelo vehicular de dos ruedas.

Finalmente, para encontrar la relación entre el ángulo de volante corregido y el ángulo de peralte se debe cumplir la condición de que el ángulo de deriva del vehículo β sea nulo (figura 3.5). En el caso de una trayectoria rectilínea, el ángulo de giro de volante calculado sería el necesario para compensar el peralte. De esta manera, cuando β es nulo, las proyecciones ortogonales del centro de curvatura cumplen la igualdad:

$$\frac{\tan(\delta^+ - \alpha_f)}{l_1} = \frac{\tan(\alpha_r)}{l_2} \quad (3.9)$$

De la anterior ecuación, se puede calcular el giro de rueda δ^+ en función del ángulo de peralte ξ , y con la desmultiplicación rueda-volante del vehículo se obtiene la corrección del giro $\delta_{SW}^+ = K_{SW} \cdot \delta^+$. Los valores de giro de volante corregidos en función del peralte se calculan mediante la ecuación 3.10. Obteniendo estos valores, es posible actualizar la tabla con los nuevos valores de giro de volante.

$$\delta_{SW} \cdot corr = \delta_{SW} - K_{sw} \cdot \delta^+ \quad (3.10)$$

3.1.4.3. Resultados con extrapolación mediante modelo dinámica vehicular

Para validar este método de reconstrucción de la trayectoria, se han llevado a cabo ensayos en un tramo de 1,4 km en entorno interurbano bajo las mismas condiciones descritas en los ensayos del subapartado 3.1.3. Mediante el sistema de adquisición, se han obtenido durante el recorrido los datos de velocidad instantánea y giro de volante del BUS-CAN de comunicaciones del vehículo, el ángulo de peralte mediante un giróscopo RMS FES 33 y, finalmente, el posicionamiento con precisión centimétrica a través de un receptor Trimble R4 GPS con corrección diferencial, este último para poder realizar la comparación frente a la trayectoria reconstruida.

Como se puede observar en la figura 3.6, el método utilizado realiza una estimación correcta de la trayectoria incluso para largas distancias. Por otro lado, cabe destacar que considerar la influencia del peralte en el cálculo es fundamental ya que, de otra manera, se alcanzan errores inadmisibles en la estimación de la trayectoria, como se puede observar en el trazado en verde de la misma figura 3.6. Además, queda reflejado cómo el error es acumulativo y la estimación se aleja cada vez más del valor real, por lo que sería necesario rectificar la estimación cada cierto tiempo. Sin embargo, para el problema que se trata, solo es necesaria la reconstrucción del elemento de referencia durante 1,5 segundos, lo que equivale a un tramo entre 18 y 25 metros para las velocidades durante el ensayo. Por ello, en la figura 3.6, la reconstrucción de la trayectoria en ambos casos (con y sin influencia del peralte) se ha realizado en intervalos de este tiempo de reconstrucción y comparados con la señal GPS.

Figura 3.6: Influencia del peralte en la reconstrucción de la trayectoria en grandes distancias.

Para evaluar las desviaciones obtenidas por el método, se ha definido un punto de control cada 20 metros sobre la trayectoria del ensayo, dando lugar a 64 posiciones en total. El valor medio de estas desviaciones ha sido de 0,231 metros, con una desviación típica de 0,174 metros. Por lo que el método de reconstrucción propuesto es capaz de extrapolar el elemento de referencia de forma eficaz.

Una vez validada la metodología de reconstrucción de la trayectoria por tramos de 1,5 segundos, es posible extrapolar el elemento de referencia hacia la parte posterior del vehículo, donde no se obtiene información de los puntos LiDAR cuando el sensor es situado en el frontal del vehículo. Para ello, al igual que se realizó en los ensayos preliminares de este apartado, se ha utilizado una configuración donde el sensor se encuentra en el techo del vehículo para obtener el rango completo de visión horizontal. Para simular la integración del láser en el frontal del vehículo, se restringe la nube de puntos solo hacia delante. De esta manera es posible comparar los resultados obtenidos para ambas configuraciones y la influencia de la extrapolación.

La figura 3.7 muestra un tramo de la trayectoria del ensayo donde se compara la detección del elemento de referencia en tres casos: *a)* detección del entorno solo hacia delante, simulando una integración del láser en el frontal del vehículo; *b)* detección del entorno con todo el

rango de visión horizontal, caso en el que el LiDAR esté situado en el techo del vehículo; *c)* integración en el frontal del vehículo, con una detección de la nube de puntos solo en la zona delantera, pero con extrapolación del elemento de referencia hacia la zona no visible por el sensor.

Figura 3.7: Comparativa de comportamiento del algoritmo con distintas configuraciones de colocación del láser.

Se puede observar que el ajuste de la regresión en la configuración *a)* se aleja sustancialmente del ajuste para la configuración del láser sobre el techo (figura 3.7.b). Sobre todo, esta desviación es notable en los valores obtenidos de error angular y curvatura. Por otro lado, en el ajuste de la configuración *c)*, donde se aplica la extrapolación según la metodología descrita, se consigue unos resultados cercanos a la situación con el láser en el techo.

Para valorar el comportamiento de la metodología a lo largo de toda la trayectoria del ensayo, se elabora la tabla 3.1. En esta tabla se comparan las dos configuraciones de láser situado en el frontal, obteniendo solo datos hacia delante y aplicando el método de extrapolación del algoritmo, frente al resultado que se obtendría con el láser situado en el techo. Las diferencias son analizadas dos a dos para los casos *a)-b)* y *b)-c)* para la duración total del ensayo.

Tabla 3.1: Desviaciones de los parámetros de control según la colocación del láser.

	a-b		b-c	
	Media	Desv. típica	Media	Desv. típica
Error Lateral (m)	0,167	0,081	0,062	0,017
Error Angular ($^\circ$)	2,668	1,293	0,886	0,241
Curvatura (m^{-1})	$4,157 \cdot 10^{-3}$	$2,355 \cdot 10^{-3}$	$0,950 \cdot 10^{-3}$	$0,948 \cdot 10^{-3}$

A la vista de las desviaciones mostradas en la tabla anterior, se puede concluir que la extrapolación permite alcanzar niveles de robustez similares a poner el sensor láser en el techo, pero sin las dificultades que esta integración puede conllevar en un vehículo. Por tanto, se facilitaría, mediante un método de bajo coste computacional, integrar el sensor en el frontal sin interferir ni reducir la precisión del cálculo de los parámetros de control para un sistema de navegación autónoma.

3.1.5. Ponderación de puntos para relevancia homogénea del elemento de referencia

El propio funcionamiento del sensor LiDAR da lugar a la divergencia entre puntos respecto a la distancia, tal y como se indicaba en la ecuación 2.9 en el subapartado 2.2.2.1. Así, la identificación del elemento de referencia estará definida con un mayor número de puntos cuanto más cerca esté del vehículo. Este hecho afecta a la regresión cuadrática utilizada en el ajuste del elemento de referencia ya que confiere mayor importancia a la zona más cercana al sensor.

Podría resultar razonable que el ajuste al elemento de referencia tenga la misma relevancia en toda su longitud y tratar de homogeneizar la importancia relativa de los puntos detectados. Para ello, se busca una ponderación en función de la distancia.

Se puede definir que la densidad de puntos de una capa láser por unidad de longitud es inversamente proporcional a la distancia, por tanto, la densidad de puntos sobre un objeto identificado a una distancia d , atenderá a una expresión como la indicada en la ecuación 3.11.

$$\rho = \frac{K}{d^2} \quad (3.11)$$

Siendo ρ , la densidad de puntos y K una constante que dependerá de la resolución horizontal y vertical del sensor LiDAR utilizado.

Teniendo en cuenta una zona ciega para una distancia menor a 2 metros (esta zona dependerá del modelo de LiDAR utilizado), se calcula una función de ponderación según la distancia al sensor, tal como se observa en la figura 3.8. Esta función está definida como la diferencia entre la densidad teórica en el límite de la zona ciega y la densidad de puntos medida, ajustada por mínimos cuadrados. La función ponderación indica el número de veces que debe repetirse cada punto para que la relevancia de todos ellos sea homogénea.

Figura 3.8: Densidad de los puntos detectados (diagrama de barras), ajuste de la densidad medida (rojo) y función de ponderación (azul) respecto a la distancia al sensor.

Para comprobar si efectivamente la función de ponderación mejora la estabilidad del algoritmo, se implementa y se analizan las señales del error angular y el error lateral durante un tramo del ensayo en carretera utilizado en los ensayos previos de este apartado. La configuración de ubicación del sensor que se utiliza es la que obtiene el rango de visión horizontal total (LiDAR ubicado en el techo del vehículo) por ser el más favorable.

En la figura 3.9 se muestra cómo afecta la integración de esta función de ponderación en las variables de salida del sistema.

Figura 3.9: Comportamiento de las variables de salida del algoritmo utilizadas para el control de la navegación autónoma: a) sin usar la función de ponderación; b) con la función de ponderación integrada.

Al aplicar esta función, el comportamiento presenta un rizado en la señal (figura 3.9.b), lo que hace patente la sensibilidad del resultado a errores de detección de puntos lejanos que no pertenecen realmente al elemento de referencia.

El problema de este comportamiento reside en la identificación de un elemento externo

al elemento de referencia como parte de él, en este caso la barrera. Es decir, es más fácil que obstáculos, vehículos, señales de tráfico o cualquier otro elemento, queden integrados en la barrera si se encuentran a mayor distancia debido a una menor definición de estos objetos. Además, al ponderar de forma homogénea todos los puntos identificados, este error de identificación tendrá la misma relevancia que los puntos que verdaderamente pertenezcan al elemento de referencia. Por otro lado, es complicado que en puntos cercanos se tengan detecciones erróneas y por ello, se obtienen mejores resultados manteniendo la relevancia de los puntos según el propio funcionamiento del láser.

3.1.6. Conclusiones finales

Los sistemas de posicionamiento actuales utilizados para la navegación en vehículos no garantizan una precisión lo suficientemente robusta como para permitir un guiado autónomo. Este último requiere de sistemas de posicionamiento fiables y precisos bajo cualquier situación. En vehículos de alto nivel de automatización o completamente autónomos suelen integrarse sensores láser 3D para la identificación de objetos del entorno mediante técnicas como las que se han desarrollado en el Capítulo 2 de esta tesis. En este apartado, sin embargo, se ha pretendido usar esos mismos sensores, pero con un objetivo diferente: servir de apoyo a la navegación autónoma, optimizando los recursos implementados en el vehículo y aportando una mejora en el posicionamiento.

En este sentido, se ha propuesto un método que, a partir de la identificación de un elemento de referencia, es capaz de calcular los parámetros de control transversal utilizados en un sistema de navegación como son el error angular y el error longitudinal. Además, se ha valorado qué ubicación del láser favorece una mejor respuesta de este sistema, donde resulta que una ubicación en el techo del vehículo favorece el cálculo estable de las variables de salida del algoritmo. Sin embargo, a veces esta ubicación puede no estar accesible o bien, se facilitaría la integración del sensor en otra posición como en el frontal del vehículo.

Si se requiere integrar el láser en el frontal, el rango de visión se restringe a solo la parte delantera. Para evitar que esta falta de información influya en el cálculo de los parámetros de control mencionados, se ha definido una metodología haciendo uso de un modelo matemático de dinámica vehicular. Con este método se garantiza un resultado similar al que se obtendría si el láser se posiciona en el techo. Esta metodología se basa en la reconstrucción del elemento de referencia para extrapolarlo hacia la parte trasera del vehículo donde no se obtienen datos.

Este método propuesto, sin llegar a ser estrictamente un algoritmo de SLAM, no necesita de altas prestaciones ni carga computacional para obtener una salida correcta. La extrapolación del elemento de referencia hacia la parte trasera está acotada en un periodo corto de tiempo, por lo que errores de deriva acumulativos que puedan aparecer por los sensores inerciales utilizados, no son significativos. Por otro lado, cabe destacar que, aunque es posible una reconstrucción del elemento de referencia en coordenadas globales a partir de los datos de un receptor GNSS, esta solución no sería válida en entornos complejos. En estos casos, es más preciso y fiable hacer uso de una fusión sensorial entre sensores inerciales, receptores GNSS (mediante filtros de Kalman), modelos de dinámica vehicular o incluso mediante técnicas de odometría visual.

En ciertas ocasiones resulta de gran utilidad reconstruir los datos obtenidos según la trayectoria seguida. A través de esta reconstrucción incremental del entorno, la densidad de información disponible aumenta considerablemente, pudiendo generar mapas del entorno donde se pueda ubicar el propio vehículo o permitir un mejor entendimiento del entorno. Estas ventajas son propias de las técnicas de SLAM y ponen de manifiesto el potencial de estos algoritmos en el desarrollo de nuevas capacidades del vehículo autónomo, tal y como se describe en el siguiente apartado.

3.2. Mapeado y localización simultánea para entornos con características específicas

Como se ha descrito anteriormente, dentro de los sensores utilizados en una configuración convencional de prototipos de vehículos totalmente autónomos (grado 5 según escala SAE (SAE, 2016)), el LiDAR 3D está siendo utilizado cada vez más como sensor principal de reconocimiento del entorno debido a las ventajas que ofrece y su continuo desarrollo (Jiménez, 2017). No solo cumpliendo funciones de seguridad al detectar obstáculos, peatones o los límites de la carretera, sino que, además, como se veía en el apartado 3.1 ha abierto otro campo de aplicación como es el apoyo al posicionamiento del vehículo. Siguiendo en esta línea, pero como una metodología alternativa o complementaria a otros sistemas de localización, surgen diferentes algoritmos, como son la Odometría Visual y algoritmos para la Localización y Mapeo Simultáneos, conocidos también, por sus siglas en inglés, SLAM (Simultaneous Localization And Mapping).

Es posible distinguir algoritmos que tratan de resolver el problema de SLAM basándose en sistemas de localización por satélite fusionado con datos de la odometría del vehículo. Sin embargo, estos sistemas son susceptibles de una baja precisión si la calidad de la señal GPS disminuye o incluso no está disponible. Por ello, los sistemas de SLAM basados en sensores de percepción, también conocidos como Visual SLAM, pueden ofrecer un buen rendimiento bajo circunstancias donde los primeros sistemas fallan. A pesar de las ventajas que aportan estas soluciones de SLAM, también pueden sufrir algunas limitaciones en su precisión debido principalmente a errores acumulativos.

En primer lugar, es necesario aclarar que SLAM, formalmente, no se refiere a un tipo específico de algoritmo, más bien es un concepto heredado de robótica (en concreto, robótica móvil) que trata de resolver el problema de posicionamiento en un entorno, a priori desconocido, construyendo incrementalmente el mapa por donde circula el “agente/robot” mientras hace uso de este para posicionarse. En este sentido, las técnicas que abordan el problema de SLAM pueden ser implementadas de diversas formas donde, además, cada etapa del método podrá utilizar un número de algoritmos concretos. Por otro lado, cabe destacar que el problema de SLAM como tal, no ha sido resuelto completamente ya sea por desviaciones en la reconstrucción de mapas de gran tamaño, errores de deriva en el cálculo del posicionamiento o bien por alta carga computacional que no permiten una ejecución y actualización del mapa en tiempo real Pierzchała et al. (2018). Por estos motivos, existe una considerable investigación en este ámbito.

Aplicado al ámbito del vehículo autónomo, los algoritmos que tratan de resolver el SLAM han demostrado su importancia en la navegación al proporcionar capacidades de localización en áreas donde otros métodos tienen dificultades o bien, al apoyar y mejorar el posicionamiento proporcionado por estos. En este sentido, se han desarrollado técnicas de SLAM que hacen uso de la información de la odometría disponible por el BUS-CAN del vehículo, o bien utilizando la combinación GPS-INS, donde una unidad inercial (IMU) proporciona datos sobre la orientación del vehículo en los 6 gdl y se fusiona con la localización proporcionada por la señal de satélite. Por otro lado, tal como se adelantaba anteriormente, también se encuentran metodologías de Visual SLAM que hacen uso de los sensores de percepción, como cámaras, stereo-vision, radar, láser 2D o LiDAR 3D para extraer características del entorno (*landmarks*) a través de las cuales poder calcular el posicionamiento relativo. En todas estas técnicas, la reconstrucción del mapa del entorno genera información complementaria que puede ser usada por otros algoritmos al obtener un mayor conocimiento del escenario (figura 3.10), por ejemplo, para optimizar la extracción de características, identificación de obstáculos estáticos, número de carriles, bifurcaciones, etc.

Figura 3.10: Ejemplo de reconstrucción del entorno a partir de la fusión de cuatro LiDAR VLP-16, utilizando posiciones GPS-INS.

A pesar de que, actualmente existen numerosas soluciones de SLAM basadas en LiDAR, en la mayoría de los casos se requiere de una alta carga computacional, tal y como se señalaba anteriormente. Sin embargo, existen entornos que presentan una infraestructura particular que permitiría el uso de algoritmos más simplificados. Bajo esta premisa, se desarrolla un algoritmo de posicionamiento básico basado exclusivamente en LiDAR 3D orientado a escenarios interiores o exteriores donde se toman como referencia planos verticales. Esta solución es bastante eficiente en escenarios como parkings interiores o entornos urbanos angostos donde la cobertura GPS es mala o nula. Además, se exploran varias alternativas que mejoran la estabilidad del método y la estimación del posicionamiento. El algoritmo ha sido implementado en entornos virtuales ideales en una primera etapa de diseño y, posteriormente, ensayado en situaciones reales obteniendo resultados satisfactorios.

3.2.1. Trabajos previos

Solucionar el problema de SLAM consiste en generar la estimación del movimiento del propio vehículo y la reconstrucción del mapa del entorno de forma simultánea. Aunque su

aplicación en robótica móvil es muy amplia, se destaca el uso orientado a los sistemas de guiado en vehículos autónomos.

El reciente auge del vehículo autónomo ha hecho adaptar este concepto surgido en robótica hace unos 30 años (Leonard y Durrant-Whyte, 1991; Smith et al., 1990) ante la necesidad de un posicionamiento con precisión centimétrica y la de obtener un mayor entendimiento del entorno para ejecutar correctamente cálculos de *path planning* o de toma de decisiones, entre otros. Así pues, debido a la gran variedad de escenarios donde el vehículo autónomo debe desenvolverse y la casuística de cada uno, las técnicas de SLAM encuentran un campo de aplicación fundamental.

Como se indicó en el apartado 3.1, obtener una localización precisa no es trivial. Soluciones basadas en sistemas de posicionamiento GNSS, incluso utilizando receptores con corrección diferencial, no garantiza una precisión centimétrica bajo cualquier circunstancia. Otros enfoques contemplan la integración de unidades inerciales (IMU) mediante filtro de Kalman a dichos sistemas GNSS, obteniendo una solución parcial ya que, para medias y largas distancias, se obtiene un error de deriva incremental con el tiempo. Otros métodos como el desarrollado en el apartado 3.1, se basan en la identificación de un elemento de referencia para favorecer la navegación autónoma, sin embargo, esto no es suficiente cuando se trata de entornos complejos, cruces o sin una infraestructura de apoyo. La tendencia general ante esta problemática es buscar un equilibrio en la fusión de metodologías donde se beneficie de las ventajas que aporte cada una individualmente. Sin embargo, existe otra alternativa que puede minimizar las carencias anteriormente detectadas, como son las técnicas basadas en Odometría Visual o Visual SLAM, ambas apoyadas en tecnología láser 3D. A pesar de poder obtener un error de deriva, se pueden considerar como una mejor solución en entornos complejos como áreas urbanas reducidas, entornos interiores con baja o nula cobertura GPS o donde otros sensores encuentren dificultades en su funcionamiento.

Estas técnicas de posicionamiento visual (Odometría Visual y Visual SLAM) se pueden englobar en lo que se denominará como “técnicas con enfoque clásico”. Estas se caracterizan por estar formadas por unas etapas o pasos generales (figura 3.11): extracción de puntos característicos, conocidos como *landmarks*; asociación de datos, estimación de la posición y actualización o corrección del estado. Si bien, ambas metodologías tienen en común estas etapas, cada una de ellas tiene sus peculiaridades. Mientras las técnicas de Odometría Visual se centran en una consistencia local para calcular incrementalmente la trayectoria, las de Visual SLAM deben garantizar una consistencia global. Para ello, pueden apoyarse de técnicas de *loop-closure*, dando lugar a una reducción del error de deriva en las estimaciones de la trayectoria y la reconstrucción del mapa (Yousif et al., 2015). La estimación del posicionamiento en ambos casos se realiza a partir de la asociación de los *landmarks* de la observación actual y los obtenidos previamente. De forma general, tras obtener la estimación del movimiento relativo desde distintos sensores, se debe corregir o actualizar la posición. Las técnicas de corrección de la posición pueden ser separadas en dos grandes grupos: con un enfoque basado en filtros derivados de métodos Bayesianos (por ejemplo, filtro extendido de Kalman (EKF) (Newman et al., 2006; Davison, 2003; Montiel et al., 2007), filtro de partículas (Montemerlo et al., 2002; Mohan y Krishna, 2010; Hähnel et al., 2003) o filtro de información (IF), una variante del EKF (Thrun et al., 2004; Gutmann et al., 2011) y, por otro lado, métodos basados en optimización (por ejemplo, mediante Bundle adjustment (Z. Zhang y Shan, 2004)).

Figura 3.11: Etapas generales en algoritmos de Odometría Visual y Visual SLAM.

Como se indicaba, estas técnicas se pueden denominar clásicas al tener una base común analítica que puede ser implementada mediante el uso de diversos algoritmos específicos para cada etapa y a su vez, emplear distintas tecnologías. En oposición, se abordarán más adelante en esta tesis técnicas basadas en inteligencia artificial, las cuales se alejan de este enfoque clásico.

Centrándose en el problema de SLAM para vehículos autónomos, se pueden distinguir dos ramas principales atendiendo a la fuente principal de generación del mapa: SLAM con un solo vehículo y SLAM multi-vehicular (*Single-vehicle SLAM* y *Multi-vehicle SLAM*, respectivamente). En el primer caso, se trata de un campo más amplio y establecido, donde se pueden distinguir algoritmos de SLAM que por un lado utilizan técnicas de *loop-closure* o relocalización (Eade y Drummond, 2008), con las que se reduce el error de deriva; o bien, obtienen el posicionamiento a través del mapa reconstruido como en Bresson et al. (2016) o finalmente, aquellos que aprovechan un mapa electrónico creado previamente con información adicional para el posicionamiento (Joshi y James, 2015; Guo et al., 2016). Por otro lado, la dificultad de crear mapas a escala mundial requiere de la cooperación entre vehículos, por ello surge el concepto Multi-vehicle SLAM (Nettleton et al., 2006), donde es posible diferenciar entre los denominados como centralizados (Kehoe et al., 2015), cuando parte de la computación de los algoritmos individuales de SLAM se extienden a un servidor; mientras que por otro lado, se clasifican como descentralizados (H. Li y Nashashibi, 2013; Indelman et al., 2014) si los datos son intercambiados directamente entre vehículos sin la necesidad de una infraestructura dedicada, haciendo uso de las capacidades de las comunicaciones V2X.

En cuanto a la tecnología que se puede emplear, los sensores LiDAR 3D son ampliamente utilizados ya que, debido a la información tridimensional del entorno, la extracción de características puede resultar más robusta en comparación con otros sensores, tanto de visión como de distancia. Sin embargo, la mayoría de estos métodos sufren de una gran carga

computacional. Aun así, si el entorno presenta una infraestructura representativa, permite simplificar estos algoritmos. En concreto, se desarrolla a continuación un algoritmo de Visual SLAM basado en LiDAR 3D con un enfoque clásico. Haciendo uso de un solo láser, con el que se extraen planos perpendiculares al suelo como *landmarks*, se consigue un buen desempeño en entornos tanto interiores como exteriores con la limitación de que estas zonas deben estar caracterizadas por una infraestructura específica. Para tratar dotar de mayor robustez al método, se exploran distintas alternativas que mejoren la salida del sistema sobre todo cuando la extracción de características del entorno no se logra con suficiente certeza. En estas situaciones podrían suponer una ventaja utilizar métodos alternativos basados en inteligencia artificial, por ello, se abordará esta metodología en profundidad en apartados posteriores.

3.2.2. Desarrollo de la metodología del algoritmo clásico de Visual SLAM

El algoritmo desarrollado toma los planos verticales del entorno como elementos característicos, por lo que está orientado hacia escenarios particulares donde exista un alto número de estos elementos. Se describe a continuación según las etapas que componen un algoritmo genérico de SLAM:

- Extracción de elementos característicos
- Asociación de datos
- Estimación y corrección de la posición
- Reconstrucción del mapa

3.2.2.1. Extracción de elementos característicos

En general, los elementos del entorno que se extraigan para ser utilizados como *landmarks* deben tener ciertas propiedades que los diferencien del resto de elementos de la escena. Deben ser geoméricamente representativos, de tal manera que permita una alta repetibilidad en la identificación desde distintas observaciones. Por otro lado, deben permitir una identificación única, es decir, además de ser fácilmente identificable, esta debe tener una representación bien definida en el espacio para que el emparejamiento durante distintos *frames* sea unívoco y preciso. Esta última característica está íntimamente relacionada con la exactitud global del algoritmo. Además de ello, evidentemente, el elemento característico elegido debe ser estático, para poder realizar la etapa de emparejamiento (*matching*) correctamente y que sea posible identificar el elemento característico frecuentemente en el entorno, ya que un mayor número de *landmarks* extraídos, favorecerá a la convergencia a un resultado preciso.

Tanto en entornos urbanos como industriales, para los cuales está enfocado el uso de este método, se pueden encontrar numerosos muros, columnas o secciones planas. Debido a la dificultad de detectar las esquinas verticales con la información dispersa del LiDAR, se detectarán planos verticales, a partir de los cuales, se extraerán las intersecciones entre ellos. Por tanto, estas dos características principales del entorno se seleccionan como *landmarks* al cumplir con los criterios anteriormente definidos y permiten una correcta asociación a lo largo de la trayectoria seguida. A partir del *matching* de estos elementos, es posible estimar la evolución de las posiciones del vehículo a partir del movimiento relativo de los elementos característicos detectados.

- Método 1: Identificación a partir del análisis por capas del láser:

Para la identificación del elemento característico elegido se plantea, en primer lugar, implementar el coeficiente propuesto en J. Zhang y Singh (2016), el cual estudia la monotonía en cada capa del láser, logrando evaluar la suavidad de esta entre puntos adyacentes. De esta manera se pretende detectar esquinas o cambios drásticos de curvatura dentro de una misma capa. Sin embargo, se descarta esta primera aproximación ya que el coeficiente es demasiado sensible a discontinuidades y, por tanto, extrae un número demasiado alto de *landmarks* en entornos reales, lo que dificulta su ajuste e implementación.

Manteniendo este enfoque de análisis por capas del sensor, se estudia extraer estos cambios bruscos entre puntos consecutivos, comparando las pendientes entre líneas constituidas por estos. Para ello, se calcula, centrado en cada punto (pivote), el ajuste a una recta formada por un conjunto de puntos adyacentes a este. Posteriormente, comparando las pendientes de las rectas obtenidas, se agrupan en función de la similitud de este valor, identificando así los puntos que pertenecen a una misma superficie plana. Tras esta agrupación, se descartan en primer lugar los *outliers* que por su pendiente hayan quedado aislados y, posteriormente, también aquellos conjuntos formados por un número insuficiente de puntos. Los *clusters* restantes corresponden a grupos de puntos pertenecientes a tramos continuos y sin cambios bruscos de curvatura. Este proceso se repite por cada capa del LiDAR y, una vez se han extraído todos los tramos rectos válidos, se comparan entre capas para encontrar posibles emparejamientos entre una y otra. Finalmente, una vez se obtiene esta correspondencia, es posible obtener la traza del plano que representa a cada agrupación de rectas. En la figura 3.12 se muestra un ejemplo de la detección de estas superficies en una zona interior donde existen columnas, paredes y vehículos. Se puede observar que este método identifica varias superficies y cómo las trazas de estos planos se intersecan en varios puntos del entorno.

Figura 3.12: Ejemplo de identificación de superficies planas mediante Método 1.

- Método 2: Identificación de superficies planas a partir del análisis de nubes de puntos:

De forma alternativa al Método 1, se ha valorado la extracción de superficies planas a través del tratamiento directo de la nube de puntos 3D, evitando así, que el método dependa en exceso de la variabilidad los puntos cuando estos se identifican a distancias mayores. A partir la utilización del algoritmo M-SAC (Torr y Zisserman, 2000), una variante del algoritmo RANSAC, para el ajuste a superficies planas con nubes de puntos, se extraen de forma iterativa los planos del entorno, que pasan a formar parte del conjunto de elementos característicos.

Se obtiene, por tanto, un set de superficies planas y solo aquellas que cumplen las condiciones de perpendicularidad con el plano horizontal y un error de ajuste por debajo de un umbral asignado, son de interés para la siguiente etapa de *matching*, el resto son descartados. En la figura 3.13 se muestra como el algoritmo utilizado de ajuste a plano identifica correctamente un conjunto de superficies planas (cada una de un color distinto) sobre la nube de puntos (en rojo). Este conjunto de superficies cumple con las dos condiciones mencionadas anteriormente, la perpendicularidad con el plano del suelo y un error de ajuste máximo admisible.

Figura 3.13: Ejemplo de identificación de superficies planas mediante Método 2.

Las secciones planas extraídas que forman parte del conjunto de *landmarks* son proyectadas sobre el plano horizontal pasando de la representación 3D del entorno, a un mapa 2D donde son representadas las características identificadas (figura 3.14). Al igual que en el Método 1, se trabajará con las trazas de los planos.

Figura 3.14: Identificación de los *landmarks* elegidos (Fuente: Im et al. (2016).)

Una vez extraídas las trazas de los planos, ya sea con un método u otro de los descritos anteriormente, se calculan las posibles intersecciones entre ellas para utilizar también estos puntos como *landmarks*. Algunos de estos puntos corresponderán a esquinas reales entre planos (paredes, columnas, etc.) si se encuentran en el campo de visión del LiDAR, mientras que otros serán puntos virtuales, resultado de la prolongación de la traza calculada. Las intersecciones localizadas a más de 100 metros del sensor son descartadas. El número máximo de intersecciones en función del número de trazas empleadas está determinado por la ecuación 3.12.

$$n_{intersections} = \frac{n_{trazas}}{2} \cdot (n_{trazas} - 1) \quad (3.12)$$

3.2.2.2. Asociación de datos

El objetivo de esta etapa es emparejar los *landmarks* observados previamente con los de la identificación del *frame* actual. También se suele conocer a este paso como “reobservación de *landmarks*”.

En primer lugar, se implementa el algoritmo Iterative Closest Point (ICP) (Besl y McKay, 1992; Yang y Medioni, 1992), como método de asociación de los *landmarks* extraídos en el paso anterior. A pesar de ser un método iterativo, ha demostrado un gran desempeño en diversas aplicaciones, siendo uno de los más ampliamente empleados para asociar dos nubes de puntos con diferentes orientaciones: una de ellas permanece fija, mientras que la otra, de forma iterativa, es transformada para encontrar el mejor emparejamiento (combinando translación y rotación) que minimice la métrica del error elegida. Sin embargo, este método no proporciona una buena precisión cuando se lleva a cabo la asociación de las dos nubes de puntos que tan solo contienen los *landmarks* extraídos anteriormente, por lo que se propone adaptar este método a las condiciones del problema actual.

De este modo, el nuevo método se basa en el método anterior (ICP), pero simplificado y optimizado para el *matching* de los *landmarks* identificados en la etapa previa. Se realiza una primera aproximación desplazando y girando los *landmarks* del *frame* anterior con la última estimación que se haya calculado del movimiento relativo. Respecto a la translación, los desplazamientos serán Δx_0 y Δy_0 ; mientras que la rotación vendrá dada por $\Delta \varphi_0$. Esto

es posible debido a que, por la dinámica del sistema, no puede existir grandes diferencias de movimiento entre cálculos consecutivos. A continuación, se busca el punto característico más cercano a cada uno de los *landmarks* de la observación actual minimizando el error en distancia. Si se obtiene un error por debajo del umbral establecido, ambos puntos característicos se considerarán el mismo.

La identificación de elementos característicos en cada *frame* no se realiza con la misma certeza para cada uno de ellos, por ello es necesario discernir entre los que aportan información para la estimación de la posición, de los que no. Al final de este paso del algoritmo, los *landmarks* que no han sido asociados se eliminan de la lista de elementos clave. De esta manera, aquellos que no consiguen ser relacionados en nuevas observaciones, se consideran que no son útiles para la estimación del movimiento.

3.2.2.3. Estimación y corrección de la posición

A partir del paso anterior, se procede a la estimación del movimiento relativo del vehículo. Para ello se han tenido en cuenta dos alternativas diferentes que calculan el desplazamiento y la rotación entre dos observaciones consecutivas.

- Método A: Resolución de un sistema lineal de ecuaciones

Este método calcula la matriz de rotación y translación del movimiento del vehículo a partir de resolver un sistema lineal de ecuaciones. En primer lugar, se calcula el movimiento relativo entre aquellos puntos que han sido detectados en dos *frames* consecutivos y han sido correctamente asociados en el paso anterior. Para ello, se calculan los desplazamientos longitudinales y laterales, así como el ángulo de guiñada relativo a partir de la resolución del sistema de ecuaciones 3.13.

$$\begin{bmatrix} \cos \Delta\varphi & -\sin \Delta\varphi \\ \sin \Delta\varphi & \cos \Delta\varphi \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \quad (3.13)$$

Donde x_1 y y_1 corresponden a las coordenadas de los puntos en la observación actual, mientras que x_0 y y_0 , son las coordenadas en el *frame* anterior y $\Delta\varphi$, Δx y Δy , las incógnitas del sistema.

El anterior sistema de ecuaciones es no lineal y, por tanto, sería necesario recurrir a métodos numéricos para resolverlo. Sin embargo, el tiempo entre observaciones es reducido (0,1 segundos en esta implementación), y como se adelantaba anteriormente, por la dinámica del sistema, se podría considerar que la variación del ángulo de guiñada $\Delta\varphi$ tendrá valores pequeños. De esta manera, se puede considerar simplificar y convertir el sistema en uno lineal (ecuación 3.14).

$$\begin{bmatrix} 1 & 0 & -y_0 \\ 0 & 1 & x_0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\varphi \end{bmatrix} = \begin{bmatrix} x_1 & -x_0 \\ y_1 & -y_0 \end{bmatrix} \quad (3.14)$$

- Método B: Cálculo independiente de translación y rotación

Para evitar realizar la simplificación indicada en el método anterior, se propone una alternativa para la resolución del cálculo del movimiento relativo. Este método, a diferencia del anterior, calcula la rotación y la translación de forma independiente.

Al igual que en el Método A, se parte del último movimiento relativo calculado como primera aproximación. Sin embargo, en este caso, para calcular la translación entre el *frame* actual y el anterior, se realiza a partir de las intersecciones extraídas como *landmarks*, mientras que la rotación se calcula a partir de las trazas de los planos utilizadas como *landmarks* (figura 3.14). Este método hace un uso en paralelo de los *landmarks* según la tipología (puntos o líneas) para calcular cada variable del movimiento que ha realizado el vehículo entre dos instantes de tiempo consecutivos. La principal justificación de este método se basa en que una medición precisa de ángulos se realiza entre líneas y por ello, las trazas de los planos son utilizadas para calcular la rotación, $\Delta\varphi$. De igual manera, los puntos de intersección sirven como referencia para calcular los desplazamientos en las dos dimensiones (Δx_0 y Δy_0).

Figura 3.15: Cálculo de las variables del movimiento relativo según el Método B.

En este punto, el resultado obtenido es la estimación de un movimiento relativo por cada pareja de *landmarks* asociada. Sin embargo, el objetivo final de esta etapa del algoritmo es el posicionamiento real del sistema (el vehículo). Este es función de la probabilidad de que cada estimación obtenida sea correcta. Por ello, es necesario realizar una etapa de actualización o corrección de la posición, tal como se describe en el estado del arte.

Independientemente del método seguido para la estimación del movimiento (Método A o B), en este desarrollo, la “Corrección de la posición” se realiza con un enfoque basado en optimización donde, tras obtener el conjunto de posibles posiciones del vehículo por cada emparejamiento realizado, se calcula la media de estas estimaciones, priorizando en función de la certeza con la que se haya realizado el emparejamiento y la similitud de valores obtenidos. La certeza de la asociación de *landmarks* está relacionada con aspectos geométricos como por ejemplo la distancia a la que se han identificado o el ángulo entre las trazas que definen una intersección. Estos aspectos se describen en profundidad en el subapartado 3.2.2.5.

3.2.2.4. Reconstrucción del mapa

Finalmente, una vez se obtiene la matriz de translación y rotación relativa a la nueva posición del vehículo, es posible reconstruir el mapa. Para ello, se desplaza y rota la nube de puntos asociada a cada matriz calculada, generando de esta manera el mapa del entorno. En la figura 3.16 se muestra un ejemplo de dicha reconstrucción sobre un entorno interior en el que se podían distinguir diferentes paredes, vehículos estacionados y columnas.

Figura 3.16: Ejemplo de reconstrucción del mapa en entorno interior.

Esta reconstrucción servirá, por un lado, para obtener el posicionamiento global del vehículo en el entorno por donde circula y, por otro, se empleará para valorar la precisión de la estimación de la trayectoria. Además, la reconstrucción obtenida es utilizada para comparar los diferentes métodos alternativos implementados y también evaluar las mejoras sobre el algoritmo que se describen a continuación.

3.2.2.5. Propuesta de mejoras del algoritmo

La durante la etapa de “Asociación de datos” se crea un set con las características principales extraídas durante la etapa previa “Extracción de elementos característicos”. Al inicio, esta lista está vacía y se van agregando puntos clave a medida que el vehículo los identifica en el entorno. A pesar de que la extracción de características sea robusta, existirán *landmarks* con mayor o menor afinidad para ser emparejados en observaciones sucesivas. Este problema puede afectar en gran medida a la precisión del posicionamiento. Por ello, es fundamental durante la asociación de *landmarks*, considerar aquellos que reporten cierta fiabilidad.

Por otro lado, este algoritmo se basa en la extracción de planos perpendiculares a uno de referencia que corresponde con el suelo. Según el posicionamiento del láser sobre el vehículo, este plano de referencia puede quedar orientado con cierta desviación respecto al sistema de coordenadas del láser e interferir en la identificación de características. Por este motivo,

algunos algoritmos de Visual-SLAM, como en Shan y Englot (2018), requieren introducir el vector normal donde se sitúa el plano del suelo. Sin embargo, en el desarrollo propuesto se pretende corregir de forma automática la orientación relativa entre láser y suelo cuando estas desviaciones son pequeñas.

Estas dos razones llevan a plantear dos mejoras sobre el algoritmo descrito previamente:

- Ponderación de puntos y líneas usadas como *landmarks*.
- Corrección de la posición del láser sobre el vehículo.

La primera mejora propuesta trata de asignar un valor de confianza a cada punto o línea que se extrae como elemento característico. Esta ponderación se lleva a cabo, incrementando el número de estimaciones asociadas al *matching* entre dos *landmarks* en función del valor de confianza asignado. Así, en el proceso de “Corrección de la posición”, la media de todas las estimaciones obtenidas queda ponderada según los coeficientes de confianza que se describen a continuación.

En primer lugar, a partir de las intersecciones entre trazas de los planos identificados, se extraen los puntos utilizados como *landmarks*. Se establece el primero de los coeficientes de confianza en función de la distancia a la que se encuentran estas intersecciones del láser, ya que, a mayor distancia, menor precisión en su identificación. Así, este coeficiente se define como la inversa a la distancia de la siguiente forma:

$$C_{puntos_1} = \frac{1}{\sqrt{x_p^2 + y_p^2}} \quad (3.15)$$

Donde x_p y y_p son las coordenadas del punto de intersección.

Además, cuando el ángulo formado entre las dos trazas es mayor, la localización del punto intersección está definida con mayor precisión. Este indicador se cuantifica mediante el segundo coeficiente:

$$C_{puntos_2} = 1 - \frac{2 \cdot (\frac{\pi}{2} - \alpha)}{\pi} \quad (3.16)$$

Donde α , es el ángulo formado por las líneas que definen el punto de intersección.

Finalmente, se define un tercer coeficiente como la combinación de los dos anteriores:

$$C_{puntos_3} = C_{puntos_1} \cdot C_{puntos_2} \quad (3.17)$$

Por otro lado, se ponderarán también las trazas usadas como *landmarks*. Para ello, se tendrá en cuenta dos factores: el número de puntos contenidos en la superficie plana asociada a la traza y el error de ajuste a dicha superficie. Un mayor número de puntos asociados a una superficie definirá en mayor medida a esta. Por otro lado, si la desviación relativa entre puntos es baja, se traduce en una buena definición del plano y un error de ajuste bajo. Por tanto, se define un coeficiente en función del valor del error cuadrático medio (RMSE) obtenido durante el ajuste (definido como ε) y el número de puntos que contiene (n_{puntos}) de

la siguiente manera:

$$C_{trazas} = n_{puntos} \cdot (1 - \varepsilon) \quad (3.18)$$

En cuanto a la segunda mejora: “Corrección de la posición del láser sobre el vehículo”, se trata de mejorar la identificación y extracción de planos verticales. Tal como se comentaba anteriormente, pequeñas diferencias de orientación entre el sistema de referencia del láser y el plano considerado de referencia pueden afectar a la identificación de los planos verticales. También, debido a terrenos irregulares o inclinados (por ejemplo, rampas de acceso a parking subterráneo), se pueden producir desviaciones en la perpendicularidad entre los planos verticales y el plano que identifica el suelo.

Para evitar este efecto, durante la primera extracción de planos se identifican las orientaciones predominantes de los vectores normales tal y como se desarrolló en el apartado 2.1 mediante el uso de una esfera de radio unitario donde se proyectan los vectores normales del entorno. En este caso, el problema se simplifica en gran medida, ya que solamente requiere encontrar las orientaciones de los planos verticales. Estos se dispondrán sobre la esfera a lo largo de una de sus circunferencias máximas, tal como se puede observar en la figura 3.17. Los vectores correspondientes a esta disposición tendrán cualquier orientación en dos de sus ejes, pero su tercera componente será común en todos ellos, por lo que finalmente se calcula el plano perpendicular a estos. De esta manera, el plano calculado servirá de referencia y es perpendicular a los planos que posteriormente se extraen como *landmarks*.

Figura 3.17: Identificación de orientaciones predominantes de planos verticales. La zona delimitada por la línea roja identifica la disposición de los vectores normales de los planos verticales.

3.2.3. Ensayos preliminares y configuración final del algoritmo

En una primera fase, se han llevado a cabo varias simulaciones en diferentes entornos ideales, modelados con paredes y columnas, y un ensayo preliminar en entorno real controlado.

El objetivo es evaluar el desempeño de las alternativas de los métodos propuestos en cada paso del algoritmo y valorar cuáles de las mejoras se implementarán en la versión final.

En cuanto al entorno real controlado, se trata de la nave industrial situada en las instalaciones del INSIA-UPM que se caracteriza por ser un entorno interior compuesto por paredes, columnas y vehículos apartados. Este escenario también se utilizará durante el análisis de resultados y comparativa con otros entornos reales, tal y como se describirá en el subapartado 3.2.4.

Tanto en los escenarios virtuales como en el ensayo preliminar en la nave del INSIA-UPM, se han establecido siete puntos de control localizados en diferentes puntos estratégicos a lo largo de la trayectoria seguida, donde se evaluarán dos indicadores de la precisión obtenida. Por un lado, la media de los errores cometidos en cada punto como la distancia máxima entre las observaciones de un mismo elemento. Por otro lado, el error relativo, como el cociente entre el error medio y la distancia total recorrida.

Comparación de métodos para extracción de elementos característicos:

En primer lugar, se analizan los dos métodos de extracción de *landmarks*, descritos en el 3.2.2.1. El primer método ("Método 1"), basado en el análisis de cada capa láser, comparaba la pendiente de las rectas formadas por puntos adyacentes. Sin embargo, debido a la variabilidad que ofrecen estas pendientes, cuando los puntos se encuentran a mayor distancia, no ofrece una identificación robusta de superficies planas, tal como se puede observar en la figura 3.18. Sin embargo, el "Método 2", procesando la nube de puntos directamente, permite un buen ajuste a estas superficies en cualquier circunstancia y con baja carga computacional.

Figura 3.18: Comparativa Método 1 (izquierda) y Método 2 (derecha), tanto en escenario ideal (arriba) como en entorno real (abajo). En rojo aparecen los puntos no identificados como planos.

Comparación de métodos para la estimación y corrección de la posición:

En segundo lugar, se estudia el método preferente para el cálculo de la estimación de la trayectoria, es decir, la matriz de translación y rotación del movimiento relativo entre observaciones. Para ello, en el subapartado 3.2.2.3, se proponía un método basado en la resolución de un sistema lineal de ecuaciones ("Método A") y, por otro lado, el cálculo independiente de la translación y la rotación en función del *landmarks* utilizado: puntos intersecciones o trazas de planos identificados, respectivamente ("Método B").

En la figura 3.19 se puede observar el comportamiento de cada método en un entorno virtual, evaluando el error en los siete puntos de control del entorno designados con un cuadrado amarillo.

Figura 3.19: Comparación del Método A (izquierda) y Método B (derecha) para la estimación de la trayectoria en entorno ideal.

Los puntos de control indican que la superposición de las nubes de puntos en observaciones sucesivas se realiza correctamente según el cálculo del "Método B". Mientras que con el "Método A" no se llega a obtener la misma precisión debido a las simplificaciones que implica este método y la integración de estos errores. A raíz de esta comparativa, se implementa el segundo método en el algoritmo final, ya que se espera que este dé los mejores resultados en entornos reales.

Evaluación de las propuestas de mejora:

Por último, una vez establecidos los métodos que se utilizarán en cada paso del algoritmo final, se analizan las propuestas de mejora del algoritmo descritas en el subapartado 3.2.2.5, esta vez solo en el entorno real controlado de la nave del INSIA-UPM. En concreto, para verificar las mejoras propuestas sobre la ponderación de puntos y líneas usadas como *landmarks* y la corrección de la posición del láser sobre el vehículo, se lleva a cabo un ensayo donde la trayectoria seguida por el vehículo describe un tramo recto, seguido de una curva de 90° hacia la derecha. Cabe recordar que el entorno está constituido por diversos muros, columnas y vehículos, sobre el cual se han establecido siete puntos de control distintos para evaluar la precisión que aporta cada mejora (figura 3.20).

Figura 3.20: Estimación de la trayectoria en Nave del INSIA-UPM.

Los resultados obtenidos aplicando los coeficientes de ponderación o realizando los cálculos necesarios para la corrección de la inclinación del láser, se resumen en la tabla 3.2.

Tabla 3.2: Resultados de los métodos de mejora propuestos

Método de mejora propuesto	Coefficiente aplicado / Cálculo realizado	Error medio	Error relativo
Caso inicial sin mejoras	-	0,76 m	1,65 %
Ponderación puntos según distancia	C_{puntos_1}	0,61 m	1,32 %
Ponderación de puntos según ángulo entre líneas	C_{puntos_2}	0,60 m	1,30 %
Ponderación combinada	C_{puntos_3}	0,59 m	1,28 %
Ponderación de trazas de los planos	C_{tramas}	0,86 m	1,86 %
Corrección inclinación con plano de referencia	<i>Cálculo vectores normales</i>	0,57 m	1,24 %
Corrección combinada (excepto ponderación de líneas)	C_{puntos_3} <i>Cálculo vectores normales</i>	0,54 m	1,17 %

Por tanto, el modelo final del algoritmo integra los métodos seleccionados anteriormente y las mejoras de ponderación de los puntos combinada y la corrección de la inclinación. Queda descartada, por tanto, la ponderación de las trazas según el RMSE obtenido durante el ajuste, debido a que tras el ensayo introduce un error adicional.

3.2.4. Pruebas y resultados

3.2.4.1. Escenarios seleccionados

Una vez definido la estructura final de algoritmo y para poder validarlo, se ha utilizado un vehículo instrumentado donde el sensor LiDAR VLP-16 ha sido posicionado en el techo y se han llevado a cabo los ensayos en tres escenarios distintos:

- Nave industrial en las instalaciones del INSIA-UPM figura 3.21.a: Se trata del escenario descrito anteriormente utilizado durante los ensayos preliminares. Es un escenario adecuado para el algoritmo ya que dispone de un alto número de muros y columnas, por lo que se podrán extraer una cantidad considerable de *landmarks*. En este escenario se han llevado a cabo dos maniobras: una maniobra recta y otra con un giro de 90°.

- Parking interior (figura 3.21.b): Este escenario presenta un número elevado de columnas. Sin embargo, se compromete la correcta identificación de estas debido a que, al ser muy estrechas, el número de puntos que las definen es menor y, por tanto, se dificulta el ajuste a un plano. Además, muchos de los puntos láser son perdidos, bien en el techo del propio parking o bien, en vehículos aparcados.
- Entorno urbano (figura 3.21.c): El ensayo se ha realizado en una calle con un reducido ancho de calzada. Cuenta con intersecciones con otras calles, lo que puede favorecer la identificación de elementos clave. Este tipo de escenario es adecuado para el algoritmo para verificar su aplicabilidad en entornos abiertos con cobertura GPS, pero donde la señal puede ser pobre o nula en algunos casos.

Figura 3.21: Escenarios de ensayo. a) Nave industrial en las instalaciones del INSIA-UPM, b) parking interior, c) entorno urbano.

3.2.4.2. Resultados en entornos reales

De igual manera que en el ensayo preliminar, se seleccionan varios puntos de control estratégicos a lo largo de las zonas de ensayo para el cálculo del error cometido y, por tanto, la precisión de la reconstrucción del mapa. En la tabla 3.3 se pueden observar los indicadores del error cometido para cada ensayo. En las figuras 3.22-3.25, se muestra la reconstrucción resultante.

Tabla 3.3: Resultados en entornos reales.

Escenario	Distancia recorrida	Error medio	Error relativo
Nave INSIA-UPM (Trayectoria rectilínea)	64,99 m	0,35 m	0,54 %
Nave INSIA-UPM (Trayectoria en forma de L)	46,13 m	0,54 m	1,17 %
Parking interior	11,94 m	0,15 m	1,24 %
Entorno urbano	70,91 m	0,27 m	0,38 %

Figura 3.22: Nave INSIA-UPM (Trayectoria rectilínea) **Figura 3.23:** Nave INSIA-UPM (Trayectoria en forma de L)

Figura 3.24: Parking interior.

Figura 3.25: Zona urbana.

El menor error relativo ha sido obtenido en el escenario de INSIA-UPM (test con trayectoria rectilínea) y en el entorno urbano. Ambos escenarios se caracterizan por disponer de grandes superficies planas como son las paredes de la nave industrial en INSIA-UPM o las fachadas de edificios, en el caso de la zona urbana. En este sentido, el algoritmo estima una trayectoria y una reconstrucción del mapa más precisas cuando los elementos característicos son identificados con mayor certeza y, por tanto, un mayor valor del coeficiente de confianza definido previamente.

Por otro lado, cabe destacar que en los resultados obtenidos en el caso del parking interior, se obtiene el menor error medio y, sin embargo, el mayor error relativo. Esto se debe a que, a pesar de que las desviaciones en los puntos de control durante la reconstrucción del mapa han

sido pequeñas, cuando se compara frente a la distancia recorrida (reducida en este caso), da lugar al escenario con error relativo más alto. Además, como se comentaba anteriormente, el alto número de columnas, pero de reducidas dimensiones, dificulta la certeza de identificación de los planos.

3.2.5. Conclusiones finales

Actualmente son muchas las soluciones propuestas al problema de SLAM, aunque en general, son algoritmos sofisticados que sufren de una alta carga computacional o no garantizan la precisión en cualquier situación. Sin embargo, ciertos entornos con una infraestructura específica pueden dar lugar a un buen desempeño de algoritmos simplificados que requieren una carga computacional más liviana. Bajo estos antecedentes, se ha desarrollado un algoritmo de Visual SLAM basado en sensores LiDAR 3D. Su funcionamiento está orientado a entornos interiores o exteriores que dispongan de un elevado número de superficies planas (muros, columnas, fachadas, etc.), como pueden ser parkings interiores, calles estrechas o naves industriales. El algoritmo se vale de esta infraestructura para la estimación del movimiento relativo del vehículo.

El método empleado utiliza exclusivamente datos LiDAR por lo que permite que sea totalmente autónomo para el cálculo del desplazamiento y la guiñada del vehículo. De este modo, no depende de señales externas como GPS o sensores inerciales. Gracias a esto, el algoritmo puede proporcionar una estimación aceptable con una baja carga computacional en aquellos escenarios, interiores o exteriores, para los que está dirigido.

La principal limitación que presenta el algoritmo es la reducción en la precisión de la estimación de la trayectoria cuando no se dispone de una alta cantidad de *landmarks* en el entorno. En escenarios con bajo número de superficies planas, el algoritmo encuentra dificultad para conseguir realizar el cálculo de posicionamiento. Para tratar de solucionar la escasez de *landmarks* otros autores suelen recurrir a una fusión sensorial. De esta manera, se pueden contrarrestar los inconvenientes de cada sensor implicado en la fusión y obtener un mayor número de características singulares del entorno. Sin embargo, para continuar con la premisa de mantener una baja carga computacional se puede estudiar la viabilidad de utilizar modelos de inteligencia artificial que, una vez entrenados, ofrecen un gran rendimiento de procesamiento.

En apartados sucesivos se abarca este enfoque y se estudia en profundidad las capacidades que pueden ofrecer respecto a la precisión en la estimación de la trayectoria del vehículo.

3.3. Aplicación de modelos basados en redes de convolución para estimación de la odometría

En los apartados 3.1 y 3.2 se ha abordado, desde un punto de vista clásico, el problema del posicionamiento del vehículo autónomo en un entorno desconocido. Analíticamente, se ha tratado de resolver el problema de SLAM y la estimación de la odometría utilizando para ello la información del proporcionada por sensores LiDAR 3D. Sin embargo, en la mayoría de los

casos, tanto de la literatura como los propuestos en esta tesis, bien no obtienen resultados definitivos o sufren de una alta carga computacional que impide su funcionamiento en tiempo real.

Por otro lado, los nuevos desarrollos en el campo de la Inteligencia Artificial han dado lugar a avances significativos en la disciplina conocida como Aprendizaje Automático o *Machine Learning*, donde se dota a un sistema (software, algoritmo, etc.) de la capacidad de aprender una tarea sin estar explícitamente programado para ello (Samuel, 1959). Asimismo, hoy en día, la facilidad de disponer y procesar grandes cantidades de datos ha dado lugar al avance de técnicas de Aprendizaje Profundo (*Deep Learning*) que han demostrado en los últimos años su validez en campos muy diversos, uno de ellos, la percepción del entorno.

Por ello, en este apartado se valorará la capacidad de un sistema basado en modelos de *Machine Learning* para estimar la odometría del vehículo a partir de datos LiDAR.

3.3.1. Introducción

Las nuevas técnicas de *Deep Learning* están demostrando, cada día más, ser capaces de resolver una gran variedad de problemas complejos. Entre ellas, cabe destacar su aplicación en el desarrollo del vehículo autónomo, ya sea desde la segmentación semántica del entorno, la detección de obstáculos, el cálculo de la estrategia de conducción o en la toma de decisiones ante cualquier eventualidad.

Sin embargo, no siempre es recomendable aplicar este enfoque para dar solución a cualquier problema. Por ello, las técnicas de Aprendizaje Automático deben ser utilizadas en aquellos casos en los que el condicionamiento del problema lo permita. En este sentido, el *Machine Learning* proporciona herramientas adecuadas cuando la solución requiere demasiado ajuste manual o para problemas complejos, donde un enfoque tradicional no consigue aportar una solución definitiva. También su aplicación es apropiada en entornos cambiantes, donde se necesita llegar a una generalización de funcionamiento. Por otro lado, estas técnicas de inteligencia artificial ofrecen una serie de limitaciones. En concreto, aquellas basadas en *Deep Learning* requieren un gran número de datos que posibiliten modelar el problema y, según el paradigma de aprendizaje, que dichos datos estén relacionados con el resultado esperado, lo cual no siempre es accesible. Además, la utilización de estos algoritmos da lugar a una pérdida del entendimiento a nivel interno sobre la resolución del problema. Para evitar este fenómeno de "caja negra", existen recientes estudios en los que se trata de establecer pautas para alcanzar una interpretación del proceso interno de estos algoritmos (Doshi-Velez y Kim, 2017).

En el caso particular del vehículo autónomo, la percepción del entorno puede ser sensible a numerosos factores externos. Cambios en las condiciones meteorológicas o imprevistos durante la conducción, la incertidumbre en la previsión de la trayectoria de peatones o ciclistas, son algunos ejemplos donde una pequeña variación puede dar lugar a resultados muy dispares en los algoritmos programados en el vehículo. La dependencia de un posicionamiento preciso sobre estos factores, permite que este tipo de técnicas de *Machine Learning* tengan cabida en el problema que se aborda en este capítulo.

Se desarrolla a continuación un método de Odometría Visual basado en modelos de *Ma-*

chine Learning usando un sensor LiDAR 3D como única información de entrada. A partir de las nubes de puntos obtenidas, se pretende inferir la velocidad y la variación de guiñada del vehículo y, así, poder estimar la trayectoria del mismo. De igual modo, se comparan distintas topologías de modelos de *Machine Learning* para estudiar su adecuación al problema. En un estudio más profundo sobre el ajuste óptimo, se valoran diferentes alternativas respecto al tratamiento previo de datos, ajuste de parámetros del modelo y cómo influyen en el rendimiento y precisión de la estimación.

3.3.1.1. Importancia del modelo en técnicas basadas en *Machine Learning*

La conducción autónoma pretende dar solución a una amplia gama de problemas dinámicos que se pueden considerar muy sensibles a las variaciones de las condiciones iniciales y, por tanto, difíciles de modelizar. A lo largo de la historia, mediante el conocimiento adquirido a través del desarrollo de la ciencia, se ha tratado de buscar patrones que puedan explicar estos modos de funcionamiento caóticos, creando para ello modelos. Estos modelos son una reconstrucción conceptual y simplificada de una realidad más compleja. En general, un modelo trata de encontrar un equilibrio entre representar fielmente dicha realidad y ser simple para poder aplicarlo. Unos de los objetivos básicos del *Machine Learning* es encontrar estos modelos que permiten conceptualizar, predecir, generalizar y aprender de esta realidad.

Para el diseño y construcción de un modelo basado en *Machine Learning*, existen tres elementos clave relacionados transversalmente, que se utilizarán a lo largo de este apartado:

- **Datos:** A partir de los cuales se extrae la información de la realidad para construir el modelo.
- **Parámetros:** Los valores que se modifican para ajustar el modelo a los datos. Como se estudiará, no siempre es recomendable dotar al modelo de demasiada flexibilidad cuando se persigue que este pueda generalizar correctamente. La tarea principal de los algoritmos de *Machine Learning* es encontrar los valores óptimos de los parámetros a partir de los datos.
- **Optimización:** Definir una función de error, indica cómo el modelo se ajusta a los datos. En aprendizaje supervisado, se busca la relación existente entre unos datos de entrada y sus correspondientes salidas. La señal de error se calculará en función del ajuste de estos datos e irá recalculando los parámetros del modelo durante el proceso de optimización (también llamado, entrenamiento o ajuste del modelo).

Estos tres elementos son básicos y están relacionados íntimamente entre sí para definir el mejor modelo representativo del problema.

Por otra parte, los modelos que mayor impacto están teniendo dentro de la percepción del entorno del vehículo autónomo se basan en redes neuronales convolucionales (*Convolutional Neural Networks*, CNN). Esta técnica es la más utilizada para el reconocimiento de imágenes debido al potencial y la efectividad que han demostrado en diversas aplicaciones en este campo. Así, han permitido solucionar parcial o totalmente algunos de los problemas planteados sobre todo en lo referido a la identificación de objetos (Christiansen et al., 2016).

Por tanto, en el desarrollo del este apartado, se describen las alternativas valoradas en cada uno de los tres elementos descritos anteriormente orientados a encontrar modelos CNN.

3.3.1.2. Trabajos previos sobre Odometría Visual

En el apartado 3.2, se estudió la capacidad que tienen los sensores LiDAR 3D para proporcionar información del entorno y que esta pueda ser útil para el cálculo del posicionamiento a través de algoritmos de Odometría Visual o SLAM. Asimismo, se definió el concepto de Odometría Visual como la estimación del movimiento relativo entre observaciones consecutivas, es decir, considerando una consistencia local. De esta manera, no sería posible obtener una optimización global, como ocurre cuando se aplican técnicas de *loop-closure* que minimizan el error de deriva generado.

A pesar de ello, las técnicas de Odometría Visual constan de una serie de ventajas, donde cabe destacar un mayor tiempo de *dead-reckoning* preciso frente a otros sistemas de localización, y su versatilidad de implementación por lo que se encuentran aplicadas en un amplio marco de la robótica móvil. En este sentido, han sido implementadas en misiones espaciales a Marte (Yang Cheng et al., 2005) donde los vehículos de exploración, también conocidos como *rovers*, mandados en 2003, debían explorar la geología y superficie del planeta. También, en proyectos como el que presentan Scaramuzza y Siegwart (2008) donde se implementa en vehículos terrestres para entornos exteriores empleando visión artificial. Se extraen, en primer lugar, características principales del entorno que se emparejan empleando descriptores SIFT, para finalmente filtrar los *outliers* mediante el uso del algoritmo RANSAC.

Por otro lado, se destacan varios desarrollos basados en sensores LiDAR 3D, como V-LOAM (J. Zhang y Singh, 2016), IMLS-SLAM (Deschaud, 2018) y MC2SLAM (Neuhaus et al., 2019) situados en las primeras posiciones del ranking de odometría de KITTI Dataset. En el caso del V-LOAM, se trata del método *state-of-the-art* para la estimación del posicionamiento basado en sensores LiDAR. Su método logra una gran precisión con una baja carga computacional. En este sentido, consigue reducir el error de deriva implementando dos algoritmos que se ejecutan simultáneamente. El primero realiza la estimación a alta frecuencia (10 Hz) pero con baja precisión y un segundo algoritmo, realiza el *matching* preciso de las nubes de puntos para reconstruir el mapa. De igual modo, el desarrollo IMLS-SLAM consigue una buena precisión, donde su principal aportación se encuentra en el método utilizado de *matching*. Para ello, realizan un estudio de los vectores normales del entorno que asocian a un modelo de superficies definido (una estrategia *scan-to-model*). A pesar de los resultados que obtienen, requiere una alta carga computacional con un tiempo de ejecución de 1,25 segundos. En el desarrollo presentado como MC2SLAM, se propone un método que obtiene un error de deriva bajo y consigue ejecutarse en tiempo real. Previo a realizar el *matching* entre observaciones, realizan una compensación del movimiento del vehículo sobre los datos obtenidos del LiDAR.

Como se ha mencionado previamente, modelos de *Machine Learning* basados en redes convolucionales han demostrado su eficacia en la identificación de obstáculos. Sin embargo, estas redes no tienen por qué ser usadas exclusivamente en aplicaciones de clasificación o segmentación, también pueden utilizarse en problemas de regresión. En este caso, la salida de la red es una señal continua y, por ello, es posible estudiar su utilización como método de Odometría Visual estimando los valores de translación y rotación relativa.

Concretamente, los modelos basados en CNN son adecuados en el desarrollo del vehículo autónomo debido, entre otras cosas, al gran volumen de datos obtenidos por los sensores y por la necesidad de funcionamiento en tiempo real. Todo ello se hace posible debido, por un lado, al avance en la capacidad de computación y, por otro, a la gran cantidad y calidad de *datasets* disponibles para procesos de aprendizaje de modelos. En este sentido, varios desarrollos han tratado de estudiar esta viabilidad, como en Konda y Memisevic (2015) donde se usa visión estéreo para la extracción de características, pero basando la estimación en valores discretos. También se destaca el desarrollo de Nicolai et al. (2016), donde se utiliza un sensor LiDAR, esta vez, con estimaciones de valores continuos. O bien, el desarrollo de S. Wang et al. (2017), donde tratan el problema mediante visión artificial y la combinación de redes de convolución y recurrentes (RNNs).

3.3.1.3. Organización del apartado

Al comienzo de este apartado se han destacado los tres elementos básicos en el desarrollo de modelos relacionados de forma transversal: los datos, los parámetros y la optimización. De este modo, el resto del apartado se estructura en base a estos conceptos donde se proponen y valoran las diferentes alternativas estudiadas en cada uno de ellos.

En primer lugar, se realiza un estudio de los datos disponibles que servirán de entrada al modelo desarrollado. Se proponen distintas configuraciones de representación del entorno para evaluar cuál es la más adecuada para obtener una mejor estimación de la red. Para ello, es necesario también definir los *datasets* utilizados para medir el rendimiento del modelo, así como los entornos urbanos donde se han llevado a cabo los ensayos.

Posteriormente, respecto a la valoración de modelos adecuados para el problema en cuestión, se realiza una comparación entre distintas topologías de Perceptrón Multicapa (MLP) y redes de convolución para verificar la aplicabilidad de esta herramienta en la estimación de odometría. Asimismo, es necesario establecer diversos hiper-parámetros y condiciones de entrenamiento para optimizar el ajuste del modelo.

En cuanto a los resultados obtenidos, estos pueden ser estudiados bajo dos enfoques diferentes: un enfoque de rendimiento y un enfoque geométrico. Por el lado del rendimiento, el uso de técnicas de *Deep Learning* requieren de mucho menos procesado una vez que el modelo está ajustado y, además, se facilita la paralelización de procesos. De igual modo, comparado con métodos clásicos, se evitan procesos de optimización sobre la propagación de incertidumbre, o bien, métodos iterativos como es el caso del ICP (Segal et al., 2009). Además, el hecho de utilizar solo datos LiDAR, empleado para la detección de obstáculos, da lugar a una optimización del uso de los recursos del vehículo. Por la parte del enfoque geométrico, la extracción de características y su posterior *matching* (dos de las fases básicas en técnicas de Odometría Visual clásica), no siempre es posible mediante el uso de LiDAR debido a posibles oclusiones que dan lugar a vistas parciales de objetos. Sin embargo, si se dispone de un *Ground Truth* preciso, el modelo no debería encontrar dificultades y generalizar de forma correcta.

El desarrollo ha sido implementado en vehículos instrumentados en situaciones reales para la adquisición de datos y generación de *datasets*. De la misma forma, también se han utilizado bases de datos de referencia para una mejor generalización del método. Los resultados de

los entrenamientos expuestos en este apartado se llevan a cabo bajo el mismo hardware y software. Se ha utilizado una máquina GNU/Linux con Intel Xeon CPU E3-1200 (family) v3/4th Gen y una NVIDIA GeForce GTX 980 Ti con 2816 núcleos CUDA y 6GB de RAM. Se ha programado utilizando Python 3.6, TensorFlow 1.15.2 de *Google* (Abadi et al., 2016) y Keras 2.3.1, con lo que se logra beneficiarse del uso de las GPUs.

3.3.2. Selección y tratamiento de datos

Tanto para la selección del modelo como para definir el proceso de aprendizaje, es necesario conocer previamente la tipología de los datos que se utilizarán y la cantidad de procesamiento necesario.

El objetivo es realizar un método de Odometría Visual basada en LiDAR 3D, por tanto, los únicos datos de entrada que representan el entorno serán en forma de nubes de puntos. Estos sensores generan una representación tridimensional del entorno por cada *frame* (observación) del láser rotativo con una distribución por capas. Según el modelo LiDAR que se emplee, el número de capas será distinto, así como la resolución horizontal y vertical. De cualquier forma, dicha nube de puntos contiene la información de las coordenadas cartesianas y la reflectividad de cada punto que haya incidido en alguna superficie, por lo que se trabaja en el dominio del espacio.

Por otro lado, la estimación de la odometría es un problema geométrico en el que se debe calcular la translación y la rotación relativa del movimiento. Por tanto, los datos iniciales correspondientes al espacio tridimensional del LiDAR 3D se ajustan, a priori, como un buen punto de partida para resolver un problema de este tipo. Estos datos estarán sujetos a un tratamiento previo para que puedan ajustarse a la entrada de las redes y que se pueda inferir el movimiento realizado entre dos observaciones consecutivas.

3.3.2.1. Datos de entrada y salida

El problema actual contempla el movimiento relativo entre dos observaciones, por tanto, los datos se deben presentar de forma que contengan la información de ambos *frames* a partir de los cuales extraer patrones que permitan inferir el movimiento.

Las nubes de puntos, las cuales están en un espacio tridimensional, se transformarán a un dominio matricial para poder trabajar con estas. De tal manera que se creará una matriz de datos por cada observación y a continuación, se apilará una tras la otra, consiguiendo una matriz multidimensional. A su vez, estas matrices pueden ser representadas como imágenes, las cuales contendrán en cada píxel la información correspondiente del entorno.

El tamaño de estas matrices será $H \times W \times C$, donde H y W serán las dimensiones alto y ancho respectivamente, correspondientes a la resolución de la imagen-dato; y C corresponderá a la suma del número de canales que utilice cada una. Estas dimensiones variarán en función de la representación utilizada tal y como se describe en el siguiente apartado.

Las salidas del modelo serán, como se ha indicado anteriormente, la velocidad lineal y la variación de guiñada. Cabe destacar que se valorará la estimación de los dos parámetros de forma conjunta por la red y de forma individual, es decir, ajustando un modelo específicamente

para estimar cada uno de los parámetros.

3.3.2.2. Representación de datos

Cabe mencionar que el procesamiento de los datos LiDAR para crear las imágenes-dato, se realiza en paralelo al ajuste de los hiper-parámetros y el entrenamiento del modelo, que se estudiará en el subapartado 3.3.3. En este caso, se trata de valorar diferentes alternativas de representación de datos que mejor se adecuen al problema. Así se han definido tres tipos de imágenes-dato: Mapas de Profundidad (*Depth Maps*), de Coordenadas (*Coordinate Maps*) y de Ocupación (*Grid-occupancy Maps*). Todas las imágenes-dato descritas a continuación normalizan de 0 a 1 un umbral fijo de distancias (según la configuración varía de 0 – 25 metros a 0 – 50 metros). Por esta razón, no se pierde información en el escalado, lo que sí ocurriría en el caso de normalizar los valores máximos y mínimos de cada *frame* por separado.

- Mapas de Profundidad

Para representar la información del láser en este tipo de mapa, se ha proyectado el dato de distancia sobre una matriz bidimensional. El eje horizontal corresponde a los valores del ángulo horizontal o azimut de los puntos, mientras que el eje vertical de la matriz corresponde con cada una de las capas láser (figura 3.26). En el caso de que se obtengan varios puntos en un mismo píxel, es decir, que se encuentren en el mismo intervalo de resolución horizontal y vertical de la matriz, prevalece el dato de distancia más cercana, debido a que se intuye que será más representativo del movimiento.

Figura 3.26: Proyección de los puntos LiDAR en matriz de datos.

En este caso, se ha ido incrementando gradualmente el rango de visión horizontal y vertical en función de los resultados obtenidos. Así, en las primeras pruebas se acotó el campo de visión a las cuatro esquinas del vehículo (figura 3.27.a) y las capas del láser orientadas hacia la calzada. De esta manera, se definió una primera configuración debido a que, por un lado, se esperaba que en esta zona existiera mayor cantidad de información relevante para inferir los desplazamientos del vehículo, y por otro, limitar la carga computacional durante el proceso de entrenamiento de la red. Sin embargo, se observó que estas imágenes-dato no generaban suficiente calidad para el proceso de aprendizaje del modelo y se planteó incluir mayor cantidad de información del entorno. Por ello, se añadió la información de los laterales del vehículo donde se esperaba observar elementos estáticos que proporcionaran información sobre el movimiento relativo a partir de ellos (figura 3.27.b). Se descarta incluir la parte delantera y trasera del vehícu-

lo, ya que en dichas áreas se espera encontrar obstáculos móviles que induzcan a error. Además, se observa que el incremento de los datos de entrada no supone una penalización excesiva en el tiempo de entrenamiento del modelo. Por tanto, se amplía el campo de visión horizontal a dos sectores de 150° orientados hacia los laterales del vehículo, obteniendo una resolución horizontal de 300 píxeles. Por la misma razón, también se decide aumentar el número de capas verticales, en un rango de -11° a $+3^\circ$.

Figura 3.27: : Comparación de los rangos de visión horizontal entre la primera (a) y la segunda configuración (b) consideradas.

Una tercera alternativa de configuración de este tipo de mapas, amplía el rango de visión horizontal a 360° , con una discretización de $0,5^\circ$. Igualmente, se aumenta el rango vertical, incluyendo mayor número de capas. Se propone esta configuración debido fundamentalmente a que, por la eficiencia del entrenamiento, se posibilita el incremento de variables de entrada. Por otro lado, se observa que el modelo es capaz de discernir de forma correcta aquella información del entorno de que se puede extraer patrones que infieran el movimiento del vehículo.

Una vez se crean las imágenes-dato, se agrupan por pares las correspondientes a dos instantes de tiempo consecutivos. El formato final de entrada a la red resulta del acoplamiento de las dos imágenes, obteniendo una matriz de dimensiones $H \times W \times 2$. En la figura 3.28, se muestran varios ejemplos de los pares de imágenes-dato según la segunda configuración descrita.

Figura 3.28: Ejemplos Mapas de Profundidad. Configuración con resolución de las imágenes-dato 8 x 300 x 2.

Por otro lado, la tercera configuración, en la que se abarca todo el rango horizontal de visión, se obtiene una mayor resolución de las imágenes que pasan a ser de 28 x 720 x 2. En la figura 3.29, se puede observar un ejemplo de esta configuración.

Figura 3.29: Ejemplos Mapas de Profundidad. Configuración con resolución de las imágenes-dato 28 x 720 x 2.

El principal problema que surge al crear estos Mapas de Profundidad es la falta de información en algunos píxeles, debido principalmente a la divergencia entre puntos. Debido a que los sensores LiDAR no pueden generar información densa del espacio, algunos píxeles no tendrán datos disponibles para ciertas resoluciones angulares. La cantidad de puntos no válidos tiene suficiente peso como para interferir en el proceso de aprendizaje. Para evitar dicho problema, se realiza una interpolación lineal entre píxeles adyacentes horizontalmente. Como resultado, se obtiene una imagen-dato densa con la resolución acorde a la configuración establecida originalmente.

Por otro lado, como ya se estudió en el apartado 3.1 de esta tesis, la densidad de puntos es inversamente proporcional al cuadrado de la distancia. Por tanto, la distribución de los valores de los píxeles de las imágenes no está equilibrada, la mayoría de los valores se concentra en un rango concreto. Para compensar esta distribución y obtener un histograma de valores mayormente repartido en todo el rango, se aplica el logaritmo como función de escalado. Esta función logarítmica mejora el contraste entre valores similares, resaltando las características a partir de las cuales se pueden extraer patrones (figura 3.30). La aportación de esta función se refleja principalmente en las configuraciones de

mapa con menor resolución.

Figura 3.30: Ejemplo de distribución de valores de la imagen-dato antes de aplicar logaritmo (rojo) y después de ello (verde).

- Mapa de Coordenadas

De manera similar a los mapas de distancias se generan unas imágenes-dato, esta vez, a partir de las coordenadas X, Y, Z de los puntos LiDAR. La justificación de esta configuración de imágenes se basa en la hipótesis de un mejor condicionamiento geométrico del problema. Incluir la información de las coordenadas por separado en la matriz de entrada, puede ayudar a inferir los desplazamientos lineales y la rotación a través de la variación de las coordenadas y no de las distancias como en el caso anterior.

La diferencia respecto al proceso de generación de Mapas de Profundidad radica en que, en este caso, cada imagen contiene tres canales de información, es decir, un canal para cada coordenada de forma individual. Se obtiene, por tanto, una imagen-dato donde cada píxel contiene las coordenadas X en el primer canal, las coordenadas Y en el segundo y de igual forma, las coordenadas Z en el tercer canal. Asimismo, durante la generación de este tipo de mapa se tendría en cuenta la interpolación entre valores horizontalmente adyacentes, debido a la ausencia de información en ciertos píxeles.

Este tipo de representación del entorno solo se ha aplicado a partir de la tercera configuración de los Mapas de Profundidad. Por ello, al apilar dos imágenes-dato de este tipo, la matriz de entrada tiene una resolución de $28 \times 720 \times 6$. Un ejemplo de estos mapas de coordenadas se puede observar en la figura 3.31.

Figura 3.31: Ejemplos Mapas de Coordenadas. Configuración con resolución de las imágenes-dato $28 \times 720 \times 6$.

- Mapa de Ocupación

La tercera alternativa de representación son los Mapas de Ocupación o *Grid-occupancy maps*. Estos mapas consisten en una vista en planta del entorno del vehículo de una zona delimitada y dividida en cuadrículas según una resolución. Cada casilla corresponde a un píxel de la imagen-dato que contiene el valor del número de puntos que han sido obtenidos a la distancia correspondiente a la cuadrícula. Por tanto, a mayor número de puntos obtenidos en una cuadrícula, mayor será el valor numérico de esta.

En esta proyección sobre el plano XY, se puede observar cómo se desplazan, relativos al láser, los elementos principales del entorno, es decir, aquellos en los que ha incidido una mayor cantidad de puntos. Esta representación es la más intuitiva de las tres, ya que se puede apreciar cómo es el movimiento entre observaciones, lo que indica una posible ventaja para el aprendizaje del modelo. Las imágenes generadas, en este caso, tienen una resolución de $150 \times 150 \times 1$. Finalmente, cuando se apilan los mapas de dos observaciones consecutivas, la matriz de entrada tiene un tamaño de $150 \times 150 \times 2$. La figura 3.32 muestra un ejemplo de secuencia de movimiento representada mediante estas imágenes-dato.

Figura 3.32: Ejemplo de secuencia durante 6 observaciones seguidas representadas mediante *Grid-occupancy Maps*.

3.3.2.3. Aumentación de datos (*Data Augmentation*)

Para un correcto entrenamiento que permita al modelo generalizar correctamente, es necesario una gran cantidad de datos, en este caso, observaciones. De hecho, existen distintos métodos para incrementar el número de datos de entrenamiento conocidos como técnicas de aumentación de datos o *Data Augmentation*. Estas técnicas son utilizadas sobre todo cuando

el problema es complejo o bien, no se dispone de suficientes datos, evitando además problemas de sobreajuste (*overfitting*).

En este caso, se han implementado dos técnicas, la primera denominada *Mirroring* y otra llamada *Shaking*.

Con el *Mirroring* se consigue duplicar el número de observaciones al calcular la imagen inversa. Se basa en girar las imágenes 180°, y de la misma forma, calcular la ruta inversa del nuevo entorno que se utilizará como *Ground Truth*. Esta técnica se ilustra en la figura 3.33.

Figura 3.33: Ejemplo de aplicación de la técnica *Mirroring*. a) Representación original del entorno. b) Representación espejo equivalente.

En la técnica denominada *Shaking*, se utiliza la propia incertidumbre del LiDAR para generar nubes de puntos similares. Considerando que estos sensores suelen tener una precisión del orden de $\pm 3\text{cm}$ (Velodyne Lidar, 2016; OUSTER, 2014), se traduce en que cada punto puede situarse de forma aleatoria en una esfera de 3 cm de radio. Por tanto, se pueden generar nubes de puntos similares añadiendo un ruido a cada punto acotado por este umbral. Por cada secuencia de nubes de puntos, se genera otra con un mismo *Ground Truth* pero ligeramente diferente a la original. El resultado de aplicar esta técnica se puede observar en la figura 3.34.

Figura 3.34: Ejemplo de aplicación de la técnica *Shaking*. a) Representación original del entorno. b) Representación con *Shaking* aplicado.

El objetivo perseguido al implementar esta última técnica es extraer patrones relevantes para poder inferir el movimiento entre dos imágenes consecutivas y que este no dependa de una alta precisión en cada *frame*. De esta manera, se consigue que varias parejas de nubes de puntos puedan obtener un mismo movimiento relativo entre ellas.

3.3.2.4. Descripción de *datasets*

Es preciso describir los *datasets* utilizados durante el ajuste del modelo ya que son parte fundamental del proceso de entrenamiento y de los resultados obtenidos. En concreto, los ensayos realizados se han llevado a cabo en dos *datasets* distintos. Por un lado, un conjunto de datos generado a partir de los sensores y vehículos disponibles en la unidad de investigación del INSIA-UPM. Por otro, se utiliza el *dataset* KITTI, disponible por parte de la Universidad Karlsruhe Institute of Technology, en el cual utilizan un modelo diferente de LiDAR y sensores con los que proporcionan un *Ground Truth* preciso.

Un aspecto común en ambos *datasets* es la repartición entre conjunto de entrenamiento y conjunto de validación. Por la naturaleza de los datos, existe una dependencia temporal entre observaciones, por lo que no es posible hacer una desordenación aleatoria de todo el conjunto de datos y dividir posteriormente los conjuntos de entrenamiento y validación en una proporción 90%–10%. Para garantizar que exista un tipo de datos en ambos conjuntos y un correcto aprendizaje de la red, se reserva para validación una secuencia de ensayos que suponga en torno al 10% sobre el total de imágenes-dato. Una vez hecha esta repartición, el conjunto de entrenamiento se desordena de forma aleatoria para suministrar los datos a la capa de entrada, mientras que en el conjunto de validación no, de tal manera que la predicción de la red sobre este conjunto se obtiene de forma ordenada según la secuencia.

Dataset Campus Sur – UPM:

En el primer caso, el *dataset* ha sido generado a partir de los sensores instalados en el vehículo de pruebas. Se ha utilizado un sensor LiDAR VLP-16 para recoger la información del entorno y, para el *Ground Truth*, un giróscopo RMS FES 33 de alta resolución con el que se obtiene la variación de guiñada y un sensor de velocidad sin contacto L-CE Correvit, con el cual se obtiene la velocidad lineal del vehículo sin pérdida de precisión por posibles deslizamientos. El conjunto de datos de entrenamiento está formado por los datos adquiridos durante una hora de conducción por entorno urbano situado en el Campus Sur – UPM (Madrid) y alrededores, lo que equivale a 25,6 km. Cabe mencionar que, aunque se haya utilizado un giróscopo y, por tanto, la señal generada pueda sufrir de errores de deriva, la medición de variaciones de guiñada es precisa y este error solo se manifiesta al integrarse en el tiempo. Además, si el modelo empleado es capaz de adecuarse correctamente a los valores de *Ground Truth*, se espera un comportamiento similar ante un *Ground Truth* sin desviaciones. La velocidad de adquisición y sincronización de los datos ha sido a 10 Hz.

Durante la adquisición de datos, el comportamiento del conductor es variado, de manera que se obtienen diferentes velocidades, paradas o cambios de carril, favoreciendo la generalización de los datos. El entorno adquirido consta de elementos estáticos como parte de la

infraestructura cercana y situaciones con un tráfico dinámico. Además, mezcla entornos diáfanos con calles más estrechas y giros más acusados. Todo ello, hace que el *dataset* contenga una gran diversidad en las observaciones recogidas. En la figura 3.35, se muestran algunas escenas ejemplo del entorno de entrenamiento.

Figura 3.35: Ejemplo del entorno urbano incluido en el *dataset* Campus Sur – UPM.

Para el conjunto de evaluación, se ha llevado a cabo una nueva adquisición de datos en una ruta que comprende el trazado anterior y un tramo intermedio no incluido en el conjunto de entrenamiento (figura 3.36). A pesar de mantener los mismos inicios y final de ruta, los datos corresponden a condiciones de tráfico y conducción distintas. En estas primeras valoraciones, se pretende estudiar la capacidad de generalización del modelo sobre un entorno observado en el entrenamiento, así como en un tramo totalmente nuevo. En este caso, el conjunto de evaluación comprende 2,85 km que equivalen a 4.756 observaciones.

Figura 3.36: Rutas incluidas en *dataset* Campus Sur - UPM. a) Conjunto de entrenamiento (varias vueltas). b) Conjunto de evaluación.

Después de aplicar las técnicas de *Data Augmentation*, el *dataset* completo se divide en un 93% entrenamiento y un 7% de datos de evaluación.

Dataset KITTI:

El *dataset* de KITTI sirve como *benchmark* de referencia para el desarrollo de diversos algoritmos dentro del campo de investigación del vehículo autónomo. Se proporcionan datos de entornos reales con un *Ground Truth* asociado, además de disponer desde *datasets* orientados a la identificación de objetos mediante visión artificial, la segmentación semántica del entorno, hasta la odometría del vehículo a través de varios recorridos, el cual es el utilizado para este desarrollo.

Los datos proporcionados en este *dataset* constan de la nube de puntos de un LiDAR de 64 capas junto con el posicionamiento preciso asociado a cada *frame*. Los datos son adquiridos y sincronizados a una frecuencia de 10 Hz. El *dataset* se distribuye en 11 secuencias de entornos urbanos e interurbanos, con diversidad de elementos estáticos, dinámicos y condiciones de tráfico. En la figura 3.37 se muestra un ejemplo de las secuencias incluidas en este *dataset*.

Figura 3.37: Ejemplo del entorno urbano incluido en el *dataset* KITTI.

Para el entrenamiento del modelo se han utilizado 10 de las 11 secuencias de las que se dispone un *Ground Truth* de la trayectoria seguida, teniendo un total de 23.201 observaciones (nubes de puntos junto al movimiento relativo asociado). La secuencia restante se utiliza como conjunto de validación. Una vez se aplican las técnicas de aumentación de datos, se ha dividido según el número de observaciones totales en una proporción 90% - 10% para los conjuntos de entrenamiento y evaluación respectivamente.

3.3.3. Ajuste del modelo

3.3.3.1. Condicionamiento del problema

Por otro lado, se definen los aspectos que debe satisfacer el modelo para cumplir con el objetivo principal. Para ello, es preciso conocer en profundidad la naturaleza del problema y así poder enmarcar el problema que se aborda. Se definirá el tipo y modo de aprendizaje, tipo de estimación esperada por la red y el modo de medición del error.

- Tipo de aprendizaje

Según el paradigma de aprendizaje la mayor parte de los algoritmos de *Machine Learning* pueden ser clasificados en: aprendizaje supervisado, basado en encontrar las relaciones existentes entre unas variables de entrada y de salida; no supervisado, aquel que consigue generar aprendizaje a partir únicamente de los datos de entrada, sin necesidad de proporcionar qué salida se desea obtener; y por refuerzo, en los que se premia o se penaliza al sistema de aprendizaje (agente inteligente) en función de que sus acciones se acerquen al resultado esperado o no respectivamente. En particular, para que el aprendizaje supervisado encuentre esa relación entre variables entrada y salida, es necesario disponer de un conjunto de resultados previos (*Ground Truth*) para que pueda originarse el aprendizaje.

El problema planteado en este apartado se enmarca en este tipo de aprendizaje supervisado, ya que el ajuste del modelo se realizará contra un posicionamiento conocido. Por tanto, es necesario disponer de una gran cantidad de datos que relacionen los datos de entrada (las nubes de puntos mencionadas anteriormente) y los datos del posicionamiento. Por ello, tal y como se ha tratado en el subapartado 3.3.2.4, se han desarrollado los ensayos en dos *dataset* que contienen una gran cantidad de observaciones (datos de entrada) relacionadas con el movimiento relativo del vehículo entre dos instantes de tiempo (datos de salida). Además, para que el entrenamiento y, por tanto, la generación de conocimiento del modelo, sean eficaces se han propuesto varias técnicas de *Data Augmentation*.

El objetivo de este tipo de aprendizaje es ajustar los parámetros del modelo a partir de un conjunto de datos de entrenamiento y que este sea capaz de generalizar una salida cuando se presenten nuevos casos. Precisamente, la conducción tanto en zonas urbanas como interurbanas se caracteriza por entornos altamente cambiantes y, por ello, esta capacidad de generalizar resulta de gran interés.

- Tipo de predicción

En este caso, el modelo debe inferir el valor de una señal continua que represente la translación y rotación relativas del vehículo entre dos observaciones. Por tanto, el tipo de predicción será de regresión, dejando de lado el enfoque de clasificación o segmentación donde se estiman valores discretos.

En concreto, para la translación se estimará la velocidad lineal seguida por el vehículo entre dos *frames* consecutivos. Por su parte, para la rotación relativa, se estimará la señal de variación de guiñada. De esta manera se cumple con el principio de consistencia local de los métodos de odometría. No se estima un posicionamiento absoluto, ni el giro acumulado.

Se valorará, además, la diferencia entre estimar las dos señales en conjunto por la misma red (problema de regresión multivariable) o empleando un modelo ajustado para estimar cada señal individualmente (problema de regresión univariable)

- Medición del rendimiento del aprendizaje

Otro elemento fundamental durante el proceso de optimización es la medición del rendimiento del aprendizaje, es decir, definir una función de error. La métrica que se elija servirá de indicador para valorar la precisión del modelo para representar la realidad del problema.

Ya que el problema que se aborda es de regresión, el error se medirá a partir del cálculo del error cuadrático medio (Root Mean Square Error, RMSE). Esta métrica del error acentúa las diferencias mayores entre la estimación y el valor real.

$$RMSE(X, h) = \sqrt{\frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)})^2} \quad (3.19)$$

Donde m es el número de observaciones o *samples* del conjunto de datos sobre los que se está midiendo el error; $x^{(i)}$ es un vector con los datos de entrada a la red para la observación i ; mientras que $y^{(i)}$ es la salida real. X es la matriz de entrada de todo el *dataset* de entrenamiento y h es la hipótesis del modelo, es decir, la salida del sistema ante unos datos de entrada.

Otras métricas consideradas, como el error medio absoluto (MAE), no son tan sensibles a errores mayores, por lo que se descarta para la aplicación.

3.3.3.2. Ajuste de hiper-parámetros

El ajuste de los hiper-parámetros de un modelo consiste en la asignación de todos aquellos parámetros que no son ajustados automáticamente por el modelo durante el proceso de optimización. Existe una gran cantidad de estos y, en general, no es posible conocer con antelación el mejor valor para cada uno. Normalmente no hay una estrategia definida, por lo que el ajuste de hiper-parámetros es un proceso cíclico.

Uno de los factores más relevantes durante el proceso de aprendizaje es el algoritmo de optimización utilizado. En este caso, se ha utilizado el optimizador Adam en todos los ensayos por tratarse de un algoritmo equilibrado en precisión y eficiencia respecto a otros como SGD o RMSProp, a pesar de que estos últimos pueden llevar a mejores resultados si se realiza un riguroso ajuste del resto de hiper-parámetros (Wilson et al., 2017).

Asimismo, cabe destacar las interacciones existentes entre hiper-parámetros. En este sentido, a partir del optimizador elegido (Adam), se selecciona un método basado en *mini-batch* como modo de computar la optimización. Esta estrategia conlleva elegir un tamaño de *mini-batch* (a veces también llamado *batch-size*) mediante el cual se calcula y actualizan los pesos del modelo. Este tamaño de *mini-batch* tiene una fuerte interrelación con el hiper-parámetro de la tasa de aprendizaje. En general, se puede afirmar que *batches* reducidos pueden dar lugar a una mejor generalización del modelo. Sin embargo, el proceso de entrenamiento con tamaños de *mini-batch* tan reducidos implica utilizar una tasa de aprendizaje baja para mantener

la estabilidad debido a la gran variación en la estimación calculada tras cada paso. Además, como resultado se incrementa notablemente el tiempo de entrenamiento debido a esta baja tasa de aprendizaje y se requiere un mayor número de pasos para observar el conjunto de entrenamiento completo (Goodfellow et al., 2016). Por ello, según la topología del ensayo, se buscará un equilibrio entre el tamaño de *mini-batch* y la tasa de aprendizaje, para dar lugar a un entrenamiento que sea eficiente en tiempo y que a su vez generalice correctamente.

En la tabla 3.4, se han resumido y categorizado algunos de los hiper-parámetros definidos en los distintos modelos valorados:

Tabla 3.4: Hiper-parámetros a tener en cuenta durante el ajuste del modelo.

Entrenamiento	Arquitectura	
<ul style="list-style-type: none"> • Optimizador • Tasa de aprendizaje (Learning rate) • Kernel initializer • Tamaño de mini-batch • Iteraciones (epochs) 	Red	<ul style="list-style-type: none"> • Función de activación • Capas ocultas • Unidades ocultas • Número de filtros • Tamaño de filtros • Regularización: Dropout
	Datos	<ul style="list-style-type: none"> • Stride • Padding • Normalización

3.3.3.3. Alternativas valoradas para las arquitecturas

Una vez se ha descrito el problema planteado, es posible definir las características generales de los modelos. Para confirmar la validez de las redes CNN, se compara el rendimiento de estas con otro tipo de modelo de redes neuronales bien conocido, como son los modelos basados en Perceptrón Multicapa (se utilizará la nomenclatura MLP para referirse a estas redes). Este tipo de modelos es ampliamente conocido y por ello, define una referencia respecto al rendimiento obtenido por las CNN aplicadas a este problema. Por otro lado, permiten adaptar la capa de entrada al tipo de representación de datos descrita en el subapartado 3.3.2.2.

Los modelos MLP son una red neuronal con arquitectura *feed-forward* compuesta por capas de neuronas. Estas capas se estructuran de tal manera que todas las neuronas de una capa se conectan con todas las de la siguiente (capas *fully-connected* figura 3.38). Cada conexión da lugar a un parámetro de peso. Estos son los parámetros que se deberán de ajustar automáticamente para optimizar el error cometido.

Figura 3.38: Capa *fully-connected*. Las flechas de la izquierda representan las entradas a la red, mientras que aquellas situadas a la derecha, las salidas. Cada conexión entre neuronas es un parámetro de peso. (Fuente: Ramsundar y Zadeh (2018)).

Por su parte, los modelos basados en CNN, aunque su planteamiento es distinto a los MLP, siguen manteniendo una arquitectura *feed-forward*, pero las operaciones se basan en convoluciones. Estas redes se estructuran igualmente en capas de procesamiento, donde una serie de filtros, formados por neuronas, se aplican a toda ella y, a su vez, se conecta a la siguiente (figura 3.39). En las primeras capas, aquellas más cercanas a los datos de entrada, se identifican patrones básicos, que son utilizados en capas posteriores para extraer características o patrones más genéricos de forma jerarquizada.

Figura 3.39: Ejemplo de estructura de las capas de procesamiento en modelos CNN. (Fuente: Geron (2019)).

Los datos de entrada en ambos modelos es el mapa 3D del entorno en dos observaciones consecutivas. Se espera que este mapa contenga patrones reconocibles, donde las CNN tienen un gran desempeño extrayéndolas y usándolas para conseguir un mejor ajuste que en el caso

de los MLP.

La comparación entre modelos se lleva a cabo entre dos topologías MLP y seis topologías de redes convolucionales utilizando el *dataset* del Campus Sur – UPM con la segunda configuración de Mapas de Profundidad descrita en el apartado 3.3.2.2 (cabe recordar que estas imágenes-dato tienen una resolución de 8 x 300 x 2). Los hiper-parámetros utilizados varían en función del modelo empleado.

- Topologías Perceptrón Multicapa:

Los modelos MLP planteados utilizan el mismo **optimizador**: Adam Optimization algorithm (Kingma y Ba, 2014), con una **tasa de aprendizaje** de 10^{-5} . En cuanto a técnicas de **regularización**, para tratar de evitar *overfitting*, se ha utilizado *dropout* (Srivastava et al., 2014) en la última capa *fully-connected*. Esta técnica se basa en inhibir aleatoriamente la operación de un conjunto de neuronas en cada iteración durante el proceso de entrenamiento, evitando la sobre-especialización de las neuronas para ejemplos concretos y mejorando, por tanto, la calidad del entrenamiento y la generalización del modelo. En el caso de los modelos MLP, se ha utilizado un *dropout* de 0,4, es decir, de una capa donde se aplica esta regularización a la siguiente se mantienen activas el 60% de las neuronas.

Respecto a los principales hiper-parámetros sobre la arquitectura de red que se han tenido en cuenta, se ha seleccionado la **función de activación** ReLU (Nair y Hinton, 2010), donde cada neurona queda definida como en la figura 3.40.a. Después de varias pruebas llevadas a cabo con diferentes funciones de activación, como la función sigmoidea o tangente hiperbólica, se observó que el proceso de entrenamiento se acelera mediante una activación ReLU, por lo que se consigue un aprendizaje más rápido.

Figura 3.40: a) Red neuronal artificial con función de activación ReLU; b) Topología genérica de Perceptrón Multicapa.

Los modelos MLP utilizan un vector de datos de entrada que da lugar a una **primera capa** de 4.800 neuronas, acorde al tamaño de la matriz de entrada. Por un lado, la primera topología presentada (MLP-1), con una apariencia como la figura 3.40.b, contiene **dos capas ocultas** con **300 neuronas** cada una.

Durante el entrenamiento de este modelo, tras 300 iteraciones, el RMSE comenzó a crecer, por lo que se detuvo el proceso consiguiendo unos errores para el mejor de los epoch de 10,17 km/h para la velocidad lineal y 5,75 °/s para la variación de guiñada. La validación consiguió un error en velocidad de 13,15 km/h y 5,75 °/s en variación de guiñada.

El segundo modelo MLP-2, basado igualmente en la red descrita en la figura 3.40.b, mantiene todos los hiper-parámetros de entrenamiento iguales al modelo MPL-1. Sin embargo, la arquitectura de red añade una capa oculta adicional, también con 300 neuronas en esta.

Durante el entrenamiento de este modelo, el error RMSE mantiene una tendencia decreciente hasta el epoch 320. El error obtenido en entrenamiento y validación es peor que en el modelo MLP-1. Los errores obtenidos en entrenamiento son 10,26 km/h y 5,76 °/s, mientras que en validación son de 13,23 km/h y 5,99 °/s, para estimación de velocidad lineal y variación angular respectivamente.

- Topologías Redes de Convolución:

En el caso de las redes de convolución, la arquitectura utilizada se puede dividir en dos bloques: bloque de extracción de características y bloque de estimación de la señal (figura 3.41). El primer bloque, enfocado en la extracción de las características, se compone de un conjunto de capas donde se aplican sucesivamente convoluciones y reducciones del espacio donde se posibilita aprender patrones clave a partir de los datos de entrada. A continuación, los patrones más generales, obtenidos en las últimas capas, se modifican a forma de vector y se conectan al bloque de estimación de valores. Este bloque se define como una topología *fully-connected*, similar a la utilizada en los modelos MLP.

Figura 3.41: Topología de la red neuronal de convolución utilizada.

El primer bloque se compone de capas convolucionales, donde a partir de los datos

de entrada, un conjunto de filtros aplica la función de convolución para generar una matriz de salida que se conectará a la entrada de las capas sucesivas. Estas funciones se representan como neuronas en cada filtro con una **función de activación** ReLU. El proceso de **optimización** se realiza igualmente con Adam y se ha seleccionado una **tasa de aprendizaje** de 10^{-3} , mayor que la utilizada en las topologías de MLP. Para la **regularización** se ha tenido en cuenta aplicar un *dropout* entre 0,4 y 0,1 tras las capas *fully-connected*, según la topología utilizada.

Las capas de convolución tienen asociadas varias sub-capas u operaciones que se consideran, normalmente, como parte de ellas. De este modo quedan integradas en la arquitectura:

Capa Local Response Normalization (LRN) (Krizhevsky et al., 2012): Se ha añadido tras la capa de convolución. Esta capa incrementa las diferencias entre valores de la matriz, es decir, incrementa el contraste de la salida.

Capa Pooling: Ejecuta un filtro tras la capa de convolución que reduce la dimensión de la matriz de entrada. En concreto, se ha utilizado MaxPooling, de manera que considera el mayor valor de cada subconjunto de datos que abarca el filtro.

El entrenamiento llevado a cabo para las seis topologías definidas fue realizado hasta el epoch 3.000.

Por conveniencia de nomenclatura, se especifica como $C_{H,W,N}$ a cada capa de convolución con N filtros de dimensiones $H \times W$ con la correspondiente capa de normalización LRN. La capa de Pooling se designa como $P_{H,W}$ para la operación de MaxPool con filtros de dimensiones $H \times W$. Finalmente, el término FC_N describe las capas *fully-connected* compuestas por N neuronas. La tabla 3.5 muestra los modelos utilizados y los resultados obtenidos en cada uno.

Tabla 3.5: Resultados obtenidos durante entrenamiento y validación (test) de las topologías CNN.

Red	Topología	Dropout	RMSE	Entren.	RMSE	Eval.
			Velocidad (km/h)	Variación guiñada (°/s)	Velocidad (km/h)	Variación guiñada (°/s)
CNN-1	$C_{3,15,32}, P_{2,2},$ $C_{2,5,64}, P_{2,2},$ FC_{512}, FC_{512}	0,1	0,918	1,534	3,766	3,002
CNN-2	$C_{3,15,32}, C_{3,15,32},$ $P_{2,2}, C_{2,5,64},$ $C_{2,5,64}, P_{2,2},$ FC_{512}, FC_{512}	0,1	0,917	0,945	3,542	2,758
CNN-3	$C_{3,15,32},$ $C_{3,15,32}, P_{2,2},$ $C_{2,5,64}, C_{2,5,64},$ $P_{2,2}, C_{2,3,128},$ $C_{2,3,128}, P_{1,3},$ $FC_{512}, FC_{512},$ FC_{512}	0,1	1,043	1,298	3,332	2,718
CNN-4	$C_{3,15,32}, P_{2,2},$ $C_{2,5,64}, P_{2,2},$ FC_{512}, FC_{512}	0,4	2,769	1,324	3,582	3,055
CNN-5	$C_{3,15,32},$ $C_{3,15,32}, P_{2,2},$ $C_{2,5,64}, C_{2,5,64},$ $P_{2,2}, FC_{512},$ FC_{512}	0,4	1,708	1,326	2,905	2,577
CNN-6	$C_{3,15,32},$ $C_{3,15,32}, P_{2,2},$ $C_{2,5,64}, C_{2,5,64},$ $P_{2,2}, C_{2,3,128},$ $C_{2,3,128}, P_{1,3},$ $FC_{512}, FC_{512},$ FC_{512}	0,4	1,419	1,927	3,229	2,778

Nótese que las topologías CNN-1 y CNN-4; CNN-2 y CNN-5 y CNN-3 y CNN-6 son equivalentes dos a dos respecto a la estructura de capas, sin embargo, se ha variado su valor de *dropout*.

Analizando los resultados obtenidos, se puede observar que aumentando la complejidad de la red (es decir, añadiendo más capas de convolución) de las topologías CNN-1 a la

CNN-2 (y, por tanto, de la CNN-4 a la CNN-5), hay una mejora de la precisión en la estimación de las dos variables de salida. Sin embargo, cuando se sigue aumentando el tamaño de la red, estas mejoras dejan de ser significantes e incluso empeoran los resultados. En el caso de pasar de las topologías CNN-2 a la CNN-3 empeoran los resultados en entrenamiento y en el paso de la CNN-5 a la CNN-6, empeora tanto en entrenamiento (solo variación de guiñada), como en validación. Por otro lado, comparando topologías equivalentes, pero con distinto *dropout*, al aumentar este valor mejora el error en validación en la mayoría de los casos, no así en los errores obtenidos durante entrenamiento.

Las diferentes pruebas llevadas a cabo indican que la mejor topología se compone de 3 capas de convolución y un *dropout* de 0,4. Así, la mejor topología obtenida para estos ensayos ha sido la CNN-5. En la figura 3.42, se muestra la estimación de esta arquitectura sobre las dos variables de salida, comparando su resultado con la salida esperada en cada caso.

Figura 3.42: Resultados de la estimación de la topología CNN-5. En rojo, la salida esperada (*Ground Truth*) de cada una de las variables estimadas y en azul, la estimación de la red para cada una de ellas.

Como se describe en el apartado 3.3.2.4, este *dataset* comparte el inicio y final de la ruta seguida en los conjuntos de entrenamiento y evaluación. Sin embargo, a partir de las gráficas de la figura 3.42, se prueba la capacidad de generalizar por parte del modelo en aquellas zonas que no son conocidas. En las estimaciones correspondientes a las observaciones entre los instantes 150 y 250 s, se observan algunas velocidades lineales residuales o pequeñas desviaciones en la variación de guiñada mientras el vehículo estaba parado (velocidad real, en rojo, 0 km/h), debido principalmente al entorno dinámico donde se desarrolla el ensayo. Sin embargo, es preciso destacar que, en el conjunto de

entrenamiento de este *dataset*, no se incluía ninguna situación con el vehículo parado y, a pesar de ello, la red es capaz de inferir una velocidad próxima a cero.

- Comparación de resultados entre arquitecturas MLP y CNN:

La figura 3.43 resume los errores de cada topología MLP y CNN, diferenciando entre los resultados obtenidos durante entrenamiento y validación sobre el mismo *dataset* y configuración de datos.

Figura 3.43: a) RMSE Velocidad; b) RMSE Variación de guiñada. En azul, error en entrenamiento y en naranja, error en validación.

A la vista de la comparación entre los modelos MLP y CNN, se concluye que el problema está bien condicionado para aplicar modelos basados en redes convolucionales. El error obtenido en los modelos MLP es muy alto en cualquiera de los casos comparado con el obtenido en los modelos CNN. En concreto, se obtienen errores entre 3 y 5 veces mayores, según sea error en validación o entrenamiento respectivamente, para la estimación de la velocidad. Mientras que, respecto al error de la variación de guiñada, estos errores son entre 2 y 4 veces superiores según se trate del error cometido durante validación o entrenamiento.

Por otro lado, se observa que el error obtenido en los diferentes modelos de CNN es contenido. Las topologías CNN, gracias a los filtros empleados, tienen en cuenta la información espacial local a cada píxel. Esta información parece ser importante en el problema objeto ya que las CNNs cometen un error más bajo. Tras la evaluación de los resultados y tener de referencia la mejora de los modelos CNN respecto a los MLP, el estudio se centra la optimización de los modelos con redes convolucionales.

3.3.4. Selección del modelo final

Una vez se ha verificado el condicionamiento del problema y la aplicabilidad de las redes de convolución para obtener una estimación de la odometría del vehículo, se persigue encontrar el modelo que mejor defina el problema real. Este proceso se lleva a cabo de manera cíclica como se hacía mención en el subapartado 3.3.3.2. En este apartado se describen las alternativas valoradas y ensayos llevados a cabo para definir la topología final del modelo en función de la mejora sobre el error obtenido.

3.3.4.1. Ensayos realizados para obtener modelo CNN

- Alternativas sobre la configuración de los datos de entrada:

La configuración final elegida de los datos de entrada descrita en el subapartado 3.3.2.2, ha sido obtenida a partir de diferentes ensayos modificando el modo de su integración en la capa de entrada de la red. A partir de los resultados obtenidos, ha sido posible descartar las diferentes alternativas, entre las que se destacan:

1. Información de variación de distancias entre instantes de tiempo consecutivos.
2. Imágenes-dato con valor de la reflectividad.
3. Incluir los datos de las distancias de dos *frames* consecutivos en capas independientes.
4. Concatenando diferencias de distancias con diferencias de intensidades.
5. Incluyendo tres *frames* consecutivos: t , $t - 1$ y $t - 2$.
6. Incluyendo intervalos más amplios entre instantes de tiempo: t y $t - 2$.

Así, por ejemplo, en una configuración incluyendo tres observaciones consecutivas como se indica en el caso 5, fue descartada por la baja precisión general obtenida. En concreto, en la tablas 3.6 y 3.7, se comparan los resultados obtenidos incluyendo los instantes de tiempo t , $t - 1$ y $t - 2$, y los obtenidos para esa misma topología de red, incluyendo solo dos instantes de tiempo consecutivos (estos resultados ya han sido descritos anteriormente en la tabla 3.5):

Tabla 3.6: Comparación de resultados en entrenamiento según configuración de datos de entrada. Incluyendo 3 (t , $t-1$, $t-2$) y 2 observaciones consecutivas (t , $t-1$).

Red	RMSE Entrenamiento			
	Velocidad (km/h)		Variación guiñada (°/s)	
	t , $t-1$ y $t-2$	t y $t-1$	t , $t-1$ y $t-2$	t y $t-1$
CNN-5	0,804	1,708	1,305	1,326
CNN-6	1,589	1,419	2,759	1,927
CNN-3	1,123	1,043	2,027	1,298

Tabla 3.7: Comparación de resultados en evaluación según configuración de datos de entrada. Incluyendo 3 (t , $t-1$, $t-2$) y 2 observaciones consecutivas (t , $t-1$).

Red	RMSE Evaluación			
	Velocidad (km/h)		Variación guiñada ($^{\circ}/s$)	
	t , $t-1$ y $t-2$	t y $t-1$	t , $t-1$ y $t-2$	t y $t-1$
CNN-5	3,093	2,905	4,911	2,577
CNN-6	2,992	3,229	5,562	2,778
CNN-3	3,126	3,332	4,515	2,718

A la vista de los resultados del error obtenido en evaluación, se puede concluir que al incluir tres instantes de tiempo consecutivos no supone un gran aumento del error en la estimación de la velocidad, pero sí en la variación de guiñada obtenida siendo entre 1,6 y 2 veces error obtenido en cuando se incluyen solo dos instantes de tiempo.

- Valoración del valor de *dropout* como método de regularización:

Uno de los hiper-parámetros utilizados para la evitar el sobreajuste del modelo, tal y como se indicó anteriormente en el subapartado 3.3.3.2 ha sido la técnica de *dropout*. A partir de los resultados presentados en tal apartado, se ha observado que el error obtenido es muy sensible a pequeñas variaciones de este hiper-parámetro. En particular, los resultados presentados en la figura 3.43 hacían patente dicha variabilidad.

Existe una gran dependencia sobre la consistencia global cuando se utilizan varios valores de *dropout* ya que una pequeña variación influye sobre el valor de RMSE obtenido. Así, una pequeña desviación en la variación de guiñada puede suponer obtener resultados totalmente diferentes debido a la acumulación del error cuando se integra esta señal.

Por un lado, el resultado obtenido en los ensayos mencionados con la arquitectura CNN-5 con *dropout* de 0,4 fueron unos valores RMSE de 2,90 km/h y 2,57 $^{\circ}/s$ en velocidad y variación de guiñada respectivamente, tal como se ilustraba en la figura 3.42. A pesar de que estos errores son pequeños respecto a la señal de referencia, su integración da lugar a una mala reconstrucción de la trayectoria. La consistencia global no es acertada, mientras que la red sí que realiza una estimación del movimiento relativo correcta, garantizando una consistencia local.

- Problema de regresión multivariable frente a regresión univariable:

Tal y como se adelantaba en el subapartado 3.3.3, se baraja la posibilidad de entrenar dos modelos independientes para estimar la velocidad y variación de ángulo, pasando a considerarse un problema de regresión univariable.

La variación de guiñada es el valor con mayor importancia para el ajuste de la trayectoria. Sin embargo, el error obtenido en la estimación de la velocidad es asumible y una pequeña mejora de este, no se refleja en un mejor ajuste de la trayectoria sobre la de

referencia. Por esta razón, se pretende buscar un modelo que se especialice en optimizar el error de la variación de guiñada.

En unos resultados preliminares, manteniendo la arquitectura y condiciones del entrenamiento como en el modelo descrito anteriormente CNN-3, los errores obtenidos han sido de 0,63 °/s durante entrenamiento y 3,26 °/s en validación. Mientras que con la red CNN-5, 0,63 °/s en entrenamiento y 3,13 °/s en validación.

A partir de estos primeros resultados obtenidos sobre la variación de guiñada únicamente, se obtienen errores del mismo orden que en modelos multivariable. Sin embargo, la duración del entrenamiento es mucho más corta, necesitando menos iteraciones para obtener estos valores. Por otro lado, cabe destacar que, a partir de estas iteraciones, el error de validación comienza a subir mientras que el de entrenamiento continúa bajando, comenzando a distanciarse entre sí estos errores, lo que se traduce en un sobreajuste del modelo. Para evitar este *overfitting*, se pueden adoptar diferentes medidas como reducir la complejidad de la red, aumentar el valor de *dropout*, asumir otras estrategias de regularización o llevar a cabo un escalado (amplificación) de los valores utilizados como *Ground Truth*.

Por tanto, se decide seguir con esta estrategia de regresión y tratar de encontrar un modelo que optimice cada valor de salida por separado.

- Adaptación de datos e hiper-parámetros de entrenamiento del modelo al *dataset* de KITTI:

A partir de las conclusiones derivadas de los ensayos realizados sobre el anterior conjunto de datos (Campus Sur – UPM), se persigue mejorar los resultados utilizando esta vez un conjunto de datos mayor y con mejor resolución, manteniendo en la medida de lo posible las configuraciones anteriores y adaptando los hiper-parámetros precisos en el caso que no sea posible.

Respecto a la configuración de los datos, las primeras consideraciones a tener en cuenta son que debido a la una mayor resolución del LiDAR utilizado en este caso (64 capas frente a las 16 utilizadas anteriormente), deja de tener relevancia la interpolación de valores en píxeles horizontalmente adyacentes, tal como se describía anteriormente en el subapartado 3.3.2.2. Esto se debe a que cuando la resolución es menor, la distribución de píxeles no válidos en la imagen-dato inducía a error. Sin embargo, en las imágenes-dato con una mayor resolución del láser y, por tanto, de la imagen creada, da lugar a una representación más cercana a la realidad y, por tanto, los valores no válidos se distribuyen de forma coherente en la imagen. La figura 3.44 ilustra esta explicación comparando dos imágenes-dato de cada configuración:

Figura 3.44: Comparación de resolución en Mapas de Profundidad con distintos LiDAR. Rodeados en rojo, identificación de vehículos en ambas imágenes.

Por la misma razón, no es necesario aplicar la función logaritmo en este caso. En el conjunto de datos anterior (Campus Sur – UPM), con una baja resolución, era necesario amplificar el contraste entre píxeles con valores similares. En el caso de obtener imágenes con mayor resolución, la distribución de valores está mayormente repartida y, por tanto, no es necesaria dicha amplificación.

En cuanto a la adaptación del modo de aprendizaje, uno de los hiper-parámetros que tiene mayor influencia en el resultado obtenido y que, además tiene una fuerte interrelación con el tamaño de *mini-batch* utilizado es la tasa de aprendizaje, como se comentaba en el subapartado 3.3.3.2. Al utilizar este conjunto de datos, se ha utilizado un tamaño de *mini-batch* distinto, lo que ocasiona modificar la tasa de aprendizaje. En concreto, se ha implementado un *Learning Rate* decreciente exponencial según la ecuación 3.20 y figura 3.45

$$Learning\ rate\ (epoch) = lr_{min} + (lr_{max} - lr_{min}) \cdot e^{\frac{-epoch}{decay_{speed}}} \quad (3.20)$$

Figura 3.45: Tasa de aprendizaje en función del epoch utilizada durante los ensayos.

Donde lr_{min} es el mínimo valor de la tasa de aprendizaje, con un valor de 10^{-4} , lr_{max} es

la tasa máxima de aprendizaje, establecida en 10^{-3} , $decay_{speed}$ es la velocidad de decremento, definida en 2.000 y $epoch$ es la iteración durante el entrenamiento del modelo.

- Adaptación de la arquitectura del modelo para *dataset* del KITTI:

Como se había observado en los resultados con el anterior conjunto de datos, al elegir un modelo específico que realice el aprendizaje de una sola variable era necesario tomar ciertas medidas para evitar sobreajuste. Para ello se pueden seguir varias estrategias para adecuar la arquitectura de red al conjunto de datos de KITTI. La primera medida tomada es disminuir la complejidad de la topología, es decir, reducir el número de parámetros que se optimizarán durante el entrenamiento.

Así, la red mantiene la estructura básica descrita en el subapartado 3.3.3.3, donde se pueden seguir diferenciando dos bloques, uno para la extracción de características y otro para la estimación de la señal, tal como se ilustraba en la figura 3.41. En este caso, el tamaño de las capas de convolución y de los filtros de cada una quedan reducidos en tamaño, así como la capa *fully-connected*. Manteniendo la nomenclatura utilizada para describir las topologías, quedaría:

$$\text{CNN-KITTI} = C_{10,10,32}, C_{8,8,64}, FC_{512}$$

Donde cada capa de convolución incluye las operaciones de la capa LRN. Tras la capa *fully-connected*, se aplica la técnica de *dropout* como regularización antes del cálculo del valor de salida, que en este caso se computa a través de una sola neurona.

La complejidad de la arquitectura se reduce en un alto grado y, además, el número de iteraciones necesarias hasta obtener el resultado pasa de ser del orden de 3.000 epochs a 350 – 700 epochs según el ensayo, lo que conlleva a una disminución del tiempo de entrenamiento.

- Resultados obtenidos con representación de datos como Mapas de Profundidades en KITTI:

Para el ajuste de estos modelos univariantes que estiman la velocidad o bien la variación de guiñada, se ha utilizado en un primer momento la representación de datos como Mapas de Profundidad en dos instantes de tiempo consecutivos. Con esta configuración se mantienen las conclusiones obtenidas en ensayos anteriores, comprobando su viabilidad para el conjunto de datos de KITTI.

La estimación de la variación de guiñada, como se había descrito anteriormente, es la variable crítica para obtener un mejor ajuste de la trayectoria respecto a la de referencia. De este modo, se llevan a cabo una serie de ensayos centrados en optimizar el error, modificando para ello el valor de *dropout* por su influencia observada sobre el resultado. En la tabla 3.8 se observan los resultados obtenidos de RMSE en función del valor del *dropout* elegido.

Tabla 3.8: RMSE variación de guiñada frente al valor de dropout.

Valor de dropout	RMSE Variación de guiñada ($^{\circ}/s$)
0,4	1,903
0,45	1,956
0,5	1,809
0,6	1,906
0,7	1,922
0,8	2,031
0,9	2,124

Por lo que el mejor valor de *dropout* encontrado es de 0,5, un valor alto teniendo en cuenta la arquitectura reducida que se está utilizando, sin embargo, necesario para evitar un *overfitting* prematuro. Para este valor, se obtiene un RMSE de la variación de guiñada de 1,809 $^{\circ}/s$ (en ensayos anteriores, el mejor valor obtenido había sido 2,577 $^{\circ}/s$)

Por otro lado, para la estimación de la velocidad, al no ser el parámetro crítico y una pequeña desviación en su estimación, no influye demasiado en la trayectoria reconstruida, se llevan a cabo dos verificaciones de la influencia del *dropout*. Así, los resultados obtenidos han sido con *dropout* 0,9, un RMSE de 1,566 km/h y con *dropout* de 0,5, el RMSE disminuye a 1,499 km/h (en ensayos anteriores, el mejor valor obtenido había sido 2,905 km/h). Por lo que, a la vista de la poca diferencia en la mejora de este error, se asigna que este valor es el óptimo para estimar la velocidad del vehículo.

- Resultados obtenidos con las representaciones alternativas de datos en KITTI:

En el subapartado 3.3.2.2 se han descrito diferentes configuraciones de representación de datos junto a la motivación que ha llevado a cada una de ellas. Sin embargo, los resultados obtenidos utilizando *Coordinate Maps* y *Grid-occupancy Maps* han sido descartados debido a la falta de precisión obtenidos en ellos. Así, el error cometido en un modelo específico para la estimación de la velocidad utilizando los mapas de coordenadas alcanzaba un error de 21,6 km/h, mucho mayor al obtenido por ese mismo modelo utilizando Mapas de Profundidad. De igual modo, en modelos entrenados específicamente para estimar la variación de ángulo de guiñada utilizando *Grid-occupancy Maps*, se obtiene 2,1 $^{\circ}/s$ en el mejor de los casos.

3.3.4.2. Resultados obtenidos con el modelo CNN seleccionado

A partir de los ensayos llevado a cabo, se llega a la conclusión de utilizar dos modelos univariable con la topología designada como CNN-KITTI descrita anteriormente, motivada principalmente para evitar sobreajuste y una mayor eficiencia en el proceso de entrenamiento de la red. Asimismo, se han modificado los hiper-parámetros necesarios durante la fase de optimización como la tasa de aprendizaje, implementando en este caso un *Learning Rate Decay*. Un estudio sobre la influencia del *dropout* en el error obtenido, ha llevado a elegir un

valor de 0,5 para ambos modelos (modelo de estimación de velocidad y modelo de estimación de variación de guiñada).

Por otro lado, se utiliza la representación como Mapas de Profundidad como configuración de los datos de entrada por la precisión obtenida en la medida del error frente a las alternativas propuestas.

- Proceso de aprendizaje:

El proceso de aprendizaje del modelo para la estimación de la velocidad alcanza el mejor valor de RMSE (1,499) en este paso en la iteración 357. En la figura 3.46 se observan las gráficas de aprendizaje tanto para el conjunto de entrenamiento como para el conjunto de validación.

Figura 3.46: Evolución del error en la estimación de la velocidad durante entrenamiento y validación.

Por otro lado, el modelo que estima la variación de ángulo de guiñada obtiene su mejor RMSE en la iteración 629. La figura 3.47 corresponde con el proceso de entrenamiento sobre el conjunto de entrenamiento y validación.

Figura 3.47: Evolución del error en la estimación de la variación de guiñada durante entrenamiento y validación.

- Evaluación del resultado obtenido:

Los resultados obtenidos en validación han sido de un RMSE de 1,4994 km/h para el modelo de estimación de la velocidad lineal y un RMSE de 1,809 °/s en el caso del modelo para la variación de guiñada. La figura 3.48 representa la predicción de cada modelo frente a las señales de referencia para la secuencia de validación completa.

Figura 3.48: Estimación del modelo CNN-KITTI para la secuencia de evaluación. En azul, la señal de referencia para cada variable estimada, en rojo la salida de la red.

Es preciso indicar que, aunque la selección del RMSE como métrica del error durante el proceso de optimización del modelo es correcta, esta medida no representa la precisión de la trayectoria reconstruida a partir de la integración de los datos de odometría. El error medido como RMSE es eficiente en la optimización del error durante el entrenamiento, pero es un indicador del error cometido respecto a la señal de referencia. Esta señal de referencia, dependiendo del modo de adquisición o tratamiento previo de datos, podrá diferir al utilizar un conjunto de datos u otro y, por tanto, el resultado no es extrapolable al obtenido bajo otras condiciones.

Por tanto, la medida de la precisión de la odometría se basará en su validación mediante los algoritmos que evalúan el error de posición absoluto y relativo: APE (*Absolute Pose Error*) y RPE (*Relative Pose Error*) (Kümmerle et al., 2009; Lu y Milios, 1997; Sturm et al., 2012; Umeyama, 1991). Por un lado, la métrica APE se emplea como medida de la consistencia global de una trayectoria obtenida por SLAM, se basa en las posiciones absolutas entre varios instantes de tiempo. Por el contrario, la métrica RPE se utiliza para medir la consistencia local, comparando las posiciones relativas entre la trayectoria estimada y la de referencia. A partir de estos algoritmos se obtiene una valoración del

ajuste de la trayectoria obtenida frente a una de referencia y que, además, es independiente de las condiciones de adquisición de los datos suministrados a la red.

La evaluación de las trayectorias mediante los algoritmos APE y RPE anteriormente citados, dan como resultado:

- APE: 237,19 m
- RPE (translación): 1,02 m

Lo que pone de manifiesto la hipótesis previa de que la red es capaz de garantizar una consistencia local, mientras que en la evaluación de la trayectoria global pierde la precisión por los errores acumulados, lo que no garantiza la consistencia global.

La trayectoria de evaluación tiene una longitud total de 3.722 metros donde la velocidad máxima ha sido de 46 km/h. El error medio de translación ha sido de 77,69%, mientras que el error medio de rotación ha sido de 0,319 °/m. La figura 3.49 muestra el desglose de estos resultados en distintos tramos de longitud y para distintos intervalos de velocidades.

Figura 3.49: Errores de translación y rotación en función de tramos de longitud y velocidades.

A la vista de los resultados, se observa que las estimaciones de velocidad y rotación alcanzan una mejor precisión cuando la velocidad se encuentra entre 20 – 35 km/h. También cabe destacar que el error de la estimación incrementa a velocidades bajas (por debajo de 20 km/h).

3.3.5. Conclusiones finales

La estimación del movimiento a través de métodos de Odometría Visual es una tarea crucial para el vehículo autónomo, ya que no siempre es posible obtener un posicionamiento mediante otros métodos (GPS) y, además, sirve de base para posteriores etapas como *Path Planning* o cálculo de la zona transitable, entre otras.

La aplicación de *Deep Learning* propuesta en este apartado es un paso complementario para resolver el problema de la Odometría Visual en el vehículo autónomo. Aprovechando las ventajas y el potencial de las redes de convolución para el procesamiento de imágenes, se ha utilizado la información LiDAR para crear imágenes-dato que sirvan de entrada al modelo CNN presentado.

La propuesta de implementar un modelo basado en CNN para la estimación de la odometría se justifica desde un enfoque de la mejora del rendimiento y, por otro lado, desde un enfoque del problema geométrico. Este último encuentra dificultades bajo ciertas condiciones, por ejemplo, por oclusiones de los objetos o poca información del entorno para extracción de características. Mediante el sistema propuesto, el modelo de CNN debe inferir el movimiento del vehículo en cualquier situación.

Se ha evaluado en primer lugar el condicionamiento del problema para aplicar técnicas de Aprendizaje Automático, concluyendo que estas ofrecen un gran potencial para resolver un problema geométrico. Por otro lado, se han comparado diferentes topologías de CNN frente al rendimiento de las bien conocidas arquitecturas de Perceptrón Multicapa, verificando la aplicabilidad de las primeras al problema que se quiere resolver.

Una vez comprobada la adecuación de modelos CNN, se han llevado a cabo una serie de ensayos con el objetivo de definir el modelo que mejor represente la odometría real del vehículo. Por un lado, se ha observado la gran influencia sobre el error cometido que tiene el valor seleccionado de *dropout* como método de regularización. Por otro, se ha comprobado que, de las dos variables estimadas, la variación del ángulo de guiñada es crítica para obtener un buen ajuste de la trayectoria y, por tanto, se han buscado dos modelos univariable que se especialicen en estimar la velocidad o la variación de guiñada de forma independiente.

Finalmente, a la vista de los resultados, se puede concluir que este método garantiza una buena consistencia local para la estimación de la odometría. Mientras que, respecto a la trayectoria reconstruida, no se alcanza un mínimo de precisión que garantice la consistencia global en este caso.

Por ello, se propone este método como complementario a técnicas de SLAM que consiguen un buen ajuste de forma global. En general, los algoritmos de SLAM basados en sensores de visión o distancia requieren de una etapa de emparejamiento de *landmarks*, tal como se aclaraba en el apartado 3.2 de esta tesis. Por tanto, durante esta etapa, ya sea mediante técnicas de filtrado u optimización, el método propuesto puede ofrecer ventajas significativas en la convergencia hacia la solución correcta. Además, en función del entorno pueden existir ocasiones en las que estos algoritmos fallen en la extracción de características, lo que da lugar a una baja precisión del resultado. En tal caso, el desarrollo planteado, puede servir de apoyo al aportar una estimación aproximada. Estas ventajas que complementan a los algoritmos clásicos de SLAM y Odometría Visual, son estudiadas en profundidad en el siguiente apartado.

3.4. *Matching* robusto de nubes de puntos asistido por modelo de regresión de *Deep Learning* para métodos de Visual SLAM

A lo largo de este capítulo se ha hecho énfasis en la importancia de obtener un posicionamiento preciso en el vehículo autónomo ante cualquier situación, así como las ventajas que pueden aportar soluciones basadas en SLAM o *Visual Odometry*. Ambas soluciones pueden abordar el problema del posicionamiento desde un punto de vista principalmente analítico, tal y como se detallaba en el apartado 3.2, donde se denominaban a estas soluciones como “técnicas clásicas”. Tanto la solución propuesta con este enfoque, como aquellas que se pueden encontrar en la literatura, pueden alcanzar un buen funcionamiento. Sin embargo, los principales inconvenientes que pueden darse se relacionan con la precisión y el rendimiento. La precisión puede empeorar en estimaciones de trayectorias de largo recorrido o bien, cuando el entorno no ofrece suficientes características para extraer *landmarks*. En cuanto al rendimiento, estos sistemas pueden no garantizar una ejecución en tiempo real debido a la alta carga computacional. Por ello, estas limitaciones llevan a plantear soluciones alternativas que basen su funcionamiento en otro principio distinto al estrictamente geométrico.

En concreto, en el apartado 3.3, se proponía un método de estimación de la odometría del vehículo basada en un modelo de *Deep Learning*. Este enfoque difiere del denominado clásico, ya que su base analítica para la extracción de características y el posterior emparejamiento entre ellas para calcular el movimiento del vehículo, quedan incluidas en el proceso de optimización de los parámetros del modelo durante la fase de entrenamiento. Además, este nuevo enfoque no se estructura con las etapas convencionales que caracterizan un sistema de SLAM (figura 3.11 del apartado 3.2). Por otro lado, una de las conclusiones que se destacaba de estas técnicas, indicaba que estos modelos no garantizan una consistencia global, propia de las soluciones de SLAM, ya que la reconstrucción de la trayectoria a partir de las estimaciones de la red, no alcanzan la precisión requerida. Sin embargo, estos modelos sí que consiguen garantizar una consistencia local y, por tanto, generar una señal funcional de la odometría.

A la vista de las conclusiones de estos dos enfoques, se propone en este apartado fusionar ambos para llevar a cabo una solución de SLAM basada en LiDAR 3D en la que se complementen las ventajas de la estimación analítica y la obtenida mediante un modelo de *Deep Learning*. La metodología plantea asistir a la etapa de asociación de los métodos convencionales de Visual SLAM, mediante la estimación de modelos de Deep Learning.

3.4.1. Justificación y objetivo del método de SLAM asistido por modelos de Deep Learning

Las soluciones de SLAM basadas en sensores de percepción quedan englobadas en las denominadas técnicas de Visual SLAM. Estas técnicas requieren de una serie de etapas secuenciales generales que pueden resumirse en: determinación de la zona de interés, extracción de *landmarks*, *matching* de *landmarks*, estimación del movimiento y, finalmente, actualización y corrección de la posición. En función de los sensores implicados en el método, podrá existir alguna interrelación entre etapas o realimentación de algún resultado parcial, tal y

como se detallaba en el apartado 3.2. Además, cabe recordar que cada una de estas etapas se lleva a cabo mediante algoritmos específicos y que el conjunto de estos conforma el sistema de SLAM. Sin embargo, este apartado se centra en la etapa de *matching* de características ya que es la más relevante respecto a la precisión y rendimiento del algoritmo.

La etapa de *matching* de características en algoritmos basados en LiDAR, se encarga de realizar el emparejamiento de nubes de puntos en distintos instantes de tiempo (figura 3.50). Este emparejamiento, también llamado asociación o registro de datos, es una etapa crítica dentro del sistema de SLAM. Partiendo de la extracción de *landmarks*, el algoritmo de *matching* se encarga de minimizar el error de asociación entre las características actuales y los *landmarks* de la observación anterior.

Figura 3.50: Ejemplo de etapa de asociación de nubes de puntos. a) Antes de *matching*; b) después de *matching*.

Esta etapa, generalmente es un proceso cíclico donde en cada iteración una nube de puntos con las características extraídas de la observación actual se traslada y rota, para tratar de superponerla a la nube de puntos con los *landmarks* de la observación anterior, optimizando el error de asociación obtenido. Para un mejor rendimiento del algoritmo, se suele proporcionar una primera aproximación de la transformación basada en la última matriz de movimiento calculada. Este hecho es posible en la navegación autónoma debido a la naturaleza del problema, ya que dinámicamente, no se espera obtener valores muy diferentes a los obtenidos en la estimación anterior, donde normalmente ha transcurrido entre 0,1 y 0,2 segundos (dependiendo de la implementación).

Respecto a los métodos de Visual SLAM existen principalmente dos limitaciones durante el proceso de asociación de *landmarks*.

Por un lado, estos algoritmos suelen encontrar dificultades cuando se producen grandes variaciones de movimiento (ya sean variaciones de translación o rotación). Por ejemplo, variaciones en el ángulo de guiñada durante curvas cerradas, aceleraciones o deceleraciones pronunciadas, repercutirán en el movimiento calculado por los algoritmos entre dos observaciones consecutivas. Este cálculo no se considera fiable debido a la propia incertidumbre del método y las medidas de los sensores. Por ello, se suele establecer un umbral mínimo de desplazamiento para que la asociación sea correcta o bien, delimitar una variación máxi-

ma factible por la dinámica del problema. Las pequeñas desviaciones obtenidas ocasionarán errores de deriva. Además, si el movimiento relativo calculado es utilizado como primera aproximación en la iteración siguiente, puede llegar a ocasionar un mal condicionamiento de la etapa de *matching*.

Por otro lado, es evidente la dependencia de esta etapa del sistema de SLAM a las características que se puedan extraer del entorno. Para obtener un buen funcionamiento general, debe existir un número elevado de *landmarks* y que, a su vez, estos se caractericen por cumplir una serie de requisitos (subapartado 3.2.2.1). Por tanto, una falta de características relevantes o que estas sean de baja calidad para el *matching*, ocasiona una baja robustez durante el emparejamiento. Concretamente, usando LiDAR 3D, estas características pueden sufrir de la dispersión entre puntos y, por tanto, no estar suficientemente definidas o bien que, debido a oclusiones, se pierda información relevante para extraer estas características.

Estas dos limitaciones son propias de las técnicas tratadas en el apartado 3.2 como clásicas. Su enfoque analítico basado en la correspondencia geométrica entre nubes de puntos depende en gran medida de los problemas descritos anteriormente. Por tanto, recurriendo a métodos que obtengan una estimación de la translación y la rotación, sin tener una fuerte dependencia geométrica, se piensa que se podrían mitigar estas limitaciones. En concreto, el modelo basado en CNN planteado en el apartado 3.3, cambia el enfoque del tratamiento de características y su posterior *matching*, al no recurrir a etapas explícitamente geométricas como son la extracción de *landmarks* y su asociación.

Para que este método sea integrable como asistencia a la etapa de *matching* clásico, debe aportar la información sobre el movimiento entre observaciones consecutivas. Este movimiento, al ser relativo, requiere una consistencia local mientras que la consistencia global no es necesaria en esta etapa del SLAM. El proceso iterativo de asociación de nubes de puntos se encarga de calcular el movimiento entre observaciones para, posteriormente, ser integrado en las etapas posteriores del sistema de SLAM. Esta característica es fundamental para que se pueda emplear el método de modelos basados en CNN propuesto en el apartado 3.3. Con este método se alcanzaba una buena precisión respecto a las estimaciones de velocidad y variación de guiñada. Sin embargo, se concluía que dicho método no alcanzaba la consistencia global debido a los errores incrementales que se producían.

El modelo planteado de CNN, al no depender geoméricamente de características extraídas, se piensa que puede proporcionar una estimación bajo cualquier circunstancia a pesar de que el entorno sufra de oclusiones o no ofrezca suficientes características relevantes. Estas situaciones pueden dar lugar a un aumento del error cometido por algoritmos convencionales debido a la dependencia geométrica del entorno (poca cantidad o baja calidad de *landmarks*) o incluso por la poca definición de los *landmarks* por dispersión entre puntos.

Se propone un método que, mediante la asistencia de la estimación de un modelo de *Deep Learning*, consiga una asociación robusta durante la etapa de *matching* en un sistema de SLAM basado en LiDAR 3D (figura 3.51). Este método propuesto complementa la estructura general de este tipo de algoritmos ilustrada mediante la figura 3.11 en el apartado 3.2. En primer lugar, se debe verificar que dicha estimación de la red es aplicable al problema propuesto. Posteriormente, se plantean varios escenarios donde la red pueda complementar las desventajas propias de métodos clásicos.

Figura 3.51: Estructura general del algoritmo de SLAM junto a la integración de la asistencia del modelo CNN (en rojo).

Además, el método propuesto es extrapolable a cualquier método de SLAM cuya etapa de *matching* utilice un algoritmo que pueda admitir una aproximación inicial. En tal caso, la estimación de la red se plantea como alternativa a la estimación propia del sistema de SLAM cuando esta última no está disponible o carece de precisión. No se pretende, sin embargo, mejorar la precisión del sistema, ya que la estimación de la red asiste al inicio de la etapa de asociación como primera aproximación. La precisión de la salida de esta etapa, así como de etapas sucesivas del sistema completo de SLAM, dependerá exclusivamente de cómo hayan sido implementadas estas.

3.4.2. Trabajos previos

Como ya se indicó, el campo del *Machine Learning* y, en concreto, el uso de redes neuronales de aprendizaje profundo (Deep Neural Networks, DNNs) han conseguido posicionarse como métodos del estado del arte en cuanto a clasificación semántica, sobre todo, dentro del campo de la visión artificial. Sin embargo, aún no han conseguido alcanzar la misma relevancia resolviendo problemas geométricos, como la estimación de movimiento entre observaciones o

el mapeado de un escenario, debido a la precisión limitada que se logra.

En el primer caso, una clasificación semántica del entorno trata de resolver un problema fundamentalmente descriptivo, donde un análisis cualitativo del entorno puede generar una estimación correcta y, por tanto, la utilización de métodos de *Deep Learning* alcanzan una gran eficiencia. Sin embargo, cuando se trata de resolver un problema con gran dependencia de variables geométricas, este suele estar mejor condicionado de forma analítica. Por ejemplo, en el problema de mapeado y, concretamente, durante su etapa de asociación de datos, es un problema definido analíticamente y, por tanto, se deberían obtener resultados más precisos resolviendo numéricamente dicho problema que desde un enfoque basado en la experiencia, como sería el caso de utilizar DNNs. Puede plantearse que la precisión lograda se deba a una falta de ejemplos en el *dataset* de entrenamiento, y que, proporcionando una mayor cantidad de observaciones y variedad de escenarios, se alcanzaría mejor precisión, sin embargo, este planteamiento puede ser cuestionable si se quiere encontrar un equilibrio entre precisión y eficiencia del método.

Aun así, existen trabajos previos en los que se integran métodos basados en *Deep Learning* para resolver problemas geométricos. Estos métodos, al igual que el desarrollado en el apartado 3.3, tratan de resolver un problema de regresión que obtenga el movimiento relativo a partir de una representación del entorno (ya sea mediante visión artificial o LiDAR 3D). En concreto, en Zhou et al. (2017); Ummenhofer et al. (2017), mediante aprendizaje no supervisado, tratan de resolver este problema relacionando variables de profundidad y de movimiento. Otro algoritmo, más orientado a Visual SLAM, como el planteado en Bloesch et al. (2018), utilizando como variables de entrada profundidades e intensidad, optimizan posteriormente la estimación del movimiento mediante Bundle-Adjustment. Otros métodos, como los planteados en Parisotto et al. (2018); Henriques y Vedaldi (2018), debido a la dependencia temporal del problema, utilizan redes recurrentes (RNNs) para inferir el movimiento utilizando una secuencia de imágenes a color con una capa de profundidad mediante aprendizaje supervisado.

El problema que se trata de resolver en este apartado se centra en la optimización durante la asociación de nubes de puntos. Como se indicaba, este proceso suele caracterizarse por minimizar el error de asociación de forma iterativa. Estos algoritmos de asociación o registro de datos pueden clasificarse en dos grupos: métodos de asociación local o global.

Respecto a los métodos que realizan la optimización de forma local, se encuentran algoritmos como ICP (Segal et al., 2009), NDT (Biber y Straßer, 2003; Magnusson, 2009) o métodos probabilísticos (Danelljan et al., 2016; Myronenko y Song, 2010; Jian y Vemuri, 2005). Estos se caracterizan por depender de una inicialización para obtener una buena convergencia del método. En el caso concreto del vehículo autónomo, esta inicialización se basa en la última estimación correcta y registrada del sistema de SLAM.

Por otro lado, los métodos de optimización globales tratan de establecer correspondencias entre dos observaciones utilizando descriptores SIFT (Scovanner et al., 2007), SHOT (Tombari et al., 2010), NARF (Steder et al., 2010) o ajustes geométricos (Zeng et al., 2017). Estos métodos, además, suelen gestionar la identificación de outliers como es el caso del algoritmo de RANSAC y sus variaciones (Torr y Zisserman, 2000) y, al contrario que los métodos de asociación locales no requieren una aproximación inicial, pudiendo llegar a una convergencia

a partir de una inicialización aleatoria, como por ejemplo el algoritmo Coherent Point Drift (CPD) (Myronenko y Song, 2010) usado para la asociación de nubes de puntos 3D empleando un método probabilístico de estimación de densidades. Sin embargo, estos métodos suelen requerir más iteraciones para converger a un resultado.

Trabajos previos como Ding y Feng (2019) utilizan una pre-asociación de las nubes de puntos mediante el algoritmo ICP, para posteriormente refinar el resultado con una arquitectura basada en un modelo de aprendizaje no supervisado y otro de clasificación binaria que optimiza la reconstrucción del mapa. Otro método, presentado en Ramezani et al. (2020), utiliza de nuevo una alineación previa de las nubes de puntos mediante ICP, para posteriormente utilizar un modelo pre-entrenado que realice *loop-closure* en su sistema propuesto de SLAM. Ambos trabajos utilizan el primer ajuste grueso mediante ICP, el cual se ha indicado que se clasifica como de optimización local y que necesita una inicialización para obtener una buena convergencia. En caso de no disponer de una buena aproximación inicial, puede llevar a aumentar el tiempo de cómputo total, lo que no es operativo para aplicaciones de vehículo autónomo, donde el tiempo de ejecución es crucial.

Por ello, en este apartado se plantea un sistema que combine un enfoque convencional de asociación de nubes de puntos junto a las capacidades que ofrece un modelo de DNN, pero al contrario que estos trabajos, el modelo basado en *Deep Learning* es utilizado para realizar la pre-asociación de las nubes de puntos, ya que por un lado no necesita inicialización para converger y, además, no alcanzaría una alta precisión final, como sí puede lograr un método analítico.

Como se ha comentado, el método planteado es extrapolable a cualquier método de SLAM que utilice métodos de asociación de datos con optimización local, es decir, que requieran de una inicialización para mejorar su eficiencia. Esta primera aproximación puede ser asistida con la estimación de la red o incluso ser sustituida completamente si el método convencional no logra converger correctamente.

En primer lugar, se debe verificar que, utilizando una estimación inicial proveniente de DNNs, se logra alcanzar una convergencia adecuada y una salida global que, se espera, tenga una precisión similar a la del método clásico. También, en este primer estudio, evaluar en qué medida los métodos de asociación de nubes de puntos dependen de una estimación previa correcta.

Por otro lado, la precisión del sistema de SLAM depende de la etapa de asociación que, a su vez, dependerá de la información proporcionada del entorno. Por tanto, se debe verificar la correlación existente entre la información del entorno y la precisión durante la asociación. Identificando esta relación, se plantea que la estimación de la red pueda ayudar a converger al sistema de SLAM, cuando el resultado de la asociación no sea posible o fiable.

3.4.3. Metodología propuesta para etapa de asociación asistida por modelo CNN

Debido a que el funcionamiento de la red es independiente al resto del sistema de SLAM, la estimación del modelo estará disponible al comienzo de la etapa de asociación del sistema de SLAM. La asistencia se realiza en dos fases. Previo al comienzo de la etapa de asociación,

se comprueban que los valores de inicialización se encuentran dentro el orden de magnitud esperado a partir de su comparación con la estimación de la red. La segunda fase tiene lugar a la salida del método de asociación. En función de la calidad de la asociación, considerando el rendimiento o la precisión alcanzada, se realimenta de nuevo el algoritmo de asociación convencional inicializando, esta vez, con la estimación de la red ya que, como se justificaba en el subapartado 3.4.1, el enfoque analítico obtendrá mejores resultados. Sin embargo, si a pesar de ser inicializado con la estimación de la red, en una segunda evaluación, el error de asociación sigue siendo elevado, se descarta dicha estimación y se utiliza la de la red, ya que así se evitaría una posible pérdida del orden de magnitud en el peor de los casos. El principal indicador para evaluar la calidad de la asociación es el RMSE de asociación, tal y como se observará en los resultados de cada ensayo. La figura 3.52 ilustra esta metodología indicando las dos fases descritas y se corresponde el desglose de la etapa de "Asociación de datos" indicada en la figura 3.51

Figura 3.52: Diagrama de flujo de la metodología propuesta de asistencia a la etapa de asociación.

3.4.4. Modelo de *Deep Learning* utilizado para la asistencia a la asociación de datos

Primeramente, para poder describir el tipo de asistencia que se proporcionará al sistema de SLAM, se describe el procedimiento del diseño del modelo de *Deep Learning* y los resultados obtenidos.

El enfoque para diseñar la arquitectura de red del modelo utilizado en esta aplicación es el mismo que el descrito en el apartado 3.3, por lo que no es necesario describir en profundidad todo el proceso. Sin embargo, cabe recordar algunos aspectos más relevantes o que se hayan adaptado para esta aplicación.

Respecto a los datos, se utilizará como entrada a la red las imágenes-dato en formato mapa de profundidades, ya que esta configuración resultó ser la que mejor condicionaba el problema geométrico que se trata. En este caso se emplea un LiDAR de 64 capas (modelo Ouster OS-1), con el que será posible reconstruir imágenes de mayor resolución (figura 3.53).

Figura 3.53: Ejemplo de mapa de profundidad del dataset del parking de La Gavia.

Como datos de salida, se adquiere la velocidad del vehículo, a partir de la información disponible por BUS-CAN, y el ángulo de guiñada, mediante un giróscopo de alta resolución (modelo RMS FES 33). Con estos datos sincronizados con las nubes de puntos entre dos instantes, es posible llevar a cabo un aprendizaje supervisado.

El *dataset* utilizado en este caso ha sido elaborado a partir de una serie de ensayos llevados a cabo en un entorno específico. Debido a que el objetivo subyacente es mejorar la precisión en el posicionamiento bajo cualquier circunstancia, se requiere buscar las capacidades de los sistemas de Visual SLAM basado el LiDAR. Para ello, se ha planteado crear un *dataset* en un entorno de parking subterráneo, donde por un lado tienen aplicación directa los algoritmos de Visual SLAM al no disponer de señal de posicionamiento desde receptores por satélite. Por otro lado, en el apartado 3.2, donde se planteaba un método convencional de SLAM, las conclusiones sobre los resultados obtenidos en parking apuntaban a un peor rendimiento. Este hecho es debido a la información poco fiable o “ruidosa” que generaban las superficies planas de las columnas, ya que, al caracterizarse por ser normalmente estrechas, no quedaban definidas correctamente. Por todo ello, se piensa que este entorno es el idóneo para poner a prueba el método planteado en este apartado.

En concreto, la adquisición de datos para crear el *dataset* ha sido realizada en el parking subterráneo del Centro Comercial de La Gavia (Madrid) en 15 secuencias distintas, lo que da lugar a una duración total de unos 25 minutos. Las velocidades durante la conducción han sido en torno a 10 – 15 km/h, por lo que esta velocidad baja puede poner en compromiso la fiabilidad del sistema de asociación de datos y, además, los giros han sido pronunciados

en algunos casos al girar esquinas de 90° . El conjunto de datos completo consta de 15453 observaciones, las cuales se han dividido en el conjunto de entrenamiento y evaluación. En cuanto al conjunto de entrenamiento, tras una llevar a cabo la técnica de *Mirroring* para la aumentación de datos (descrito en el subapartado 3.3.2.3), da lugar a un total de 25448 observaciones. Se ha seleccionado una secuencia completa de 2.729 *samples* como conjunto de evaluación. Esta distribución supone una proporción de 90% - 10% para los conjuntos de entrenamiento y evaluación respectivamente. La figura 3.54 muestra la trayectoria elegida para el conjunto de evaluación, donde se puede observar que existe una variedad entre tramos rectos y giros pronunciados.

Figura 3.54: Trayectoria seleccionada para evaluación.

En cuanto al modelo de *Deep Learning*, se basa en la misma topología que la planteada en el subapartado 3.3.3.3, donde el modelo se constituía por una serie de capas convolucionales (bloque de extracción de características) que se conectaban a un conjunto de capas *fully-connected* (bloque de estimación de valores). Los datos de entrada del modelo son dos imágenes-dato de los mapas de profundidad en dos instantes de tiempo consecutivos, mientras que los datos de salida será la velocidad del vehículo y su variación de guiñada entre t y $t - 1$. Como método de regularización, se sigue utilizando *dropout*, donde se verifica su gran influencia sobre la precisión del resultado al igual que ocurría en el desarrollo anterior (apartado 3.3).

Otros aspectos importantes respecto al entrenamiento del modelo son la tasa de aprendizaje y el tamaño designado de *mini-batch*. Ambos parámetros tienen una fuerte interrelación entre ellos y una gran influencia sobre el resultado. En este desarrollo se han designado unos valores de 10-3 con *learning rate decay* y un tamaño de *mini-batch* de 500 ejemplos.

La figura 3.55 muestra los resultados de la estimación de la red. Se comparan las señales estimadas de velocidad y variación de guiñada con la proporcionada como *Ground Truth*, obteniendo unos RMSE de 1,018 km/h y 4,153 $^\circ$ /s, respectivamente.

Figura 3.55: Estimación del modelo para la secuencia de evaluación. En azul, la señal de referencia para cada variable estimada, en rojo la salida de la red.

Como se indicaba en el subapartado 3.3.4.2, utilizar la métrica de error de RMSE durante el proceso de entrenamiento del modelo es correcto para optimizar el ajuste de los parámetros, sin embargo, no es un buen indicador de la precisión global, ya que depende de la naturaleza de los datos utilizados como *Ground Truth*. En este sentido, se puede apreciar como la señal de referencia para la velocidad, no sufre del ruido que sí aparece en la señal de referencia de la variación de guiñada. Por ello, cabe destacar que, ante una señal con mayor ruido, es posible obtener un mayor RMSE y, sin embargo, la precisión de la estimación sea mejor.

Para ello, al igual que se describe en el subapartado 3.3.4.2, se plantea validar la precisión de la estimación mediante unos indicadores que sean independientes de la naturaleza de las señales de referencia. En concreto mediante los algoritmos APE y RPE se mide la consistencia global y local, respectivamente. Los resultados obtenidos mediante estas métricas han sido:

- APE: 10,38 m
- RPE: 0,35 m

La figura 3.56 muestra la trayectoria reconstruida de la estimación de la red y la de referencia.

Figura 3.56: Reconstrucción de la trayectoria de la secuencia utilizada como evaluación. En azul, la trayectoria de referencia y en rojo, la reconstrucción de la trayectoria a partir de la estimación de la red.

Se puede observar cómo, efectivamente, el método no llega a obtener la suficiente precisión como para garantizar el posicionamiento en un mapa global. Sin embargo, localmente, puede poder aportar una primera aproximación para la convergencia de un método analítico.

La longitud de la trayectoria de evaluación es de 747 metros, donde la estimación de la red ha obtenido un error medio de translación del 48,38%, mientras que el error medio de rotación ha sido de 0,446 °/m. La figura 3.57 ilustra estos resultados pormenorizado por tramos de evaluación.

Figura 3.57: Errores de translación y rotación en función de tramos de longitud.

A la vista de los resultados obtenidos, se vuelve a verificar que este método garantiza una buena consistencia local, mientras que, al reconstruir la trayectoria, no se logra la suficiente

precisión como para una consistencia global del método.

Nótese que estos resultados aquí descritos son obtenidos a partir del modelo exclusivamente, ya que este tiene un funcionamiento aislado del resto del algoritmo de SLAM sobre el que se pretende asistir en la etapa de *matching*. Una vez se conoce el resultado y precisión esperado por el modelo de *Deep Learning* por sí mismo, se plantean los ensayos preliminares para verificar la dependencia de los métodos convencionales de asociación a una primera aproximación y comprobar que la estimación de la red es aplicable en cada caso.

3.4.5. Valoración de la dependencia de la inicialización para métodos de asociación local

Los métodos de asociación local convencionales requieren una aproximación inicial para lograr un buen funcionamiento y, en caso de no disponer dicha inicialización, pueden obtenerse errores de convergencia, requerir un mayor número de iteraciones o una baja precisión. Para valorar el grado de dependencia de una aproximación inicial, se plantea realizar unos ensayos preliminares sobre dos algoritmos de optimización local de asociación de nubes de puntos: ICP y NDT. Además, por otro lado, se requiere verificar que una estimación aportada por la red puede asistir a estos métodos y proporcionar una salida similar.

Aunque la metodología aquí descrita puede ser empleada en cualquier algoritmo de asociación local de nubes de puntos, la implementación y ensayos realizados se aplicarán sobre estos dos para ilustrar el método, ya que son dos de los algoritmos más utilizados en la etapa de asociación. Además, debido a su facilidad de implementación y flexibilidad, permite realizar todas las verificaciones que se llevarán a cabo. Al tratarse de dos metodologías diferenciadas, es interesante analizar el comportamiento de ambas para que el método propuesto sea generalizable al resto de métodos de asociación.

El ICP es un método conocido y bastante empleado en la etapa de asociación de nubes de puntos, como se ha podido observar en los trabajos previos mencionados en el apartado 3.2. Su funcionamiento se basa en la búsqueda iterativa de la transformación entre dos nubes de puntos que minimice la distancias entre puntos o planos, tras un filtrado y eliminación de *outliers*. Este método se caracteriza por su eficiencia y baja carga computacional.

Por otro lado, el algoritmo NDT (por sus siglas en inglés, Normal-Distributions Transform), basa la asociación entre nubes de puntos que previamente han sido transformadas a *voxels*. A partir del cálculo de la distribución normal de los puntos de cada celda, logra estimar la transformación de una nube de puntos a otra mediante métodos de optimización numérica convencionales como el método de Newton. La principal diferencia respecto al algoritmo ICP es que este método no utiliza los puntos láser directamente para su *matching* evitando, por tanto, la búsqueda del punto más cercano. Este algoritmo, a pesar de tener una velocidad de cómputo más baja que el ICP, puede obtener resultados más precisos.

Se plantean tres situaciones distintas:

- Situación 1. Sin utilizar ninguna realimentación previa:

Los algoritmos, a pesar de poder utilizar una inicialización para mejorar el cálculo del *matching* entre nubes de puntos, en esta situación no se proporciona ninguna inicialización. Se pretende observar en qué medida puede afectar al método respecto al

rendimiento (iteraciones) y precisión (RMSE de asociación). Se observará en qué grado, el método es susceptible de una buena aproximación inicial.

- Situación 2. Utilizando la propia estimación del sistema de SLAM:
En este caso, la primera aproximación vendrá dada por la última estimación de movimiento correcta que ha registrado el sistema de SLAM. Como se adelantaba, en conducción autónoma es posible realizar esta aproximación ya que, por la propia dinámica del problema, se espera que la variación en el siguiente instante de tiempo no difiera demasiado de la anterior.
- Situación 3. Utilizando la estimación de la red como valor de inicialización:
En esta situación, se pretende valorar si la estimación de la red, como proceso en paralelo, puede proporcionar una primera estimación que asista a la convergencia del método. La precisión final dependerá del método en sí. Sin embargo, es interesante evaluar también si existen condiciones donde exista un comportamiento diferente a la situación 2, ya sea por baja variación de desplazamiento o, al contrario, en situaciones con una alta variación de translación o rotación.

Se espera que exista una correlación entre el tipo de realimentación que se introduzca a la etapa de asociación y la convergencia y precisión obtenida durante la misma. Como se indicaba en el subapartado 3.4.1, los métodos convencionales pueden tener alguna limitación en función de la variación de movimiento entre *frames*. Por ello, es de especial interés verificar si existe una relación entre las iteraciones del método de asociación frente a la velocidad y la variación de guiñada. Asimismo, valorar si estas situaciones pueden afectar a la precisión, a partir de la evaluación del RMSE de asociación. De obtener una relación entre iteraciones y RMSE frente a la velocidad o variación de guiñada, se plantea una estrategia para la fase 2 de la metodología propuesta, donde se la inicialización de la etapa de asociación pueda ser llevada a cabo por la propia estimación del sistema de SLAM o bien, de la estimación proveniente de la red.

3.4.5.1. Ensayos preliminares sobre la dependencia del tipo de inicialización en algoritmo ICP

Se estudia el comportamiento del algoritmo ICP en los tres supuestos diferentes: 1) utilizando la propia estimación del sistema de SLAM, 2) sin ninguna realimentación y 3) con la estimación de la red. El objetivo principal de estas pruebas es comprobar que la estimación del modelo de CNN puede ser utilizada como inicialización y obtener un comportamiento similar del sistema. Además, se pretende realizar un análisis de dónde los métodos de asociación de nubes de puntos pueden tener limitaciones en cuanto al rendimiento y la precisión en función de la variación de desplazamiento entre *frames*. Esta limitación podría minimizarse utilizando la estimación de la red al no tener una fuerte dependencia geométrica para la asociación.

Para medir el rendimiento y la precisión en función de la variación de desplazamiento, se analiza el número de iteraciones y el RMSE obtenido tras la etapa de asociación. Los resultados obtenidos de estos valores para las tres situaciones se describen en la tabla 3.9.

Tabla 3.9: Rendimiento y precisión en algoritmo ICP en las tres situaciones planteadas.

	Situación 1	Situación 2	Situación 3
Media de iteraciones	5,09	5,09	3,58
RMSE de asociación medio (m)	0,248	0,248	0,248
RMSE translación (m)	0,039	0,039	0,039
RMSE rotación (°)	0,403	0,401	0,399

A la vista de los resultados, se puede destacar que una variación en la inicialización del método no afecta a la precisión de la salida (figura 3.58). Sin embargo, sí que tiene influencia en el rendimiento. Así, mientras que la precisión de la estimación es prácticamente igual en las tres situaciones, el número de iteraciones medio requerido en cada situación es variable. En el caso de no realimentar el algoritmo con una inicialización, baja el rendimiento a 5 iteraciones de media durante el trazado completo. Cabe destacar que entre la Situación 2 y 3, no existe una gran diferencia en el número de iteraciones, pero es ligeramente mejor en el caso de utilizar la propia estimación, por lo que se plantea buscar si existen situaciones específicas donde haya un mejor rendimiento de una u otra.

Figura 3.58: Precisión obtenida con ICP en las tres situaciones en estudio.

Tanto el rendimiento como la precisión durante la etapa de asociación, puede depender de la estimación previa. Asimismo, esta influencia dependerá de la transformación que se debe calcular en la iteración actual. Por ello, se realiza un análisis en cuanto a dos indicadores: número de iteraciones, para el rendimiento y RMSE de asociación, para la precisión. Ambas en función de la variación de desplazamiento.

Se determina, en primer lugar, la correspondencia entre la velocidad llevada por el vehículo y el número de iteraciones que requiere el método para converger a un resultado. Cabe indicar que la salida se obtiene una vez que el método alcanza cierta tolerancia mínima de error o bien, llega a un número máximo de iteraciones preestablecido. Representando el número de iteraciones en función de la velocidad como un histograma bidimensional, es posible identificar los rangos de funcionamiento en cada situación.

La figura 3.59 ilustra mediante dicho histograma 2D, la distribución del número de itera-

ciones según la velocidad a lo largo del recorrido. Se puede identificar cómo en el caso de no utilizar una realimentación de inicialización, se requiere más de 3 iteraciones para prácticamente todo el rango de velocidades. En cambio, las situaciones 2 y 3, tienen una distribución muy similar y, por tanto, el rendimiento de ambos métodos. Además, esta representación, también permite identificar la distribución del número de iteraciones para cada velocidad, como si de un mapa de calor se tratase. Así, por ejemplo, se mencionaba anteriormente que, aunque la situación 2 y 3 tienen rendimientos muy similares, la situación 2 es ligeramente mejor. Esta diferencia se aprecia en el rango de velocidades cercado a 10 km/h, donde el método 2 necesita, normalmente, menos iteraciones que en la situación 3 (zonas en amarillo – naranja).

De igual modo, respecto a la variación del ángulo de guiñada (figura 3.59), se identifica que en la situación 1, se requiere un mayor número de iteraciones, mientras que la situación 2 y 3, es bastante similar. Se puede observar, sin embargo, que en la situación 3, existe una zona comprendida entre una variación de guiñada de 20 °/s a 30 °/s, donde debido a esta inicialización, el método requiere de más iteraciones. Por tanto, se identifican las zonas donde cada situación encuentra más limitaciones en cuanto a rendimiento.

Figura 3.59: Representación de rendimiento del algoritmo ICP. De naranja a azul, se representa la mayor o menor repetición del valor de número de iteraciones, respectivamente.

Por otro lado, respecto a la precisión obtenida por este método, se había identificado a partir

de la tabla 3.9, que el algoritmo de ICP obtenía los mismos resultados independientemente de la inicialización utilizada. Valorando esta precisión mediante el RMSE de asociación, la distribución de este valor en función de la velocidad y la variación de guiñada ofrece un mismo comportamiento como era de esperar (figura 3.58). Sin embargo, cabe destacar que existe una relación entre la precisión obtenida y el movimiento realizado. En este sentido, se obtienen peores resultados cuanto mayor es la velocidad (valor más elevado de RMSE), mientras que, en el caso de la variación de guiñada, ocurre al contrario, logrando una mejor asociación para variaciones de ángulo mayores.

Figura 3.60: Representación del RMSE de asociación del algoritmo ICP. De naranja a azul, se representa la mayor o menor repetición del valor de error de asociación obtenido, respectivamente.

Tras el análisis realizado con el método ICP, cabe destacar que se obtiene siempre la misma precisión, independientemente de la realimentación que se tenga y, además, existe una correlación entre el RMSE obtenido y la variación de desplazamiento. Por otro lado, el rendimiento del algoritmo es dependiente en función de la realimentación, de este modo, se ha comprobado que el método ofrece un mejor rendimiento siempre que tenga disponible una primera aproximación, como era de esperar. Además, tanto en la medición del rendimiento como en la precisión, se logran unos resultados similares en el caso de asistir con la estimación de la red, por lo que se hace posible el utilizar este sistema con un funcionamiento en paralelo al resto del sistema de SLAM.

3.4.5.2. Ensayos preliminares sobre la dependencia del tipo de inicialización en algoritmo NDT

Se procede igualmente para el método de asociación NDT. Es interesante, de cara al objetivo de extrapolar el método propuesto en este apartado para cualquier método de SLAM, realizar el análisis para este algoritmo porque, aunque se trate igualmente de un método de asociación local, su funcionamiento difiere del empleado por el ICP. De esta manera, tras una comparación de las correlaciones del método ICP y las obtenidas en este, se pretende generalizar el comportamiento.

La tabla 3.10 resume los valores obtenidos durante el trazado de evaluación. El valor medio del número de iteraciones es más dependiente de la realimentación que en el caso de utilizar el ICP. En concreto, al no utilizar realimentación, el método obtiene de media más de 9 iteraciones (este valor está limitado por el número máximo de iteraciones preestablecido en 30), mientras que en las situaciones 2 y 3 se obtiene un valor considerablemente más bajo.

Respecto a la precisión de la asociación, esta se ve comprometida en función de la realimentación. Así, se obtiene un valor superior en el error cometido en la situación 1, tanto en la asociación como en el error de translación y rotación. Sin embargo, en el caso de la situación 2 y 3, se obtienen valores similares, llegando a ser ligeramente mejores en el caso de una realimentación inicial de la red.

Tabla 3.10: Rendimiento y precisión en algoritmo NDT en las tres situaciones planteadas.

	Situación 1	Situación 2	Situación 3
Media de iteraciones	9,48	1,51	2,84
RMSE de asociación medio (m)	0,303	0,248	0,247
RMSE translación (m)	0,192	0,014	0,014
RMSE rotación (°)	0,970	0,210	0,203

Por tanto, se pone de manifiesto de nuevo la premisa inicial, en la que estos métodos dependen de una aproximación previa que afectará al rendimiento y la precisión. Además, la asistencia de la red mantiene la precisión del método, como era de esperar, aunque aumenta ligeramente el número de iteraciones. La figura 3.61 ilustra la precisión de cada situación sobre la reconstrucción de la trayectoria.

Figura 3.61: Precisión obtenida con NDT en las tres situaciones en estudio.

Se observa cómo este método, al contrario que ocurre con el método ICP, tiene una fuerte dependencia de la inicialización sobre la precisión obtenida.

De igual modo, se procede a realizar un análisis pormenorizado del rendimiento y la precisión frente a la variación de desplazamiento entre *frames*.

La figura 3.62 muestra como la situación 1, obtiene el máximo número de iteraciones para cualquier rango de velocidades o variación de guiñada. En cuanto a las situaciones 2 y 3, cabe destacar que se obtiene un peor rendimiento para velocidades superiores, pero también para desplazamientos cercanos a cero, llegando a ser más sensible en estos casos, la situación 2. Respecto al comportamiento del método frente a la variación de guiñada, se observa que el método 2 requiere de menor número de iteraciones que la situación 3 en la mayoría de los casos.

Figura 3.62: Representación de rendimiento del algoritmo NDT. De naranja a azul, se representa la mayor o menor repetición del valor de número de iteraciones, respectivamente.

En cuanto a la representación de la precisión frente a las variaciones de desplazamiento, tal y como se observaba en la tabla 3.10, la situación 2 y 3 obtienen resultados similares. Esto se refleja en que los histogramas 2D de la figura 3.63 tengan una apariencia similar tanto en el caso del RMSE en función de la velocidad, como en función de la variación de guiñada. Sin embargo, es interesante que el valor obtenido de error es superior para velocidades en el rango superior del ensayo, mientras que ocurre de forma opuesta para variaciones de guiñada. Este comportamiento es igual al estudiado en el algoritmo ICP.

Figura 3.63: Representación del RMSE de asociación del algoritmo NDT. De naranja a azul, se representa la mayor o menor repetición del valor de error de asociación obtenido, respectivamente.

3.4.5.3. Discusión de los resultados sobre la dependencia de la realimentación

Se han comparado dos algoritmos frecuentemente utilizados para la asociación de nubes de puntos dentro de algoritmos de SLAM basados en LiDAR. Además, estos algoritmos tienen un modo de funcionamiento muy diferente, por lo que resulta interesante analizar su comportamiento de cara a que el método propuesto sea extrapolable a la mayoría de las soluciones de SLAM que consideren métodos de asociación local.

En cuanto al algoritmo ICP, se observa que la precisión obtenida de asociación no se ve comprometida por la inicialización utilizada, pero sí que existe una dependencia respecto al

rendimiento obtenido.

Por otro lado, el algoritmo NDT, es dependiente en mayor medida tanto en la precisión como el rendimiento según la inicialización que se utilice.

Es importante destacar que, tanto en la situación 2 y 3, se obtienen comportamientos similares en cuanto a rendimiento y precisión. Este hecho, hace posible que la estimación de la red pueda asistir al método de asociación. Además, se ha verificado que existe un mismo comportamiento en ambos métodos en función de la variación de desplazamiento. En este sentido, ambos métodos obtienen un mayor error para velocidades situadas en el rango superior del ensayo y, sin embargo, para valores de variación de guiñada cercanas a cero.

Por tanto, se verifica que la inicialización basada en la red es posible integrarla en el sistema y, además, esta estimación se obtiene de forma independiente al resto del sistema de SLAM. Puede implementarse para delimitar valores atípicos de inicialización como se plantea en la fase 1 de la metodología propuesta y que se estudia en profundidad en el siguiente subapartado.

Sin embargo, no es posible apreciar diferencias significativas según la variación de desplazamiento entre utilizar la inicialización del propio sistema de SLAM y la que proporciona la red, por lo que la estrategia a seguir en la fase 2 de la metodología propuesta se basará en el resultado obtenido de una asociación llevada a cabo, en primer lugar, por la estimación previa del sistema.

3.4.6. Verificación del funcionamiento del método ante perturbaciones de información de entorno

Una vez se ha verificado que la estimación proporcionada por la red se puede considerar válida para asistir al algoritmo de asociación, es necesario evaluar si el método propuesto cumple con los objetivos e hipótesis iniciales y conseguir una etapa de *matching* robusta ante las limitaciones que el entorno puede generar sobre los métodos convencionales.

El inconveniente principal de los algoritmos de asociación está relacionado con la información disponible del entorno. Según sea el algoritmo de extracción de características del sistema de SLAM, estos métodos podrán ser más sofisticados o genéricos, reportando en cada caso una precisión distinta. Como se indicaba, el objetivo principal del método propuesto en este apartado no es mejorar la precisión, ya que dependerá del funcionamiento intrínseco de la solución de SLAM. Sin embargo, esta precisión puede tener una gran dependencia de la calidad de *landmarks* que estén disponibles del entorno. Concretamente, en los algoritmos basados en LiDAR 3D, estos problemas de extracción pueden ocurrir principalmente por dos razones: poca información relevante del entorno o debido a oclusiones.

Por tanto, en este subapartado se comprueba la validez del método propuesto para una asociación de datos robusta, simulando para ello las situaciones que puedan dar lugar a una baja calidad de los *landmarks* del entorno. Estas simulaciones se llevarán a cabo sobre los algoritmos de asociación bajo estudio: ICP y NDT. Se busca, por un lado, comprobar cómo afecta cada simulación al rendimiento y precisión de cada algoritmo y, al mismo tiempo, al modelo pre-entrenado de CNN. Evaluando el comportamiento de ambos, se puede diseñar una estrategia para que la salida de la red asista cuando exista una mala estimación por

parte del algoritmo de asociación.

En primer lugar, para simular un entorno con baja información relevante para extracción de características se contemplan dos tipos de ensayos. La calidad de la información del entorno y la de los *landmarks* está relacionada con su frecuencia de identificación y, por otro lado, con el grado de definición de estos. Por ello, se ha llevado a cabo una simulación donde se acota el alcance de visión del láser con el objetivo de disminuir el número de *landmarks* disponibles, pero manteniendo el grado de definición de cada uno (figura 3.64.a). Para el segundo caso, se realiza un *downsampling* con una distribución aleatoria uniforme, de esta forma, se disminuye el grado de definición de los elementos característicos, pero sigue estando disponible el mismo número de ellos (figura 3.64.b).

Figura 3.64: Ejemplos de los entornos simulados. a) Limitación de alcance; b) *downsampling* aleatorio.

Por otro lado, la segunda problemática descrita se relaciona con las oclusiones producidas por los elementos más próximos al láser y que, por tanto, delimitan el rango visible a aquellas zonas donde podrían extraerse *landmarks*. Esta situación, en última instancia, correspondería a una zona del rango de visión por la que no se adquieren datos. Por ello, esta situación se ha simulado restringiendo la adquisición de puntos en un sector circular (figura 3.65). De nuevo, se plantean dos situaciones distintas. La primera, en la que la zona ciega se sitúa siempre en el mismo lugar y otra, donde esta región va cambiando de posición aleatoriamente. De esta forma, se pretende poner en compromiso la robustez del algoritmo delimitando la zona de adquisición de datos desde distintas posiciones.

Figura 3.65: Ejemplo de entorno simulando situaciones con oclusiones.

A partir de los resultados obtenidos en el subapartado anterior, se ha observado una correlación entre la velocidad, la variación de guiñada y el error de asociación obtenido. Por tanto, es interesante que las perturbaciones simuladas del entorno se lleven a cabo en un tramo que combine estas situaciones. En concreto, el tramo situado entre el *frame* 1.100 y el

1.500 (correspondiente a los instantes 110 y 150 segundos), enlaza dos giros de 90° y un tramo recto, por lo que existirán variaciones tanto en la velocidad como en el ángulo de guiñada. Todas las simulaciones son evaluadas en el caso de que la etapa de asociación inicialice con la propia estimación previa y en el caso de utilizar la estimación de la red.

Para el seguimiento del resto del apartado, las simulaciones llevadas a cabo se definen en la tabla 3.11.

Tabla 3.11: Designación de las simulaciones propuestas.

Ensayo	Descripción	Objetivo
Limitación alcance (10 y 15 metros)	Rango máximo de visión limitada a la distancia definida	Disminuir disponibilidad de <i>landmarks</i> , manteniendo definición
<i>Downsampling</i> 65%*	Eliminación aleatoria uniforme del 35% de los puntos	Disminuir definición de <i>landmarks</i> , manteniendo la disponibilidad
Limitación sector circular fijo	Rango de valores de ángulo horizontal sin puntos	Restricción de información por oclusión fija
Limitación sector circular random	Rango de valores de ángulo horizontal sin puntos variable de forma aleatoria	Restricción de información por oclusión desde distintas posiciones

* El valor de 65% de *downsampling* es seleccionado como un valor acorde a una disminución de la definición del entorno. Un valor mayor de *downsampling* (mantener mayor número de puntos) no pondría en compromiso ninguno de los métodos evaluados. En caso contrario, un valor menor distorsionaría demasiado la información del entorno como para poder considerarse semejante a la realidad.

3.4.6.1. Comportamiento del modelo de CNN en los entornos simulados con perturbaciones en la información del entorno

Se espera que la estimación de la red sea más inmune a estas perturbaciones de la nube de puntos y se pueda utilizar dicha estimación cuando el sistema de SLAM detecte que el entorno no proporcione suficiente información relevante como para realizar una asociación correcta.

El modelo pre-entrenado, definido en el subapartado 3.4.4, es utilizado para evaluar su comportamiento ante las nuevas situaciones simuladas. La tabla 3.12 muestra los resultados obtenidos respecto al RMSE de la estimación de la velocidad y de la variación de guiñada. Cabe aclarar que estos valores pueden ser utilizados para la comparación de la precisión obtenidos previamente, ya que se trata del mismo *dataset* y la misma configuración.

Tabla 3.12: Resultados del modelo CNN pre-entrenado para las situaciones simuladas.

Ensayo	RMSE Velocidad (km/h)	RMSE variación de guiñada (°/s)
<i>Entorno sin perturbación simulada (referencia)</i>	<i>1,0181</i>	<i>4,153</i>
Limitación alcance 10 m	1,242	4,558
Limitación alcance 15 m	1,027	4,244
<i>Downsampling 65%</i>	1,312	4,714
Limitación sector circular fijo	1,471	4,371
Limitación sector circular random	1,198	4,982

A la vista de estos resultados, se verifica que el modelo de *Deep Learning* mantiene la salida en el orden de magnitudes esperable, por lo que refleja que su dependencia geométrica con el entorno es menor. En cuanto a la estimación de la variación de guiñada, se mantiene robusta en la mayoría de los casos proporcionando un ajuste similar al entorno sin perturbaciones simuladas, mientras que la estimación de la velocidad sí que varía de forma sustancial en algunas situaciones. Sin embargo, como se estudió en el subapartado 3.3.4.2, la variación de la guiñada es el parámetro con mayor relevancia de cara a la precisión de la estimación, por lo que esta mayor sensibilidad de la estimación de la velocidad puede ser asumible.

La figura 3.66 ilustra, a modo de ejemplo, la señal de las salidas (velocidad y variación del ángulo de guiñada) para el caso de la limitación del alcance a 10 metros. Se observa que la desviación que se produce en el tramo donde se introduce la limitación del entorno (entre el instante 110 y 150 segundos), es mayor en la estimación de la velocidad.

Figura 3.66: Comportamiento del modelo de CNN en la situación de limitación de alcance de 10 m. Las líneas discontinuas indican el tramo de limitación simulada.

Por otro lado, se destaca el comportamiento del modelo en la situación de la limitación del sector circular fijo, donde se ha obtenido la mayor desviación en la estimación de la velocidad (figura 3.67). A pesar de la desviación, la red es capaz de proporcionar una salida dentro del orden de magnitud esperado, por lo que generaliza de forma adecuada.

Figura 3.67: Comportamiento del modelo de CNN en la situación de limitación de oclusión fija. Las líneas discontinuas indican el tramo de limitación simulada.

Se observa que la estimación de la variación de guiñada sigue siendo precisa en cualquier situación, siendo además la variable más relevante para una mejor reconstrucción de la trayectoria.

3.4.6.2. Comportamiento de los algoritmos de asociación en los entornos simulados con perturbaciones en la información del entorno

De igual modo, se evalúan los dos algoritmos de asociación bajo estudio sobre cada una de las situaciones simuladas. El objetivo es identificar cómo afectan las perturbaciones sobre la información del entorno en el rendimiento y la precisión de asociación en cada caso. Realizando este análisis, se podrá diseñar correctamente la fase 2 de la metodología propuesta.

Método de asociación mediante ICP:

La tabla 3.13 muestra los valores obtenidos de los indicadores tanto de rendimiento como de precisión para cada situación. La primera fila de la tabla, indica los resultados obtenidos previamente en el entorno sin perturbación como referencia para los resultados obtenidos en las nuevas situaciones.

Tabla 3.13: Resultados del algoritmo ICP para las situaciones simuladas.

Ensayo	Media de iteraciones	RMSE de asociación medio (m)	RMSE translación (m)	RMSE rotación (°)
<i>Entorno sin perturbación simulada (referencia)</i>	3,06	0,2480	0,0389	0,4008
Limitación alcance 10 m	9,08	1,807	5,250	63,256
Limitación alcance 15 m	3,03	0,241	0,044	0,537
<i>Downsampling 65%</i>	3,08	0,253	0,040	0,394
Limitación sector circular fijo	3,11	0,253	0,044	0,435
Limitación sector circular random	4,06	Inf.	Inf.	12,067

Se destaca que este no converge a una salida correcta en las situaciones de “Limitación alcance 10 m” y en la “Limitación sector circular random”, obteniendo unos valores fuera del orden de magnitud que dan lugar a un mal condicionamiento para la siguiente iteración del sistema de SLAM. El algoritmo ICP muestra un comportamiento inestable cuando varía la perturbación provocada en la información del entorno. Así, por ejemplo, mientras que, para una limitación del alcance a 15 metros, el algoritmo logra alcanzar unos valores semejantes al de referencia, cuando la limitación se restringe a 10 metros, el comportamiento es totalmente diferente, llegando a la inestabilidad. La figura 3.68 ilustra la situación para el caso de una limitación a 10 metros, donde se muestran tanto los valores alcanzados en número de iteraciones como en el error de asociación en función de las variaciones de movimiento (velocidad y variación de guiñada).

Figura 3.68: Comportamiento algoritmo ICP ante limitación de la información del entorno a 10 metros. A la izquierda, el número de iteraciones y el error de asociación frente a la velocidad; a la derecha, frente a la variación de guiñada.

Se puede observar cómo el algoritmo estima durante gran parte de los *frames*, valores fuera del orden de magnitud esperado tanto para la velocidad como para la variación de guiñada. Además, se alcanza el número máximo de iteraciones (20 en el caso del algoritmo ICP) y valores de RMSE de asociación muy por encima de la media obtenida en el entorno de referencia.

Método de asociación mediante NDT:

En cuanto al algoritmo NDT, a pesar de que en los ensayos preliminares mostraba una mayor sensibilidad al tipo de realimentación que utilizaba como primera aproximación en etapa de asociación, muestra unos resultados muy robustos ante las distintas situaciones simuladas, tal y como se muestra en la tabla 3.14.

Tabla 3.14: Resultados del algoritmo NDT para las situaciones simuladas.

Ensayo	Media de iteraciones	RMSE de asociación medio (m)	RMSE translación (m)	RMSE rotación (°)
<i>Entorno sin perturbación simulada (referencia)</i>	1,51	0,2481	0,0138	0,2097
Limitación alcance 10 m	1,57	0,239	0,014	0,215
Limitación alcance 15 m	1,49	0,238	0,013	0,206
<i>Downsampling</i> 65%	1,50	0,252	0,014	0,207
Limitación sector circular fijo	1,53	0,251	0,014	0,211
Limitación sector circular random	1,51	0,263	0,014	0,209

Se observa cómo los valores obtenidos de los indicadores en cada caso son similares al obtenido en el caso de referencia (primera fila). Se destaca que incluso, en el caso de delimitar el alcance a 15 metros, se mejora ligeramente los resultados, ya que este algoritmo en concreto mejora su rendimiento y precisión cuando se limita la cantidad de puntos del entorno. Sin embargo, si se analiza el error de asociación en función del tiempo, es posible identificar un aumento de este cuando existe alguna perturbación de la información. En concreto, la figura 3.69 muestra el comportamiento del algoritmo en la situación “Limitación sector circular random”, donde cada vez que la oclusión cambia de posición, da lugar a un aumento considerable en el error de asociación.

Figura 3.69: Error de asociación en función del tiempo en la situación con una oclusión variable. Las líneas discontinuas indican el tramo de limitación simulada.

3.4.6.3. Aplicación de la metodología para etapa de asociación asistida por modelo CNN

A partir de las anteriores evaluaciones sobre el comportamiento tanto del modelo de CNN, como de los algoritmos convencionales de asociación, es posible aplicar la metodología propuesta. El modelo de CNN, por su parte, ha verificado la hipótesis inicial y no obtiene una fuerte dependencia geométrica del entorno, por lo que es posible utilizar la estimación de este para llevar a cabo la fase 1 y 2 del método propuesto.

Método de asociación mediante ICP:

Las tablas 3.15 y 3.16 muestran los resultados de los mismos indicadores anteriormente utilizados, aplicando, en este caso, la metodología completa para lograr el *matching* robusto de las nubes de puntos sobre el algoritmo ICP. Destacado en negrita, se puede observar los valores obtenidos mediante el método propuesto junto al valor obtenido para el mismo caso cuando no es aplicado.

Tabla 3.15: Comparativa de rendimiento y error de asociación del ICP para las situaciones simuladas.

Ensayo	Media de iteraciones		RMSE de asociación medio (m)	
	Sin asistencia	Con asistencia	Sin asistencia	Con asistencia
Limitación alcance 10 m	9,08	3,03	1,807	0,241
Limitación alcance 15 m	3,03	2,94	0,241	0,241
Downsampling 65 %	3,08	3,06	0,253	0,253
Limitación sector circular fijo	3,11	3,06	0,253	0,253
Limitación sector circular random	4,06	3,13	Inf.	0,264

Tabla 3.16: Comparativa de error de translación y rotación del ICP para las situaciones simuladas.

Ensayo	RMSE translación (m)		RMSE rotación (°)	
	Sin asistencia	Con asistencia	Sin asistencia	Con asistencia
Limitación alcance 10 m	5,250	0,024	63,256	0,388
Limitación alcance 15 m	0,044	0,022	0,537	0,297
Downsampling 65 %	0,040	0,023	0,394	0,299
Limitación sector circular fijo	0,044	0,025	0,435	0,309
Limitación sector circular random	Inf.	0,025	12,067	0,316

Cabe destacar que las dos situaciones descritas anteriormente donde el algoritmo ICP no converge a soluciones coherentes (“Limitación alcance 10 m” y “Limitación sector circular random”) en este caso, mediante la integración del método planteado, se logra una solución precisa y con el mismo rendimiento que en el resto de las situaciones. Por otro lado, aunque no es objeto del método planteado, se observa que en el resto situaciones donde el ICP conseguía alcanzar un resultado coherente, la aplicación del método mejora el rendimiento (menor número de iteraciones medias) y la precisión, tanto en asociación como en la reconstrucción. La figura 3.70 ilustra esta última en la reconstrucción de la trayectoria para la situación donde se limita el alcance a 15 metros.

Figura 3.70: Reconstrucción de la trayectoria en la simulación de la limitación de alcance a 15 m con ICP. En azul sin aplicación del método propuesto. En naranja, aplicando el método.

Método de asociación mediante NDT:

De igual modo, se procede con el estudio de integrar la estimación del modelo de CNN sobre la asociación llevaba a cabo por el algoritmo NDT. En este caso, los resultados obtenidos previamente eran bastante robustos y similares al entorno que no presentaba perturbaciones en la información. Por tanto, es esperable que no se produzcan desviaciones significativas aplicando la metodología, tal y como se observa en las tablas 3.17 y 3.18.

Tabla 3.17: Comparativa de rendimiento y error de asociación del algoritmo NDT para las situaciones simuladas.

Ensayo	Media de iteraciones		RMSE de asociación medio (m)	
	Sin asistencia	Con asistencia	Sin asistencia	Con asistencia
Limitación alcance 10 m	1,57	1,89	0,239	0,239
Limitación alcance 15 m	1,49	1,88	0,238	0,238
Downsampling 65 %	1,5	2,02	0,253	0,253
Limitación sector circular fijo	1,53	1,99	0,251	0,251
Limitación sector circular random	1,51	2,03	0,263	0,263

Tabla 3.18: Comparativa de error de translación y rotación del algoritmo NDT para las situaciones simuladas.

Ensayo	RMSE translación (m)		RMSE rotación (°)	
	Sin asistencia	Con asistencia	Sin asistencia	Con asistencia
Limitación alcance 10 m	0,014	0,015	0,215	0,213
Limitación alcance 15 m	0,013	0,014	0,206	0,202
Downsampling 65 %	0,014	0,014	0,207	0,203
Limitación sector circular fijo	0,014	0,015	0,211	0,206
Limitación sector circular random	0,014	0,015	0,209	0,221

Como se indicaba, no se persigue mejorar la precisión del sistema de SLAM, ya que esta dependerá de implementación completa de cada sistema de SLAM. Este hecho se ilustra con los resultados obtenidos en con el algoritmo NDT, que partían de un buen resultado en las situaciones simuladas y, cuando se implementa el método, se obtienen resultados muy similares. La figura 3.71 pone de manifiesto este comportamiento del algoritmo en la situación donde se limita el alcance a 15 metros.

Figura 3.71: Reconstrucción de la trayectoria en la simulación de la limitación de alcance a 15 m con NDT. En azul sin aplicación del método propuesto. En naranja, aplicando el método.

Sin embargo, en este caso, la metodología propuesta puede proporcionar una primera aproximación del movimiento relativo al inicio de la etapa de asociación que puede ser utilizada en el caso de que no se garantice el tiempo de ejecución, paralelizando procesos.

Finalmente, tras la etapa de *matching*, donde se centra la solución planteada, se designa la estimación del movimiento relativo del vehículo, el cual será integrado al resto de posiciones previas, actualizando la posición global en el mapa y posibilitando la reconstrucción de éste (figura 3.72).

Figura 3.72: Ejemplo de reconstrucción del mapa sobre la trayectoria de evaluación.

3.4.7. Conclusiones finales

En este apartado se ha abordado la problemática principal respecto al rendimiento y la precisión de muchas de las soluciones de SLAM basadas en LiDAR 3D. Los métodos convencionales basados en láser encuentran dificultades durante la etapa de asociación para lograr un emparejamiento entre nubes de puntos robusto. El problema puede ser debido al entorno por donde se circula, si este no contiene cierta cantidad de características relevantes o bien, estas no son fácilmente extraíbles al no estar bien definidas. Otros problemas son intrínsecos a la utilización del LiDAR, como puede ser la dispersión entre puntos u oclusiones, dando lugar a una mala asociación igualmente. Sin embargo, estas técnicas convencionales han demostrado lograr una mejor precisión cuando no existen dichas limitaciones, ya que el problema a resolver está analíticamente definido.

Por tanto, se ha planteado una metodología donde se complementen, por un lado, las ventajas de las técnicas clásicas y, por otro lado, las ventajas que ofrece un enfoque basado en modelos de *Deep Learning*. Estos últimos no tienen una fuerte dependencia geométrica y, por tanto, pueden mitigar las carencias del enfoque convencional en este sentido. Contrariamente a los métodos clásicos, los modelos de *Deep Learning* no consiguen obtener una precisión global, pero sí aseguran una buena consistencia local, por lo que pueden ser utilizados como primera aproximación durante la etapa de asociación de datos mediante un algoritmo clásico.

El objetivo principal es conseguir una asociación robusta en algoritmos de asociación de datos con optimización local, ya que estos requieren de una aproximación inicial. Además, la solución planteada es extrapolable a cualquier sistema de SLAM que utilice estos métodos en su etapa de *matching*. Para ello, se ha descrito la metodología planteada y se han realizado las verificaciones y ensayos sobre dos algoritmos convencionales: ICP y NDT.

En primer lugar, se ha validado que la estimación aportada por un modelo basado en CNN puede lograr un comportamiento similar en ambos algoritmos de asociación. Se concluye que estos algoritmos son susceptibles de la realimentación que obtengan, por lo que, si la red es capaz de asegurar un buen condicionamiento de la primera aproximación, los algoritmos mejorarán su salida, filtrando posibles valores que no sean coherentes con el valor esperado.

Una vez realizada la anterior validación, es necesario conocer si la estimación de la red es menos susceptible que los métodos analíticos en cuanto a las limitaciones que pueda ofrecer el entorno y así, corroborar la hipótesis inicial y poder aplicar el método propuesto. Para ello, se han simulado una serie de situaciones donde se pone en compromiso la calidad de la información del entorno. Se han obtenido resultados muy destacables, sobre todo, en algunas situaciones donde, en concreto el algoritmo ICP, mostraba inestabilidad en los resultados. Mediante la aplicación del método, se consigue lograr valores similares de precisión y rendimiento que obtendrían los algoritmos convencionales de no sufrir de estas limitaciones.

Aunque se ha comprobado que el modelo de *Deep Learning* generaliza de forma correcta ante las situaciones simuladas, sí existe una desviación en la salida. Esta desviación es mayor en la estimación de la velocidad, mientras que la estimación de la variación de guiñada mantiene su error de forma más contenida, por lo que no implica mayor problema en la precisión alcanzada. Sin embargo, para mejorar la generalización del modelo, podría plantearse nuevas técnicas de *Data Augmentation* para generar datos de entrenamiento que simulen estas situaciones donde se limita la información del entorno.

Capítulo 4.

Aportaciones, conclusiones y futuros desarrollos

En este capítulo se exponen las principales conclusiones y aportaciones que se pueden extraer del trabajo realizado en la presente Tesis Doctoral.

4.1. Aportaciones y conclusiones

Las conclusiones de la Tesis se pueden estructurar en varios bloques:

Uso del LiDAR:

- Seguridad:

El desarrollo de la tecnología LiDAR ha posibilitado en los últimos años el desarrollo de nuevos algoritmos para la percepción del entorno del vehículo autónomo. Nuevos sensores con mayores capacidades, facilidad de integración en el vehículo y abaratamiento de costes, favorecen su implementación tanto en vehículos de alto grado de automatización, como en vehículos prototipo totalmente autónomos.

La introducción de estos sensores láser en el campo de la percepción complementa las ventajas de otros sensores ya establecidos en este campo, como visión artificial, stereo-vision o radar. Gran parte de su potencial se fundamenta en los algoritmos que utilizan esta tecnología como base principal de funcionamiento.

Estos algoritmos basados en LiDAR 3D para la percepción del entorno son, cada día más, un pilar fundamental para garantizar la seguridad de la navegación autónoma terrestre, ya que la información precisa y robusta del entorno, permite lograr un mejor entendimiento de este. Estos desarrollos han demostrado un correcto funcionamiento incluso en zonas complejas, *off-road* o con menor grado de infraestructura, como se ha tratado a lo largo de la tesis.

- Versatilidad:

Además de la función más evidente y establecida de los sensores láser rotativos como es la identificación de obstáculos en el entorno, estos sensores han demostrado un gran nivel de versatilidad siendo implementados con otros objetivos.

En este sentido, se establecen dos grandes campos de aplicación: caracterización del entorno y posicionamiento.

En el primer caso, estos sensores permiten obtener capacidades adicionales a la identificación y evitación de obstáculos (apartado 2.2), como es la caracterización horizontal de la calzada, al extraer las líneas viales o los límites de la zona transitable (apartado 2.3).

En el segundo caso, el LiDAR 3D demuestra su aplicabilidad como herramienta para el posicionamiento, ya sea apoyando una navegación autónoma a partir de la caracterización de un elemento del entorno (apartado 3.1) o como sensor principal de una solución de SLAM (apartado 3.2). Incluso a partir del tratamiento de nubes de puntos, se posibilita desarrollar modelos de *Deep Learning*, que obtienen una buena estimación de la odometría del vehículo (apartado 3.3). A su vez, esta estimación de la odometría consigue servir de asistencia a otros algoritmos convencionales de posicionamiento, para mejorar su rendimiento o precisión (apartado 3.4).

Por todo ello, se comprueba la versatilidad y capacidad que se puede obtener a partir del uso de estos sensores.

- Rendimiento:

A pesar de que los algoritmos basados en esta tecnología láser pueden sufrir de una alta carga computacional, se han conseguido diseñar diversos algoritmos en esta tesis con la premisa de mantener la carga computacional baja, obteniendo resultados significativos en cada caso.

Así, a partir de la implementación de un solo LiDAR sin altas prestaciones, se han diseñado algoritmos, tanto para la caracterización del entorno como para el posicionamiento, que consiguen lograr robustez y precisión en los resultados.

Por otro lado, los desarrollos basados en modelos de *Machine Learning*, permiten utilizar sensores LiDAR más sofisticados (en concreto se han utilizado de 64 capas) sin poner en compromiso el aumento de carga computacional. Una vez que el modelo de *Deep Learning* es entrenado, el tiempo de computación de la aplicación se reduce drásticamente, quedando reducido al tiempo requerido para el tratamiento previo de datos y, de la misma manera, el tratamiento de las salidas del modelo.

Principales aportaciones:

- Calibración extrínseca automática:

Se ha propuesto un método para llevar a cabo la calibración extrínseca entre sensores LiDAR 3D que permite de forma automática conocer la ubicación relativa entre sensores. La principal característica destacable de este método es que no requiere necesariamente de elementos externos de calibración. Aporta ventajas significativas, sobre todo, durante las pruebas de diferentes *set-ups* de los sensores sobre el vehículo, simplificando el proceso de calibración.

El método realiza una fusión de bajo nivel minimizando el error de asociación entre las nubes de puntos. Una vez calculada la asociación, el algoritmo evalúa el error cometido para establecer si procede realizar una nueva iteración. Una vez el error cometido es inferior a un umbral designado, las nubes de puntos quedan referenciadas a un mismo sistema de referencia. Durante los ensayos se ha comprobado los límites operativos del algoritmo mediante distintas orientaciones entre láseres, obteniendo un funcionamiento correcto para desviaciones de hasta 30° entre ellos. Las desviaciones obtenidas en el resultado son asumibles cuando se realiza una fusión de alto nivel.

- Detección de obstáculos específica para entornos *off-road*:

Se ha diseñado un algoritmo enfocado a la identificación de obstáculos negativos en terrenos *off-road* con un posicionamiento y orientación del LiDAR diferente al habitual, con el objetivo de obtener la mejor definición del terreno. La principal ventaja del método es su baja carga computacional, garantizando un funcionamiento a 10 Hz, y a pesar de su enfoque para obstáculos negativos, la solución es operativa igualmente para obstáculos positivos.

Su implementación en vehículos reales ha posibilitado validar el algoritmo en situaciones *off-road* donde se comprometía la robustez del método por la orografía irregular. Así, se destaca un bajo índice de falsos positivos y una precisión del 94,97%, entendiendo esta como el porcentaje de acierto sobre las identificaciones. La exactitud del método es de un 81,30%, donde se tienen en cuenta los resultados verdaderos ($VP + VN$). Sin embargo, se han tenido en cuenta para esta métrica cualquier identificación (total o parcial) que se encontrara dentro del ROI. Una validación más exhaustiva de este indicador debería de tener en cuenta la distancia de identificación.

- Extracción de las líneas de carril para la caracterización de la carretera:

A pesar de no existir una amplia literatura al respecto, se ha abordado la caracterización transversal de la calzada extrayendo las líneas delimitadoras de carril, estimando el ancho y número de carriles que componen el total de la vía. Por tanto, este método reporta unas ventajas considerables no solo como herramienta de caracterización del entorno, sino también para el posicionamiento a nivel de carril.

A través de distintos ensayos en entornos reales de carretera, se ha validado la operatividad del algoritmo. Sin embargo, existen dificultades en la identificación cuando se trata de escenarios que presentan alguna singularidad como cruces o incorporaciones. En estos casos, la precisión del método puede disminuir, pero si se incluye la información de un mapa electrónico, el algoritmo es capaz de contrastar esta información y mejorar el resultado.

Por otro lado, cabe destacar que el método desarrollado para el cálculo de los límites en entornos urbanos es aplicable a zonas todoterreno. En este sentido, se han llevado a cabo ensayos en caminos *off-road* donde el método es capaz de calcular la zona transitable, incluso ante bifurcaciones o delimitaciones poco definidas.

- Modelo de CNN basado en LiDAR 3D para la estimación de la odometría:

Aprovechando los avances en Deep Learning en los últimos años, se ha diseñado un modelo de CNN para inferir la odometría del vehículo, garantizando una consistencia local. Para resolver este problema de regresión se ha desarrollado una solución completa, desde el pretratamiento de nubes de puntos, al diseño de una arquitectura de red basada en CNN que infiere el movimiento relativo del vehículo, aportando un enfoque diferente a un problema estrictamente geométrico.

En cuanto a la adaptación de las nubes de puntos para su implementación en modelos de CNN, se han barajado tres configuraciones de imágenes-dato diferentes, siendo los Mapas de Profundidad, los que mejor resultado han logrado.

Tras un proceso cíclico para el diseño de la topología de red, se logra definir una arquitectura basada en CNN diferenciando dos bloques: bloque de extracción de características y bloque de estimación de la señal. Los resultados obtenidos indican que el modelo logra alcanzar una buena estimación respecto a la velocidad y la variación de guiñada, garantizando localmente el posicionamiento. Sin embargo, se hace patente que estos modelos no consiguen una consistencia global suficiente como para garantizar su posicionamiento en un mapa debido a la integración del error. Por ello, se plantea la idea de utilizar estos modelos como asistencia a métodos de posicionamiento global basado en sensores de percepción. Esta idea es desarrollada en el apartado 3.4.

- Metodología para asociación de nubes de puntos asistida por modelo de CNN:

Tras los resultados obtenidos por el modelo de CNN en el apartado 3.3 y considerando que el problema del posicionamiento global mediante SLAM está definido analíticamente, se ha desarrollado una metodología que combine las ventajas de ambos enfoques para lograr una asociación de nubes de puntos robusta.

Se ha identificado que ciertos problemas inducidos por la información del entorno pueden dar lugar a un mal condicionamiento de los métodos convencionales de asociación y que, a su vez, el modelo planteado de CNN, no se ve afectado en la misma medida. Por

un lado, se cumple con la hipótesis de que estos modelos son menos dependientes de la geometría del entorno que los métodos analíticos. Y por otro lado, se puede emplear la estimación obtenida por modelos CNN como asistencia a la etapa de asociación de datos cuando los métodos convencionales encuentran dificultades.

Para verificar el método planteado, se han llevado a cabo varios ensayos donde se evaluaba la dependencia sobre la información del entorno, tanto del modelo CNN como de los métodos analíticos. En este sentido, tras simular varias perturbaciones en las nubes de puntos, los métodos de asociación convencionales encontraban alguna dificultad al converger, mientras que el modelo CNN mantenía su respuesta dentro de los valores esperables. Esto posibilita utilizar la estimación de la red con dos objetivos. Por un lado, una verificación de que la primera aproximación que se suministre al método de asociación está dentro de unos órdenes de magnitud esperables. Por otro lado, si el propio método de asociación encuentra dificultad para converger a una solución correcta, se puede utilizar la estimación de la red como aproximación en una nueva ejecución del método de asociación.

4.2. Futuros desarrollos

A partir de las conclusiones y contribuciones de la tesis, quedan varias líneas de investigación abiertas por donde se podría avanzar en los sistemas de percepción basados en LiDAR 3D. Algunos de los aspectos donde, a través de un análisis en profundidad, se podrían ampliar o mejorar los resultados obtenidos son:

Mejoras de las prestaciones generales de los algoritmos:

- Ciertos algoritmos propuestos se han desarrollado bajo la premisa de mantener una carga computacional baja, ya que esta se considera uno de los aspectos cruciales en el vehículo autónomo. Sin embargo, para buscar un equilibrio entre rendimiento y precisión en el resultado, se podrían utilizar funciones genéricas ya desarrolladas. Aunque estas herramientas no estén optimizadas para la aplicación en concreto, puede aportar una mejora en ciertas etapas dentro de la estructura global del algoritmo.
- Los desarrollos planteados a lo largo de los capítulos se centran en el estudio de las capacidades que puede aportar un solo sensor LiDAR. Una vez conocida la eficiencia del funcionamiento con un solo sensor, se podría afrontar la integración de una fusión sensorial. Este enfoque combina las ventajas, en muchos casos complementarias, de diferentes sensores consiguiendo un sistema redundante. La gran variabilidad de escenarios por donde puede circular el vehículo autónomo pone de manifiesto la importancia de sistemas robustos y generalizables.

Alternativas de topologías de redes neuronales:

- El diseño de las arquitecturas de red empleadas en los modelos de CNN requiere un proceso cíclico para lograr el mejor ajuste. Además, el campo de investigación está en continuo cambio, lo que da lugar a una complejidad añadida. A pesar de haberse optimizado el entorno de desarrollo para el diseño de arquitecturas, sería interesante desarrollar una arquitectura basada en redes recurrentes para obtener la estimación de la odometría. Tal y como señalan algunos trabajos previos indicados en la tesis, las redes recurrentes pueden llegar a aportar buenos resultados cuando existe una dependencia temporal entre los datos. En este sentido, se ha llegado a realizar alguna prueba preliminar sin mejorar los resultados descritos en la tesis, sin embargo, un análisis más exhaustivo podría llegar a ofrecer resultados destacables.
- De igual modo, si las redes recurrentes pueden aportar un buen funcionamiento, sería interesante diseñar una arquitectura de red que combine la aportación de estas junto con la extracción de características de las redes de convolución. Un enfoque parecido, pero utilizando visión artificial, es el que se desarrolla en S. Wang et al. (2017). Basándose en el desarrollo del apartado 3.3 de esta tesis, sería fácilmente adaptable al uso de nubes de puntos.

Alternativas de representación del entorno como entrada a modelos de redes neuronales:

- Al igual que el proceso de diseño cíclico de una topología de red de *Deep Learning* es determinante durante el desarrollo, el pretratamiento de datos cobra una gran importancia para lograr una buena optimización del modelo. Así, se han presentado varias configuraciones de tratamiento de nubes de puntos para utilizarlas como entrada a redes de convolución. Sin embargo, se plantean otras configuraciones que pueden reportar un buen funcionamiento igualmente como, por ejemplo, recurriendo a la fusión sensorial, integrar la imagen RGB de cámaras junto con los mapas de profundidad desarrollados en la tesis. A pesar de aumentar la complejidad del tratamiento de datos, la información del color puede aportar ventajas al inferir la odometría del vehículo.
- Asimismo, se puede estudiar la integración entre distintos algoritmos presentados y completar, de esta manera, el sistema de percepción dentro de la estructura de un vehículo autónomo. En este sentido, la información del ancho de carril, el error lateral dentro del mismo o elementos de la carretera, se pueden codificar e incluir en los mapas utilizados en los modelos CNN. Esta capa adicional de información puede lograr una mejora de la estimación de la odometría al contener información relevante del entorno.

4.3. Publicaciones

Los desarrollos expuestos en la Tesis Doctoral han dado lugar a diversas publicaciones en revistas de alto índice de impacto, participación en capítulos de libro y divulgaciones en congresos, tanto nacionales como internacionales.

Publicaciones en revistas indexadas en el JCR:

“Improving the lane reference detection for autonomous road vehicle control”

Autores: Jimenez, F., Clavijo, M., Naranjo, J. E., Gómez, Ó.

Publicación: Journal of Sensors

Edita: Hindawi Publishing Corporation

Referencia: Año 2016, ID 9497524, 13 páginas

ISSN: 1687-725X

Índice de impacto: 1,704 (año 2016)

Cuartil: Q2

Área: 126 de 262 en Engineering, Electrical & Electronic Science

26 de 58 en Instruments & Instrumentation

“Reference element guidance of autonomous vehicles using 3D laser scanner”

Autores: Naranjo, J. E., Jiménez, F., Clavijo, M., Gómez, Ó.

Publicación: Dyna Ingeniería e Industria

Edita: Federación de Asociaciones de Ingenieros Industriales de España

Referencia: Año 2017, Volumen 92, nº 3, pp. 327-332

ISSN: 0012-7361

Índice de impacto: 0,520 (año 2017)

Cuartil: Q4

Área: 76 de 86 en Engineering, Multidisciplinary

“Modelling the human lane-change acceptance behaviour through Multilayer Perceptrons and Convolutional Neural Networks”

Autores: Díaz, A., Clavijo, M., Jiménez, F., Talavera, E., Serradilla, F.

Publicación: Transportation Research Part F: Traffic Psychology and Behaviour

Edita: Elsevier

Referencia: Año 2018, Volumen 56, pp. 134–148

ISSN: 1369-8478

Índice de impacto: 2,360 (año 2018)

Cuartil: Q2

Área: 30 de 82 en Psychology, Applied

19 de 36 en Transportation

“Accurate and detailed transversal road section characteristics extraction using laser scanner”

Autores: Jiménez, F., Clavijo, M., Castellanos, F., Álvarez, C.

Publicación: Applied Sciences

Edita: MDPI AG

Referencia: Año 2018, Volumen 8, pp. 724

ISSN: 2076-3417

Índice de impacto: 2,217 (año 2018)

Cuartil: Q2

Área: 89 de 172 en Chemistry, Multidisciplinary

151 de 293 en Materials Science, Multidisciplinary

67 de 148 en Physics, Applied

Capítulos de libro:

”Intelligent Road Vehicles: Enabling Technologies and Future Developments”

Editorial: Elsevier

ISBN: 978-0-12-812800-8

Editado por: Felipe Jiménez

Fecha: 2017

Capítulos con participación:

- Environmental Perception for Intelligent Vehicles. pp.23-101.
- Positioning and Digital Maps. pp.141-174.
- Automated Driving. pp.275-342.

Comunicaciones en congresos:

(10 comunicaciones en congresos internacionales y 4 nacionales)

“Supervision system for reversible lanes based on autonomous vehicles”

Autores: Clavijo, M., Jiménez, F., Naranjo, J. E., Gómez, Ó.

Congreso: 11nd European Congress and Exhibition on Intelligent Transport Systems and Services.

Lugar y fecha: Glasgow (Escocia), 6-9 junio 2016 Organiza: ERTICO. ITS Congress Association

Carácter: Internacional

Participación: Coautor y ponente

“Autonomous vehicle for surveillance missions in off-road environment”

Autores: Naranjo, J. E., Clavijo, M., Jiménez, F., Gómez, Ó., Rivera, J. L., Anguita, M.

Congreso: 2016 IEEE Intelligent Vehicles Symposium IV16

Lugar y fecha: Goteborg (Suecia), 19-22 junio 2016 Organiza: IEEE

Carácter: Internacional

Participación: Coautor

“Guiado de un vehículo autónomo mediante la detección de los límites del carril usando un escáner láser”

Autores: Jiménez, F., Clavijo, M., Naranjo, J. E., Gómez, O.

Congreso: XXI Congreso Nacional de Ingeniería Mecánica Lugar y fecha: Elche, 9 – 11 de noviembre de 2016 Organiza: Asociación Nacional de Ingeniería Mecánica y Universidad de Elche

Carácter: Nacional

Participación: Coautor

ISSN/ISBN: 978-84-16024-37-7

“Vehículos terrestres autónomos para misiones de vigilancia en entornos todoterreno”

Autores: Naranjo, J. E., Clavijo, M., Jiménez, F., Gómez, O., Rivera, J. L., Anguita, M.

Congreso: IV Congreso Nacional de I+D en Defensa y Seguridad

Lugar y fecha: Murcia, 16 – 18 de noviembre de 2016 Organiza: Subdirección General de Planificación, Tecnología e Innovación de la DGAM

Carácter: Nacional

Participación: Coautor

“Aplicaciones avanzadas de detección mediante LIDAR para Conducción Autónoma o Teleguiada”

Autores: Jiménez, F., Clavijo, M., Naranjo, J. E., Serradilla, F.

Congreso: XVII Congreso Español de Sistemas

Inteligentes de Transporte

Lugar y fecha: Madrid, 25-27 abril 2017

Organiza: ITS España

Carácter: Nacional

Participación: Coautor

“‘Where in the lane’ positioning for autonomous vehicles”

Autores: Clavijo, M., Jimenez, F., Castellanos, F., Labari, I.

Congreso: 12nd European Congress and Exhibition on Intelligent Transport Systems and Services.

Lugar y fecha: Estrasburgo (Francia), 19-22 junio 2017

Organiza: ERTICO. ITS Congress Association

Carácter: Internacional

Participación: Coautor

“Off-road autonomous vehicle for military missions”

Autores: Jimenez, F., Naranjo, J. E., Clavijo, M., Gómez, O., Anaya, J. J., Rivera, J. L., Anguita, M.

Congreso: 12nd European Congress and Exhibition on Intelligent Transport Systems and Services.

Lugar y fecha: Estrasburgo (Francia), 19-22 junio 2017

Organiza: ERTICO. ITS Congress Association

Carácter: Internacional

Participación: Coautor

“Application of Deep Learning to Route Odometry Estimation from LiDAR Data”

Autores: Clavijo, M., Serradilla, F., Naranjo, J. E., Jiménez, F., Díaz, A.

Congreso: VEHICULAR 2017, The Sixth International Conference on Advances in Vehicular Systems, Technologies and Applications

Lugar y fecha: Niza (Francia), 22-23 julio 2017

Organiza: IARIA, International Academy, Research, and Industry Association

Carácter: Internacional

Participación: Coautor y ponente

Reconocimiento a mejor artículo

“Toma de decisiones de un vehículo autónomo de un entorno complejo soportado por sistema de fusión sensorial”

Autores: Jiménez, F., Martínez, C., Clavijo, M., Romero, D.

Congreso: 13 Congreso Iberoamericano de Ingeniería Mecánica

Lugar y fecha: Lisboa (Portugal), 23 – 26 de octubre de 2017

Organiza: Universidad Nova de Lisboa

Carácter: Internacional

Participación: Coautor

“Aplicación de Deep Learning para la estimación de odometría de un vehículo empleando un escáner láser”

Autores: Clavijo, M., Serradilla, F., Jiménez, F., Naranjo, J. E., Díaz, A.

Congreso: 13 Congreso Iberoamericano de Ingeniería Mecánica

Lugar y fecha: Lisboa (Portugal), 23 – 26 de octubre de 2017

Organiza: Universidad Nova de Lisboa

Carácter: Internacional

Participación: Coautor

“Deep Learning Application for 3D LiDAR Odometry Estimation in Autonomous Vehicles”

Autores: Clavijo, M., Díaz, A., Serradilla, F., Jiménez, F., Naranjo, J. E.

Congreso: 7th European Transport Research Conference. TRA 2018

Lugar y fecha: Viena (Austria), 16-19 abril 2018

Organiza: Transport Research Arena (TRA)

Carácter: Internacional

Participación: Coautor y ponente

“Algoritmos para la detección de obstáculos en un entorno todoterreno mediante LiDAR”

Autores: Clavijo, M., Jiménez, F., Díaz, A., Gorak, A.

Congreso: XXII Congreso Nacional de Ingeniería Mecánica

Lugar y fecha: Madrid, 19 – 21 de Septiembre de 2018

Organiza: Asociación Nacional de Ingeniería Mecánica y UNED

Carácter: Nacional

ISSN/ISBN: 1137-2729

“Interface design for an assistance system focused on high attentional load situations”

Autores: Sofía Sánchez-Mateo, Miguel Clavijo, Alberto Díaz-Álvarez, Felipe Jiménez

Congreso: 25th ITS World Congress

Lugar y fecha: Copenhagen (Denmark), 17-21 September 2018

Carácter: Internacional

Participación: Coautor

“LiDAR-based SLAM algorithm for indoor scenarios”

Autores: Jiménez, F., Clavijo, M., Juana, J.

Congreso: VEHICULAR 2018, The Seventh International Conference on Advances in Vehicular Systems, Technologies and Applications

Lugar y fecha: Venecia (Italia), 24-28 Junio 2018

Organiza: IARIA, International Academy, Research, and Industry Association

Carácter: Internacional

Contributions, conclusions and future work

This chapter presents the main conclusions and contributions that can be drawn from the work carried out in this Doctoral Thesis.

Contributions and conclusions

The thesis conclusions can be structured in several categories:

Use of 3D LiDAR:

- Security:

The development of LiDAR technology has made possible in recent years the development of new algorithms for the environment perception of the autonomous vehicle. New sensors with greater capabilities, ease of integration into the vehicle and cheaper costs, favor their implementation in both highly automated vehicles and fully autonomous prototype vehicles.

The introduction of these laser sensors in the field of perception complements the advantages of other sensors already established in the field, such as artificial vision, stereo-vision or radar. Much of its potential is based on algorithms that use this technology as the main operating basis.

These algorithms based on 3D LiDAR for the environment perception constitute, increasingly, a fundamental pillar in order to guarantee the security of the autonomous ground navigation, since the accurate and robust information of the environment allows them to achieve a better understanding of the same. These developments have demonstrated correct operation even in complex areas, off-road or with a lower degree of infrastructure, as explored throughout the thesis.

- Versatility:

In addition to the most evident and established function of laser scanner sensors, namely the obstacles detection in the environment, these sensors have demonstrated a high level of versatility when implemented with other objectives.

In this regard, there are two main fields of application: characterization of the environment and positioning.

In the first case, these sensors have additional capabilities to the obstacles identification and avoidance (section 2.2), such as the horizontal characterization of the road, by extracting the lane lines or the boundaries of the passable area (section 2.3).

In the second case, 3D LiDAR demonstrates its applicability as a tool for positioning. Either supporting autonomous navigation from the characterization of a reference element of the environment (section 3.1) or as the main sensor of a SLAM solution (section 3.2). Moreover, from the point cloud processing, it is possible to develop Deep Learning models, which obtain a good vehicle odometry estimation (section 3.3). In turn, this odometry estimation can be used for assistance to other conventional positioning algorithms, so to improve its performance or precision (section 3.4).

Therefore, the versatility and capacity that can be obtained from the use of these sensors is verified.

- Performance:

Despite the fact that algorithms based on this laser technology can suffer from a high computational load, various algorithms have been designed in this thesis while keeping the computational cost low, obtaining significant results in each case.

Thus, from the implementation of a single LiDAR without high capacities, algorithms have been designed, both for the characterization of the environment and for positioning, which achieve robustness and precision in the results.

Likewise, the developments based on Machine Learning models allow the use of more sophisticated LiDAR sensors (specifically 64-layer ones have been used) without compromising the increased computational cost. Once the Deep Learning model is trained, the application runtime is drastically reduced to the time required for the pre-treatment of data and, in the same way, the treatment of the model's outputs.

Main contributions:

- Automatic extrinsic calibration:

The proposed method to carry out an extrinsic calibration between 3D LiDAR, allows to detect the relative location between sensors automatically. The main remarkable feature of this method is that it does not necessarily require external calibration elements. It provides significant advantages, especially during tests of different sensor set-ups on the vehicle, thereby simplifying the calibration process.

The method performs a low-level fusion minimizing the matching error between the point clouds. Once the association has been calculated, the algorithm evaluates the error obtained to establish whether to proceed with a new iteration or not. Once the error is below a designated threshold, the point clouds are referenced to the same reference system. During the tests, the operating limits of the algorithm were verified by means of different orientations between lasers, obtaining correct operation for deviations of up to 30° between them. The deviations obtained in the result are acceptable when performing a high-level fusion.

- Obstacle detection for specific off-road environments:

An algorithm focused on identifying negative obstacles on off-road scenarios has been designed with a different LiDAR positioning and orientation than usual, with the aim of obtaining the best ground definition. The main advantage of the method is its low computational cost, which guarantees operation at 10 Hz, and despite its focus on negative obstacles, the solution is equally effective on positive obstacles.

Its implementation in real vehicles has made it possible to validate the algorithm in off-road areas, where the robustness of the method was compromised by irregular orography. Thus, a low false positive rate and an accuracy of 94.97% stand out, understanding this as the percentage of valid identifications. The accuracy of the method is 81.30%, where the true results are taken into account (positives and negatives). However, any identification (total or partial) found within the ROI has been taken into account for this metric. A more exhaustive validation of this indicator should take into account the identification distance.

- Extraction of lane lines to characterize the road:

Despite the lack of an extensive literature on the subject, the horizontal characterization of the road has been addressed by extracting the lane lines, estimating the width and number of lanes that make up the whole road. Therefore, this method provides considerable advantages not only as a tool for characterizing the environment, but also for positioning at the lane-level.

The algorithm's operability has been validated through different tests in real road en-

vironments. However, there are difficulties in identification when it comes to scenarios that present some singularities such as crossings or merging lanes. In these cases, the accuracy of the method may decrease, but if the information from a digital map is included, the algorithm is able to compare this information and improve the result.

On the other hand, it should be noted that the method developed for the boundaries detection in urban environments is applicable to off-road areas. In this sense, tests have been carried out on off-road paths where the method is capable of calculating the passable area, even in the case of slightly defined bifurcations or limits.

- 3D LiDAR-based CNN model for odometry estimation:

Taking advantage of advances in Deep Learning in recent years, a CNN model has been designed to infer vehicle odometry, ensuring local consistency. To solve this regression problem, a complete solution has been developed, from the preprocessing of point clouds cloud to the design of a CNN-based network architecture that infers the relative movement of the vehicle, providing a different approach to a strictly geometric problem.

Regarding the adaptation of the point clouds for their implementation into CNN models, three different image-data configurations have been considered, where Depth Maps resulted in the best results.

After a cyclical process for the design of the network topology, it is possible to define an architecture based on CNN by differentiating two blocks: feature extraction and signal estimation block. The results obtained indicate that the model achieves a good estimate regarding speed and yaw rate, ensuring local positioning. However, it is clear that these models do not achieve sufficient global consistency to ensure positioning on a map due to the integration of the error. For this reason, the idea of using these models to assist global positioning methods based on perception sensors is proposed. This idea is developed in section 3.4.

- CNN model-assisted Point Cloud registration methodology:

Following the results obtained by the CNN model in section 3.3 and considering that the problem of global positioning using SLAM is analytically defined, a methodology has been developed that combines the advantages of both approaches to achieve a robust point cloud registration.

It has been identified that certain problems induced by environmental information can lead to poor conditioning of conventional point cloud registration methods, where the proposed CNN model is not affected to the same extent. On the one hand, the hypothesis that these models are less dependent on the geometry of the environment than analytical methods is fulfilled. On the other hand, the estimation obtained by CNN models can be used to assist the data association step when conventional methods encounter difficulties.

To verify the proposed method, several tests have been carried out evaluating the dependence on information from the environment, both from the CNN model and from analytical methods. In this sense, after simulating several disturbances in the point clouds, the conventional registration methods found some difficulty when converging, while the CNN model kept its response within the expected values. This makes it possible to use network estimation for two purposes. On the one hand, a verification that the first approximation to be provided to the registration method is within expected orders of magnitude. On the other hand, if the registration method itself finds difficulty to converge to a correct solution, network estimation can be used as an approximation in a new execution of the registration method.

Future work

On the basis of the conclusions and contributions of the thesis, there remains several open lines of research, where progress could be made in perception systems based on 3D LiDAR. In fact, the results obtained could be expanded or improved upon through an in-depth analysis, in the following areas of application:

Improvement of the overall performance of the algorithms:

- Some of the proposed algorithms have been developed under the premise of maintaining a low computational cost, as it is considered one of the crucial aspects of the autonomous vehicle. However, in order to find a balance between performance and precision in the result, already developed generic functions could be used. Although these tools are not optimized for the specific application, they can provide improvement at certain steps within the overall structure of the algorithm.
- The developments presented throughout the chapters focus on the study of the capabilities that a single LiDAR sensor can provide. Once the efficiency of the operation with a single sensor is proved, the integration of a sensory fusion could be subsequently faced. This approach combines the advantages, in many cases complementary, of different sensors, thereby achieving a redundant system. The great variability of scenarios through which the autonomous vehicle can circulate highlights the importance of robust and generalizable systems.

Alternatives to neural network topologies:

- The design of the network architectures used in the CNN models requires a cyclical process to achieve the best fit. Moreover, such research field is constantly changing, thereby leading to added complexity. Despite having optimized the development environment for the design of architectures, it would be interesting to develop an architecture based on recurrent neural networks to obtain the estimation of the odometry. As indicated in some previous works mentioned in the thesis, recurrent networks can bring good results when there is a temporal dependence among data. In this respect, some prelimi-

nary tests have been carried out without improving the results described in the thesis, however, a more exhaustive analysis could eventually offer remarkable results.

- Similarly, if recurrent neural networks can provide a good performance, it would be interesting to design a network architecture that combines the contribution of these together with the feature extraction of convolutional neural networks. A similar approach, but using artificial vision, is the one developed in S. Wang et al. (2017). Based on the development of the section 3.3 of this thesis, it would be easily adaptable to the use of point clouds.

Alternatives for representing the environment as input to neural network models:

- In the same way that the process of cyclical design of a Deep Learning network topology is decisive during development, data preprocessing is very important to achieve a good model optimization. Thus, several configurations of the point cloud processing have been presented to be used as input to convolutional neural networks. However, there are other configurations proposed that can report a good operation as well, for example, resorting to sensory fusion, integrating the RGB image of cameras along with the depth maps developed in the thesis. Despite increasing the complexity of data processing, color information can provide benefits by inferring vehicle odometry.
- Likewise, the integration between the different algorithms presented can be studied and, in this way, the perception system within the structure of an autonomous vehicle can be completed. In this sense, the information of the lane width, the lateral error within the same or elements of the road, can be encoded and included in the maps used in CNN models. This additional layer of information can achieve an improvement in odometry estimation by containing relevant information from the environment.

Referencias

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... Brain, G. (2016). TensorFlow: A System for Large-Scale Machine Learning TensorFlow: A system for large-scale machine learning. En *12th usenix symposium on operating systems design and implementation (osdi '16)* (pp. 265–284).
- Alvarez, J. M. Á., y Lopez, A. M. (2010). Road detection based on illuminant invariance. *IEEE transactions on intelligent transportation systems*, 12(1), 184–193.
- Anderson, S. J., Karumanchi, S. B., y Iagnemma, K. (2012). Constraint-based planning and control for safe, semi-autonomous operation of vehicles. En *Ieee intelligent vehicles symposium, proceedings* (pp. 383–388).
- Ankerst, M., Breunig, M. M., Kriegel, H.-P., y Sander, J. (1999, junio). Optics: Ordering points to identify the clustering structure. *SIGMOD Rec.*, 28(2), 49–60.
- Aparicio, F., Jiménez, F., y Sanchez, J. (2009). Development and use of vehicle dynamics simulation software as support for road vehicles theory teaching. *Computer Applications in Engineering Education*, 17(4), 467–478.
- Asvadi, A., Premevida, C., Peixoto, P., y Nunes, U. (2016, 9). 3D Lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes. *Robotics and Autonomous Systems*, 83, 299–311.
- Atia, M. M., Hilal, A. R., Stellings, C., Hartwell, E., Toonstra, J., Miners, W. B., y Basir, O. A. (2017). A low-cost lane-determination system using GNSS/IMU fusion and HMM-based multistage map matching. *IEEE Transactions on Intelligent Transportation Systems*, 18(11), 3027–3037.
- Bergenheim, C., Hedin, E., y Skarin, D. (2012). Vehicle-to-Vehicle Communication for a Platooning System. *Procedia - Social and Behavioral Sciences*, 48(2011), 1222–1233.
- Besl, P. J., y McKay, N. D. (1992, 2). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), 239–256.
- Biber, P., y Straßer, W. (2003). The normal distributions transform: A new approach to laser scan matching. En *Proceedings 2003 ieee/rsj international conference on intelligent robots and systems (iros 2003)(cat. no. 03ch37453)* (Vol. 3, pp. 2743–2748).
- Blackman, S. S. (2004). Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1), 5–18.
- Bloesch, M., Czarnowski, J., Clark, R., Leutenegger, S., y Davison, A. J. (2018).

- CodeSLAM—learning a compact, optimisable representation for dense visual SLAM. En *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 2560–2568).
- Bresson, G., Rahal, M. C., Gruyer, D., Revilloud, M., y Alsayed, Z. (2016, 12). A Cooperative fusion architecture for robust localization: Application to autonomous driving. En *Ieee conference on intelligent transportation systems, proceedings, itsc* (pp. 859–866). Institute of Electrical and Electronics Engineers Inc.
- Broggi, A., Bertozzi, M., Fascioli, A., Bianco, C. G. L., y Piazzzi, A. (1999). The ARGO Autonomous Vehicle’s Vision and Control Systems. *International Journal of Intelligent Control and Systems*, 3(4), 409–441.
- Broggi, A., Bertozzi, M., Fascioli, A., Guarino Lo Bianco, C., y Piazzzi, A. (2000). Visual Perception of Obstacles and Vehicles for Platooning. *IEEE Transactions on Intelligent Transportation Systems*, 1(3), 164–176.
- Buehler, M., Iagnemma, K., y Singh, S. (2009). *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic* (Springer, Ed.). Springer.
- Campello, R. J., Moulavi, D., y Sander, J. (2013). Density-based clustering based on hierarchical density estimates. En *Pacific-asia conference on knowledge discovery and data mining* (pp. 160–172).
- Caraffi, C., Cattani, S., y Grisleri, P. (2007). Off-road path and obstacle detection using decision networks and stereo vision. *IEEE Transactions on Intelligent Transportation Systems*, 8(4), 607–618.
- Castro, M., Lopez-Cuervo, S., Paréns-González, M., y de Santos-Berbel, C. (2016). LIDAR-based roadway and roadside modelling for sight distance studies. *Survey Review*, 48(350), 309–315.
- Christiansen, P., Nielsen, L. N., Steen, K. A., Jørgensen, R. N., y Karstoft, H. (2016, 11). DeepAnomaly: Combining background subtraction and deep learning for detecting obstacles and anomalies in an agricultural field. *Sensors*, 16(11), 1904.
- Chu, K., Lee, M., y Sunwoo, M. (2012). Local path planning for off-road autonomous driving with avoidance of static obstacles. *IEEE Transactions on Intelligent Transportation Systems*, 13(4), 1599–1616.
- Clavijo, M., Jimenez, F., Castellanos, F., y Labari, I. (2017). “Where in the lane” positioning for autonomous vehicles. En *12th its european congress* (pp. 19–22).
- Cox, I. J., y Hingorani, S. L. (1996). An efficient implementation of Reid’s multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on pattern analysis and machine intelligence*, 18(2), 138–150.
- Danelljan, M., Meneghetti, G., Shahbaz Khan, F., y Felsberg, M. (2016). A probabilistic framework for color-based point set registration. En *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1818–1826).
- Davison, A. J. (2003). Real-time simultaneous localisation and mapping with a single camera. En *Proceedings ninth ieee international conference on computer vision* (pp. 1403–1410). IEEE.

- Day, T. D. (1995, 2). An Overview of the HVE Vehicle Model. *SAE*(950308), 55–68.
- Deschaud, J. E. (2018, 9). IMLS-SLAM: Scan-to-Model Matching Based on 3D Data. En *Proceedings - iee international conference on robotics and automation* (pp. 2480–2485). Institute of Electrical and Electronics Engineers Inc.
- Ding, L., y Feng, C. (2019). DeepMapping: Unsupervised map estimation from multiple point clouds. En *Proceedings of the iee conference on computer vision and pattern recognition* (pp. 8650–8659).
- Doshi-Velez, F., y Kim, B. (2017). Towards A Rigorous Science of Interpretable Machine Learning.
- Eade, E., y Drummond, T. (2008). Unified loop closing and recovery for real time monocular SLAM. En *Bmvc 2008 - proceedings of the british machine vision conference 2008*.
- Ester, M., Kriegel, H.-P., Sander, J., y Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. En *Kdd-96 proceedings*. Portland, Oregon, USA, 2–4 August.
- Fernández, C., Izquierdo, R., Llorca, D. F., y Sotelo, M. A. (2014). Road curb and lanes detection for autonomous driving on urban scenarios. *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 1964–1969.
- Fernández, C., Llorca, D. F., Stiller, C., y Sotelo, M. A. (2015). Curb Detection Method Based on Curvatures for Urban Autonomous Navigation. *Iv*.
- García, F., Jiménez, F., Anaya, J. J., Armingol, J. M., Naranjo, J. E., y de la Escalera, A. (2013). Distributed pedestrian detection alerts based on data fusion with accurate localization. *Sensors*, 13(9), 11687–11708.
- Geron, A. (2019). *Hands-On Machine Learning with Scikit-Learn and TensorFlow 2nd edition*.
- Gidel, S., Checchin, P., Blanc, C., Chateau, T., y Trassoudaine, L. (2010). Pedestrian Detection and Tracking in an Urban Environment Using a Multilayer Laser Scanner. *IEEE Transactions on Intelligent Transportation Systems*, 11(3), 579–588.
- Goodfellow, I., Bengio, Y., y Courville, A. (2016). *Deep Learning*. MIT Press.
- Guo, C., Kidono, K., Meguro, J., Kojima, Y., Ogawa, M., y Naito, T. (2016, 8). A low-cost solution for automatic lane-level map generation using conventional in-car sensors. *IEEE Transactions on Intelligent Transportation Systems*, 17(8), 2355–2366.
- Guo, C., Mita, S., y McAllester, D. (2012). Robust Road Detection and Tracking in Challenging Scenarios Based on Markov Random Fields With Unsupervised Learning. *IEEE Transactions on Intelligent Transportation Systems*, 13(3), 1338–1354.
- Gutmann, J. S., Eade, E., Fong, P., y Munich, M. (2011, 6). A constant-time algorithm for vector field SLAM using an exactly sparse extended information filter. En *Robotics: Science and systems* (Vol. 6, pp. 193–200). Robotics: Science and Systems Foundation.
- Hähnel, D., Burgard, W., Fox, D., y Thrun, S. (2003). An Efficient FastSLAM Algorithm for Generating Maps. *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, 1(October), 206 - 211.

- Han, J., Kim, D., Lee, M., y Sunwoo, M. (2012). Enhanced Road Boundary and Obstacle Detection Using a Downward-Looking LIDAR Sensor. *IEEE Transactions on Vehicular Technology*, 61(3), 971–985.
- Hastie, T., Tibshirani, R., y Friedman, J. (2009). *Springer Series in Statistics* (Vol. 27) (no 2).
- Hata, A., y Wolf, D. (2014). Road marking detection using LIDAR reflective intensity data and its application to vehicle localization. En *17th international ieee conference on intelligent transportation systems (itsc)* (pp. 584–589).
- Hata, A. Y., Osorio, F. S., y Wolf, D. F. (2014). Robust curb detection and vehicle localization in urban environments. En *2014 ieee intelligent vehicles symposium proceedings* (pp. 1257–1262).
- Heide, N., Emter, T., y Petereit, J. (2018). Calibration of multiple 3D LiDAR sensors to a common vehicle frame. En *Isr 2018; 50th international symposium on robotics* (pp. 1–8).
- Henriques, J. F., y Vedaldi, A. (2018). Mapnet: An allocentric spatial memory for mapping environments. En *proceedings of the ieee conference on computer vision and pattern recognition* (pp. 8476–8484).
- Holgado-Barco, A., Riveiro, B., González-Aguilera, D., y Arias, P. (2017, 1). Automatic Inventory of Road Cross-Sections from Mobile Laser Scanning System. *Computer-Aided Civil and Infrastructure Engineering*, 32(1), 3–17.
- Huang, Z. (1998). Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. *Data Mining and Knowledge Discovery*, 2(3), 283–304. doi: 10.1023/A:1009769707641
- Im, J. H., Im, S. H., y Jee, G. I. (2016, 8). Vertical corner feature based precise vehicle localization using 3D LIDAR in Urban area. *Sensors*, 16(8), 1268.
- Indelman, V., Nelson, E., Michael, N., y Dellaert, F. (2014, 9). Multi-robot pose graph localization and data association from unknown initial relative poses via expectation maximization. En *Proceedings - ieee international conference on robotics and automation* (pp. 593–600). Institute of Electrical and Electronics Engineers Inc.
- Jain, A. K., y Dubes, R. C. (1988). *Algorithms for Clustering Data*. USA: Prentice-Hall, Inc.
- Jian, B., y Vemuri, B. C. (2005). A robust algorithm for point set registration using mixture of Gaussians. En *Tenth ieee international conference on computer vision (iccv'05) volume 1* (Vol. 2, pp. 1246–1251).
- Jiménez, F. (2011). Improvements in road geometry measurement using inertial measurement systems in datalog vehicles. *Measurement: Journal of the International Measurement Confederation*, 44(1), 102–112.
- Jiménez, F. (2017). *Intelligent Road Vehicles: Enabling Technologies and Future Developments*. Elsevier.
- Jiménez, F., Clavijo, M., Naranjo, J. E., y Gómez, . (2016). Improving the Lane Reference Detection for Autonomous Road Vehicle Control. *Journal of Sensors*, 2016, 13.

- Jiménez, F., y Naranjo, J. E. (2011). Improving the obstacle detection and identification algorithms of a laserscanner-based collision avoidance system. *Transportation Research Part C: Emerging Technologies*, 19(4), 658–672.
- Jo, Y., y Ryu, S. (2015, 11). Pothole detection system using a black-box camera. *Sensors (Switzerland)*, 15(11), 29316–29331.
- Joshi, A., y James, M. R. (2015, 1). Generation of accurate lane-level maps from coarse prior maps and lidar. *IEEE Intelligent Transportation Systems Magazine*, 7(1), 19–29.
- Kehoe, B., Patil, S., Abbeel, P., y Goldberg, K. (2015, 4). A Survey of Research on Cloud Robotics and Automation. *IEEE Transactions on Automation Science and Engineering*, 12(2), 398–409.
- Ketchen, D. J., y Shook, C. L. (1996). The application of cluster analysis in strategic management research: an analysis and critique. *Strategic management journal*, 17(6), 441–458.
- Kingma, D., y Ba, J. (2014). Adam: A Method for Stochastic Optimization. En *International conference on learning representations* (pp. 1–13).
- Konda, K., y Memisevic, R. (2015). Learning Visual Odometry with a Convolutional Network. En *International conference on computer vision theory and applications* (pp. 486–490).
- Krizhevsky, A., Sutskever, I., y Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, 1–9.
- Kummerle, J., Kuhner, T., y Lauer, M. (2018). Automatic Calibration of Multiple Cameras and Depth Sensors with a Spherical Target. En *Ieee international conference on intelligent robots and systems* (pp. 5584–5591).
- Kümmerle, R., Steder, B., Dornhege, C., Ruhnke, M., Grisetti, G., Stachniss, C., y Kleiner, A. (2009). On Measuring the Accuracy of SLAM Algorithms. *Auton. Robots*, 27(4), 387–407.
- Larson, J., y Trivedi, M. (2011). Lidar based off-road negative obstacle detection and analysis. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 192–197.
- Laurent, J., Hérbert, J. F., Lefebvre, D., y Savard, Y. (2009). Using 3D laser profiling sensors for the automated measurement of road surface conditions (ruts, macro-texture, raveling, cracks). , 1–8.
- Leonard, J. J., y Durrant-Whyte, H. F. (1991). Mobile Robot Localization by Tracking Geometric Beacons. *IEEE Transactions on Robotics and Automation*, 7(3), 376–382.
- Li, H., y Nashashibi, F. (2013). Cooperative multi-vehicle localization using split covariance intersection filter. *IEEE Intelligent Transportation Systems Magazine*, 5(2), 33–44.
- Li, L., Luo, W., y Wang, K. C. P. (2018). Lane marking detection and reconstruction with line-scan imaging data. *Sensors*, 18(5), 1635.
- Li, Q., Yao, M., Yao, X., y Xu, B. (2009, 1). A real-time 3D scanning system for pavement distortion inspection. *Measurement Science and Technology*, 21(1), 015702.
- Lu, F., y Milios, E. (1997). Globally Consistent Range Scan Alignment for Environment Mapping. *Auton. Robots*, 4(4), 333–349.

- Magnusson, M. (2009). *The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection* (Tesis Doctoral no publicada). Örebro universitet.
- McInnes, L., y Healy, J. (2017). Accelerated hierarchical density based clustering. En *2017 IEEE international conference on data mining workshops (icdmw)* (pp. 33–42).
- Mechanical Simulation Corporation. (2001). *CARSIM 5: Math models manual*. Michigan: MSC.
- Mera, J. M. (2001). *Modelos complejos de sistemas vehiculares empleando la técnica del Bond Graph* (Tesis Doctoral no publicada). Madrid.
- Mohan, M., y Krishna, K. M. (2010, 12). Mapping large scale environments by combining particle filter and information filter. En *11th international conference on control, automation, robotics and vision, icarcv 2010* (pp. 1000–1005). IEEE.
- Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., ... Thrun, S. (2009). Junior: The stanford entry in the urban challenge. *Springer Tracts in Advanced Robotics*, 56(October 2005), 91–123.
- Montemerlo, M., Thrun, S., Koller, D., y Wegbreit, B. (2002). FastSLAM: A factored solution to the simultaneous localization and mapping problem. *Proc. of 8th National Conference on Artificial Intelligence/14th Conference on Innovative Applications of Artificial Intelligence*, 68(2), 593–598.
- Montiel, J. M., Civera, J., y Davison, A. J. (2007). Unified inverse depth parametrization for monocular SLAM. En *Robotics: Science and systems* (Vol. 2, pp. 81–88).
- Myronenko, A., y Song, X. (2010). Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12), 2262–2275.
- Nair, V., y Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. *Proceedings of the 27th International Conference on Machine Learning*(3), 807–814.
- Nalecz, A. G. (1992, 2). Development and Validation of Light Vehicle Dynamics Simulation (LVDS). *SAE*(920056), 87–108.
- Naranjo, J. E., Jiménez, F., Aparicio, F., y Zato, J. (2009). GPS and inertial systems for high precision positioning on Motorways. *Journal of Navigation*, 62(2), 351–363.
- Nettleton, E., Thrun, S., Durrant-Whyte, H., y Sukkarieh, S. (2006). Decentralised SLAM with low-bandwidth communication for teams of vehicles. En *Field service robot* (pp. 179–188).
- Neuhaus, F., Koß, T., Kohnen, R., y Paulus, D. (2019, 10). MC2SLAM: Real-Time Inertial Lidar Odometry Using Two-Scan Motion Compensation. En *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Vol. 11269 LNCS, pp. 60–72). Springer Verlag.
- Newman, P., Cole, D., y Ho, K. (2006). Outdoor SLAM using visual appearance and laser ranging. En *Proceedings - IEEE international conference on robotics and automation* (Vol. 2006, pp. 1180–1187). IEEE.

- Nicolai, A., Skeeel, R., Eriksen, C., y Hollinger, G. A. (2016). Deep Learning for Laser Based Odometry Estimation. En *Science and systems conf. workshop on limits and potentials of deep learning in robotics*.
- Oniga, F., y Nedevski, S. (2010, 3). Processing dense stereo data using elevation maps: Road surface, traffic isle, and obstacle detection. *IEEE Transactions on Vehicular Technology*, 59(3), 1172–1182.
- OUSTER. (2014). *OS-1 High Resolution Imaging LIDAR - Datasheet*.
- Ozguner, U., Stiller, C., y Redmill, K. (2007, 2). Systems for Safety and Autonomous Behavior in Cars: The DARPA Grand Challenge Experience. *Proceedings of the IEEE*, 95(2), 397–412.
- Parisotto, E., Singh Chaplot, D., Zhang, J., y Salakhutdinov, R. (2018). Global pose estimation with an attention-based recurrent network. En *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 237–246).
- Pendleton, S. D., Andersen, H., Du, X., Shen, X., Meghjani, M., Eng, Y. H., ... Ang, M. H. (2017). Perception, planning, control, and coordination for autonomous vehicles. *Machines*, 5(1), 1–54.
- Pereira, M., Silva, D., Santos, V., y Dias, P. (2016). Self calibration of multiple LIDARs and cameras on autonomous vehicles. *Robotics and Autonomous Systems*, 83, 326–337.
- Petrovskaya, A., y Thrun, S. (2009). Model based vehicle detection and tracking for autonomous urban driving. *Autonomous Robots*, 26(2-3), 123–139.
- Pierzchała, M., Giguère, P., y Astrup, R. (2018, 2). Mapping forests using an unmanned ground vehicle with 3D LiDAR and graph-SLAM. *Computers and Electronics in Agriculture*, 145, 217–225.
- Quehl, J., Yan, S., Wirges, S., Pauls, J. H., y Lauer, M. (2019, 1). Estimating Object Shape and Movement Using Local Occupancy Grid Maps. En *Ifac-papersonline* (Vol. 52, pp. 233–238). Elsevier B.V.
- Ramezani, M., Tinchev, G., Iuganov, E., y Fallon, M. (2020). *Online LiDAR-SLAM for Legged Robots with Robust Registration and Deep-Learned Loop Closure*.
- Ramsundar, B., y Zadeh, R. B. (2018). *TensorFlow for Deep Learning*.
- Raymond, R., Morimura, T., Osogami, T., y Hirose, N. (2012). Map matching with hidden Markov model on sampled road network. En *Proceedings of the 21st international conference on pattern recognition (icpr2012)* (pp. 2242–2245).
- Rowekamper, J., Ruhnke, M., Steder, B., Burgard, W., y Tipaldi, G. D. (2015). Automatic extrinsic calibration of multiple laser range sensors with little overlap. En *Proceedings - IEEE international conference on robotics and automation* (Vol. 2015-June, pp. 2072–2077).
- Roy, S., y Bhattacharyya, D. K. (2005). An approach to find embedded clusters using density based techniques. En *International conference on distributed computing and internet technology* (pp. 523–535).
- SAE. (2016). Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems. *Sae International*.

- Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3), 210–229.
- Sander, J., Ester, M., Kriegel, H.-P., y Xu, X. (1998). Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data mining and knowledge discovery*, 2(2), 169–194.
- Scaramuzza, D., y Siegwart, R. (2008, 10). Appearance-Guided Monocular Omnidirectional Visual Odometry for Outdoor Ground Vehicles. *IEEE Transactions on Robotics*, 24(5), 1015–1026.
- Scovanner, P., Ali, S., y Shah, M. (2007). A 3-dimensional sift descriptor and its application to action recognition. En *Proceedings of the 15th acm international conference on multimedia* (pp. 357–360).
- Segal, A. V., Haehnel, D., y Thrun, S. (2009). Generalized-ICP. *Proc. of Robotics: Science and Systems*, 2, 4.
- Sezer, V., y Gokasan, M. (2012). A novel obstacle avoidance algorithm: "Follow the gap method". *Robotics and Autonomous Systems*, 60(9), 1123–1134.
- Shan, T., y Englot, B. (2018, 10). LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain. En *2018 ieee/rsj international conference on intelligent robots and systems (iros)* (pp. 4758–4765).
- Shang, E., An, X., Li, J., y He, H. (2014). A novel setup method of 3D LIDAR for negative obstacle detection in field environment. *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 1436–1441.
- Shang, E., An, X., Shi, M., Meng, D., Li, J., y Wu, T. (2015, 7). An Efficient Calibration Approach for Arbitrary Equipped 3-D LiDAR Based on an Orthogonal Normal Vector Pair. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 79(1), 21–36.
- Shang, E., An, X., Wu, T., Hu, T., Yuan, Q., y He, H. (2016, 8). LiDAR Based Negative Obstacle Detection for Field Autonomous Land Vehicles. *Journal of Field Robotics*, 33(5), 591–617.
- Smith, R., Self, M., y Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. En *Autonomous robot vehicles* (pp. 167–193). Springer.
- Spies, M., y Spies, H. (2006). Automobile lidar sensorik: stand, trends und zukünftige herausforderungen. *Advances in Radio Science: ARS*, 4, 99.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., y Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15, 1929–1958.
- Steder, B., Rusu, R. B., Konolige, K., y Burgard, W. (2010). NARF: 3D range image features for object recognition. En *Workshop on defining and solving realistic perception problems in personal robotics at the ieee/rsj int. conf. on intelligent robots and systems (iros)* (Vol. 44).
- Sturm, J., Engelhard, N., Endres, F., Burgard, W., y Cremers, D. (2012). A benchmark for the evaluation of RGB-D SLAM systems. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 573–580.

- Tan, J., Li, J., An, X., y He, H. (2014). Robust Curb Detection with Fusion of 3D-Lidar and Camera Data. *Sensors*, 14(5), 9046–9073.
- Thrun, S., Liu, Y., Koller, D., Ng, A. Y., Ghahramani, Z., y Durrant-Whyte, H. (2004, 8). Simultaneous localization and mapping with sparse extended information filters. En *International journal of robotics research* (Vol. 23, pp. 693–716). SAGE Publications.
- Toledo-Moreo, R., Armingol, J. M., Clavijo, M., de la Escalera, A., Ser, J. d., Musleh, B., ... Sánchez-Cubillo, J. (2018). Positioning and Digital Maps. En *Intelligent vehicles* (pp. 141–174). Elsevier.
- Toledo-Moreo, R., Bétaille, D., y Peyret, F. (2009). Lane-level integrity provision for navigation and map matching with GNSS, dead reckoning, and enhanced maps. *IEEE Transactions on Intelligent Transportation Systems*, 11(1), 100–112.
- Tombari, F., Salti, S., y Di Stefano, L. (2010). Unique signatures of histograms for local surface description. En *European conference on computer vision* (pp. 356–369).
- Torr, P. H., y Zisserman, A. (2000). MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1), 138–156.
- Umeyama, S. (1991). Least-Squares Estimation of Transformation Parameters Between Two Point Patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(4), 376–380.
- Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., y Brox, T. (2017, 7). DeMoN: Depth and Motion Network for Learning Monocular Stereo. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Velodyne Lidar. (2016). *VLP-16 Datasheet*.
- Vera, C., y Félez, J. (1994). *Dinámica vehicular basada en la técnica del BOND GRAPH*. Madrid: Publications Section of the ETSII.
- Wang, S., Clark, R., Wen, H., y Trigoni, N. (2017). DeepVO : Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks. , 2043–2050.
- Wang, W., Yan, X., Huang, H., Chu, X., y Abdel-Aty, M. (2011, 8). Design and verification of a laser based device for pavement macrotexture measurement. *Transportation Research Part C: Emerging Technologies*, 19(4), 682–694.
- Widmann, G. R., Daniels, M. K., Hamilton, L., Humm, L., Riley, B., Schiffmann, J. K., ... Wishon, W. H. (2000). Comparison of Lidar-Based and Radar-Based Adaptive Cruise Control Systems. En *Sae technical paper*. SAE International.
- Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., y Recht, B. (2017). The marginal value of adaptive gradient methods in machine learning. En *Advances in neural information processing systems* (Vol. 2017-Decem, pp. 4149–4159).
- Wirges, S., Reith-Braun, M., Lauer, M., y Stiller, C. (2019). Capturing object detection uncertainty in multi-layer grid maps. En *Ieee intelligent vehicles symposium, proceedings* (Vol. 2019-June, pp. 1520–1526).
- Wu, M., Hao, J., Fu, H., Gao, Y., y Zhang, H. (2019). A stereo visual odometry pose optimization method via flow-decoupled motion field model. *Cehui Xuebao/Acta Geodaetica et Cartographica Sinica*, 48, 460–472.

- Yang, C., y Medioni, G. (1992, 4). Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3), 145–155.
- Yang Cheng, Mark Maimone, y Larry Matthies. (2005, 10). Visual odometry on the Mars Exploration Rovers. En *2005 ieee international conference on systems, man and cybernetics* (Vol. 1, pp. 903–910).
- Yanwen, W., Nan, Z., Tao, Z., y Wei, Y. (2018). Research of lane detection and tracking methods based on multi-sensor fusion. *Application Research of Computers*(2), 60.
- Yousif, K., Bab-Hadiashar, A., y Hoseinnezhad, R. (2015, 12). An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics. *Intelligent Industrial Systems*, 1(4), 289–311.
- Yu, Y. F., Zhao, H. J., Cui, J. S., y Zha, H. B. (2017). Road structural feature based monocular visual localization for intelligent vehicle. *Acta Automatica Sinica*, 43(5), 725–734.
- Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., y Funkhouser, T. (2017). 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. En *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1802–1811).
- Zhang, J., y Singh, S. (2016). Low-drift and real-time lidar odometry and mapping. *Autonomous Robots*(October 2014).
- Zhang, W. (2010). LIDAR-based road and road-edge detection. En *2010 ieee intelligent vehicles symposium* (pp. 845–848).
- Zhang, X., Zhang, A., y Meng, X. (2015). Automatic fusion of hyperspectral images and laser scans using feature points. *Journal of Sensors*, 2015.
- Zhang, Z., y Shan, Y. (2004). Incremental motion estimation through modified bundle adjustment. En *Proceedings 2003 international conference on image processing (cat. no.03ch37429)* (Vol. 3, pp. 343–6). IEEE.
- Zhao, G., y Yuan, J. (2012). Curb detection and tracking using 3D-LIDAR scanner. *2012 19th IEEE International Conference on Image Processing*, 437–440.
- Zhou, T., Brown, M., Snavely, N., y Lowe, D. G. (2017). Unsupervised learning of depth and ego-motion from video. En *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1851–1858).

Anexo A.

Adaptación del algoritmo de clustering DBSCAN para sensores láser rotativos

A.1. Introducción

La extracción de características durante la navegación autónoma permite identificar los patrones principales del entorno que aportan información relevante para llevar a cabo una detección, un posicionamiento o cualquier otra finalidad. Tras la identificación de estas características, en general es necesario llevar a cabo una segmentación para garantizar el correcto funcionamiento en etapas posteriores. La segmentación de características permite agrupar y clasificar la información extraída previamente. Sin esta agrupación, no será posible llevar a cabo correctamente etapas sucesivas del algoritmo. Por ejemplo, en algoritmos de detección de obstáculos, permite realizar un tracking de estos, estudiando su evolución temporal y evaluar si pueden suponer un peligro para la conducción. En algoritmos de posicionamiento visual, una correcta segmentación permite realizar el matching de características entre frames y estimar el movimiento relativo del vehículo.

La utilización del LiDAR 3D en el vehículo autónomo ha permitido aplicar técnicas de clustering empleadas tradicionalmente para segmentar valores distribuidos en el espacio. La información en forma de nubes de puntos favorece una identificación robusta de características y poder aplicar métodos de segmentación basados en variables cuantitativas.

Sin embargo, el propio funcionamiento de los sensores LiDAR rotativos limita la correcta segmentación de estos métodos de clustering. En este sentido, las nubes de puntos de los sensores láser se originan desde un mismo origen (el propio sensor), por lo que la distribución de puntos no es homogénea. Por tanto, es necesario adaptar los métodos de clustering para tener en cuenta los factores de divergencia entre puntos y la variación de la densidad en función de la distancia al láser. Estos factores pueden suponer grandes limitaciones en la distancia de identificación o incluso en detecciones parciales de los elementos del entorno. Por ejemplo, en un algoritmo de detección de obstáculos, un clustering limitado puede suponer una detección tardía del peligro.

Se propone en este anexo una adaptación del método de clustering DBSCAN, empleado en varios desarrollos de la tesis, para ser utilizado correctamente con sensores láser rotativos, donde la aplicación del método original origina dificultades en la segmentación de la información del entorno.

Se han llevado a cabo diversos ensayos sobre datos reales que verifican la mejora significativa del funcionamiento de la modificación del método. Se ha comprobado, además, que el método propuesto funciona correctamente con varios modelos LiDAR, en concreto el VLP-16 de Velodyne y el OS-1 64 de Ouster, donde el número de capas, resolución horizontal y variación de la densidad de puntos es distinto.

A.2. Principales algoritmos de clustering empleados en la conducción autónoma

Los métodos de clustering son técnicas de aprendizaje no supervisado (o semi-supervisado en algunos casos) que llevan a cabo una agrupación o segmentación sobre un conjunto de datos, dividiéndolo en subgrupos o clusters (Hastie et al., 2009). Los elementos incluidos dentro de cada cluster tendrán mayor grado de similitud que los elementos de otros clusters.

Atendiendo a la forma de creación estos clusters, se podrán diferenciar dos técnicas: clustering de partición y clustering jerárquico.

En el primer caso, se realiza la segmentación en un número predefinido de clusters. Conociendo previamente el número de agrupamientos que se deben realizar, se crean las agrupaciones en torno a unos centroides. Dependiendo del algoritmo, este centroide se define minimizando una función de similitud con el resto de los elementos del conjunto de datos.

En el caso de los métodos jerárquicos, esta definición responde a los algoritmos que representan las similitudes entre elementos del conjunto de datos en forma de árbol o dendograma. La creación de los clusters se realiza en función de la métrica de similitud empleada que servirá como guía a través del árbol para incluir nuevos elementos al cluster. La principal característica de estos algoritmos es que no requieren definir previamente el número de clusters esperado.

Respecto al empleo de sensores LiDAR para la percepción del entorno, existe una amplia gama de algoritmos de clustering que pueden ser aplicados a variables cuantitativas (figura A.1). La propia representación del entorno en forma de nubes de puntos proporciona las coordenadas de cada punto, por lo que los elementos a segmentar están descritos por su posición en un espacio euclídeo y no es necesario una categorización adicional.

Figura A.1: Ejemplos del funcionamiento de algunos métodos de clustering (Fuente: scikit-learn).

Uno de los algoritmos ampliamente utilizados en la segmentación de conjuntos de datos es K-means. Se caracteriza por su eficiencia en grandes conjuntos de datos y para su funcionamiento se requieren elementos con variables cuantitativas (Jain y Dubes, 1988). Los agrupamientos se crean a partir de la similitud entre los elementos y un centroide definido. La similitud se basa en la distancia euclídea centroide-elemento. Por ello, su utilización con datos LiDAR es directa, sin embargo, al tratarse de un método de clustering de partición su principal limitación es definir a priori el número de cluster esperados.

Algunas variantes del K-means, son el K-prototypes y el K-medoids. Ambos siguen siendo métodos de clustering no jerárquicos, por lo que se debe de indicar previamente el número de clusters. En el caso de K-prototypes vuelve a utilizar la distancia euclídea entre elementos, pero aplica una medida combinada de disimilitud (Huang, 1998). En el caso de K-medoids, trata de mitigar los problemas que se producen con el algoritmo de K-means cuando existen grandes distancias entre elementos. En este caso, K-medoids agrupa los outliers en grupos diferentes (Hastie et al., 2009).

Aunque la necesidad de definir previamente el número de clusters esperado pueda suponer una limitación, estos algoritmos son ampliamente utilizados por su rendimiento. Además, existen diferentes métodos analíticos que permiten estimar el número de clusters requerido. Métodos como el que se presenta en Ketchen y Shook (1996) pueden proporcionar un número óptimo de clusters según el dataset a partir de la representación gráfica del coeficiente de aglomeración (también conocido como inercia del modelo (Geron, 2019)). Este indicador corresponde con la media del cuadrado de la distancia de cada elemento al centroide más cercano. Si se representa este valor en función del número de cluster designados (número de centroides), se puede identificar el número de cluster óptimos para un conjunto de datos. Este método ofrece una aproximación genérica poco precisa, por ello se plantean otras métricas

como Silhouette Score que se ajusta mejor al número de clusters del dataset, pero supone un mayor coste computacional (Geron, 2019).

Por otro lado, los métodos jerárquicos evitan esta limitación al no precisar de la definición previa del número de agrupamientos, por ello su versatilidad a la hora de integrarse en algoritmos más complejos. En tal caso, se pueden encontrar algunos ejemplos de algoritmos basados en agrupamientos por densidad como DBSCAN, OPTICS o Mean-shift.

El algoritmo de DBSCAN, utilizado en varias ocasiones a lo largo de la tesis, se caracteriza por tener una gran eficiencia en la segmentación, permitir identificar clusters de cualquier apariencia y mantener una carga computacional baja, a pesar de que esta etapa suele demandar mayor tiempo de ejecución. Se basa en la cercanía y la densidad de puntos como métrica de similitud para formar nuevos clusters. Su integración requiere elementos cuantitativos sobre los que evaluar la conectividad por pares. Además, es necesario designar dos parámetros para llevar a cabo la segmentación: la distancia entre elementos que se considerarán del mismo grupo y la mínima cantidad de elementos para formar parte del cluster. Si no se cumple alguna de estas condiciones, el elemento se clasifica como ruido. Existen numerosas variantes de este algoritmo como pueden ser EnDBSCAN (Roy y Bhattacharyya, 2005), HDBSCAN (McInnes y Healy, 2017; Campello et al., 2013) o GDBSCAN (Sander et al., 1998), entre otras.

El algoritmo OPTICS (Ankerst et al., 1999) se basa en el funcionamiento del DBSCAN y puede considerarse como una generalización de este. Sus principales ventajas respecto al DBSCAN son un mejor funcionamiento con clusters que presentan densidades diferentes entre ellos y, por otro lado, no requiere especificar el parámetro de la distancia entre elementos de la misma agrupación. Tanto el algoritmo OPTICS como DBSCAN encuentran dificultades en los límites que definen los clusters, donde la densidad del agrupamiento suele decrecer.

Por su parte, el algoritmo Mean-Shift tiene algunas características comunes con DBSCAN al basarse igualmente en la densidad de elementos para crear los clusters y que estos puedan adoptar cualquier apariencia (Geron, 2019). Para ser implementado solo requiere introducir, como parámetro, el radio del círculo donde se realizará la evaluación de la densidad de elementos. El mayor inconveniente de este método es una mayor carga computacional y la tendencia a dividir un mismo cluster en distintos grupos cuando existen variaciones de densidad.

Todos los algoritmos citados anteriormente pueden ser aplicados para la segmentación de nubes de puntos ya que, como se ha descrito, funcionan con elementos de variables cuantitativas y suelen utilizar la distancia euclídea como métrica para la formación de clusters. Sin embargo, los láseres rotativos generan una nube de puntos con una distribución variable en función de la distancia al origen. Por ello, se observa una limitación de estos algoritmos al no estar orientados al procesamiento de nubes de puntos generadas desde un foco fijo. Se propone una variación del algoritmo DBSCAN para su utilización con LiDAR 3D en aplicaciones de vehículo autónomo.

A.3. Descripción de la adaptación del método de clustering DBSCAN

El algoritmo DBSCAN, como método de agrupamiento basado en densidad, evalúa la cantidad de elementos (instancias) que se encuentran en cierta área definida. Como se ha indicado, este algoritmo requiere predefinir dos parámetros además de la matriz de elementos que se trata de segmentar: ε (épsilon) y $minPts$ (mínimo de puntos). Si un elemento es densamente alcanzado por otro, pasan a formar parte del mismo cluster. Es decir, si existen al menos $minPts$ a una distancia ε , todos los elementos formarán parte del mismo agrupamiento.

La figura A.2 ilustra el funcionamiento básico del algoritmo para un valor 4 del parámetro $minPts$. Los puntos B y C son densamente alcanzables, mientras que el punto N es considerado como ruido al no cumplir con estas condiciones de densidad y cercanía.

Figura A.2: Esquema de funcionamiento algoritmos DBSCAN (Fuente: De Chire - Trabajo propio, CC BY-SA 3.0)

Por tanto, se puede considerar que el parámetro ε estará relacionado con la divergencia entre los puntos láser obtenidos y $minPts$ con la densidad de estos. Siendo los dos factores clave del funcionamiento de los LiDAR 3D a tener en cuenta para la adaptación del método.

La figura A.3 ilustra cómo se definen los parámetros de la resolución horizontal ($\Delta\theta$), la resolución vertical, entre capas ($\Delta\gamma$) y la distancia del punto al origen de coordenadas (d).

Figura A.3: Definición de los parámetros $\Delta\theta$, $\Delta\gamma$ y d . a) Vista en planta; b) vista de perfil.

A partir de estos parámetros la divergencia entre puntos de una misma capa láser se había definido en el apartado 2.2 según la expresión A.1.

$$s = 2 \cdot \sin \frac{\Delta\theta}{2} \cdot d \quad (\text{A.1})$$

Se observa que existe una relación lineal entre la divergencia y la distancia. De igual modo, la divergencia que se produce entre capas correspondería a la misma expresión anterior (ecuación A.1), sustituyendo $\Delta\theta$ por la resolución vertical ($\Delta\gamma$).

Por otro lado, la densidad de puntos en función de la distancia se había estimado en el apartado 3.1 como se indica en la ecuación A.2.

$$\rho = \frac{K}{d^2} \quad (\text{A.2})$$

Por lo que, en este caso, la relación de la densidad y la distancia varía de forma cuadrática. En concreto, para una resolución horizontal definida por $\Delta\theta$ y una resolución vertical dada por $\Delta\gamma$, la expresión anterior quedaría definida según la ecuación A.3.

$$\rho = \frac{4/(\Delta\theta \cdot \sin(\Delta\gamma))}{d^2} \quad (\text{A.3})$$

Asumiendo la aproximación por ángulos pequeños, la expresión anterior se puede simplificar como se indica en la ecuación A.4.

$$\rho = \frac{4/(\Delta\theta \cdot \Delta\gamma)}{d^2} \quad (\text{A.4})$$

Cuando se aplica el algoritmo DBSCAN en algoritmos orientados a la navegación autónoma se suelen definir constantes los parámetros de *épsilon* y *minPts* con unos valores que permitan un correcto funcionamiento. Los valores de *épsilon* suelen estar comprendidos entre 0,4 y 0,7 metros, mientras que para el valor de *minPts* suele considerarse entre 20 y 35 puntos. Estos valores, como se indica, suelen ser obtenidos a partir de la experimentación y son dependientes tanto del modelo LiDAR como del objetivo perseguido por el algoritmo global.

En el ejemplo concreto de utilizar un LiDAR VLP-16, la resolución horizontal a 10 Hz es $\Delta\theta = 0,2^\circ$ y la resolución vertical es $\Delta\gamma = 2^\circ$. Si se asignan unos valores normalmente empleados como $\varepsilon = 0,5$ y $\text{minPts} = 20$, según las ecuaciones A.1 y A.4, puede existir una limitación por divergencia entre puntos a partir de una distancia de 10 – 15 metros y debido a la variación de densidad a partir de los 40 metros. Por tanto, la elección de un valor u otro limitará la zona de segmentación efectiva, y por tanto, las prestaciones del algoritmo global donde se integre la etapa de clustering.

La variación del algoritmo de DBSCAN propuesta, trata de compensar estas variaciones de los parámetros según la distancia. De tal manera que estableciendo una variación lineal del parámetro *épsilon* con la distancia, es posible establecer la compensación de la divergencia. Igualmente, con el parámetro *minPts*, al estar relacionado con la densidad, se puede

compensar la relación cuadrática descrita.

En concreto la variación del algoritmo DBSCAN se aplica según se den las siguientes condiciones:

- El nuevo elemento evaluado se considera alcanzable si la distancia a un punto del cluster ($distancia_{alcanzable}$) cumple:

$$distancia_{alcanzable} \leq \epsilon + distancia \cdot coeficiente_{divergencia} \quad (A.5)$$

- El nuevo cluster será creado si los elementos que lo componen ($n_{elementos}$) cumple:

$$n_{elementos} \geq minPtos \cdot \frac{coeficiente_{densidad}}{distancia^2} \quad (A.6)$$

A.4. Resultados

A.4.1. Verificación en entorno ideal

En primer lugar, se lleva a cabo una verificación del comportamiento de la divergencia entre puntos y densidad frente a la distancia. El objetivo es comprobar, a través de una simulación en entorno ideal, que las deducciones analíticas indicadas en A.1 y A.4 son válidas para poder establecer las compensaciones descritas anteriormente.

Para ello, se ha simulado el funcionamiento del láser VLP-16 en un entorno donde un volumen de control de $1m^3$, se desplaza y rota siguiendo una trayectoria variable (figura A.4). El láser simulado se encuentra en el centro (circulo negro). Los puntos en rojo indican aquellos que inciden sobre el volumen de control. Los puntos azules muestran la trayectoria seguida por el volumen de control. Se registra el número de puntos que inciden sobre su superficie, así como la divergencia entre puntos.

Figura A.4: Simulación en entorno ideal.

En las figuras A.5 y A.6 se observa la variación de la dispersión entre puntos y la densidad obtenidos en la simulación del entorno ideal en función de la distancia al láser, respectivamente.

Figura A.5: Variación de la divergencia en función de la distancia obtenida en simulación de entorno ideal.

Figura A.6: Variación de la densidad en función de la distancia obtenida en simulación de entorno ideal.

A la vista de los resultados en entorno ideal, se identifica una variación lineal de la divergencia entre puntos y una variación cuadrática de tipo K/d^2 para la densidad. Las posibles desviaciones que se producen respecto al valor obtenido analíticamente son debidas principalmente a la proyección de los haces sobre las superficies con distinta inclinación. Por tanto, cuando se aplique la compensación sobre datos reales es necesario realizar un ajuste basado en los resultados experimentales.

Por tanto, las deducciones analíticas son representativas del comportamiento del láser y será posible aplicar una compensación lineal para el parámetro de ϵ y una compensación cuadrática para el parámetro de $minPts$.

A.4.2. Resultados obtenidos en entornos reales

Se han realizado diversas simulaciones sobre datos reales del entorno aplicando la modificación del algoritmo DBSCAN y contrastando el resultado con el obtenido en el caso de utilizar el algoritmo sin modificar. Para ello, se ha implementado con datos adquiridos con el modelo VLP-16 de Velodyne y sobre el LiDAR OS-1 64 de Ouster, comprobando de esta forma que el método propuesto es aplicable a uso con distintos láseres.

La figura A.7 muestra la comparación del resultado del clustering obtenido con el modelo VLP-16 aplicando la compensación por divergencia y densidad descrita anteriormente y utilizando el algoritmo con unos valores fijos ($\epsilon = 0,5$ metros y $minPtos = 20$). A la izquierda se muestra la modificación propuesta. La columna de la derecha corresponde al algoritmo con su configuración estándar. La fila de abajo muestra el detalle de la zona más cercana al láser. Las figuras muestran cada cluster de un color diferente, identificando de esta manera cómo se ha llevado a cabo la segmentación. Los puntos que no han sido incluidos en ningún cluster se identifican como ruido (gris).

Figura A.7: Comparativa de funcionamiento del algoritmo DBSCAN con VLP-16.

Se puede observar cómo el funcionamiento general mejora sustancialmente en el caso de aplicar la variación del método, ampliando la distancia de segmentación a toda la región de interés (figura A.7, izquierda). Un análisis de la zona más cercana al láser (figura A.7, fila inferior), muestra cómo a partir de los 10 metros comienzan a aparecer segmentaciones

parciales (se rodean en rojo algunos ejemplos), mientras que el método planteado consigue agrupar correctamente los elementos del entorno. La compensación de los parámetros del algoritmo DBSCAN supone un aumento de la distancia de identificación, para este caso, de 10 a 40 metros (distancia máxima definida por ROI).

Se ha obtenido experimentalmente que los valores para una compensación de divergencia y densidad correcta son: $\text{coeficiente}_{\text{divergencia}} = 0,06$ y $\text{coeficiente}_{\text{densidad}} = 350$. Cabe indicar que valores superiores de estos coeficientes supone una compensación demasiado alta, provocando la fusión de diferentes clusters en uno solo. El resto de los parámetros requeridos por el algoritmo de DBSCAN se han mantenido igual ($\varepsilon = 0,5$ metros y $\text{minPts} = 20$).

En el caso de utilizar el modelo de LiDAR OS-1 64, la figura A.8 muestra la comparación del resultado de la segmentación aplicando el método (figura A.8, izquierda) y utilizando los parámetros fijos del algoritmo DBSCAN (figura A.8, derecha). Al igual que en caso anterior, la fila inferior muestra la zona más cercana al láser. Cada cluster se ha representado de un color diferente y los *outliers* de color gris. En este caso, el láser con 64 capas tiene una resolución horizontal $\Delta\theta = 0,35^\circ$ y una resolución vertical $\Delta\gamma = 0,53^\circ$.

Figura A.8: Comparativa de funcionamiento del algoritmo DBSCAN con OS-1 64.

De nuevo, el funcionamiento del clustering mejora notablemente con la modificación del algoritmo. Un detalle de la zona más cercana al láser (figura A.8, fila inferior) muestra algunas segmentaciones parciales rodeadas en rojo a partir de los 15 - 20 metros. Por otro lado, con el método propuesto, la zona donde se logra un clustering correcto alcanza los 50 metros de

distancia máxima.

Los valores para la segmentación con parámetros fijos se mantienen igual que en el caso anterior ($\varepsilon = 0,5$ metros y $minPtos = 20$). Para la optimización del clustering mediante el método propuesto, se ha obtenido experimentalmente un $coeficiente_{divergencia} = 0,04$ y $coeficiente_{densidad} = 350$.

Nótese que la compensación de la divergencia es menor en el caso del modelo de OS-1 64 al tratarse de un modelo láser con mayor resolución que el VLP-16. Por otro lado, aunque el coeficiente para la compensación de densidad no varía de un modelo a otro, su aportación al método es relevante para rangos de alcance mayores como se puede observar en la figura A.9.

Figura A.9: Aportación de la compensación de densidad de puntos. A la izquierda, solo con compensación de la divergencia. A la derecha, el método con compensación de divergencia y densidad.

Como se puede observar, en el caso de aplicar solo una compensación por divergencia (figura A.9, izquierda) la distancia de identificación alcanza los 20 - 25 metros (en una configuración sin compensación se obtendría entre 10 - 15 metros). Sin embargo, añadiendo la compensación por densidad de puntos (figura A.9, derecha) se logra mejorar esta distancia máxima de identificación hasta los 40 - 50 metros, correspondiente con la distancia máxima definida por el ROI.

A.5. Conclusiones finales

La mayoría de los algoritmos que aportan soluciones a la percepción del entorno del vehículo autónomo se componen, a su vez, de algoritmos específicos para fases concretas. Una de estas fases es la segmentación de datos, fundamental en algoritmos para la detección de obstáculos o el posicionamiento visual. Por otro lado, muchas de estas soluciones de percepción se basan

en la tecnología LiDAR, la cual representa el entorno en forma de nubes de puntos. Para la segmentación de estas nubes de puntos existe una gran gama de algoritmos de clustering que funcionan correctamente con variables cuantitativas dispuestas en un espacio euclídeo. Sin embargo, estos algoritmos no están orientados a trabajar con nubes de puntos que han sido creadas desde un foco fijo.

En este sentido, se ha planteado una variante del algoritmo DBSCAN que compense los dos factores principales que es necesario considerar cuando se trabaja con sensores LiDAR 3D: la divergencia entre puntos y la variación de la densidad frente a la distancia.

El algoritmo DBSCAN ha demostrado una gran eficiencia y versatilidad de uso, como se ha podido observar a lo largo de la tesis. Sin embargo, encuentra grandes limitaciones cuando se emplea para la segmentación de nubes de puntos LiDAR. La variación de la divergencia y la densidad dan lugar a una reducción del alcance para una segmentación correcta debido a que se suele predefinir unos valores de ϵ y $minPts$ constantes.

La variante propuesta del algoritmo DBSCAN para su uso con láser rotativo compensa estas variaciones implementando un ϵ y $minPts$ variable. Para ello, se ha deducido analíticamente la correlación entre la divergencia y la densidad frente a la distancia. Relacionando la divergencia con el parámetro ϵ y la densidad con el parámetro $minPts$, se han definidos ambos con una variación lineal y cuadrática, respectivamente.

A partir de varios ensayos con datos reales, se ha verificado que esta propuesta mejora notablemente el rendimiento del algoritmo DBSCAN convencional. En concreto, en el caso de utilizar el modelo VLP-16 se puede lograr un alcance correcto de clustering de 10 - 15 metros, mientras que aplicando el método se logra alcanzar hasta 40 metros. Igualmente, en el caso del láser OS-1 64, donde la aplicación de un DBSCAN convencional puede lograr un clustering correcto a una distancia de 20 metros, con la propuesta se ha alcanzado distancias de hasta 50 metros para la identificación de objetos. Asimismo, estos ensayos han permitido encontrar experimentalmente qué coeficientes son los que optimizan el funcionamiento de clustering para dos modelos diferentes de LiDAR. Aunque debido a la importancia sobre la mejora, se podría encontrar una metodología analítica para calcular el valor óptimo de los parámetros de compensación definidos en la propuesta.

Esta mejora del alcance de segmentación puede dar lugar a mejoras significativas sobre las prestaciones generales de los algoritmos de percepción. Como se sabe, la etapa de segmentación de datos es fundamental en una gran cantidad de algoritmos de percepción ya sea para detección de obstáculos, caracterización o posicionamiento. En concreto, una mejora notable en el proceso de clustering como la planteada en este anexo, puede permitir detectar objetos a mayor distancia, lo que conlleva mayor anticipación ante posibles peligros. También puede posibilitar la extracción de un mayor número de elementos característicos (*landmarks*) en un algoritmo de SLAM, mejorando así el posicionamiento obtenido.