



**EDUCACIÓN**

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO

# Tecnológico Nacional de México

Centro Nacional de Investigación  
y Desarrollo Tecnológico

## Tesis de Maestría

Detección de obstáculos y planos durante el  
desplazamiento vehicular

presentada por

**Ing. Andrea Palacios Cervantes**

como requisito para la obtención del grado de  
**Maestra en Ciencias de la Computación**

Director de tesis  
**Dr. Dante Mújica Vargas**

Cuernavaca, Morelos, México. Enero 2023

Cuernavaca, Morelos, **17/enero/2023**

No. de Oficio: DCC/010/2023

Asunto: Aceptación de documento de tesis  
CENIDET-AC-004-M14-OFICIO

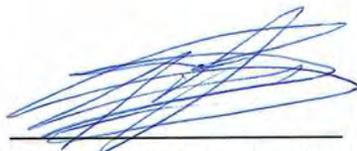
**JUAN GABRIEL GONZÁLEZ SERNA**  
JEFE DEL DEPARTAMENTO DE CIENCIAS COMPUTACIONALES  
PRESENTE

Por este conducto, los integrantes de Comité Tutorial de ANDREA PALACIOS CERVANTES, con número de control M20CE086, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis de grado titulado "DETECCIÓN DE OBSTÁCULOS Y PLANOS DURANTE EL DESPLAZAMIENTO VEHÍCULAR" y hemos encontrado que se han atendido todas las observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.



DANTE MÚJICA VARGAS

Director de tesis



NOÉ ALEJANDRO CASTRO SÁNCHEZ

Revisor 1



NIMROD GONZÁLEZ FRANCO

Revisor 2

c.c.p. Archivo



Cuernavaca, Mor.,  
No. De Oficio:  
Asunto:

**18/enero/2023**  
**SAC/023/2023**  
**Autorización de impresión de tesis**

**ANDREA PALACIOS CERVANTES**  
**CANDIDATA AL GRADO DE MAESTRA EN CIENCIAS DE LA COMPUTACIÓN**  
**PRESENTE**

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **“DETECCIÓN DE OBSTÁCULOS Y PLANOS DURANTE EL DESPLAZAMIENTO VEHICULAR”**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

**ATENTAMENTE**

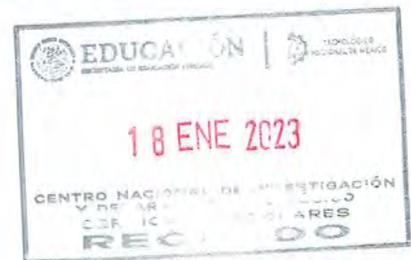
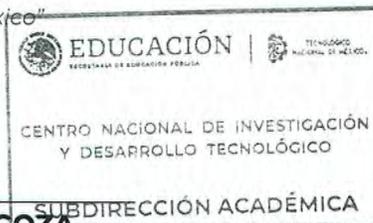
**Excelencia en Educación Tecnológica®**  
*“Conocimiento y tecnología al servicio de México”*



**CARLOS MANUEL ASTORGA ZARAGOZA**  
**SUBDIRECTOR ACADÉMICO**

C. c. p. Departamento de Ciencias Computacionales  
Departamento de Servicios Escolares

CMAZ/RMA



E32

## **Agradecimientos**

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo brindado durante mis estudios de maestría mediante su sistema de becas de posgrado. Al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), perteneciente al TECNM, por haberme permitido realizar los estudios de una Maestría en Ciencias en sus instalaciones.

Agradezco a mi director de tesis el Dr. Dante Mújica Vargas por su apoyo, esfuerzo, dedicación y guía brindada durante mi formación durante la maestría. A el comité revisor, Dr. Noé Alejandro Castro Sánchez y al Dr. Nimrod González Franco, por sus aportaciones y esfuerzo para el desarrollo de esta investigación.

## Dedicatoria

*Esta tesis se la dedico a mis padres Miguel Angel Palacios Reyes y Adriana Cervantes Badillo por todo su amor y apoyo durante esta etapa de mi vida.*

*A mis hermanos Daniel Palacios y Jimena Palacios que siempre estuvieron apoyándome y motivándome.*

*A mis amigos que me dieron siempre la motivación y apoyo durante este proceso de mi vida.*

*A mis compañeros y amigos dentro del CENIDET por todo su apoyo, consejos y asesorías que me motivaron durante el desarrollo de mi trabajo.*

# Resumen

La detección de objetos es una parte integral para la percepción del ambiente de un vehículo autónomo, ya que permite la planificación de maniobras de forma eficiente y segura, a través de sensores como LIDAR que realiza la visualización del escenario, gracias a su buen alcance y obtención de información por medio de una nube de puntos. Sin embargo, debido al ruido generado en el ambiente por virutas de polvo o gotas de lluvia la extracción de datos puede verse afectada, por lo que es recomendable combinar a LIDAR con otro sensor para disminuir la presencia de ruido.

Se utiliza el sensor RGB-D aprovechando su capacidad de trabajar en conjunto con LIDAR y disminuir el ruido para la obtención de una nube de puntos más optimizada. Para posteriormente analizar la información sensorial obtenida en tiempo real, segmentando los planos del suelo e identificando los obstáculos a través de las características obtenidas.

**Palabras clave: Detección de objetos, LIDAR, RGB-D, Conducción autónoma**

## Abstract

Object detection is an integral part for the perception the environment an autonomous vehicle, since it allows the planning of maneuvers efficiently and safely, through sensors such as LIDAR that performs the visualization the scenario, thanks to its good range and obtaining information through a point cloud. However, due to the noise generated in the environment by dust chips or raindrops, data extraction may be affected, so it is advisable to combine LIDAR with another sensor to reduce the presence of noise. The RGB-D sensor is used, taking advantage of its ability to work in conjunction with LIDAR and reduce noise to obtain a more optimized point cloud. To later analyze the sensory information obtained in real time, segmenting the ground planes and identifying obstacles through the features obtained.

**Keywords:** Object detection, LIDAR,RGB-D, Autonomous driving

# Índice General

Índice de figuras . . . . .	V
Índice de tablas . . . . .	VII
<b>1. Introducción</b>	<b>1</b>
1.1. Descripción del problema . . . . .	2
1.1.1. Delimitación del problema . . . . .	2
1.1.2. Complejidad del problema . . . . .	2
1.2. Objetivos . . . . .	2
1.2.1. Objetivo General . . . . .	2
1.2.2. Objetivos Específicos . . . . .	2
1.3. Alcances y Limitaciones . . . . .	3
1.3.1. Alcances . . . . .	3
1.3.2. Limitaciones . . . . .	3
1.4. Justificación . . . . .	3
1.5. Organización de la tesis . . . . .	4
<b>2. Marco Teórico</b>	<b>5</b>
2.1. Vehículo Autónomo . . . . .	5
2.2. Sensor LIDAR . . . . .	7
2.3. Sensor RGB-D . . . . .	10
2.4. Detección de objetos . . . . .	11
2.4.1. Agrupamiento . . . . .	12
2.4.2. Cuadro delimitador . . . . .	15
<b>3. Estado del arte</b>	<b>18</b>
3.1. Antecedentes . . . . .	18
3.2. Trabajos relacionados . . . . .	19
<b>4. Metodología</b>	<b>34</b>
4.1. Obtención de la nube de puntos . . . . .	35
4.2. Proceso de detección de objetos . . . . .	36
4.2.1. Procesamiento de la nube de puntos con RANSAC . . . . .	36
4.2.2. Agrupamiento de la nube de puntos con DBSCAN . . . . .	41
4.3. Construcción del cuadro delimitador . . . . .	44
4.4. Diagrama final del proceso de detección . . . . .	46

<b>5. Experimentación y resultados</b>	<b>48</b>
5.1. Diseño e implementación de Pruebas . . . . .	48
5.1.1. Entorno de desarrollo . . . . .	48
5.1.2. Métricas de evaluación . . . . .	52
5.2. Experimentaciones . . . . .	53
5.3. Resultados . . . . .	58
5.3.1. Resultados en zonas urbanas . . . . .	58
5.3.2. Resultados en autopistas . . . . .	62
5.3.3. Discusión de los resultados . . . . .	66
<b>6. Conclusiones</b>	<b>68</b>
6.1. Objetivos y Alcances logrados . . . . .	68
6.2. Resultados del trabajo de investigación . . . . .	68
6.2.1. Conclusiones . . . . .	69
6.2.2. Trabajos a futuros . . . . .	71
Referencias . . . . .	72

# Índice de Figuras

2.1.	Funcionamiento del sensor LIDAR. . . . .	7
2.2.	Nube de puntos 3D LIDAR (Liu et al., 2020). . . . .	8
2.3.	Estructura de la nube de puntos 3D LIDAR (Li et al., 2019). . . . .	10
2.4.	Funcionamiento de RGB-D. . . . .	10
2.5.	Estimación de la pose con RGB-D (Park et al., 2020). . . . .	11
2.6.	Ejemplo de la clasificación con el algoritmo de RANSAC (Leng et al., 2020). . . . .	12
2.7.	Ejemplo de agrupamiento de puntos con el algoritmo de DBSCAN. . . . .	13
2.8.	Representación grafica de los parámetros $\epsilon$ y $minPts$ . . . . .	14
2.9.	Tipos de puntos en el agrupamiento. . . . .	15
2.10.	Aplicación de la red <i>PointNet</i> (Qi et al., 2017) . . . . .	16
2.11.	Estructura de la <i>PointNet</i> (Qi et al., 2017). . . . .	17
3.1.	Resultados obtenidos durante la simulación (Li et al., 2019). . . . .	19
3.2.	Etapas del proceso de detección y evasión de obstáculos (Chávez Cobián, 2020). . . . .	20
3.3.	Resultados de la reconstrucción. (Ponce Herrera, 2019). . . . .	21
3.4.	Nubes de puntos de objetos con puntos clave agrupados en descriptores locales propuesto (Egi and Otero, 2019). . . . .	23
3.5.	Arquitectura del sistema del algoritmo propuesto de reconocimiento de objetos y estimación de pose (Tsai and Tsai, 2018). . . . .	24
3.6.	Modulo de sistema propuesto (Mu et al., 2020). . . . .	25
3.7.	Principales desafíos de la detección de objetos 3D en nubes de puntos con objetos incompletos (Gong et al., 2020). . . . .	28
4.1.	Proceso de detección de objetos a través de una nube de puntos. . . . .	34
4.2.	Trabajo en conjunto de los sensores LIDAR y RGB-D. . . . .	35
4.3.	Auto Ego con el sensor RGB-D y LiDAR en Matlab. . . . .	35
4.4.	Diagrama de bloques de RANSAC. . . . .	37
4.5.	Primer paso del algoritmo RANSAC. . . . .	37
4.6.	Ajuste del modelo de un subconjunto con RANSAC (Lopez Torrijos, 2021). . . . .	38
4.7.	Ajuste del modelo de un subconjunto con RANSAC (Lopez Torrijos, 2021). . . . .	39
4.8.	Funcionamiento de RANSAC (Lopez Torrijos, 2021) . . . . .	40
4.9.	Diagrama de bloques del Algoritmo de DBSCAN. . . . .	41
4.10.	Visualización de la identificación de los puntos centrales y el trazado de $\epsilon$ . . . . .	42
4.11.	Visualización de la asignación de puntos a un grupo. . . . .	42
4.12.	Visualización del agrupamiento final. . . . .	43

4.13. Diagrama de bloques de RANSAC. . . . .	44
4.14. Visualización del criterio de construcción por cercanía. . . . .	44
4.15. Visualización del criterio de construcción por área. . . . .	45
4.16. Pruebas con <i>PointNet</i> . . . . .	46
4.17. Diagrama de bloques del Proceso de detección. . . . .	47
5.1. Zonas urbanas utilizadas para realizar las simulaciones. . . . .	50
5.2. Autopistas utilizadas para realizar las simulaciones. . . . .	51
5.3. Experimentaciones realizadas en 5 pistas de zonas urbanas. . . . .	54
5.4. Proceso de detección durante las simulaciones. . . . .	55
5.5. Simulaciones realizadas en 5 pistas de vías rápidas. . . . .	56
5.6. Experimentaciones realizadas en 5 pistas de zonas urbanas. . . . .	57
5.7. Datos falsos en zonas urbanas. . . . .	58
5.8. Datos perdidos en zonas urbanas. . . . .	59
5.9. Cantidad de objetos detectados en vías urbanas. . . . .	60
5.10. Cantidad de objetos detectados en vías urbanas-continuación. . . . .	61
5.11. Cantidad de objetos detectados en vías urbanas-continuación. . . . .	62
5.12. Datos falsos autopistas. . . . .	62
5.13. Datos perdidos en vías rápidas. . . . .	63
5.14. Cantidad de objetos detectados en vías rápidas. . . . .	63
5.15. Cantidad de objetos detectados-continuación. . . . .	64
5.16. Cantidad de objetos detectados. . . . .	65
6.1. Reconocimiento del artículo presentado en la 6a. Jornada de Ciencia y Tecnología Aplicada. . . . .	76
6.2. Reconocimiento del artículo presentado en la 7a. Jornada de Ciencia y Tecnología Aplicada. . . . .	77
6.3. Aceptación de la propuesta de Maestría. . . . .	78

# Índice de Tablas

2.1. Niveles de asistencia de un vehículo autónomo al conductor (NHTSA, 2019).	6
2.2. Comparación de sensores de detección (Arya Senna Abdul Rachman, 2017).	9
3.1. Resumen de artículos. . . . .	30
5.1. Escenarios de vías urbanas ubicadas en diferentes puntos de la República Mexicana . . . . .	49
5.2. Escenarios de vías rápidas ubicados en diferentes puntos de la República Mexicana . . . . .	49
5.3. Características de los objetos detectados durante las simulaciones. . . . .	58
5.4. <i>Precision y Recall</i> de pistas urbanas. . . . .	59
5.5. <i>Precision y Recall</i> de autopistas. . . . .	66
6.1. Objetivos Realizados . . . . .	69

# Capítulo 1

## Introducción

Los vehículos autónomos buscan mejorar la experiencia de conducción, permitiendo que un automóvil sea totalmente independiente del ser humano y pueda desplazarse de forma segura en diferentes espacios, para lograr su objetivo se aplican tecnologías que apoyan a las tareas de conducción, por medio de la localización, detección de obstáculos y planos.

La identificación y el seguimiento de objetos es una parte integral de la percepción del vehículo, ya que permite estimar las trayectorias de los objetos y planificar el movimiento que realice el automóvil. Los sensores que normalmente se utilizan en la investigación para las tareas de detección son LIDAR y RGB-D, ya que proporcionan abundantes datos del ambiente alrededor del vehículo con un gran alcance en tiempo real.

No obstante, los sensores y algoritmos que se aplican tienen sus ventajas y desventajas, uno de los problemas destacables en la detección de obstáculos son su falta de capacidad en obtener datos detallados durante el desplazamiento vehicular, lo que causa un bajo rendimiento en la detección y seguimiento de los objetos. En consecuencia, aún cuando ya son existentes los automóviles que sean capaces de realizar las tareas de conducción de manera autónoma, se necesita de la asistencia de un conductor que esté al pendiente de tomar el control del automóvil en caso de que el sistema o el usuario lo considere necesario. Por este motivo, la investigación sobre este tema es amplia y constante, pues siempre se busca el trabajar y solucionar problemas en la detección de objetos, para que la tecnología aplicada sea algún día lo suficientemente madura y sofisticada para uso general en vía pública. Para el desarrollo de esta tesis, el objetivo principal es el implementar un proceso de detección de objetos durante desplazamiento vehicular, haciendo uso de los sensores LIDAR y RGB-D. Para después realizar el procesamiento de la nube de puntos y finalmente obtener la información de las dimensiones de un objeto al ser detectado durante el desplazamiento del vehículo.

## **1.1. Descripción del problema**

El vehículo autónomo durante su desplazamiento en vía urbana es más propenso a presentar errores durante la detección de objetos tales como son, peatones, motociclistas u otros vehículos o en casos más extremos objetos ajenos al camino, provocando maniobras erróneas de conducción y fallas en el movimiento, por lo tanto es necesario estudiar y corregir el conocimiento de la distancia, tamaño y orientación en la que se encuentra los objetos alrededor de su ambiente.

### **Hipótesis**

Se puede generar un plano en tiempo real con el uso de los sensores LIDAR y RGB-D e implementar un algoritmo que le permita detectar objetos, mientras el vehículo está en movimiento.

#### **1.1.1. Delimitación del problema**

En esta investigación se desarrollará un proceso de detección y ubicación en el espacio de obstáculos y planos en tiempo real durante el desplazamiento vehicular en un entorno simulado.

#### **1.1.2. Complejidad del problema**

La complejidad del problema se ubica en el análisis y evaluación de las dimensiones físicas de los obstáculos. En consecuencia, se planea aportar en la tarea de detección de objetos utilizando los sensores LIDAR y RGB-D, aplicando algoritmos de agrupamiento.

## **1.2. Objetivos**

### **1.2.1. Objetivo General**

Proponer un proceso para detectar obstáculos y planos en tiempo real, que pueda ser utilizado durante el desplazamiento vehicular.

### **1.2.2. Objetivos Específicos**

- Procesar la información sensorial en tiempo real a fin de segmentar los planos del suelo y obstáculos cercanos detectados a nivel simulación.
- Utilizar técnicas de visión por computadora y técnicas convencionales del estado del arte.

- Realizar la caracterización de los sensores a nivel simulación, para que haya una correspondencia con las medidas de los vehículos y objetos detectados.
- Implementar el proceso de detección en una simulación basada en un entorno real.

## **1.3. Alcances y Limitaciones**

### **1.3.1. Alcances**

- Identificar los requisitos para la detección de objetos.
- Describir de la dimensión del objeto.
- Realizar la implementación y ajuste en tiempo real de la detección de objetos en una simulación basada en un entorno real utilizando los sensores LIDAR y RGB-D.
- Realizar la evaluación basada en métricas establecidas en el estado del arte para los datos obtenidos, con el fin de obtener validaciones.

### **1.3.2. Limitaciones**

- No se consideran factores ajenos como el clima o la contaminación que dificulten la detección de objetos en situación urbana.
- Se limita el uso de los sensores LIDAR y RGB-D para la detección de objetos dentro de esta investigación.
- El alcance de percepción se limita al alcance de los sensores LIDAR y RGB-D.
- No se consideran afectaciones u obstrucciones que afecten la detección de objetos.

## **1.4. Justificación**

Para la detección de objetos por medio de sensores 3D LIDAR, existen diferentes modelos y algoritmos que buscan perfeccionar la percepción de las dimensiones y espacios de los objetos que se detectan. Por esta razón, se seleccionará uno de los modelos analizados, para desarrollar un proceso que realice las funciones de detección y análisis de planos u objetos, durante el desplazamiento de vehículos autónomos.

## 1.5. Organización de la tesis

La presente tesis se organiza de la siguiente forma:

En el Capítulo 2 se describen los conceptos teóricos que son necesarios para comprender el proceso y desarrollo de detección durante el desplazamiento vehicular, junto a los componentes que se encuentran involucrados.

En el Capítulo 3 se presenta un análisis de los trabajos más recientes relacionados con el proceso de detección en vehículos autónomo, el uso y calibración de sensores LIDAR y RGB-D, algoritmos de agrupamiento, fusión de datos y técnicas de visión por computadora.

En el Capítulo 4 se describe a detalle el proceso de detección propuesto que involucra la obtención de la nube de puntos, procesamiento de la nube de puntos y la construcción del cuadro delimitador.

En el Capítulo 5 se explican los experimentos realizados y resultados obtenidos a partir de los conceptos y métricas aplicadas.

Finalmente en el Capítulo 6, se presentan las conclusiones obtenidas, recomendaciones para trabajos a futuros, así como un análisis de los objetivos completados y los alcances planteados en esta investigación.

# Capítulo 2

## Marco Teórico

En este capítulo se detallan diversos conceptos básicos y específicos que son necesarios para el correcto entendimiento y desarrollo del tema de investigación propuesto.

### 2.1. Vehículo Autónomo

Los vehículos autónomos pertenecen a los sistemas inteligentes de transporte que tienen el potencial de revolucionar la conducción por medio de la automatización en funciones de seguridad, eficiencia, experiencia de usuario y la percepción del medio que lo rodea (McQueen, 2019). La automatización de los vehículos según la *National Highway Traffic Safety Administration*, se clasifica en 6 niveles como se describe en la Tabla 2.1 (NHTSA, 2019).

Para el desarrollo de estos vehículos terrestres urbanos es común que cuenten con 3 elementos (Li et al., 2018):

- **Sensores:** Estos dispositivos se encargan de la detección y transformación de magnitudes físicas o químicas en señales o pulsos electrónicos. Un ejemplo son las cámaras que se utilizan para detectar y estudiar el entorno que los rodea.
- **Sistema de computo embebido:** Con el avance de la inteligencia artificial, los sistemas embebidos en los automóviles permiten la aplicación de métodos de aprendizaje autónomo, que permite la generación de algoritmos que se aplican para ayudar al usuario con las tareas de conducción y localización.
- **Actuadores:** Su función es el proporcionar fuerza para mover otro dispositivo mecánico, convirtiendo señales eléctricas desde la unidad de control en una acción controlada. Los actuadores son una parte esencial en la electromecánica y en su mayoría son implementados en motores eléctricos y motor-reductores o válvulas electromagnéticas que regulan los sistemas de frenado y dirección.

Tabla 2.1: Niveles de asistencia de un vehículo autónomo al conductor (NHTSA, 2019).

Nivel	Concepto	Descripción
1	Asistencia del conductor	En este nivel ya existe algo de automatización. Esta puede venir en forma de que el vehículo toma control del volante o dirección, de los pedales, pero nunca de los dos a la vez.
2	Automatización parcial	En este nivel el sistema brinda asistencia continua con aceleración, frenado y dirección mientras el conductor permanece totalmente comprometido y atento.
3	Automatización de conducción condicional	En esta categoría entran en funcionamiento la velocidad constante, el asistente de conducción, tecnologías como el aparcamiento automatizado o el mantenimiento en un carril a una determinada velocidad, mientras el conductor permanece disponible para hacerse cargo de conducir.
4	Automatización de conducción alta	El automóvil maneja y conduce el coche casi por completo, pero no en todos los entornos ya que se exceptúa la climatología adversa como niebla, nieve, tormentas, etcétera. En cuanto sea posible que el coche pueda volver a coger el control el conductor debe activar la conducción automática.
5	Automatización de conducción completa	Este sistema como dice el nombre es el último nivel y aquí el conductor es poco menos que un mero espectador ya que el coche hace todo el trabajo. El sistema es independiente y se pueden prescindir de los elementos de control como el volante y los pedales.

Estos elementos ayudan a que el automóvil urbano pueda realizar las tareas de conducción de forma eficiente y segura para el conductor, y aunque los avances siguen siendo extraordinarios aun continúan existiendo detalles donde el usuario debe estar alerta para tomar el control del automóvil en caso de que el sistema se lo solicite.

Actualmente diversas compañías que apoyan el desarrollo de estos vehículos autónomos, se encuentran invirtiendo millones de dólares en empresas como Tesla Motors, Mercedes Benz y Uber Technologies Inc, las cuales su principal objetivo es ofrecer un servicio de transporte. Mientras que Toyota, Jaguar, BMW, Nissan y Renault, Audi, se encuentran mucho más comprometidos en desarrollar automóviles de uso común, taxis, carros de policías e incluso autobuses de transporte urbano (Li et al., 2018).

## 2.2. Sensor LIDAR

La percepción inteligente y la adquisición de información de objetos en 3D (posición, textura, pose y tamaño) es una guía estratégica adicional para los dispositivos inteligentes. Sin embargo para la detección de objetos 3D se presentan 3 principales desafíos los cuales son (Gong et al., 2020):

- Presencia de oclusiones: La mayoría de métodos que están basados en características 3D asumen que las nubes de puntos se encuentran incompletas debido a las oclusiones generadas por obstáculos que se encuentran en su entorno.
- Ambigüedad estructural: La ambigüedad se refiere a los objetos que llegan a tener las mismas características estructurales que causan ruido e inconsistencias.
- Métodos utilizados en la detección de objetos: Para la detección de objetos 3D, es necesario de conjuntos de entrenamiento, como son las redes neuronales profundas que necesitan muchas veces de un re-entrenamiento, por lo que en las aplicaciones prácticas el requisito de grandes conjuntos de datos de entrenamiento es un cuello de botella para los métodos basados en aprendizaje profundo.

Los sistemas de medición y detección de objetos mediante láser, más conocidos como LIDAR, son sensores que emiten rayos de luz láser infrarroja. Su funcionamiento consiste en un foco emisor de rayos láser y de un lente receptor infrarrojo capaz de percibir estos láseres, los rayos que se me emiten llegan a impactar sobre los objetos causando que reboten y sean captados por la lente, obteniendo de esta forma una nube de puntos como se aprecia en la Figura 2.1. La nube de puntos se procesará a una imagen tridimensional en tiempo real, que se actualiza mientras los objetos se desplazan.



Figura 2.1: Funcionamiento del sensor LIDAR.

Los sensores LIDAR son utilizados para el mapeo, localización y evasión de obstáculos por medio de una percepción inteligente de objetos 3D. La información que el sensor detecta es a través de una nube de puntos que tiene como principal tarea el detectar la curva de un objeto (Gong et al., 2020).

Las nubes de puntos generados por LIDAR proporcionan una descripción del entorno escaneando, pero es muy difícil diferenciar los puntos individuales, es por esta razón que los mapas producidos con LIDAR de un vehículo en movimiento como se observa en la Figura 2.2, se les aplica un método de filtrado de partículas para correlacionar la información y lograr una localización con 10 cm de precisión con mayor eficacia en entornos urbanos (Liu et al., 2020).



Figura 2.2: Nube de puntos 3D LIDAR (Liu et al., 2020).

A pesar de sus grandes, beneficios el sensor no esta exento de presentar algún defecto, sin embargo la elección de los sensores LIDAR como un sensor de detección se debe a la necesidad de encontrar un escaneo 3D envolvente, es por este motivo que al compararlo con otros sensores, se puede apreciar los beneficios de 3D LIDAR que se demuestra en la Tabla 2.2.

Tabla 2.2: Comparación de sensores de detección (Arya Senna Abdul Rachman, 2017).

Sensores Detección	Principio de operación	Complejidad computacional	Características
Cámara estéreo	Usa múltiples cámaras para generar una visión	Baja	Buen campo de visión susceptible al cambio de iluminación.
Cámara RGB-D	Detecta el ambiente por medio de la profundidad, práctica precisión hasta 50 m	Media	Alta resolución, alto campo de visión vertical, la vista es envolvente.
LIDAR 3D	Mide la distancia reflectancia de 600 a 1000 nm láser singal	Alta	Alta resolución, muy largo alcance (hasta 200 m),no susceptible a la iluminación cambio de nación, vista envolvente usando sistema de sensor de montaje único.

## Estructura de la nube de puntos

El uso del sensor LIDAR proporciona una visión de una colección de puntos multidimensionales que representan las superficies físicas de un objeto y se caracteriza por coordenadas espaciales  $XYZ$  y que opcionalmente se les puede asignar atributos adicionales, un ejemplo de como se visualiza esta estructura se aprecia en la Figura 2.3.

Matemáticamente un punto es simplemente una tupla incorporada con varios atributos y se representa como  $(p_i = x_i, y_i, z_i, I_i \dots)$ . Donde  $x_i, y_i, z_i, I_i$  representan las coordenadas espaciales en longitud, altura y ancho de cada atributo. La colección de puntos en una nube de puntos se representa en  $P = p_1, p_2, p_3, \dots, p_n$  (Li et al., 2019).

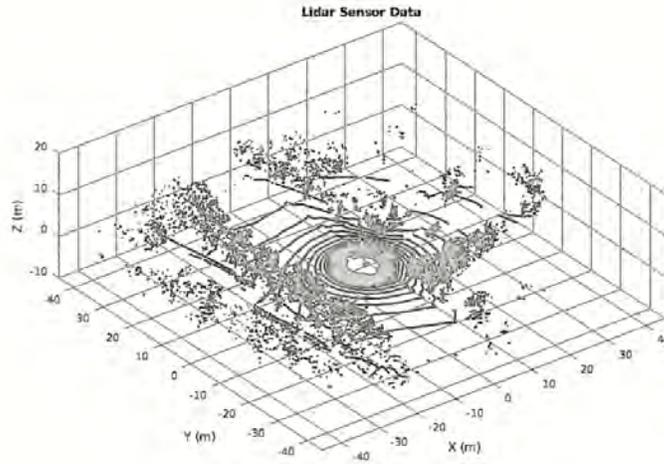


Figura 2.3: Estructura de la nube de puntos 3D LIDAR (Li et al., 2019).

### 2.3. Sensor RGB-D

Los sensores RGB-D son un tipo de dispositivo de detección de profundidad que funciona en asociación con los colores rojo, verde y azul. Su funcionamiento como se visualiza en la Figura 2.4, consiste en aumentar la imagen convencional con información de profundidad por píxel usando una cámara de color y un emisor infrarrojo que detecta la luz reflejada. La información de profundidad proporciona una contribución importante para la detección de objetos, el análisis de escenas, estimación de pose, seguimiento visual, segmentación semántica, análisis de formas y reconstrucción en 3D (Zhang et al., 2019).

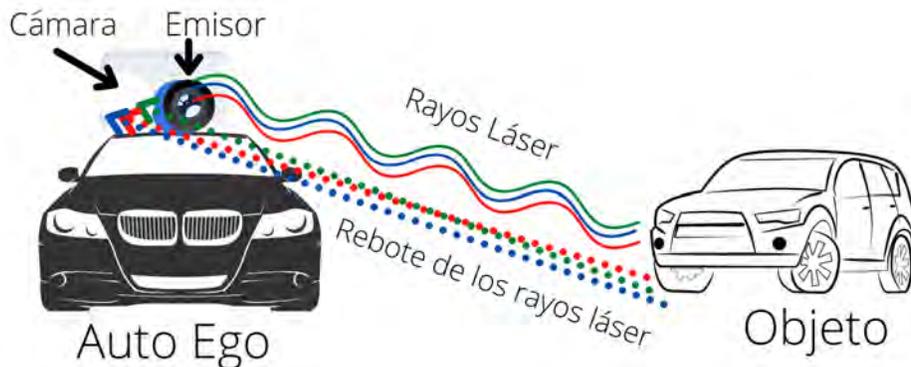


Figura 2.4: Funcionamiento de RGB-D.

Las detecciones que se realicen por medio de RGB-D a partir de las posiciones y velocidad en la que se encuentre el objeto cercano al vehículo Ego como se representa en la Figura 2.5, se hará un calculo de ángulo de estimación basado en el movimiento utilizando (2.1) y (2.2) (Park et al., 2020).

$$(X_t, Y_t) = (X_{ego}, Y_{ego}) + \mathbf{R}(\Psi_{ego}) \cdot (x_t, y_t)^T \quad (2.1)$$

$$\Psi_{target} = \tan^{-1}\left(\frac{Y_t - Y_{t-1}}{X_t - X_{t-1}}\right) \quad (2.2)$$

donde  $\mathbf{R}$  es la rotación de la matriz,  $\Psi_{ego}$  es el ángulo del vehículo ego,  $(X_t, Y_t)$  son las coordenadas del vehículo ego  $(X_{ego}, Y_{ego})$  son las coordenadas mundiales y  $(x_t, y_t)^T$  coordenadas de los objetos,  $\Psi_{target}$  es el ángulo del objeto detectado y  $(X_t, Y_t)^T$  coordenadas de los objetos.

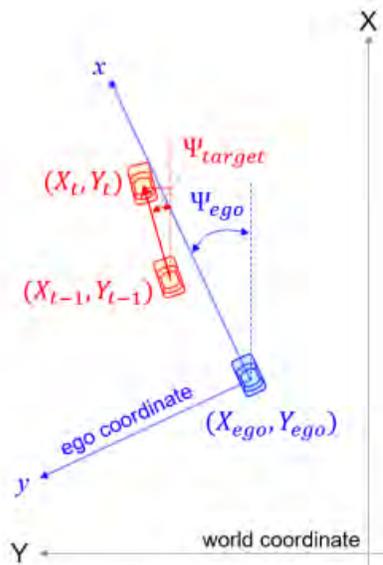


Figura 2.5: Estimación de la pose con RGB-D (Park et al., 2020).

## 2.4. Detección de objetos

La detección de objetos normalmente comienza con el proceso de las imágenes de entrada, seguido de la detección de las regiones de interés y finalmente un clasificador que identifique los objetos detectados. En los objetos 3D se proporciona cuadros delimitadores para cada fotograma como medidas para los sistemas de seguimiento de objetos múltiples en 3D (Chiu et al., 2020). Por lo tanto, la calidad del detector de objetos 3D es esencial para la precisión del seguimiento real, dado que su capacidad como sensor contiene una gran cantidad de datos sin procesar por lo que es necesario hacer una previa extracción de datos de interés como pueden ser los ciclistas, peatones, autos, entre otros objetos.

La nube de puntos que pertenece a los obstáculos se clasifican en grupos, y cada grupo se convierte en una detección mediante un cuadro delimitador, con el siguiente formato (MATLAB, 2020):

$$(x, y, z, l, w, h) \quad (2.3)$$

donde  $x, y, z$  se refieren a las coordenadas espaciales del cuadro delimitador.  $l, w, h$  se refiere a la longitud, anchura y altura del mismo. Es por esto que el detector debe abarcar las funcionalidades de agrupación y segmentación de la nube de puntos.

### 2.4.1. Agrupamiento

El agrupamiento es el proceso de reunir datos en clases donde los objetos de un grupo tengan una similitud alta entre ellos. La medida de semejanza se define normalmente por proximidad en un espacio multidimensional que pasa por un proceso de estandarización (Eduardo Morales, 2018).

## RANSAC

*Random Sample Consensus* (RANSAC) es un algoritmo iterativo que calcula los parámetros de un modelo de regresión lineal, funciona como un identificador de valores atípicos en un conjunto de datos y estimando el modelo deseado, un ejemplo de su funcionamiento se observa en la Figura 2.6 (Leng et al., 2020).

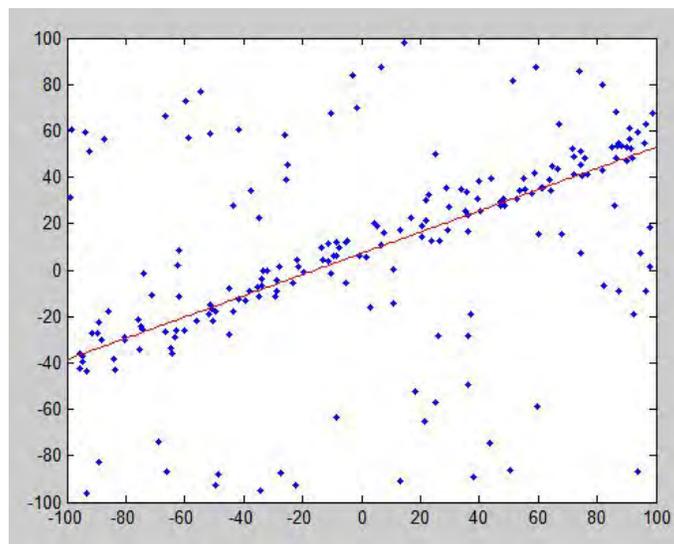


Figura 2.6: Ejemplo de la clasificación con el algoritmo de RANSAC (Leng et al., 2020).

Una de las mayores ventajas de RANSAC es su capacidad para la estimación robusta de los parámetros de un modelo, es decir, puede estimar el modelo a pesar de existir una gran cantidad de números atípicos (*Lopez Torrijos, 2021*).

Para realizar su agrupamiento aplicando este algoritmo se siguen los siguientes pasos (Leng et al., 2020):

1. Selecciona aleatoriamente un subconjunto de datos.
2. Ajustar un modelo al subconjunto seleccionado.
3. Determinar el número de valores atípicos a partir de la distancia.
4. Se repiten los pasos 1-3 hasta cumplir con un número prescrito de interacciones.

## DBSCAN

Es un algoritmo de agrupamiento basado en densidad que se basa en la muestra de decisión de proximidad. Encuentra todas las regiones densas de los puntos de muestra y trata estas regiones densas como grupo, como se aprecia en la Figura 2.7, donde cada grupo esta siendo identificado por un color diferente (Wang et al., 2019).

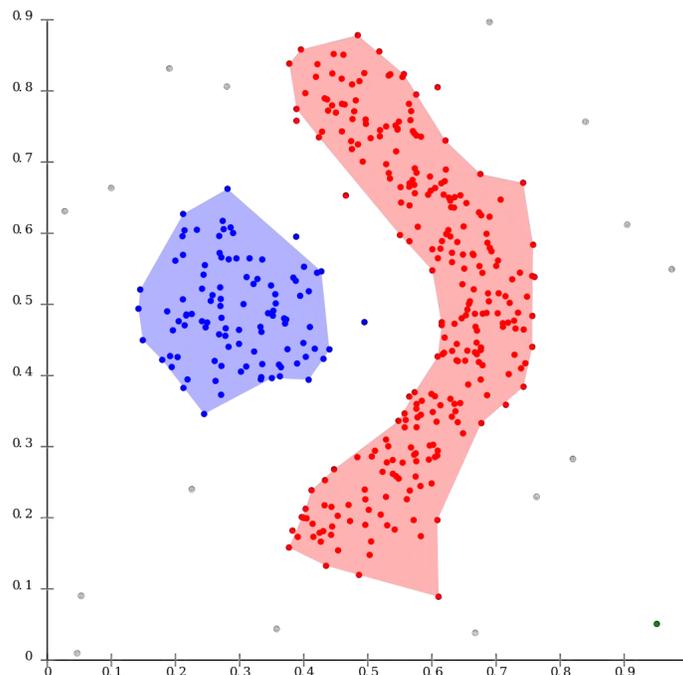


Figura 2.7: Ejemplo de agrupamiento de puntos con el algoritmo de DBSCAN.

El algoritmo tiene las siguientes características:

- Se basa en una densidad robusta a puntos de ruido alejados del núcleo de densidad.
- No es necesario agregar como parámetro el número de grupos.
- Se puede encontrar agrupaciones de forma arbitraria.

DBSCAN realiza sus observaciones a partir de un radio de proximidad y para su funcionamiento necesita de dos factores como se puede ver en la Figura 2.8, los cuales son (Li, 2020):

- *Epsilon* ( $\epsilon$ ): También llamado  $\epsilon$ -neighborhood, es el radio que define la región vecina a una observación.
- Puntos mínimos (*minPts*): Número mínimo de observaciones dentro de la región de  $\epsilon$ .

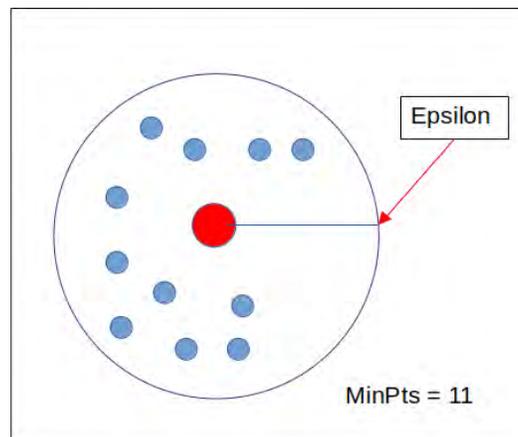


Figura 2.8: Representación grafica de los parámetros  $\epsilon$  y *minPts*.

Durante el agrupamiento DBSCAN define 3 diferentes puntos y que se aprecian en la Figura 2.9 (Li, 2020):

- Punto central: Contiene una gran cantidad de *minPts* dentro de  $\epsilon$ .
- Punto Fronterizo: La cantidad de *minPts* es menor dentro de  $\epsilon$ , pero es vecino de un punto central.
- Valor atípico: Un punto que no es central, ni fronterizo.

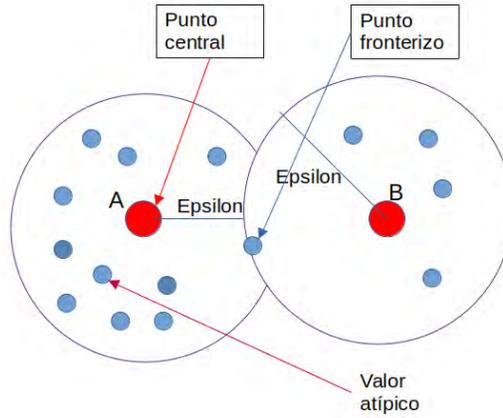


Figura 2.9: Tipos de puntos en el agrupamiento.

Tomando en cuenta los puntos anteriormente mencionados, DBSCAN realiza su agrupamiento siguiendo los siguientes pasos (Schubert et al., 2017):

- Encuentra todos los puntos vecinos dentro de ( $\epsilon$ ) e identifica los puntos centrales.
- Para cada punto central, si no hay una asignación se crea un nuevo grupo.
- Todos los puntos son conectados por su densidad y asignados al mismo grupo que el punto central. Un punto  $a$  y  $b$  se dice que son de densidad conectada si existe un punto  $c$  que tiene un número suficiente de puntos con sus vecinos  $b$  y que están dentro de  $\epsilon$ . Es un proceso de encadenamiento.
- Aquellos puntos que no pertenecen a ningún agrupamiento son considerados como ruido.

### 2.4.2. Cuadro delimitador

El cuadro delimitador es factible para la nube de puntos pues incluye explícitamente la geometría 3D que garantiza una alta objetividad de segmentos que son aprendidos durante la detección de objetos y se ajusta para tener información dimensional uniforme, aplicando un rectángulo (Kocur and Ftáčnik, 2020).

Para optimizar la construcción de cuadros delimitadores se utiliza el siguiente criterio de construcción (Zhang et al., 2017).

$$\min \sum_{i \in P} (x_i \cos \theta + y_i \sin \theta - c_1)^2 + \sum_{i \in Q} (-x_i \sin \theta + y_i \cos \theta - c_2)^2 \quad (2.4)$$

Minimizando el error del cuadro, ajustando la disyunción óptima ( $P, Q$  divide los  $m$  puntos ( $x_i | i = 1, \dots$ ) en dos conjuntos) y los parámetros óptimos ( $\theta, c_1, c_2$ ) para las dos rectas perpendiculares que corresponden a los puntos en  $P$  y  $Q$  respectivamente, las dos expresiones de recta son  $x \cos \theta + y \sin \theta = c_1$  y  $x \sin \theta + y \cos \theta = c_2$ .

### ***PointNet***

*PointNet*, es una red neuronal profunda que aprende características de puntos globales y locales proporcionando un enfoque simple, eficiente y efectivo para el reconocimiento 3D. Su modelo consta de dos principales componentes, el primero es un codificador para los datos de una nube de puntos dispersa y el segundo es un clasificador que predice la categoría de la nube de puntos que fue codificada. En la figura 2.10 se representan la aplicación que tendrá esta red dentro de la nubes de puntos en donde el principal objetivo es la clasificación e identificación de los objetos detectados (Qi et al., 2017).

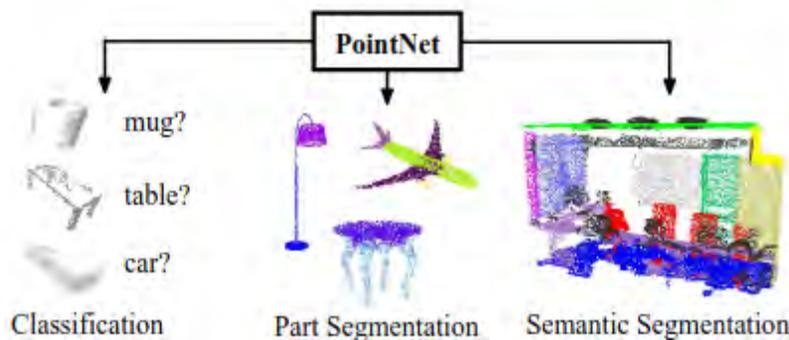


Figura 2.10: Aplicación de la red *PointNet* (Qi et al., 2017) .

La red *PointNet* está destinado a consumir un conjunto desordenado de coordenadas como datos de entrada, por lo que su arquitectura debe coincidir con las siguientes propiedades para la nube de puntos (Xie et al., 2021):

- Invarianza de permutación: Para hacer que la red sea invariante para las permutaciones de entrada se usa una función simétrica, como la agrupación máxima con los puntos de entrada que se asignan a un espacio de mayor dimensión.
- Invariancia de transformación: Los resultados de la segmentación no deben modificarse si el objeto sufre determinadas transformaciones, como la traducción o el escalado. Para una nube de puntos de entrada dada, se aplica una transformación rígida o afín apropiada para lograr la normalización de pose.
- Interacciones puntuales: La interacción entre puntos vecinos a menudo conlleva información útil, es decir, un solo punto no debe tratarse de

forma aislada. Mientras que la clasificación solo necesita hacer uso de características globales, la segmentación debe poder aprovechar las características de puntos locales junto con las características de puntos globales.

Al visualizar la estructura de la red neuronal *PointNet* en la Figura 2.11, se puede apreciar los tres módulos principales de esta red que consisten en:

- Primer módulo: Realiza la agrupación máxima que agrega la información de una nube de puntos de entrada.
- Segundo módulo: Ejecuta la combinación de información local y global.
- Tercer módulo: Alinean los puntos de entrada y las entidades de puntos.

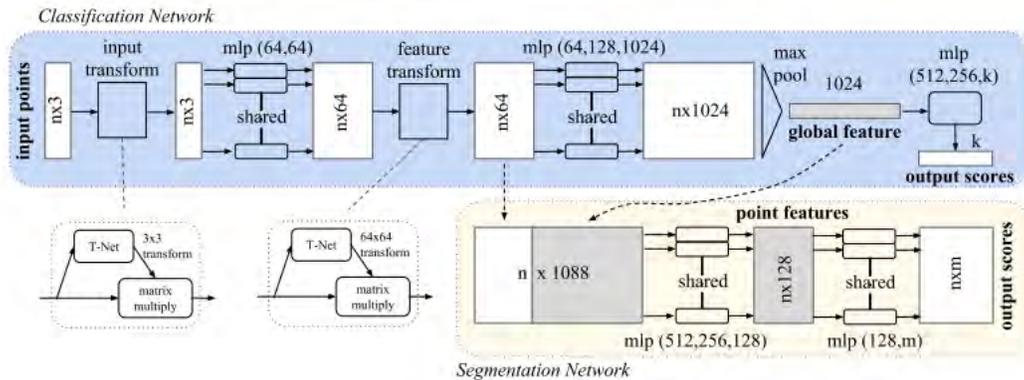


Figura 2.11: Estructura de la *PointNet* (Qi et al., 2017).

## Discusión

En este capítulo se presentó un panorama general sobre los temas básicos como el concepto de los vehículos autónomos, el cual tiene el objetivo de comprender su finalidad de aplicar en ellos el proceso de detección, sí como también el conocer los componentes que participan en el mismo. También se mencionan los conceptos y funcionamientos de los sensores LIDAR y RGB-D, al igual que los algoritmos RANSAC y DBSCAN, los cuales son fundamentales para el entendimiento en el momento de la obtención y procesamiento de la información e identificación de cada objeto durante el proceso de rastreo durante el desplazamiento vehicular. Cada concepto mencionado dentro del marco teórico serán de utilidad para desarrollar la metodología y experimentaciones dentro de esta tesis, así como también a dar una referencia inicial a la búsqueda de artículos y proyectos que serán explicados en el siguiente capítulo.

# Capítulo 3

## Estado del arte

En este capítulo se abordan los trabajos de investigación más relevantes y relacionados con el tema de tesis, que nos apoyaran a realizar la selección de algoritmos, búsqueda de métricas y apoyo de métodos para el procesamiento de la nube de puntos.

### 3.1. Antecedentes

Los trabajos mostrados a continuación fueron desarrollados en el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) y abordan temas de interés sobre conducción autónoma.

#### **Sistema Embebido para Asistencia de Conducción Basado en Lógica Difusa Tipo-2 (*Rendón Castro, 2020*)**

En este trabajo de tesis se presentó el diseño y desarrollo de un asistente de conducción basado en Lógica Difusa Tipo 2. El sistema propuesto adapta métricas a tres experimentos que evaluaron la detección de un vehículo frontal, detección de nivel de riesgo por intención en el camino y detección de nivel de riesgo de vehículo frontal en conjunto de la intención. Para la experimentación de lo anteriormente propuesto se realizó con ayuda de datos *Yano DD*, *UTA Real-life Drowsiness* y *TME Motorway*. Los resultados que se obtuvieron fueron de estructura cualitativa y cuantitativa, mostrando que el enfoque que se había propuesto es eficiente para la detección de riesgo, pues cuenta con un rendimiento mayor a 0.90 en las métricas de evaluación. Este trabajo se tomo como referencia para las métricas de evaluación de la detección de objetos.

## Conducción Autónoma de un Vehículo Simulado mediante un Modelo de Red Neuronal Convolutiva Recurrente (*Luna Álvarez, 2020*)

En este trabajo de tesis se propuso un modelo de Red Neuronal Híbrido compuesto por capas de Convolución recurrentes para realizar el control de un vehículo simulado en el Sistema Operativo Robótico. La propuesta fue hecha con base para demostrar la mejoría del modelo, analizando la evaluación en la tarea de clasificación y autonomía del vehículo. Fueron realizados entrenamientos usando una base de datos y sensores para adaptar la estructura tensorial que reciba cada modelo con datos que fueron diseñados para cuatro escenarios diferentes. Los resultados que se obtuvieron otorgaron una mayor autonomía al vehículo en escenarios libres. Este trabajo se tomó como referencia, para las tareas de clasificación y aplicación de redes neuronales al proceso de detección.

### 3.2. Trabajos relacionados

*AADS: Augmented autonomous driving simulation using data-driven algorithms* (Li et al., 2019)

En el mencionado artículo se presentó una simulación de conducción autónoma aumentada, por medio de una formulación de aumento en las imágenes del mundo real con un flujo de tráfico simulado para crear imágenes y representaciones de simulación fotorrealistas. El enfoque de esta investigación fue dar una flexibilidad en un entorno virtual con la riqueza de una mejora en la simulación efectiva del mundo real, como se puede apreciar en la Figura 3.1.

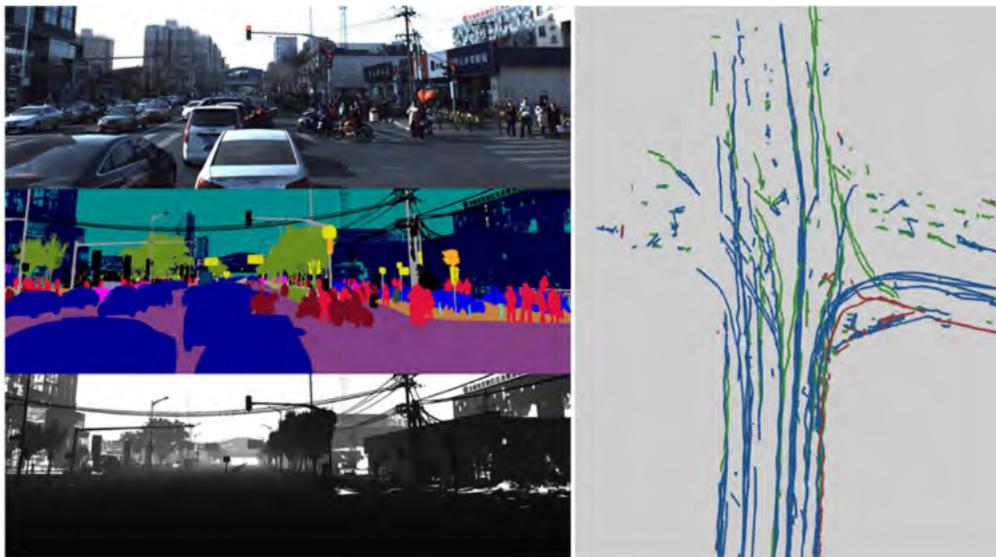


Figura 3.1: Resultados obtenidos durante la simulación (Li et al., 2019).

### Detección basada en datos 3D de objetos urbanos (Castaño Amorós, 2020)

El nombrado trabajo se centró en mejorar un problema con la detección de objetos al no diferenciar entre un vehículo o una persona, dando falsos positivos, así que en base a los sensores 3D que se usan para la detección de objetos con nubes de puntos tridimensionales, se aplicaron técnicas de *Deep Learning* similares a las que se usan en imágenes 2D. Por lo que durante el desarrollo del proyecto se entrenó una red PointNet para observar si era capaz de detectar sin tener algún tipo de modificación. Los datos obtenidos eran buenos aunque insuficientes para obtener una conclusión satisfactoria de que en un futuro su método pudiera ser aplicado.

### Sistema de Detección y Evasión de Obstáculos por medio de un LIDAR 360° para un Sistema Aéreo no tripulado (Chávez Cobián, 2020)

El presente trabajo descubrió que los vehículos aéreos no tripulados no cuenta con un modulo capaz de detectar y evadir obstáculos en un rango de 360 grados. Por lo que desarrollaron un sistema de detección y evasión de obstáculos, utilizando un sensor Sweep LiDAR 360°, una computadora Odroid C2, un controlador Pixhawk y su plataforma de trabajo Tarot FY450. Con estas herramientas se diseñó un algoritmo de adaptación de rutas basado en 4 modos de vuelos y se aplicaron las etapas de solución para la detección de objetos como se muestra en la Figura 3.2.

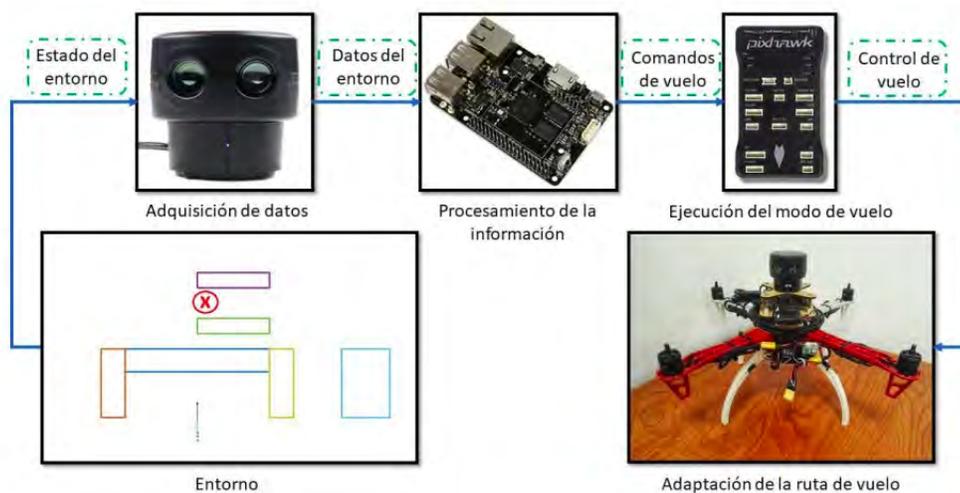


Figura 3.2: Etapas del proceso de detección y evasión de obstáculos (Chávez Cobián, 2020).

### Reconstrucción de entorno 3D mediante un sensor LIDAR (Ponce Herrera, 2019)

En el referido trabajo se realizó la reconstrucción en tres dimensiones mediante la utilización de un sensor RPLDIAR A1, un escáner láser 3D de bajo costo de desarrollador, el cual permitió realizar un barrido con un haz de luz en 360 grados produciendo una nube de puntos en 2D por lo que se diseñó una plataforma giratoria con el sensor y agrego otro grado de libertad y poder realizar una reconstrucción en 3D. El procesamiento de los datos se trabajó mediante la plataforma de MATLAB 2017b, haciendo uso de criterios matemáticos en referente a matrices de rotación, traslación y transformación homogénea, además de que se realizó una interfaz gráfica la cual es flexible para que el usuario pueda ingresar sus parámetros de reconstrucción y aplicar los filtros de visualización de las nubes de puntos.

Finalmente de los datos obtenidos se pudo eliminar ruido generado comúnmente por el hardware de muestreo en adquisición de datos y se pudo manipular de manera efectiva la nube de puntos para centrarse en áreas determinadas como se puede ver en la Figura 3.3.

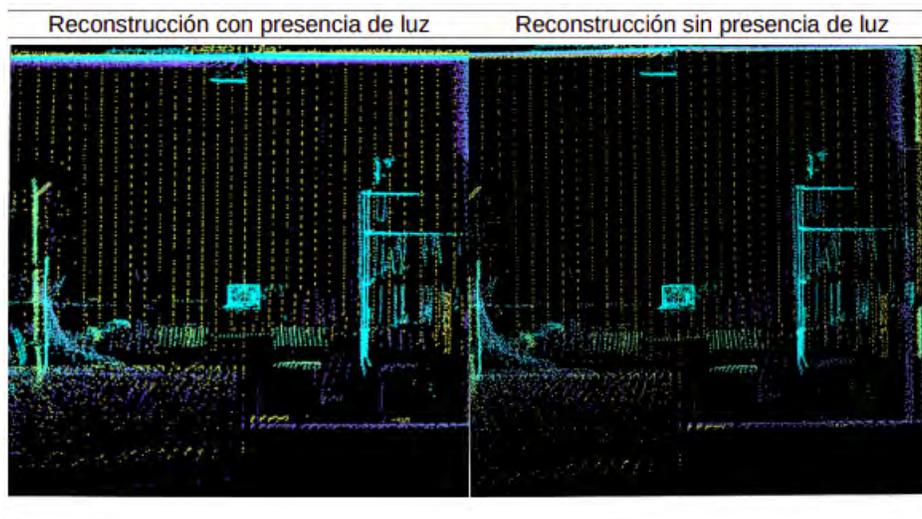


Figura 3.3: Resultados de la reconstrucción. (Ponce Herrera, 2019).

### *Object Recognition Based Interpolation with 3D LIDAR and Vision for Autonomous Driving of an Intelligent Vehicle (Weon et al., 2020)*

En el artículo revisado se desarrolló un algoritmo que permite fusionar sistemas 3D LIDAR que reciben objetos detectados, basados en aprendizaje profundo. El algoritmo aplicado permite muestras aleatorias 3D y la extracción de datos terrestres perpendiculares al plano 3D de estimación en referencia a datos y extremos a través de una estimación terrestre.

***A Decision Tree Based Road Recognition Approach Using Roadside Fixed 3D LIDAR Sensors (Zheng et al., 2019).***

En el documento analizado se presentó un algoritmo de reconocimiento de carreteras basado en árboles de decisiones que hacen uso de sensores de detección y alcance de luz fija. Los resultados obtenidos durante la experimentación y comparación con los algoritmos de reconocimiento de carreteras existentes demostraron tener ventajas de ser completamente automático, su tiempo de reconocimiento es menor y su rango de detección es más alto.

***A 3D LIDAR Data-Based Dedicated Road Boundary Detection Algorithm for Autonomous Vehicles (Sun et al., 2019)***

El citado artículo se propuso un algoritmo de cuatro pasos los cuales son:

- Primero los datos 3D LIDAR procesados previamente, emplean la información de posición y actitud del vehículo y se elimina los puntos de ruido.
- En segundo lugar, los puntos terrestres se separan rápidamente de los datos de la nube de puntos procesados para reducir la perturbación de los obstáculos en la carretera.
- En tercer lugar, los puntos candidatos de los límites de la carretera buscan a lo largo de la trayectoria pre-visualizada del vehículo y se filtran utilizando las características únicas de los puntos límite.
- Finalmente, se aplica un modelo de ajuste spline para suavizar los límites de la carretera.

***Obstacle prediction for automated guided vehicles based on point clouds measured by a tilted lidar sensor (Rozsa and Sziranyi, 2018)***

En el artículo estudiado se desarrolló un método de detección para vehículos autónomos equipados con sensores de rango 3D con fines de navegación y dar solución al problema de categorización de obstáculos para nube de puntos parciales sin modelado de formas. El enfoque se probó en una base de datos conocida, así como también para escenarios de la vida real. Los resultados obtenidos demostraron funcionar directamente en una nube de puntos, siendo invariante en la rotación, en reconocer objetos de nubes muy parciales y el método es comparable en objetos de tamaño completo de una vista a otros métodos.

*Machine-Learning and 3D Point-Cloud Based Signal Power Path Loss Model for the Deployment of Wireless Communication Systems* (Egi and Otero, 2019)

En el consultado artículo se propuso un método de implementación inteligente para minimizar la necesidad de parámetros de estimación predefinidos mediante la creación de un entorno de implementación 3D que tiene en cuenta las micro variaciones en el entorno. Por lo que su modelo propuesto se centro principalmente en el efecto de las copas de los árboles, la cual se puede aplicar a cualquier entorno, pues utiliza una clasificación de color de imagen 2D para extraer características de una nube de puntos de 3D y un algoritmo de aprendizaje automático como se muestra en la Figura 3.4.

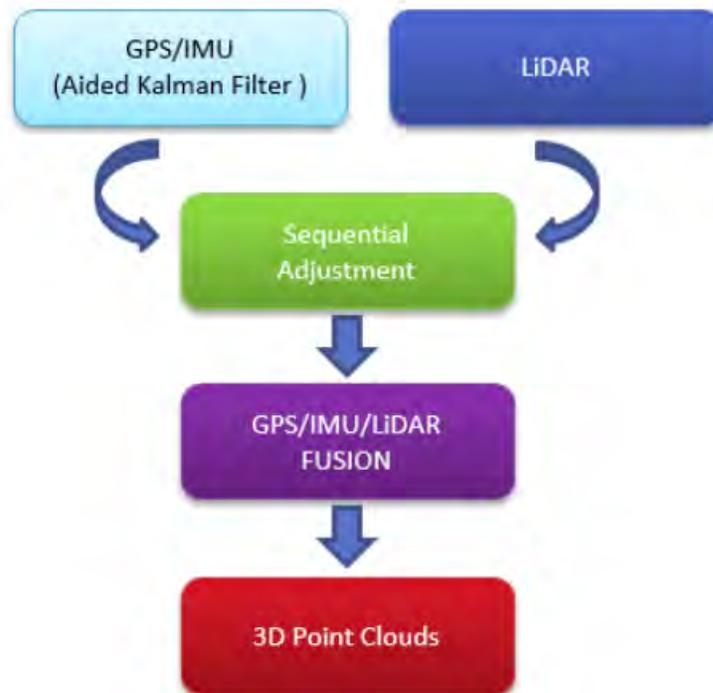


Figura 3.4: Nubes de puntos de objetos con puntos clave agrupados en descriptores locales propuesto (Egi and Otero, 2019).

### *Simultaneous 3D Object Recognition and Pose Estimation Based on RGB-D Images Speckle*(Tsai and Tsai, 2018)

En el referido artículo se propuso un diseño novedoso de reconocimiento simultaneo de objetos 3D y un algoritmo de estimación de pose basado en imágenes RGB-D. El sistema de la Figura 3.5 convierte la imagen RGB-D de entrada en datos de nubes de punto de colores y extrae características de la escena en la nube de puntos de colores.

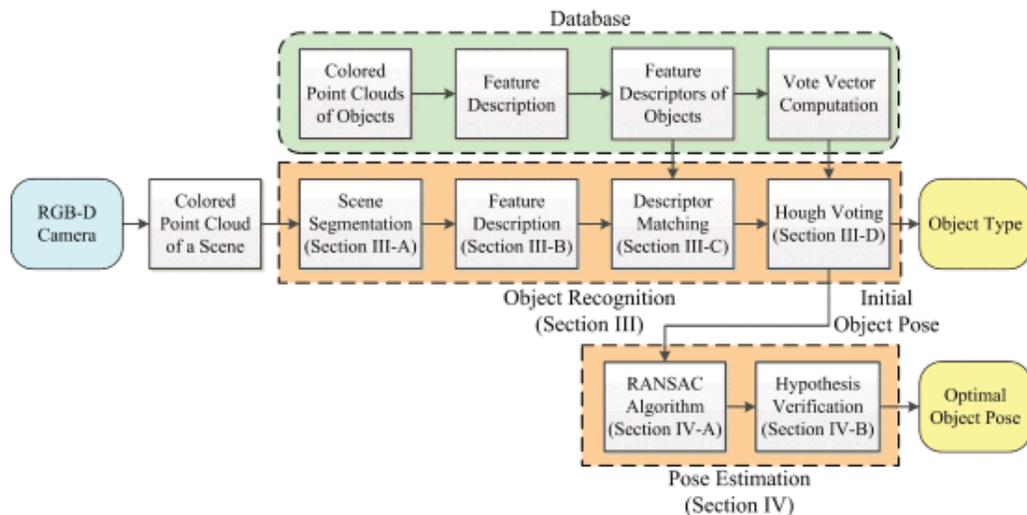


Figura 3.5: Arquitectura del sistema del algoritmo propuesto de reconocimiento de objetos y estimación de pose (Tsai and Tsai, 2018).

Dada la extracción de características llevaron a cabo el proceso de emparejamiento en dos etapas para encontrar correspondencia entre las escenas y un modelo de nube de puntos de color de un objeto, donde se utilizó un algoritmo de votación *Hough* para la filtración de errores de coincidencia en el conjunto de correspondencias y la estimación de la pose 3D inicial del objeto.

Los resultados obtenidos durante la experimentación demostraron reconocer con éxito la pose 3D del objeto en una escena compleja, además de estimar con precisión la pose 3D inicial del objeto con respecto a la cámara.

***Perspective Independent Ground Plane Estimation by 2D and 3D Data Analysis (Zhang and Czarnuch, 2020)***

En el aludido artículo se presentó un método para estimar el plano del suelo en un terreno desconocido. A partir de un conjunto de pruebas de datos RGB-D, se desarrolló un enfoque para identificar planos de tierra potenciales, agrupar objetos en las escenas y encontrar puntos clave de transformación de características invariantes en escala 2D. Los resultados muestran que el plano del suelo se puede detectar con éxito y tiene una amplia aplicación en entornos dinámicos, así como también en campos como la robótica automatizada, la localización y el mapeo.

***Research on SLAM algorithm of mobile robot based on the fusion of 2D LiDAR and depth camera (Mu et al., 2020)***

En el analizado artículo se propuso un nuevo método de localización y mapeo simultaneo basado en la optimización de gráficos mediante la combinación de detección de luz y rango, cámara RGB-D, codificador y unidad de medición inercial como se muestra en la Figura 3.6 .

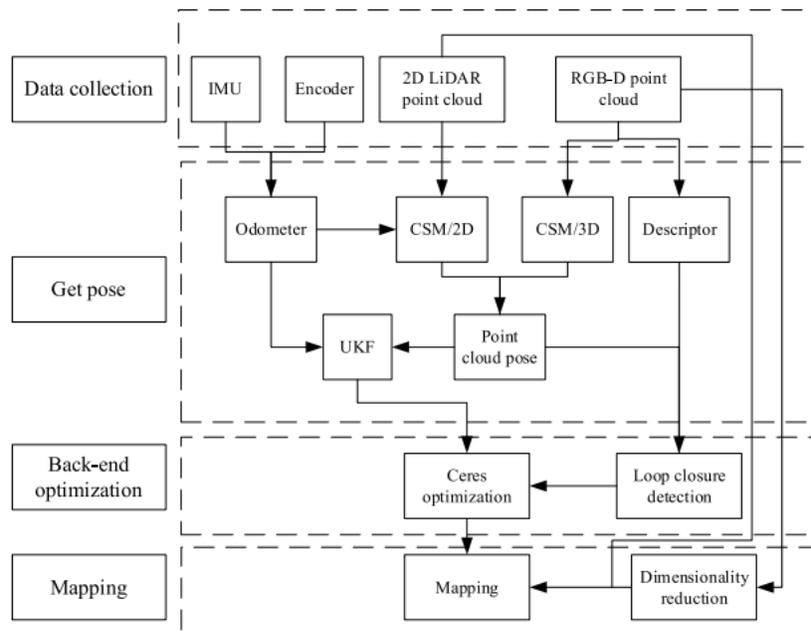


Figura 3.6: Modulo de sistema propuesto (Mu et al., 2020).

La información que genero la nube de puntos bajo el 2D LIDAR se agrego al nuevo método en la etapa de registro secuencial para verificar la precisión al describir la nube de puntos 3D. Como resultado, se muestra que el marco multisensor diseñado tiene un buen efecto de mapeo con alta precisión y exactitud.

***SpaRTA Tracking across occlusions via partitioning of 3D clouds of points (Cavagna et al., 2019)***

En el consultado documento se presentó un método de seguimiento que abordó problemas de las oclusiones dentro de grandes grupos de objetos mediante tres pasos:

1. Representa cada objetivo como una nube de puntos en 3D.
2. Una vez que ocurre un cluster 3D correspondiente a una oclusión, se define un problema de partición al introducir una función de costo que usa enlaces de proximidad espacios temporales atractivos y repulsivos.
3. Minimiza la función de costos a través de una técnica de optimización semi-definida específicamente diseñada para hacer frente a la presencia de paisajes múltiples mínimos.

El algoritmo fue diseñado para trabajar con datos 3D independientemente del método utilizando: sistemas multicamaras, lidars, radares y sistemas RGB-D. Los resultados obtenidos al realizar la experimentación demostraron que el algoritmo produce una mejora en los métodos de seguimiento, pues reduce el número de interruptores de identidad y aumenta la precisión de las posiciones estimadas de los objetos en el espacio real.

***Semantic Segmentation on 3D Occupancy Grids for Automotive Radar (Prophet et al., 2020)***

En el examinado artículo se utilizó redes de segmentación semántica para distinguir entre objetos de infraestructura en una clasificación basada en la ubicación del entorno del vehículo. Con una selección de parámetros adecuada, las cuadrículas resultantes permiten una buena asociación con las imágenes de las cámaras. Finalmente para evaluar las posibles ventajas de las grillas 3D como entrada de red con respecto al resultado de la segmentación, se creó y evaluó un conjunto de datos de simulación y dos conjuntos de datos del mundo real en diferentes aparcamientos y autopistas. Como resultado, se alcanzaron coeficientes de *Jaccard* entre 81% y 88% según del conjunto de datos, también se encontró que las imágenes de entrada 3D conduce mejoras en el conjunto de datos del estacionamiento.

### ***Multi Sensor Multu Object Tracking in Autonomous Vehicles (Manghat, 2019)***

En el consultado documento se desarrolló un método de seguimiento de objetos múltiples utilizando sensores LIDAR y una cámara 2D MOT para percibir objetos en el entorno de un vehículo. La información adquirida representó la presencia de objetos en un determinado marco establecido.

Los resultados de las experimentaciones realizadas con un conjunto de datos KITTI indicaron que el sistema de estimación propuesto para el seguimiento de objetos múltiples funcionan bien en varios entornos complejos.

### ***Detección y seguimiento de múltiples objetos en tiempo real para vehículos autónomos (Dussan Narvaez et al., 2020)***

En el analizado documento se desarrolló un sistema de detección y seguimiento de múltiples objetos en tiempo real para vehículos autónomos, su primer paso fue el seleccionar un tipo de *dataset* que permitiera entender mejor la realidad de conducción humana. El segundo paso consistió en seleccionar dos técnicas para la detección de vehículos y su seguimiento, el cual usó la red convencional *You Only Look Once* (YOLO). Después de haber realizado los pasos anteriores, el proceso finalizó con la validación de los datos, en donde se pudo observar que se obtuvieron buenos resultados y finalmente se utilizó la tarjeta Jestson NVIDIA TK1 que permitió compilar los archivos en una GPU integrada en la tarjeta embebida, la cual se puede montar a un vehículo para realizar pruebas de detección y decisión.

### ***UAV usage in the process of creating 3D maps by RGB spectrum (Pecho et al., 2019)***

En el mencionado artículo se enfocó en el uso de las UAV para el proceso de mapeo 3D, la comparación de los resultado de imágenes LIDAR y el relieve 3D utilizando el espectro RGB. Este espectro proporcionó una creación rápida y relativamente económica de mapas 3D con un uso adecuado. Los resultados obtenidos muestran un sorprendente escaneo de RGB por su alta precisión utilizando vehículos aéreos no tripulados, comparable con el escaneo LIDAR en términos de mapeo real.

***A Frustum-based probabilistic framework for 3D object detection by fusion of LiDAR and camera data (Gong et al., 2020)***

En el analizado artículo se utilizó un detector de objetos en 3D en tiempo real basado en localización simultánea basada en LIDAR y sin algún entrenamiento previo del modelo 3D CNN, obteniendo resultados de imágenes LiDAR-SLAM en un mapa de probabilidad 3D. El objetivo de esto fue resolver problemas de la dispersión y ruido de los datos LISAR-SLAM como por ejemplo en la Figura 3.7 en los que es difícil aplicar cualquier descriptor de nube de puntos.

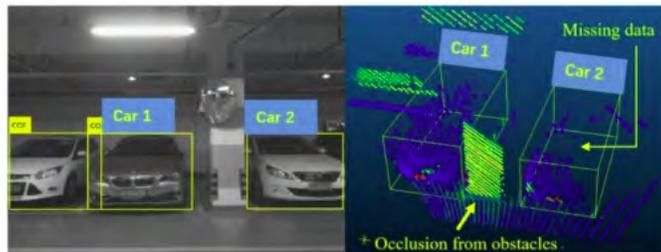


Figura 3.7: Principales desafíos de la detección de objetos 3D en nubes de puntos con objetos incompletos (Gong et al., 2020).

Es por eso que en conjunto a los datos obtenidos con LIDAR-SLAM en un mapa de probabilidad 3D, la combinación de información visual y el uso de un rango probabilístico basado en *frustum*, llegó a resolver el problema con referente a la dispersión y el ruido. Para comprobar el funcionamiento de su método se usó del conocido *KITTI Vision Benchmark Suite*, con la información de imágenes adquiridas, demostrando ser un método que supera a otros métodos de última generación.

***3D LIDAR-Based Global Localization Using Siamese Neural Network (Yin et al., 2019)***

El consultado documento se baso en el escaneo de dimensiones reducidas, aplicando el aprendizaje de redes neuronales, para dar una solución a la localización global logrando el reconocimiento y estimación de pose. Con el uso de un algoritmo para el filtro de partículas y obtener una estimación de pose precisa; Finalmente los resultados obtenidos mostraron que el sistema logra una alta eficiencia de autonomía a largo plazo.

## **Desarrollo de un asistente de conducción longitudinal mediante un Algoritmo de Aprendizaje Profundo Cotuto (Medina et al., 2020)**

En el examinado proyecto se implementó y desarrolló una simulación de un automóvil con asistente de conducción para una trayectoria longitudinal donde el automóvil es capaz de frenar o acelerar al detectar objetos, con el uso de técnicas de redes neuronales convencionales y detección de maquina. Durante las pruebas realizadas y la aplicación de las redes neuronales como clasificador para la detección de objetos demostraron ser bastante eficientes y confiables, gracias a su precisión, sensibilidad y sin ocupar una gran cantidad de recursos computacionales.

## **Discusión del Estado del Arte**

Los trabajos mencionados en el estado del arte abarcan diversos temas referentes a la detección de objetos utilizando sensores LIDAR 3D y que son de importancia para esta investigación, entre los temas de interés se encuentran: métodos de detección de objetos por medio de nubes 3D, métricas para la selección de algoritmos aplicados a nubes de puntos, métodos de disminución de ruidos en nubes 3D LIDAR. Este compendio de artículos se compone principalmente de trabajos recientes que sustentan a este trabajo. A continuación se muestra en la Tabla 3.1 información resumida de las partes clave de cada artículo que comprenden el nombre del trabajo, objetivo, algoritmos/métodos utilizados y la utilidad para esta investigación.

Tabla 3.1: Resumen de artículos.

Trabajo	Objetivo	Algoritmos/Métodos	Utilidad
AADS: Augmented autonomous driving simulation using data-driven algorithms ( <i>Li et al., 2019</i> ).	Presentar una simulación de conducción autónoma aumentada.	Algoritmo de entrenamiento en sistemas AD.	Características y mejoras para la percepción de objetos.
Detección basada en datos 3D de objetos urbanos ( <i>Castañero Amorós, 2020</i> ).	Conocer las principales herramientas y modelos de detección de objetos	Algoritmo Yolo-You Only Look Once	Uso de algoritmo para la detección de objetos.
Sistema de Detección y Evasión de Obstáculos por medio de un LIDAR 360° para un Sistema Aéreo no tripulado ( <i>Chávez Cobián, 2020</i> ).	Desarrollar un modulo electrónico capaz de detectar y evitar la colisión con obstáculos durante un vuelo	Algoritmo Go to go	Comparación de etapas de proceso.
Reconstrucción de entorno 3D mediante un sensor LIDAR ( <i>Ponce Herrera, 2019</i> ).	Reconstruir un entorno en tres dimensiones en base a la nube de puntos mediante sensor LIDAR y mostrar resultados a través de una GUI	NA	Uso del filtrado de la nube de puntos 3D en distintas condiciones luz.
Object Recognition Based Interpolation with 3D LIDAR and Vision for Autonomous Driving of an Intelligent Vehicle ( <i>Weon et al., 2020</i> ).	Desarrollo de algoritmo de fusión con sistemas 3D LIDAR que reciben objetos detectados en sensores basados en aprendizaje profundos.	Algoritmo para la detección de objetos muestras en 3D	Uso del algoritmo para la extracción de datos terrestres en 3D.

Trabajo	Objetivo	Algoritmos/Métodos	Utilidad
A Decision Tree Based Road Recognition Approach Using Roadside Fixed 3D LiDAR Sensors ( <i>Zheng et al., 2019</i> ).	Algoritmo de reconocimiento basado en arboles de decisión para la detección.	Algoritmo basado en arboles de decisión.	Comparación con los métodos de detección.
A 3D LiDAR Data-Based Dedicated Road Boundary Detection Algorithm for Autonomous Vehicles ( <i>Sun et al., 2019</i> ).	Desarrollar un algoritmo que mejore la precisión que pueda extraer de manera robusta límite de carretera en tiempo real	Algoritmo extracción de puntos en carretera	Uso del para la detección de puntos.
Obstacle prediction for automated guided vehicles based on point clouds measured by a tilted lidar senso ( <i>Rozsa and Sziranyi, 2018</i> ).	Desarrollar un metodo para la detección y categorización de obstáculos.	Metodo de detección y categorización de la nube de puntos.	Uso del método de detección de la nube de puntos .
Machine-Learning and 3D Point-Cloud Based Signal Power Path Loss Model for the Deployment of Wireless Communication Systems ( <i>Egi and Otero, 2019</i> ).	Proponer un metodo de implementación en inteligencia WCS	Metodo basada en WCS .	Comparación en el uso WCS
Simultaneous 3D Object Recognition and Pose Estimation Based on RGB-D Images Speckle ( <i>Tsai and Tsai, 2018</i> ).	Diseño de métodos de reconocimiento simultaneo de objetos 3D .	Algoritmo CJHOT.	Uso de las etapas para encontrar correspondencia entre las escenas y un modelo de nube de puntos de color

Trabajo	Objetivo	Algoritmos/Métodos	Utilidad
Perspective Independent Ground Plane Estimation by 2D and 3D Data Analysis ( <i>Zhang and Czarruch, 2020</i> ).	Desarrollo de un método de estimación del plano del suelo	Método de estimación con cámara RGB-D .	Caracterizas para la detección de objetos con RGB-D.
Research on SLAM algorithm of mobile robot based on the fusion of 2D LiDAR and depth camera ( <i>Mu et al., 2020</i> ).	Desarrollo de nuevo método de localización y mapeo simultaneo basado en gráficos mediante detección de luz y rango	Método SLAM.	Uso de la estrategia que relacione la nube de puntos LIDAR y la camra RGB-D.
SpaRTA Tracking across occlusions via partitioning of 3D clouds of points ( <i>Cavagna et al., 2019</i> ).	Desarrollo de método y algoritmo seguimiento para trabajar con datos 3D .	. Métodos SpaRTA	Uso del método propuesto para obtener mejor precisión y posición de los objetos
Semantic Segmentation on 3D Occupancy Grids for Automotive Radar ( <i>Prophet et al., 2020</i> ).	. Uso de las redes segmaticas para distinguir objetos.	NA.	Uso de las métricas para la evaluación de las entradas de imagen 3D.
Multi Sensor Multu Object Tracking in Autonomous Vehicles ( <i>Manghat, 2019</i> ).	Desarrollar un método de seguimiento de objetos multiples utilizando LIDAR y una camara 2D MOT	MÉTODO MOTA.	Uso metricas de detección de objetos.

Trabajo	Objetivo	Algoritmos/Métodos	Utilidad
<p>Detección y seguimiento de múltiples objetos en tiempo real para vehículos autónomos (<i>Dussan Narvaez et al., 2020</i>).</p>	<p>Desarrollar un sistema de detección y seguimiento de múltiples objetos basados en el estado del arte de la detección de objetos para vehículos autónomos de tipo 5</p>	<p>Algoritmo Yolo.</p>	<p>Uso de del algoritmo Yolo para la detección de objetos.</p>
<p>UAV usage in the process of creating 3D maps by RGB spectrum (<i>Dussan Narvaez et al., 2020</i>).</p>	<p>Uso de UAN en proceso de Mapeo 3D</p>	<p>Metodo UAN.</p>	<p>Comparación en métodos de detección de objetos en sensor 3D.</p>
<p>A Frustum-based probabilistic framework for 3D object detection by fusion of LiDAR and camera data (<i>Gong et al., 2020</i>).</p>	<p>Resolver problemas de la dispersión de ruido de las datos LISAR-SLAM en la que se aplica cualquier nube de puntos.</p>	<p>Método LISAR-SALM</p>	<p>Uso del método para disminuir el ruido en imágenes 3D LIDAR .</p>
<p>3D LiDAR-Based Global Localization Using Siamese Neural Network (<i>Yin et al., 2019</i>).</p>	<p>Desarrollar un método de aprendizaje de características semi-artesanales para la detección para la detección de luz 3D</p>	<p>Método de detección de características semi-artesanales.</p>	<p>Uso de métricas para nube de puntos 3D LIDAR</p>
<p>Desarrollo de un asistente de conducción longitudinal mediante un Algoritmo de Aprendizaje Profundo Cotuto (<i>Medina et al., 2020</i>).</p>	<p>Desarrollar e implementar un asistente de conducción para trayectorias longitudinales en un automóvil, utilizando un automóvil utilizando un algoritmo de aprendizaje</p>	<p>Algoritmo de CNN y Yolo</p>	<p>Uso de la evolución y selección de algoritmos de clasificación y detección.</p>

# Capítulo 4

## Metodología

En este capítulo el objetivo principal es mostrar el proceso de detección de objetos durante el desplazamiento vehicular propuesto, donde se encuentran involucrados 3 procesos fundamentales, los cuales son: obtención de la nube de puntos, la detección de objetos y la construcción del cuadro delimitador, tal como se muestra en la Figura 4.1.

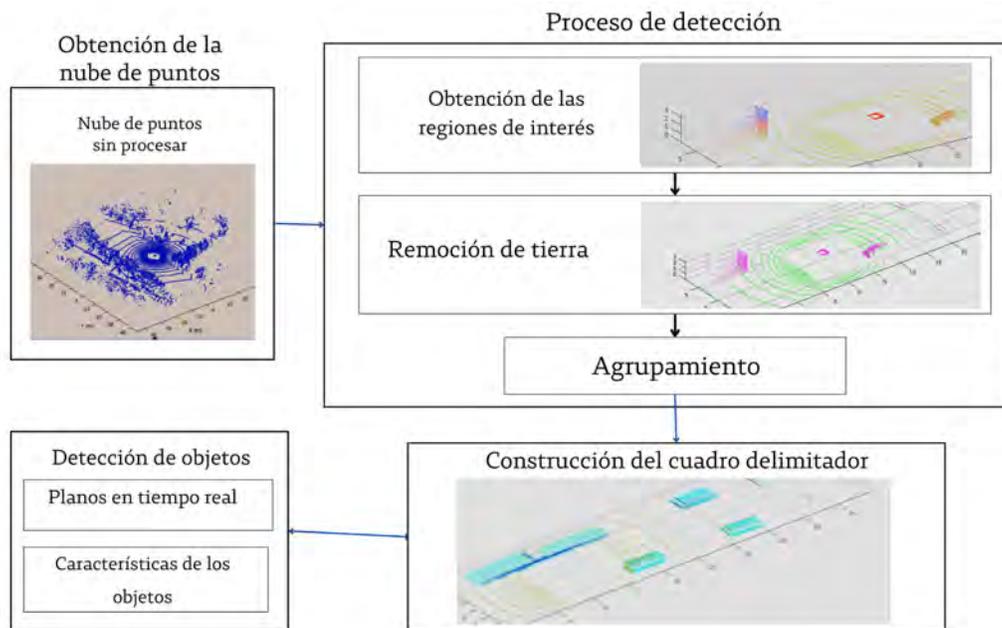


Figura 4.1: Proceso de detección de objetos a través de una nube de puntos.

## 4.1. Obtención de la nube de puntos

El primer bloque a realizado es la obtención de la nube de puntos en el que participarán dos importantes elementos, los cuales son el sensor LIDAR y el sensor RGB-D, como se observa en la Figura 4.2. El principal objetivo de combinar RGB-D con LIDAR es disminuir el ruido que puede estar presente en el ambiente, a través del mapa de profundidad y obtener una nube de puntos más optimizada.

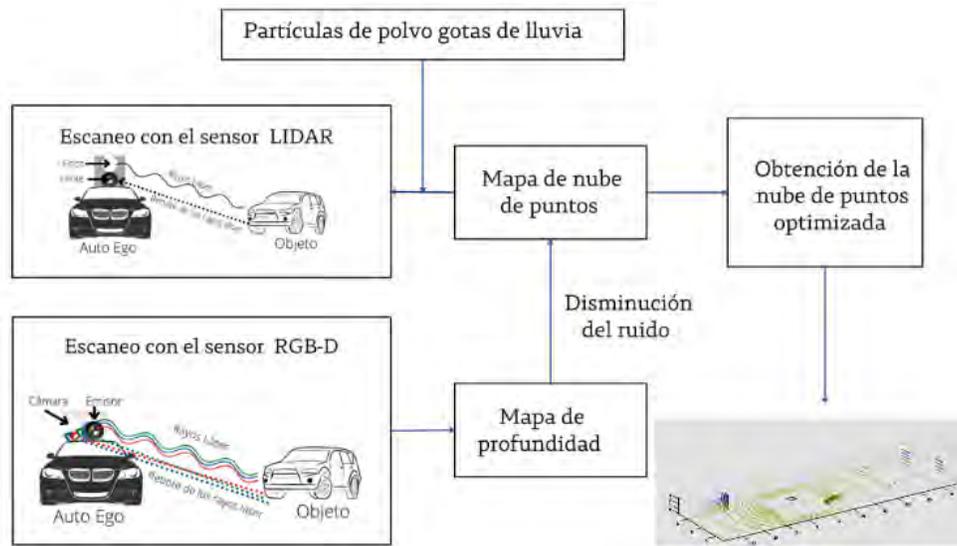


Figura 4.2: Trabajo en conjunto de los sensores LIDAR y RGB-D.

Para la obtención de datos, se creó dentro de las simulaciones un auto principal o también conocido como auto ego, que se observa en la Figura 4.3, el cual se colocó un sensor LIDAR y un sensor RGBD con las siguientes especificaciones:

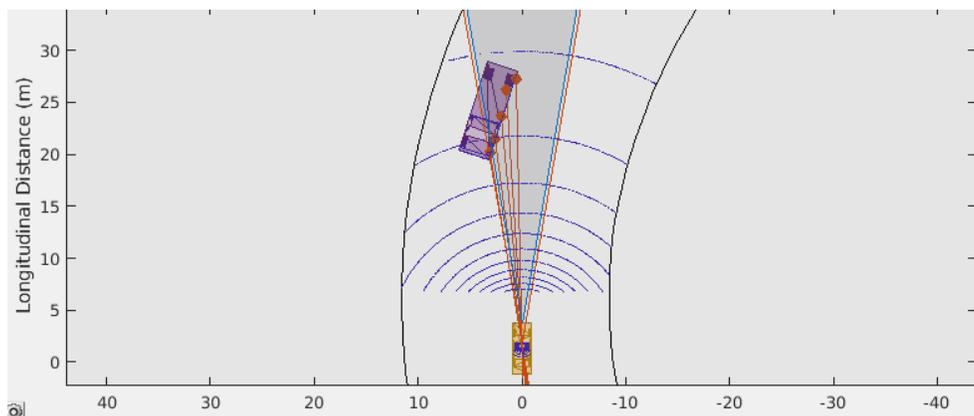


Figura 4.3: Auto Ego con el sensor RGB-D y LiDAR en Matlab.

Especificaciones de LIDAR:

- Alcance del escáner 100 m
- Precisión 0.002 m
- Elevación 1.25 m

Especificaciones de RGB-D

- Precisión: 0.9 m
- Alcance 100 m
- Elevación 1.5 m

Una vez colocados los sensores con las configuraciones pertinentes, se da inicio a las pruebas en los diferentes escenarios construidos.

## **4.2. Proceso de detección de objetos**

El siguiente bloque a realizar es el proceso de detección de objetos, utilizando la nube de puntos obtenida con los sensores LIDAR y RGB-D. En este proceso participa en un inicio, el algoritmo RANSAC para obtener los puntos útiles, la remoción de tierra, y posteriormente realizar el agrupamiento con el algoritmo DBSCAN.

### **4.2.1. Procesamiento de la nube de puntos con RANSAC**

En este proceso recibimos la nube de puntos obtenida con los sensores LIDAR y RGB-D, para poder realizar la segmentación de la nube de puntos a través del algoritmo RANSAC. Estos mismos pasos serán aplicados a la nube de puntos obtenida para realizar su procesamiento como se muestra en la Figura 4.4.

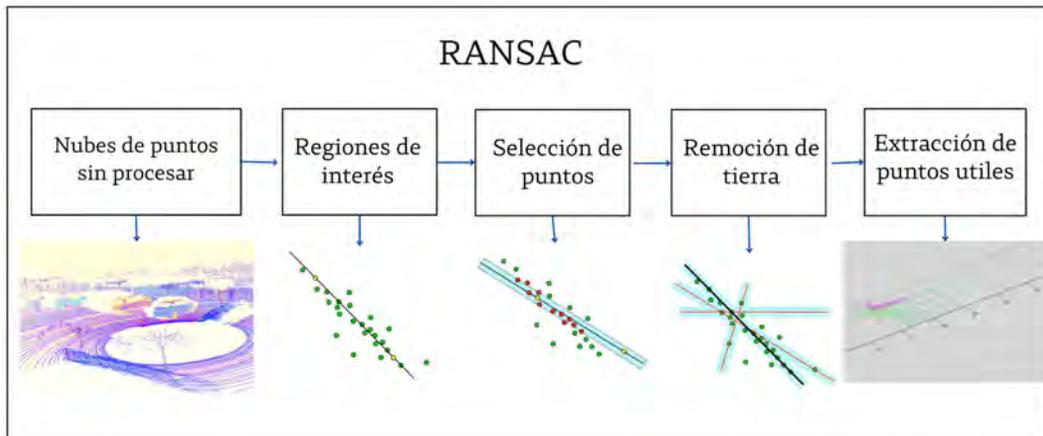


Figura 4.4: Diagrama de bloques de RANSAC.

## Regiones de Interés

En primer lugar, antes de realizar el procesamiento de la nube de puntos se lleva a cabo la remoción de tierra, cada punto que es recopilado lo divide en diferentes cuadros con una estructura  $x, y, z, i, r$ , donde  $x, y, z$  representan las coordenadas tridimensionales,  $i$  es la intensidad y  $r$  es el número de puntos (Gönültaş et al., 2020). Posteriormente necesitamos recopilar los datos obtenidos, para dividir los puntos dependiendo a una marca de tiempo.

Los puntos son seleccionados como  $P_1(x_1, y_1, z_1), P_2(x_2, y_2, z_2), P_3(x_3, y_3, z_3)$  del punto original en la nube  $P$  de forma aleatoria, los puntos de entrada se representan como  $P = (p_1, p_2, p_3, \dots, p_k)$ ,  $k$  es el número de puntos y cada punto se representa como  $p_i = (x_i, y_i, z_i)$ , ( $i = 1, 2, \dots, k$ ),  $p_i$  contiene la posición espacial en las coordenadas cartesianas de cada punto, en la Figura 4.5 se puede apreciar esta representación.

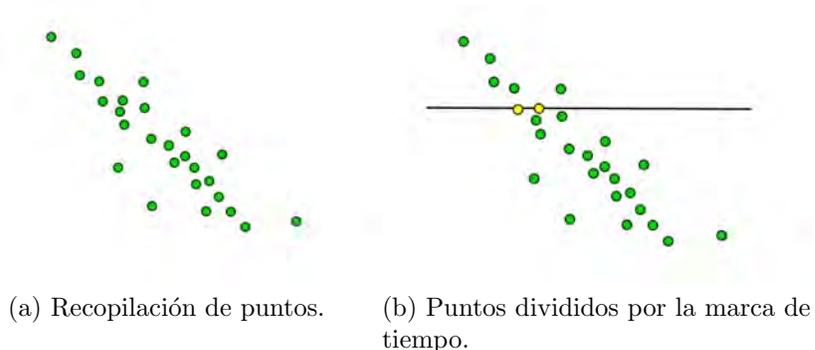


Figura 4.5: Primer paso del algoritmo RANSAC.

## Selección de puntos

Una vez obtenidas las regiones de interés es necesario descartar los datos atípicos realizando una selección de puntos mediante los parámetros de LIDAR, específicamente la resolución horizontal aplicando la formula 4.1 .

$$n_{ver} = \frac{2\pi}{r_{nor}} \times 180^\circ \quad (4.1)$$

donde  $n_{ver}$  son el recuento de las líneas verticales y  $r_{nor}$  es la resolución horizontal de LIDAR.

Además de tomar en cuenta el plano del escenario con la siguiente ecuación.

$$\begin{cases} A_{x_1} + B_{y_1} + C = Z_1 \\ A_{x_2} + B_{y_2} + C = Z_2 \\ A_{x_3} + B_{y_3} + C = Z_3 \end{cases} \quad (4.2)$$

De forma visual este proceso se puede apreciar en la Figura 4.6.

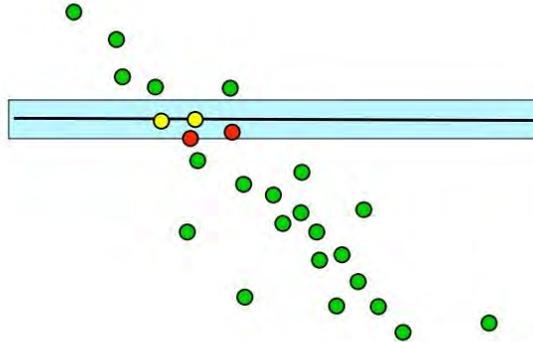


Figura 4.6: Ajuste del modelo de un subconjunto con RANSAC (Lopez Torrijos, 2021).

Una vez realizada la selección de puntos es necesario la disminución de datos innecesarios que pueden afectar el rendimiento del algoritmo, por lo que se requiere realizar un proceso de remoción de tierra.

## Remoción de tierra

Después de obtener las líneas verticales, de acuerdo con las reglas del algoritmo, se obtiene los puntos iniciales del suelo y los puntos del umbral. Estos puntos son obtenidos con apoyo de la distancia ( $d_i$ ) de cualquier punto ( $p_i$ ) en el conjunto de datos ( $P$ ) hasta el plano ( $S$ ) y se calcula con la siguiente formula:

$$d_i = \frac{|A_x i + B_y i + C - z_i|}{\sqrt{A^2 + B^2 + 1}} \quad (4.3)$$

Este paso se puede apreciar de forma visual en la Figura 4.7.

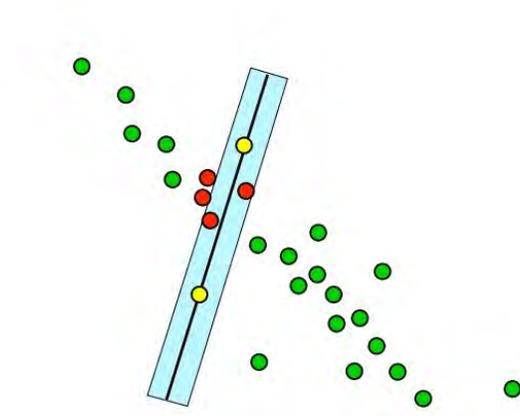


Figura 4.7: Ajuste del modelo de un subconjunto con RANSAC (Lopez Torrijos, 2021).

## Extracción de puntos útiles

Finalmente se comienza el etiquetado del primer grupo de puntos útiles y el proceso comienza de nuevo hasta que todos los puntos de la nube de puntos se encuentren etiquetados. Para comprobar el buen funcionamiento de este algoritmo calculamos el parámetro  $N$ , que representa un número mínimo de puntos con los cuales se estima la regresión, con las siguientes ecuaciones.

$$1 - (1 - \iota)^k = \varphi_1 \quad (4.4)$$

$$k = \frac{\log(1 - \varphi)}{\log(1 - (\iota)^3)} \quad (4.5)$$

donde  $\iota$  es la probabilidad de efectividad del algoritmo de los puntos fuera del plano ( $\varphi$ ) y  $K$  son la cantidad de veces del muestreo. El resultado final puede ser apreciado de forma visual en la Figura 4.8.

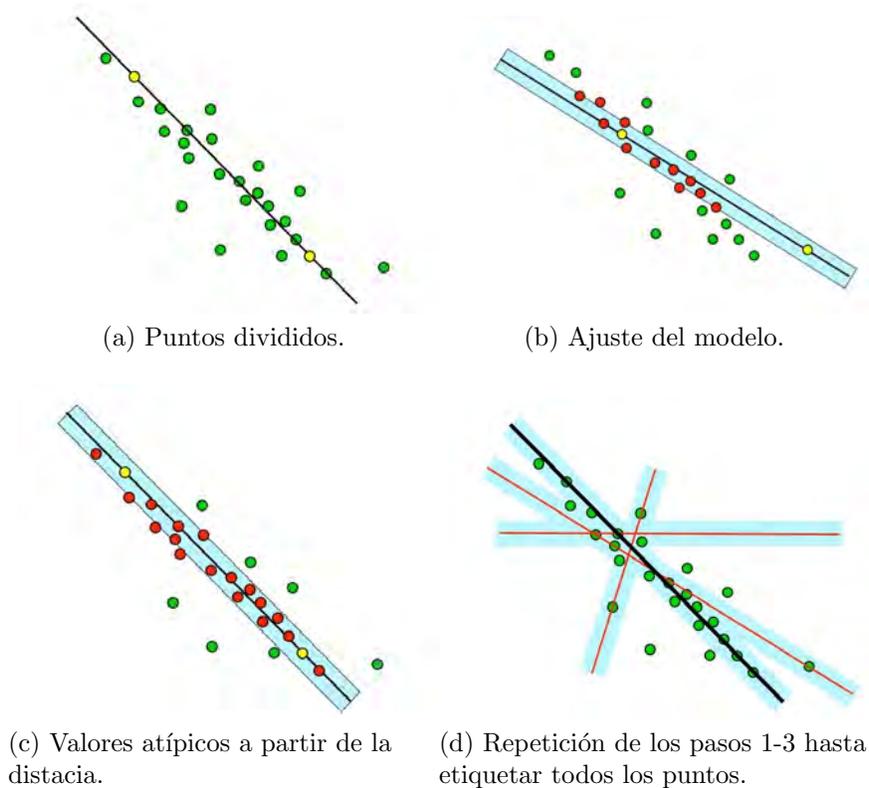


Figura 4.8: Funcionamiento de RANSAC (Lopez Torrijos, 2021)

Los pasos anteriormente mencionados pueden ser visualizados en Algoritmo 1, y el cual servirá como guía para la parte de programación.

---

**Algoritmo 1** *RANSAC*

---

**Entrada:** Nube de puntos  $N$  y regiones de interés  $K$

**Salida:** Punto más alto del plano  $S$

1: **for**  $K < N$  **do**

2: Selección de puntos

$$P_1(x_1, y_1, z_1), P_2(x_2, y_2, z_2), P_3(x_3, y_3, z_3)$$

3: **if**  $A_x + B_y + C = z$  **then**  $\left\{ \begin{array}{l} A_{x_1} + B_{y_1} + C = Z_1 \\ A_{x_2} + B_{y_2} + C = Z_2 \\ A_{x_3} + B_{y_3} + C = Z_3 \end{array} \right\}$

4: **end if**

5:  $d_i = \frac{|A_x i + B_y i + C - z_i|}{\sqrt{A^2 + B^2 + 1}}$

6: **end for**

7: Punto más alto del plano  $S$

---

Dentro de este algoritmo podemos guiarnos para realizar el proceso de selección de puntos que se muestra en el paso 2, hasta la localización del punto más alto del plano en el paso 7. Una vez terminado la aplicación del algoritmo, los resultados obtenidos pueden pasar a ser agrupados con DBSCAN.

## 4.2.2. Agrupamiento de la nube de puntos con DBSCAN

Una vez realizada la obtención de los puntos útiles con RANSAC, se procede a realizar el agrupamiento con el algoritmo de DBSCAN, en donde se realizara la identificación de puntos centrales, la asignación de los puntos a un grupo para finalmente obtener el agrupamiento de la nube de puntos, como se muestra en la Figura 4.9. Cada paso será explicado en las siguientes sub-secciones.

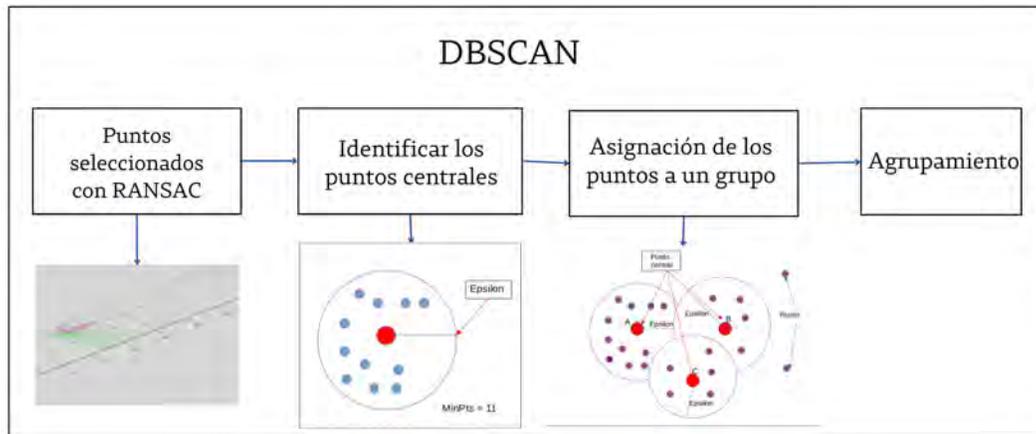


Figura 4.9: Diagrama de bloques del Algoritmo de DBSCAN.

### Identificación de puntos centrales

Al entrar la nube de puntos, con los puntos útiles a DBSCAN el primer paso es encontrar todos los puntos vecinos dentro de  $\epsilon$  e identificar los puntos centrales. Se elige trazando la distancia al vecino más cercano ordenando desde el valor más grande al más pequeño. Si se elige el valor más pequeño una gran cantidad de datos no se agruparán, mientras si se toma un valor demasiado alto las agrupaciones se fusionaran (Chauhan, 2020).

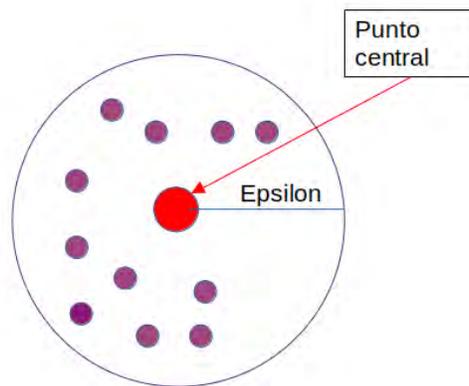


Figura 4.10: Visualización de la identificación de los puntos centrales y el trazado de  $\epsilon$ .

### Asignación de los puntos a un grupo

Para cada punto central identificado en la nube de puntos, se conectará con el resto de puntos por su densidad y asignados al mismo grupo que el punto central  $p$  como se aprecia en la Figura 4.11).

De forma matemática se puede expresar  $N \geq \text{minPts}$ , donde  $N$  representa los vecinos y  $\text{minPts}$  representa el número mínimo de observaciones dentro de la región de  $\epsilon$ . Los puntos centrales se pueden identificar como  $a$  y  $b$ , si existe un punto conocido como  $c$  que tiene un número suficiente de puntos con sus vecinos  $b$  y que están dentro de  $\epsilon$ , se le llama un proceso de encadenamiento.

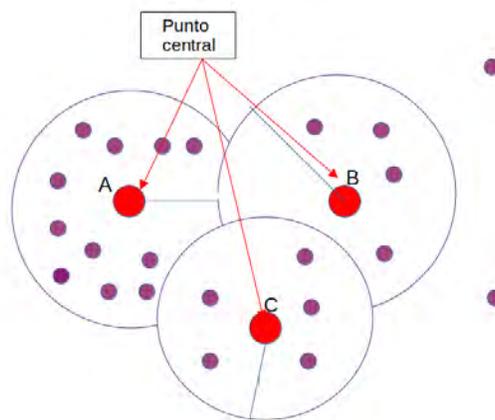


Figura 4.11: Visualización de la asignación de puntos a un grupo.

## Agrupamiento con DBSCAN

Finalmente se obtiene el agrupamiento de todos los puntos al ser colocados en cada grupo, los puntos que no lleguen a pertenecer a ningún agrupamiento, son los puntos que serán considerados como ruido, esto se puede ver como en la Figura 4.12.

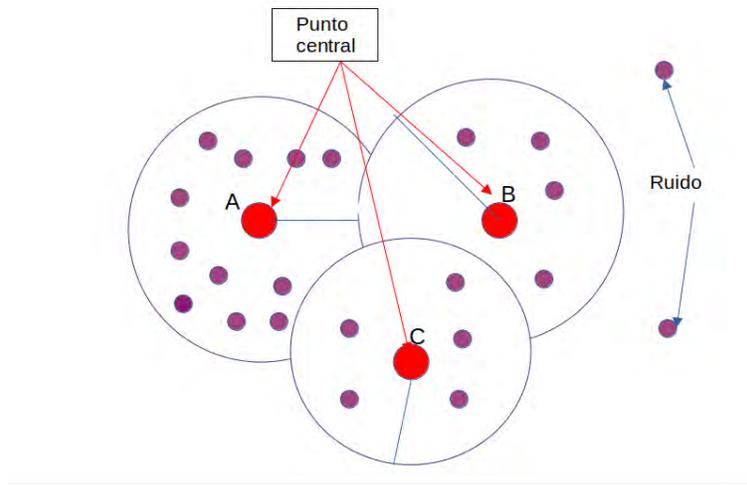


Figura 4.12: Visualización del agrupamiento final.

Para su programación se utiliza el Algoritmo 2 que se muestra a continuación:

---

### Algoritmo 2 *DBSCAN*

---

**Entrada:**  $\epsilon$ , *dataset*, *minPts*

- 1: **for**  $p$  in *dataset* **do**
  - 2: Marcar  $p$  como vecino  
Vecinos  $N$  = encontrar los vecinos de  $p$
  - 3: **if**  $N \geq minPts$  **then**  $N = N \cup p$
  - 4: **if**  $p \notin N$  **then**  $p = C$
  - 5: **end if**
  - 6: **end if**
  - 7: **end for**
- 

Dentro del algoritmo recibimos los puntos útiles obtenidos con el algoritmo anterior, para proceder al paso 2 de encontrar los vecinos más cercanos y comenzar con la asignación de grupos que señala del paso 3 al 7, hasta finalmente obtener el agrupamiento de la nube de puntos y pasar al proceso de construcción del cuadro delimitador.

### 4.3. Construcción del cuadro delimitador

El último proceso a realizar es la construcción del cuadro delimitador como se observa en la Figura 4.13, pasa a tomar dos criterios que son el área y la cercanía.

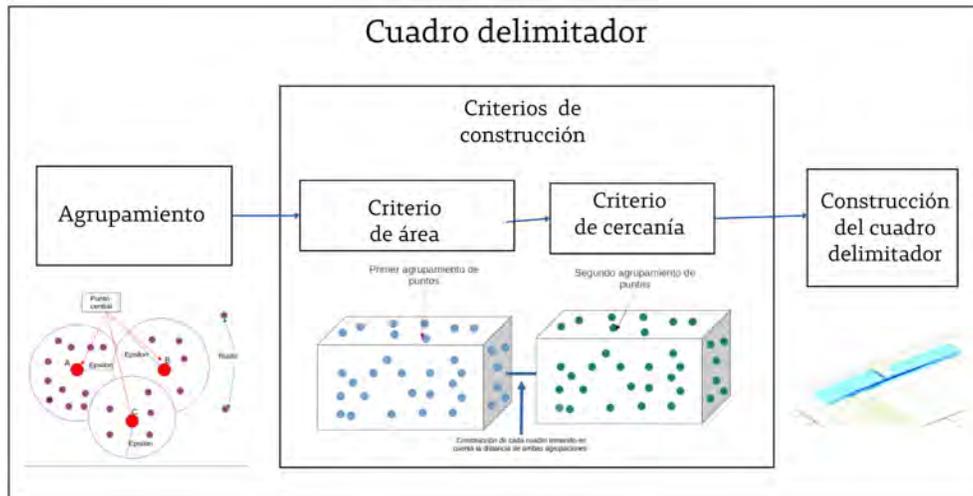


Figura 4.13: Diagrama de bloques de RANSAC.

- Criterios de cercanía: Este criterio se refiere a la cercanía en que se encuentra cada agrupación de puntos con otros para comenzar con la construcción del cuadro delimitador.

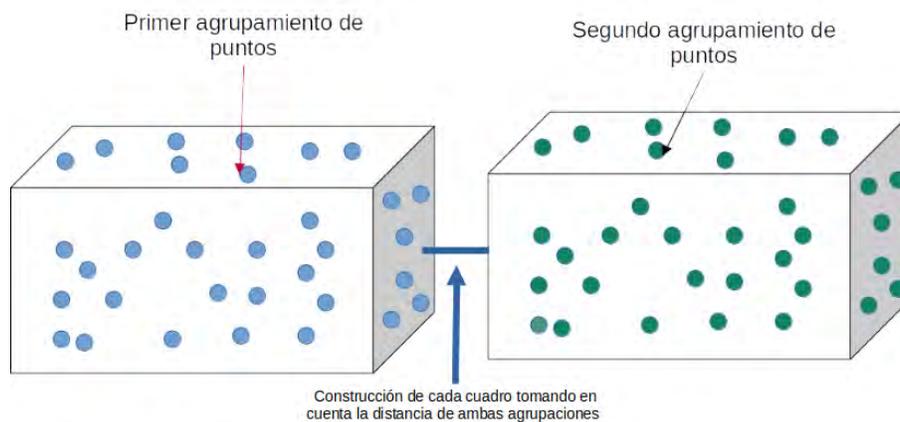


Figura 4.14: Visualización del criterio de construcción por cercanía.

- Criterio de área: Para la construcción del cuadro delimitador, se toma en cuenta la cantidad de puntos que se encuentran y estarán dentro del cuadro. Cada cuadro determinará la altura a partir de 0 a 90°.

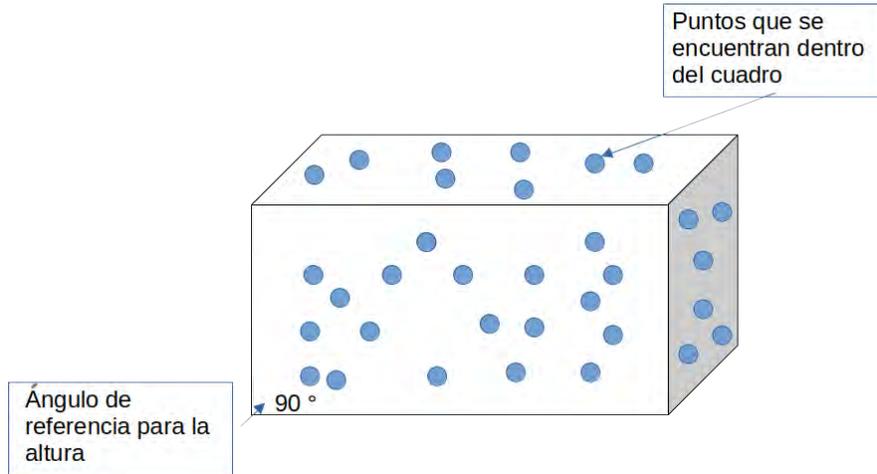


Figura 4.15: Visualización del criterio de construcción por área.

Para la programación de este proceso se utiliza el Algoritmo 3 que se muestra a continuación:

---

**Algoritmo 3** Adaptación del cuadro delimitador basado en búsqueda

---

**Entrada:** range data points  $X \in \mathbb{R}^{n \times 2}$

**Salida:** rectangle edges  $(a_i x + b_i x = c_i | i = 1, 2, 3, 4)$

- 1:  $Q \leftarrow \theta$
  - 2: **for**  $\theta = 0$  to  $\pi/2 - \delta$  **step do**
  - 3:  $C_1 \leftarrow (\cos\theta, \sin\theta)$
  - 4:  $C_2 \leftarrow (-\sin\theta, \cos\theta)$
  - 5:  $q \leftarrow$  Calculate criterion  $X(C_1, C_2)$
  - 6: insert  $q$  into with key  $(\theta)$
  - 7: **end for**
  - 8: select key  $(\theta^*)$  from  $Q$  with maximum value
  - 9:  $C_1^* \leftarrow X \cdot (\cos\theta^* \sin\theta^*)^T, C_2^* \leftarrow X \cdot (-\sin\theta^* \cos\theta^*)^T$
  - 10:  $a_1 \leftarrow \cos\theta^*, b_1 \leftarrow \sin\theta^* c_1 \leftarrow \min\{C_1^*\}$
  - 11:  $a_2 \leftarrow \cos\theta^*, b_2 \leftarrow \sin\theta^* c_2 \leftarrow \min\{C_2^*\}$
  - 12:  $a_3 \leftarrow \cos\theta^*, b_3 \leftarrow \sin\theta^* c_3 \leftarrow \min\{C_1^*\}$
  - 13:  $a_4 \leftarrow \cos\theta^*, b_4 \leftarrow \sin\theta^* c_4 \leftarrow \min\{C_2^*\}$
- 

En el anterior algoritmo de entrada tenemos la nube de puntos agrupada que paso a paso ira tomando los criterios de área, para determinar el alto, ancho y longitud de cada cuadro y posteriormente a partir del paso 10 iniciar con el criterio de cercanía, en donde el principal objetivo es confirmar los puntos encerrados dentro de cada cuadro, respetando cada agrupación con los algoritmos anteriores. A este proceso, podemos adicionar el etiquetado de cada objeto usando una red neuronal conocida como *PointNet*.

### Aplicación de PointNet

Como un adicional al proceso de detección se implementa una red neuronal conocida como *PointNet*, el cual no solo nos proporciona características de los objetos, sino además, puede etiquetar cada objeto con su debido nombre como se aprecia en la Figura 4.16, donde se muestra que el objeto detectado es un camión.

Las bondades de aplicar esta red como un adicional al proceso de detección es que se alimenta de los datos obtenidos con los algoritmos anteriormente mencionados y partir de las características obtenidas podemos agregar la etiqueta a cada objeto, es decir, podemos identificar a través de la etiqueta si el objeto fue un camión, un vehículo o un peatón.

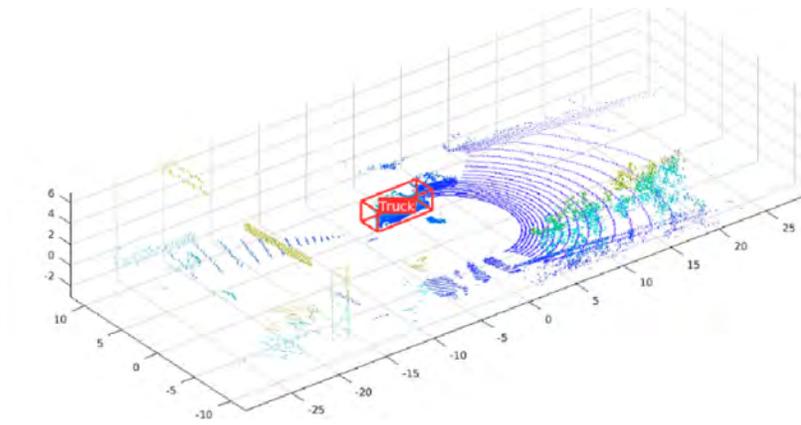


Figura 4.16: Pruebas con *PointNet*.

## 4.4. Diagrama final del proceso de detección

Tomando en cuenta cada bloque descrito anteriormente para realizar el proceso de detección, finalmente se construye un diagrama de bloques que se observa en la Figura 4.17, que describe de manera más visual el proceso anteriormente descrito y que servirá de apoyo para su implementación ya sea de forma física o virtual.

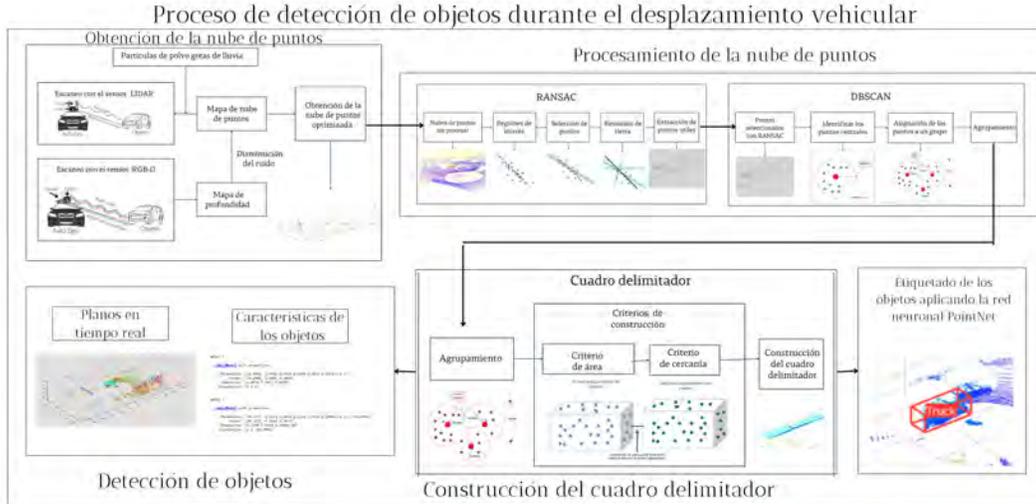


Figura 4.17: Diagrama de bloques del Proceso de detección.

## Discusión

En este capítulo se describió a detalle el proceso de detección propuesto, que involucra la obtención de la nube de puntos con los sensores LIDAR y RGB-D, el procesamiento de la nube de puntos con los algoritmos RANSAC y DBSCAN, para finalmente obtener el cuadro delimitador a partir de los criterios de área y cercanía, para obtener como primera instancia las características de cada objeto. Adicionalmente, se plantea la aplicación de una Red Neuronal conocida como PoinNet, que permite a partir de este proceso etiquetar cada objeto con su nombre correspondiente, es decir, poder señalar de forma automática si es un camión, un vehículo particular, un ciclista o un peatón.

Este procesamiento es evaluado con las métricas de detección consultadas en el estado del arte e implementado en las experimentaciones que se describen en el siguiente capítulo.

# Capítulo 5

## Experimentación y resultados

En este capítulo se presentan los experimentos diseñados e implementados para el proceso de detección, así como también las métricas de evaluación y resultados obtenidos.

### 5.1. Diseño e implementación de Pruebas

En la siguiente sección se describe el entorno y los escenarios utilizados para la experimentación, así como las métricas para la evaluación del proceso de detección.

#### 5.1.1. Entorno de desarrollo

Para el desarrollo de las experimentaciones realizadas, se utilizaron los siguientes entornos que apoyaron con la creación, visualización y evaluación de resultados en proceso de detección durante el desplazamiento vehicular.

##### ***Matlab***

*Matlab* es un entorno de escritorio combinado para el análisis iterativo y los procesos de diseño con un lenguaje de programación que expresa las matemáticas de matrices y arrays directamente, el cual permite desarrollar de forma más eficiente los algoritmos de detección (MATLAB, 2020).

##### ***Webots***

*Webots* es una aplicación de escritorio de código abierto y multiplataforma que se utiliza para simular robots. El entorno de desarrollo está diseñado para modelar, programar y simular robots. Su uso es profesional y ampliamente ocupado por la industria, la educación y la investigación. Algunas de las mayores virtudes de esta aplicación es (Michel, 2004):

- Diseño fácil de simulaciones robóticas completas.

- Bibliotecas de activos que incluye robots, sensores, actores, objetos y materiales.
- Importar mapas de *OpenStreetMap*.
- GUI moderna para editar simulación y sus controladores de robot.
- Ahorro de tiempo en el desarrollo de simulaciones.

## Escenarios de vías urbanas

Las pistas utilizadas para simular un entorno más urbano, se encuentran ubicadas en diferentes partes de la República Mexicana. Su proceso de selección fue tomando en cuenta en la viabilidad no solo de vehículos particulares, sino también del tránsito de personas y ciclistas. Las pistas seleccionadas se presentan en la Tabla 5.1.

Tabla 5.1: Escenarios de vías urbanas ubicadas en diferentes puntos de la República Mexicana

Nombre de la pista	Localización (coordenadas)
Ejidos de San Pedro Mártir- CDMX	19.2742, -99.1706
Calle ficus- Cuernavaca Morelos	18.996190, -99.20470
Centro de Tlalpan- CDMX	19.28992, -99.16968
Avenida Palmira	18.87658, -99.21862
Centro Histórico-CDMX	19.43263, -99.13324

## Escenarios en autopistas

Para la selección de estas autopistas donde la mayor cantidad de vehículos son de carga pesada y la velocidad es mayor, se tomo en cuenta la forma de la pista así como el flujo de vehículos a través de cada una. Las pistas seleccionadas se presentan en la Tabla 5.2.

Tabla 5.2: Escenarios de vías rápidas ubicados en diferentes puntos de la República Mexicana

Nombre de la pista	Localización (coordenadas)
México-Cuernavaca	19.1035,-99.2043
México-Puebla	19.3389, -98.6583
Oaxaca-Istmo	16.5497, -95.9815
Puebla-Córdoba	18.8198, -97.2353
Querétaro-Irapuato	20.6539, -101.2968

## Visualización de los escenarios en zonas urbanas

Los escenarios urbanos seleccionados que se muestran en la Figura 5.1, presentan en su mayoría calles en forma recta y que se conectan entre sí. Estas pistas fueron seleccionadas con el fin de recrear espacios urbanos.



(a) Calle Ficus- Cuernavaca.

(b) Centro de Tlalpan



(c) Colonia Ejidos de San Pedro Martir.



(d) Centro historico-CDMX.

(e) CENIDET- Cuernavaca Morelos.

Figura 5.1: Zonas urbanas utilizadas para realizar las simulaciones.

## Visualización de los escenarios en autopistas

Las autopistas seleccionadas que se muestran en la Figura 5.2, presentan diferentes formas del terreno. Estas vías fueron seleccionadas con el fin de recrear entornos más continuos y rápidos .

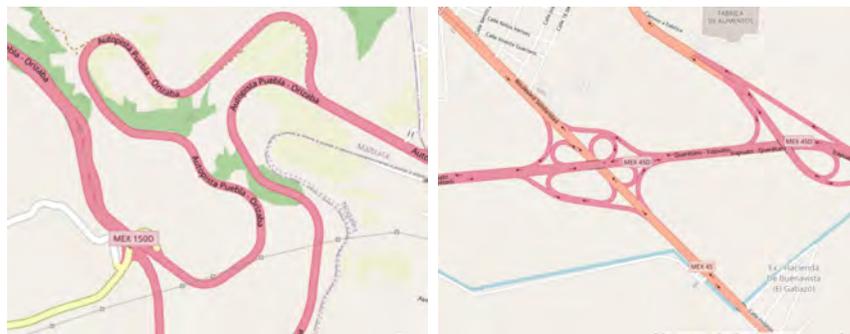


(a) Pista México-Cuernavaca.

(b) Carretera México-Puebla.



(c) Oaxaca-Itsmo.



(d) Puebla-Cordoba.

(e) Queretaro-Irapuato.

Figura 5.2: Autopistas utilizadas para realizar las simulaciones.

### 5.1.2. Métricas de evaluación

Las métricas implementadas, apoyan en la evaluación del proceso de detección y dan soporte a la eficiencia del mismo. La métrica GOSPA que tiene la finalidad de evaluar el escaneo de los sensores LIDAR y RGB-D.

#### GOSPA

La métrica GOSPA, tiene como objetivo el evaluar el rendimiento de un sistema de seguimiento proporcionando un costo escalar, siguiendo la ecuación (5.1) (García-Fernández et al., 2020).

$$GOSPA = \left[ \sum_{i=0}^m d_p^c(x_i, y_{\pi(i)}) + \frac{c^p}{a}(n - m) \right]^{\frac{1}{p}} \quad (5.1)$$

Donde  $d_c$  es la distancia basada en el corte y  $y_{m(i)}$  representan la pista asignada a la verdad  $x_i$ . La distancia de corte  $d_c$  se define como:

$$d_c = \text{minimize} \{d_b(x, y), c\} \quad (5.2)$$

donde  $d_c$  es el umbral de la distancia de corte, y  $d_b(x, y)$  es la distancia base entre la pista  $x$  y la verdad  $y$  calculada por la función de la distancia. La distancia de corte  $d_c$  es el valor más pequeño de  $db$  y  $c\alpha$  es el parámetro alfa. Cuando  $\alpha = 2$ , la métrica GOSPA se puede reducir a tres componentes:

$$GOSPA = [loc^p + miss^p + false^p]^{\frac{1}{p}} \quad (5.3)$$

El componente de localización  $loc$  se calcula como:

$$loc = \left[ \sum_{i=1}^h d_p^b(x_i, y_{\pi(i)}) \right]^{\frac{1}{p}} \quad (5.4)$$

donde  $h$  es el número de asignaciones no triviales. Una tarea trivial es cuando una pista no se asigna a ninguna verdad. El componente objetivo perdido se calcula como:

$$miss = \frac{c}{2^{\frac{1}{p}}}(n_{miss})^{\frac{1}{p}} \quad (5.5)$$

donde  $n_{miss}$  es el número de objetivos perdidos. El componente de pista falsa se calcula como:

$$false = \frac{c}{2^{\frac{1}{p}}}(n_{false})^{\frac{1}{p}} \quad (5.6)$$

donde  $n_{false}$  es el número de pistas falsas.

## Métricas para evaluar la detección de objetos

Para evaluar correctamente la detección de los objetos es preciso aplicar métricas estándar que se basan en términos de verdad, lo que quiere decir, es evaluar el conocimiento de solución del problema proporcionando una resolución ideal. Para ello deben introducirse conceptos básicos que participan en la evaluación de la detección de objetos, los cuales son (Montiel Marín et al., 2021):

- *True Positive* (TP): Los datos son correctamente detectados por el modelo.
- *False Positive* (FP): La detección es incorrecta. El modelo considera que el punto o pixel evaluado era un objetivo pero no estaba incluido en la verdad de la pista.
- *True Negative* (TN): Tanto el modelo como el terreno, considera el punto como un no objetivo.
- *False Negative* (FN): Un punto incluido en el terreno no fue detectado por el modelo.

### *Precision*

Se evalúa los valores de *precision* de cada objeto detectado, dando como resultado un valor escalar numérico para cada cuadro delimitador de clase única. En el caso de los cuadros delimitadores de varias clases, la precisión se devuelve como un vector numérico. Para su cálculo se aplica la formula:

$$Precision = \frac{TP}{TP + FP} \quad (5.7)$$

### *Recall*

Los valores de cada detección recuperan un escalar numérico para cuadros delimitadores de una sola clase. Para los cuadros delimitadores de varias clases, la recuperación se devuelve como un vector numérico. El calculo de *Recall* en general es el porcentaje de verdaderos positivos detectados y su formula es:

$$Recall = \frac{TP}{TP + FN} \quad (5.8)$$

## 5.2. Experimentaciones

Los experimentos realizados se dividieron en dos: 5 en vías urbanas y 5 en vías rápidas, esto con la finalidad de observar la eficacia del proceso de detección en entornos basados en escenarios reales. Cada escenario se construyo con diferentes obstáculos, dependiendo al entorno el cual se explica en las siguientes Secciones.

## Experimentación en zonas urbanas

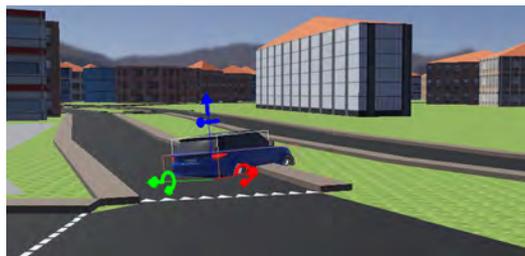
Las experimentaciones en zonas urbanas tienen la finalidad de estudiar el comportamiento del proceso de detección en escenarios donde la cantidad de obstáculos varia tanto en forma, tamaño y velocidad. Para el caso de los escenarios que se observan en la Figura 5.3 los principales objetos colocados dentro la simulación fueron ciclistas, vehículos particulares, peatones y algunos camiones. Cada objeto se les dio una velocidad estimada, como el caso de los vehículos y camiones que se les dio una velocidad no mayor a  $80 \text{ km/h}$ , mientras los ciclistas se les dio una velocidad entre  $5$  a  $10 \text{ km/h}$  y los peatones se les dio una velocidad de  $2 \text{ km/h}$ . Mientras que el vehículo principal se coloco con una velocidad de  $80 \text{ km/h}$  en un recorrido de entre  $1$  a  $2 \text{ km}$  aproximadamente, tratando de simular un ambiente típico urbano.



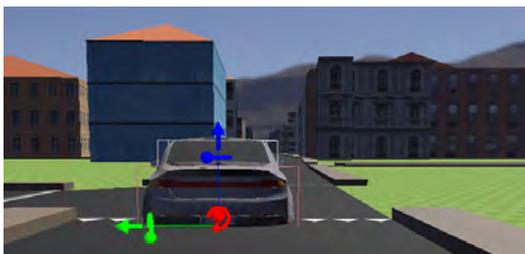
(a) Simulación de la colonia Ejidos de San Pedro Martir.



(b) Simulación de la calle Ficus.



(c) Simulación de la Avenida Palmira.



(d) Simulación del Centro Tlalpan.



(e) Simulación del Centro de la Ciudad de México.

Figura 5.3: Experimentaciones realizadas en 5 pistas de zonas urbanas.

Como se muestra en la Figura 5.4, al poner en funcionamiento el proceso de detección en cada simulación, se obtuvo la nube de puntos a través de los sensores LIDAR y RGB-D, para las 5 pistas construidas como se observa en la Figura 5.3, ambos sensores mostraron tener un buen alcance en los diferentes entornos. Posteriormente, se realizó la segmentación de la nube de puntos, mientras el vehículo está en movimiento. La tierra es identificada con los puntos color verde, mientras el vehículo encontrado en el camino es de color rosa. En cada prueba, se obtuvieron exitosamente los cuadros delimitadores de los vehículos, ciclistas y personas que estuvieron presentes en cierto punto de la simulación donde el objetivo fue tener una vista más urbana.

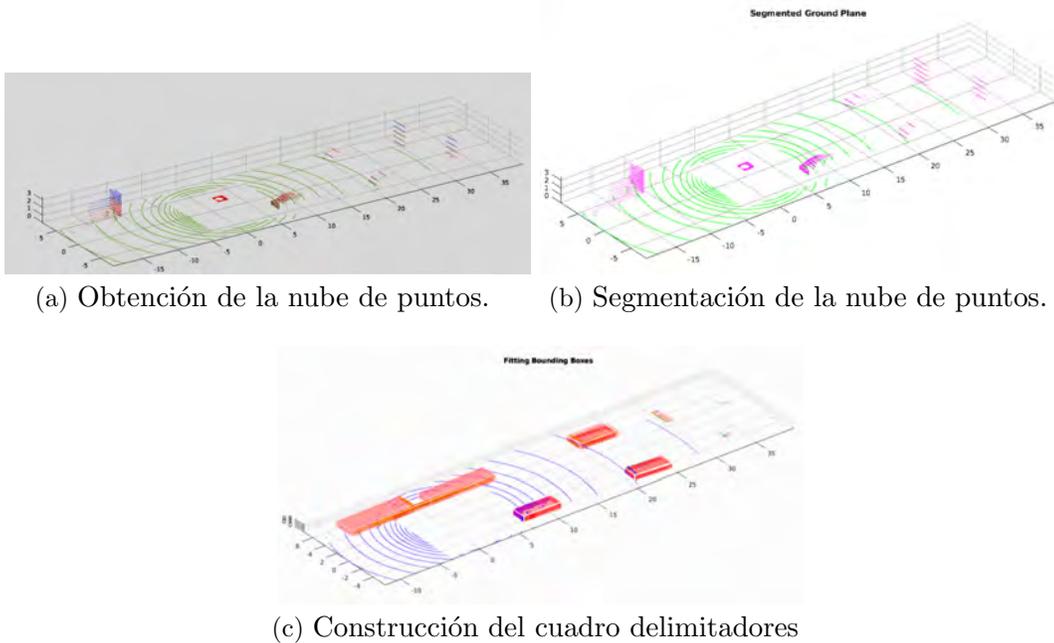


Figura 5.4: Proceso de detección durante las simulaciones.

Durante las pruebas realizadas en cada escenario, se pudo apreciar en ciertos casos que la detección de objetos pudo ser un poco más complicada debido a la cantidad de objetos presentes y la variedad de tamaños y velocidades que presentaban cada uno, incluyendo la cierta dificultad de reacción para detectar un objeto mientras el vehículo se encontraba en alguna intersección ya que la vista del escenario llegaba a ser afectada por las edificaciones que hay al rededor. Sin embargo, la reacción y alcance de los sensores fue satisfactoria, permitiendo que el proceso de detección se llevara en un rango de 0,0111 segundos aproximadamente en el caso de estos escenarios. Considerando que las condiciones de los siguientes 5 escenarios en autopistas son diferentes a las vías urbanas, en la siguiente sección explicamos los resultados obtenidos y poder realizar una debida evaluación y comparación.

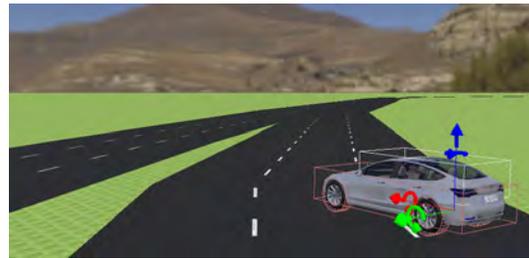
## Experimentación en autopistas

Las experimentaciones realizadas en autopistas tiene la finalidad de estudiar el comportamiento del proceso de detección en escenarios más abiertos donde si bien el tamaño de los obstáculos pueden variar, en este caso en particular lo más visible en este tipo de escenarios son diferentes vehículos a alta velocidad en un entorno más abierto como se aprecia en la Figura 5.5.

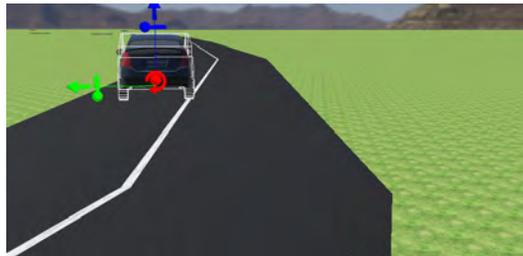
Los objetos colocados en esta simulación fueron vehículos particulares que viajaban a una velocidad de 100 a 110  $km/h$  y camiones de carga pesada que contaban con una velocidad de 80 a 95  $km/h$ , mientras que el vehículo principal fue colocado con una velocidad de 100  $km/h$  en un corrido de 1 a 3  $km$  aproximadamente.



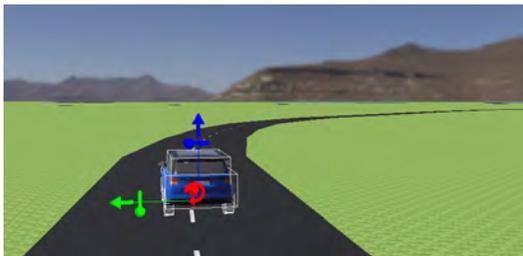
(a) Simulación de la carretera México-Cuernavaca.



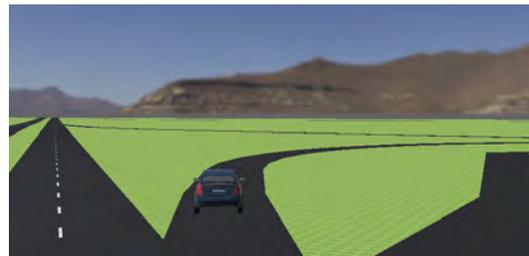
(b) Simulación de la carretera México-Puebla.



(c) Simulación de la carretera Oaxaca-Istmo.



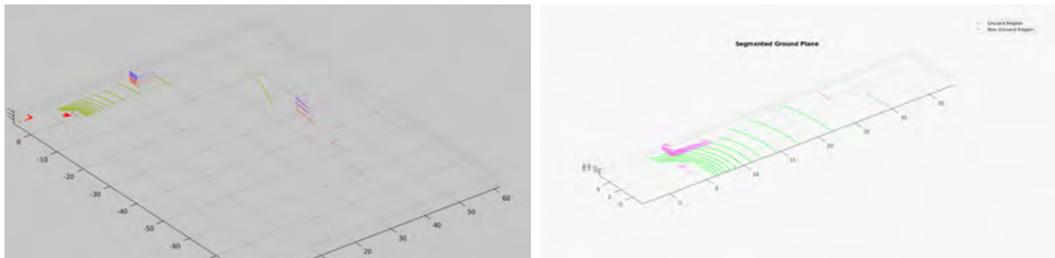
(d) Simulación del carretera Puebla-Cordoba.



(e) Simulación de la carretera Querétaro-Irapuato.

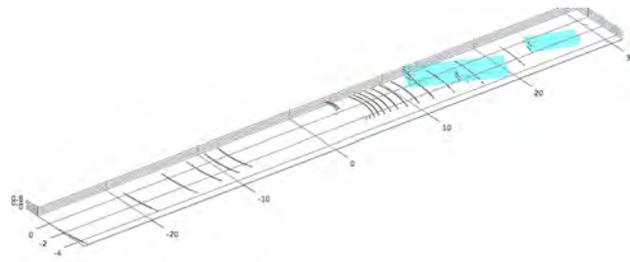
Figura 5.5: Simulaciones realizadas en 5 pistas de vías rápidas.

En la Figura 5.5, se aprecian las simulaciones donde se puso en funcionamiento el proceso de detección en cada simulación, donde, se obtuvo la nube de puntos y posteriormente, se realizó la segmentación de la nube de puntos, mientras el vehículo con los sensores está en movimiento. La tierra es identificada con los puntos color verde, mientras el vehículo encontrado en el camino es de color rosa como se aprecia en la Figura 5.6. Al obtener los cuadros delimitadores, se identifican diferentes vehículos y camiones que estuvieron presentes en cierto punto de la simulación.



(a) Nube de puntos.

(b) Nube de puntos segmentada.



(c) Cuadros delimitadores.

Figura 5.6: Experimentaciones realizadas en 5 pistas de zonas urbanas.

Los resultados obtenidos durante estas simulaciones fueron mayormente satisfactorios que en los escenarios anteriores, pues aunque los objetos iban en mayor velocidad y la visibilidad de la autopista se dificultó un poco a causa de la presencia de curvas, los obstáculos pudieron ser correctamente detectados con el buen alcance de los sensores y al proceso de detección que fue realizado en un tiempo aproximado de 0,01 segundos. En este caso la reacción fue más rápida debido a que los escenarios no tienen a la vista infraestructuras que obstruyan la visualización de los objetos que se aproximen.

Una vez observando y analizado los datos obtenidos durante el proceso de detección, se aplicó las métricas de detección establecidas, para realizar una evaluación y discusión de los resultados en la siguiente Sección.

## 5.3. Resultados

Como se mostró en las secciones anteriores las experimentaciones realizadas lograron realizar el proceso de detección satisfactoriamente y obtener las características de los objetos que se muestran en la Tabla 5.3.

Tabla 5.3: Características de los objetos detectados durante las simulaciones.

Objeto detectado	Longitud	Ancho	Altura
Camión	9 m	3 m	3.5 m
Vehículo particular	5 m	2 m	1.5 m
Persona	0.24 m	0.45 m	1.5 m
Ciclista	1.63m	0.55 m	1.53 m

Dentro de esta sección se describen de forma resumida los resultados obtenidos de la evaluación de los datos obtenidos en los diferentes escenarios.

### 5.3.1. Resultados en zonas urbanas

Aplicando la métrica de evaluación GOSPA a cada simulación, se obtuvieron resultados donde los datos falsos registrados en la grafica de la Figura 5.7 y los datos perdidos registrados en la grafica de Figura 5.8, son nulos en ambos casos y sustenta que ambos sensores realizan correctamente el escaneo del ambiente sin tener perdidas de los datos. En caso de existir un registro de datos perdidos como se aprecia en la Figura 5.8, es debido a que el vehículo ya se encontraba al ras de la simulación por lo que ya no encontraba ningún dato. Sin embargo, los resultados obtenidos de la evaluación son satisfactorios ya que sustentan el funcionamiento de los sensores en los escenarios urbanos construidos.

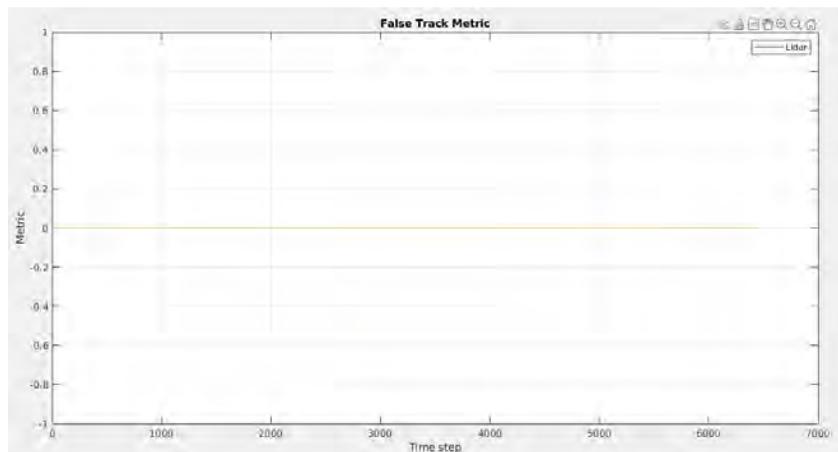


Figura 5.7: Datos falsos en zonas urbanas.

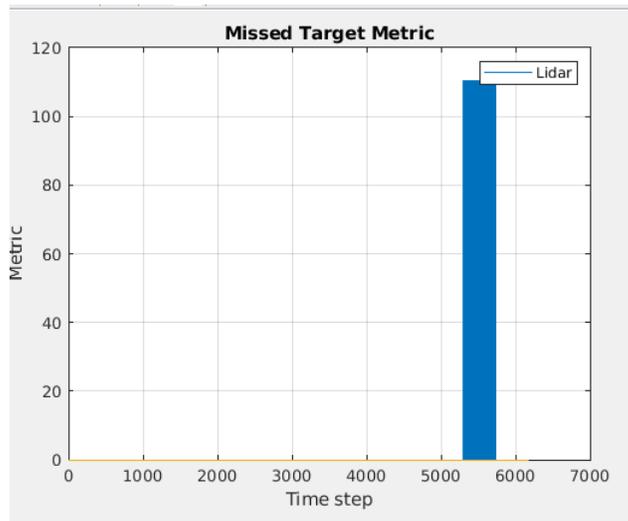


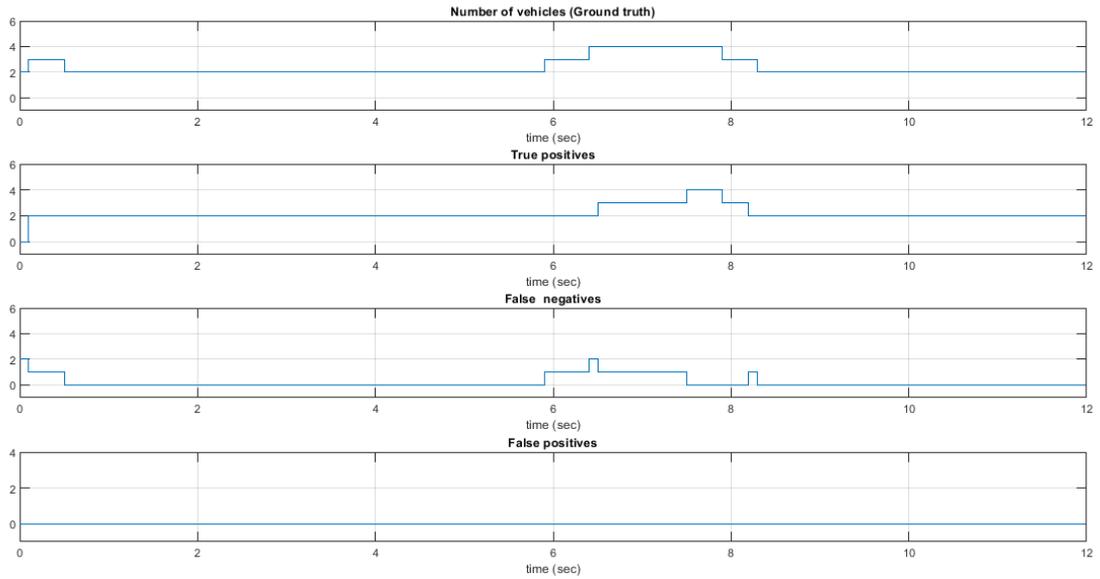
Figura 5.8: Datos perdidos en zonas urbanas.

En la Tabla 5.4 se observa la *precision* y el *recall* del proceso de detección en el escenario, dando un satisfactorios resultados confirmando el buen funcionamiento para obtener la información de los objetos en el escenario.

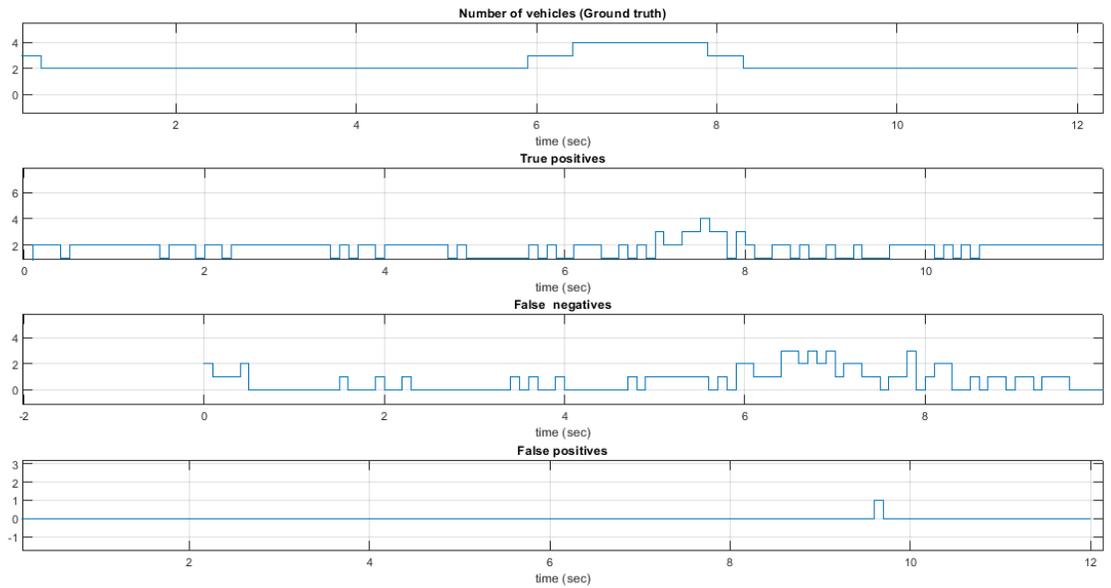
Tabla 5.4: *Precision y Recall* de pistas urbanas.

Pista	Precision	Recall
Ejidos de San Pedro Mártir- CDMX	0.9	0.899
Calle ficus- Cuernavaca Morelos	0.96	0.89
Centro de Tlalpan- CDMX	0.88	0.87
Avenida Palmira	0.88	0.87
Centro Histórico-CDMX	0.97	0.96

Los objetos detectados, pudieron ser registrados a través de las gráfica que aparecen en la Figura 5.9, ya que representa la cantidad de objetos detectados en el tiempo mientras el vehículo se deslizaba a través del escenario, registrando los datos verdaderos positivos, falsos negativos y falsos positivos. Estas graficas nos ayudan a demostrar que a pesar de que el vehículo se encuentre en movimiento, este detecta los objetos que están en dentro de su marco de visión.

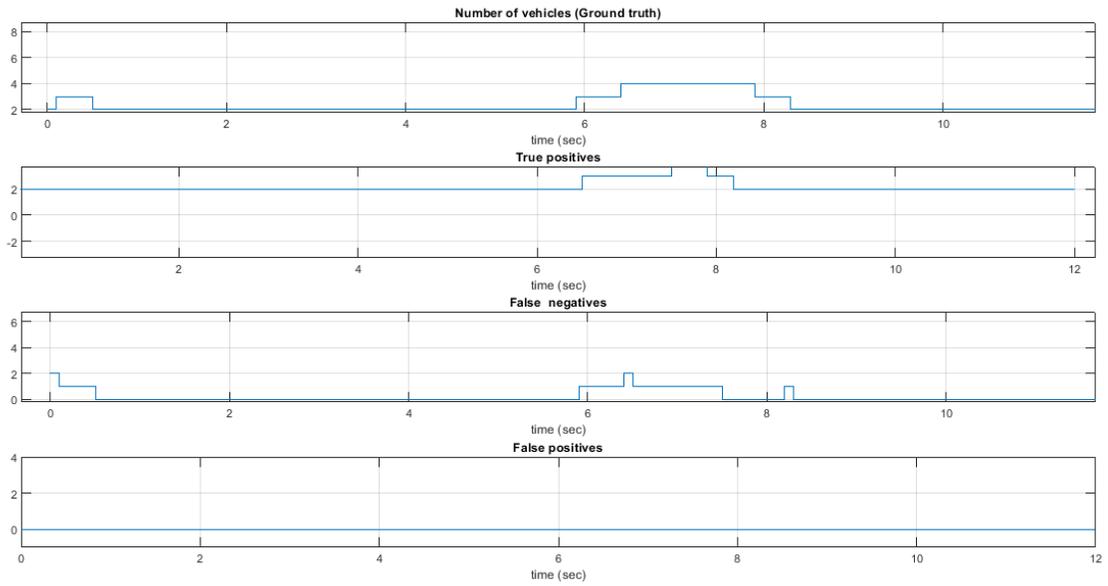


(a) Objetos detectados en un lapso de tiempo en la colonia Ejidos de San Pedro Mártir.

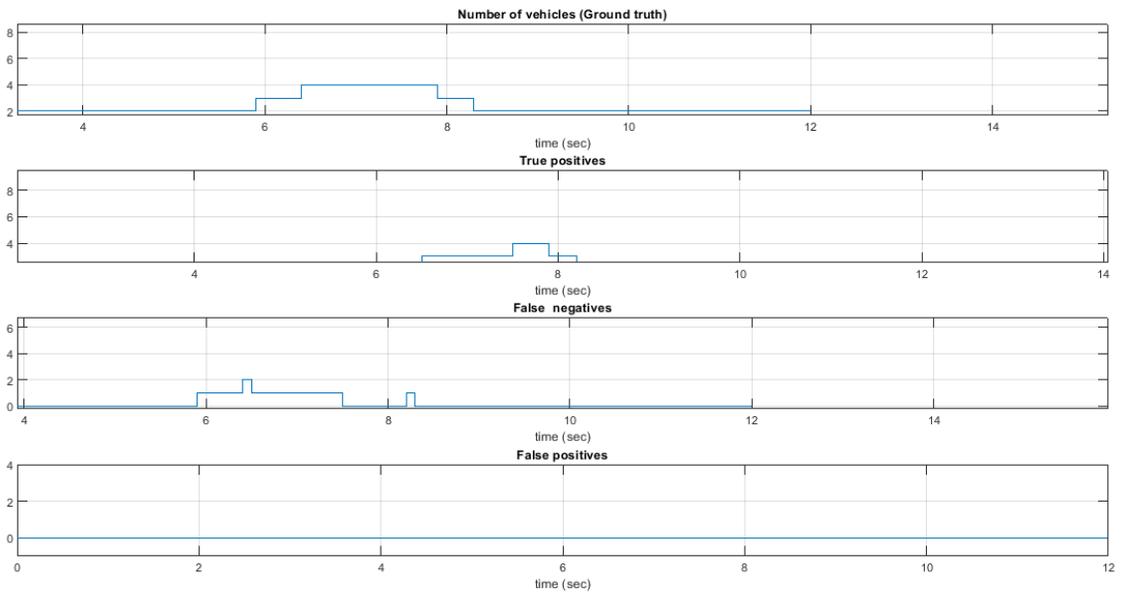


(b) Objetos detectados en un lapso de tiempo en la calle Ficus.

Figura 5.9: Cantidad de objetos detectados en vías urbanas.

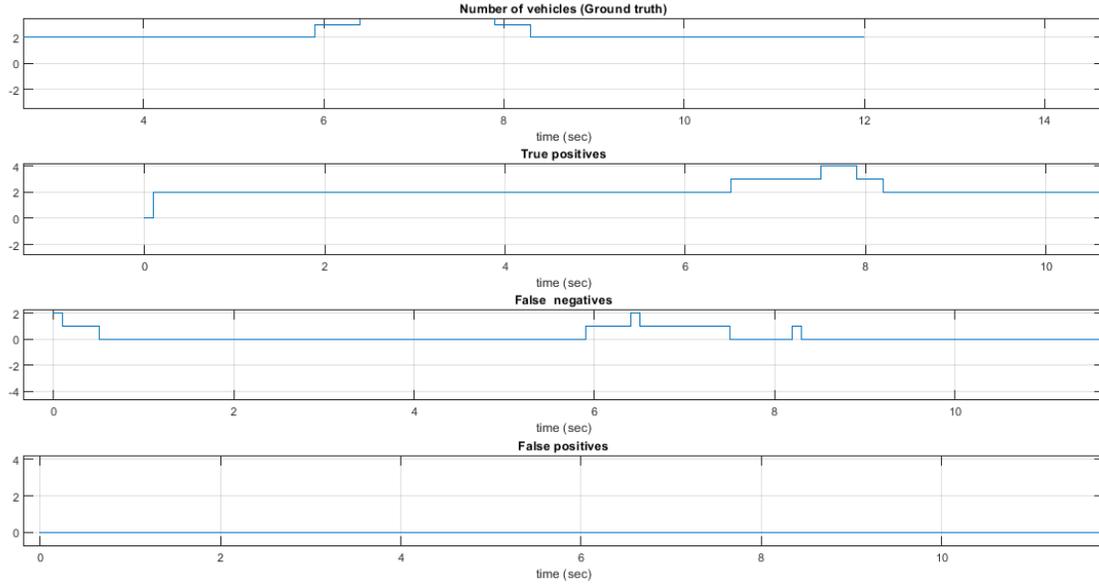


(a) Objetos detectados en un lapso de tiempo en la Avenida Palmira.



(b) Objetos detectados en un lapso de tiempo en el Centro de Tlalpan.

Figura 5.10: Cantidad de objetos detectados en vías urbanas-continuación.



(a) Objetos detectados en un lapso de tiempo en la Ciudad de México.

Figura 5.11: Cantidad de objetos detectados en vías urbanas-continuación.

### 5.3.2. Resultados en autopistas

Aplicando la métrica de evaluación GOSPA a cada simulación, se obtuvieron resultados donde los datos falsos registrados en la grafica de la Figura 5.12 y los datos perdidos registrados en la grafica de Figura 5.13, son nulos en ambos casos y sustenta que ambos sensores realizan correctamente el escaneo del ambiente sin tener perdidas de los datos. En caso de existir un registro de datos perdidos como se aprecia en la Figura 5.13, es debido a que el vehículo ya se encontraba al ras de la simulación por lo que ya no encontraba ningún dato. Sin embargo, los resultados obtenidos de la evaluación son satisfactorios ya que sustentan el funcionamiento de los sensores en los escenarios construidos en autopistas.

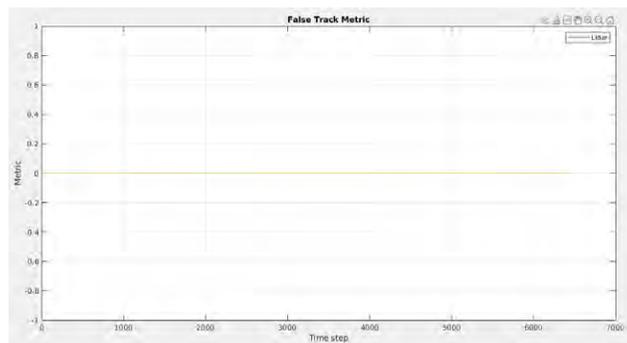


Figura 5.12: Datos falsos autopistas.

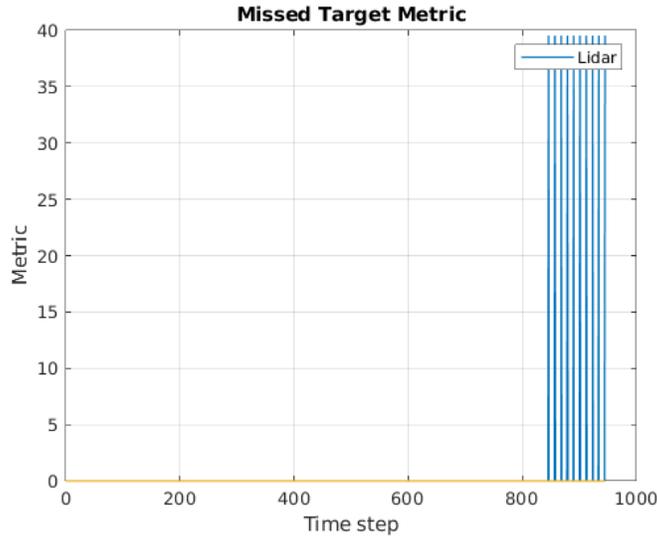
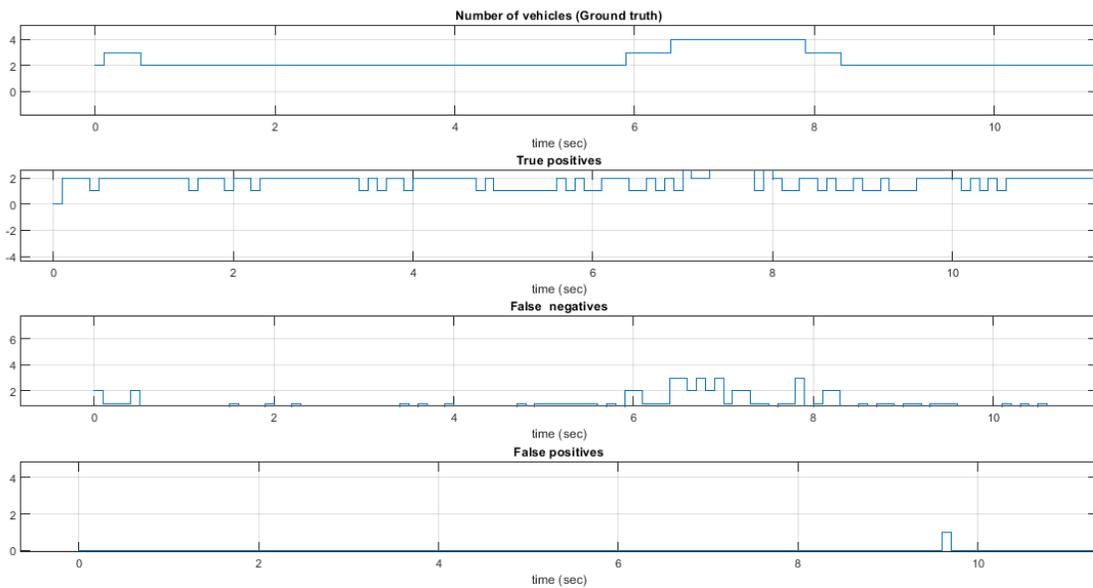


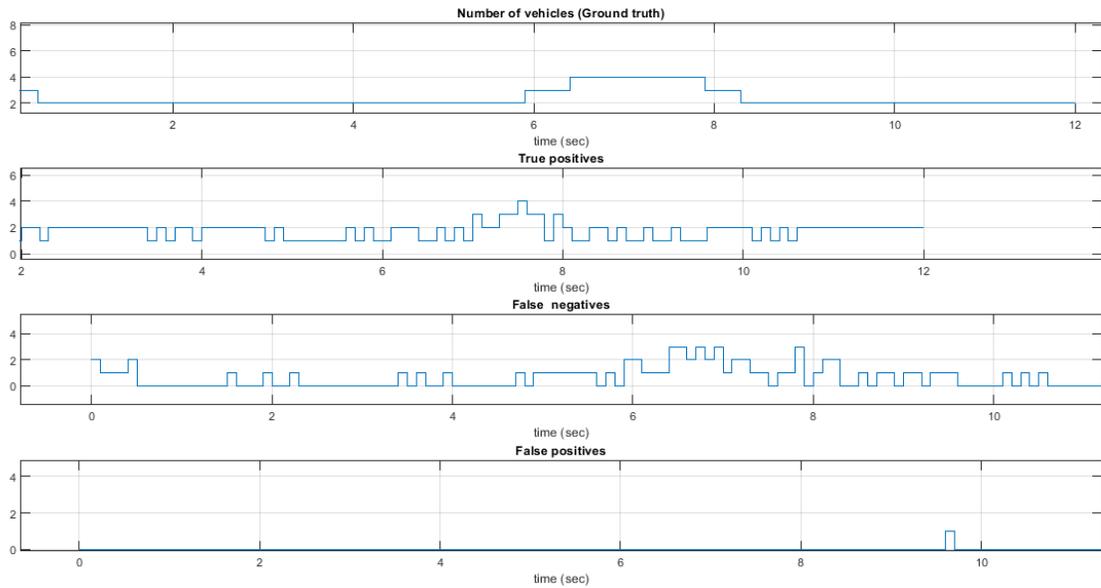
Figura 5.13: Datos perdidos en vías rápidas.

Los objetos detectados, pudieron ser registrados a través de las gráfica que aparecen en la Figura 5.16, ya que representa la cantidad de objetos detectados en el tiempo mientras el vehículo se deslizaba a través del escenario, registrando los datos verdaderos positivos, falsos negativos y falsos positivos. Estas graficas nos ayudan a demostrar que a pesar de que el vehículo se encuentre en movimiento, este detecta los objetos que están en dentro de su marco de visión.

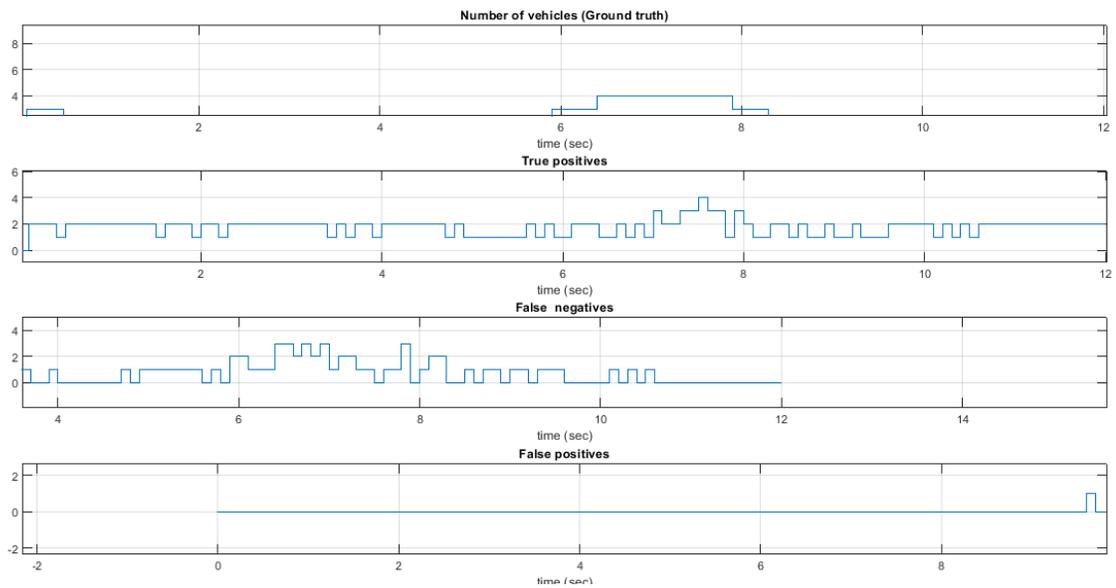


(a) Objetos detectados en un lapso de tiempo en la autopista México-Cuernavaca.

Figura 5.14: Cantidad de objetos detectados en vías rápidas.

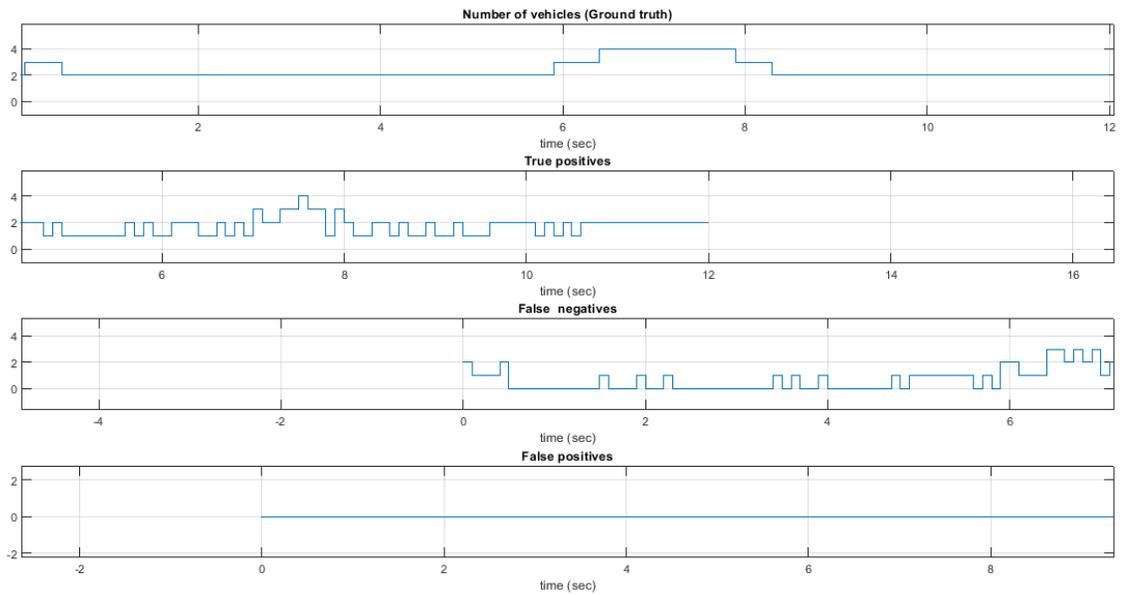


(a) Objetos detectados en un lapso de tiempo en la pista México-Puebla.

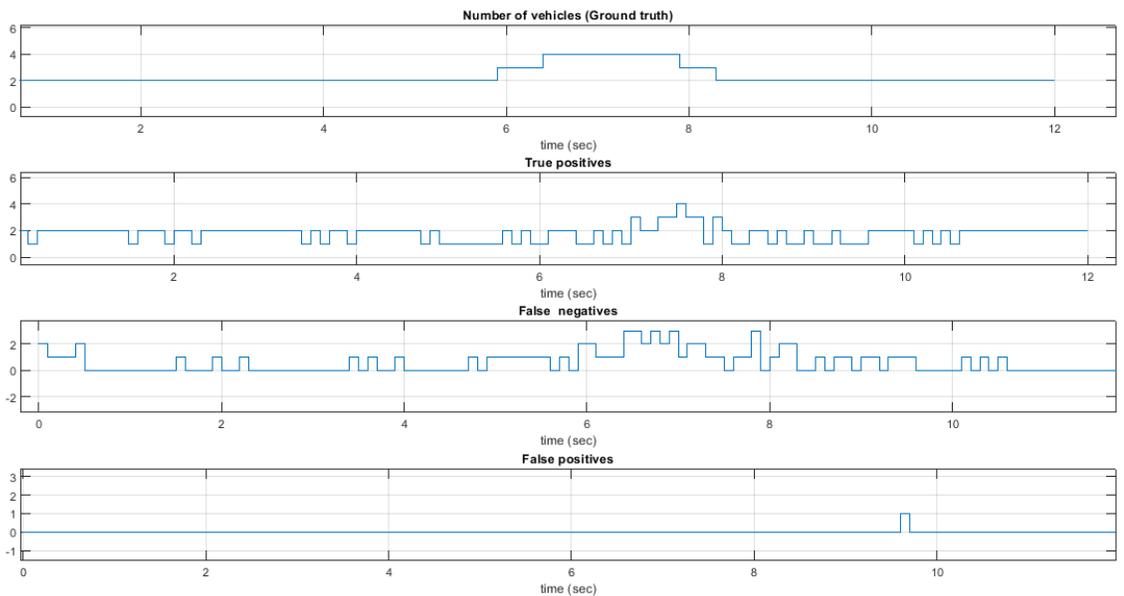


(b) Objetos detectados en un lapso de tiempo en la pista Oaxaca-Istmo.

Figura 5.15: Cantidad de objetos detectados-continuación.



(a) Objetos detectados en un lapso de tiempo en la pista Puebla-Córdoba.



(b) Objetos detectados en un lapso de tiempo en pista Querétaro-Irapuato.

Figura 5.16: Cantidad de objetos detectados.

Finalmente en la Tabla 5.5 se observa la precisión y recuperación del proceso de detección en el escenario, dando son satisfactorios los resultados, ya que la precisión y recuperación a pesar de ser menor que las experimentaciones anteriores, se sigue confirmando el buen funcionamiento para obtener la información de los objetos en los escenarios de vías rápidas. Cabe resaltar que uno de los motivos principales por las que se considera que la precisión llega a ser menor en estos escenarios, se debe a dos factores primordiales, que

son la velocidad de los vehículos y las curvas dentro de la simulación, ya que al ser mayores y el proceso debe reaccionar más rápido la precisión tiende a disminuir.

Tabla 5.5: *Precision y Recall* de autopistas.

Vías rápidas		
México-Cuernavaca	0.95	0.89
México-Puebla	0.93	0.87
Oaxaca-Istmo	0.8	0.8
Puebla-Córdoba	0.8	0.8
Querétaro-Irapuato	0.85	0.82

### 5.3.3. Discusión de los resultados

Las experimentaciones realizadas se dividieron en dos categorías, la primera en zonas urbanas y la segunda en vías rápidas. El principal objetivo en dividir las experimentaciones en dos es demostrar que se pueden detectar los diferentes objetos que participan en cada entorno.

Para los escenarios urbanos, la mayoría fueron calles que presentan formas rectas y en muy pocas ocasiones algún tipo de curva poco pronunciada, además de que los objetos tienen una velocidad menor, y de diferentes tamaños. Aunque los resultados son bastante favorables ya que los objetos se encuentran en un rango de 0.9 a 0.88 de *precision* y 0.96 a 0.87 de *recall*, el proceso llega a ser más tardado con un tiempo aproximado de 0.0111 segundos a causa de las estructuras como edificios y casas que intervienen en la visualización de los obstáculos, principalmente en una intersección o una esquina donde el vehículo puede dar vuelta para dirigirse a su destino. Sin embargo, el buen alcance de los sensores y la velocidad de los objetos permitieron realizar correctamente el proceso de detección y obtener la información de cada objeto a través del cuadro delimitador.

Por otro lado en el caso de las experimentaciones realizadas en autopistas donde el vehículo va a una mayor velocidad y se encuentra con objetos como son, vehículos particulares, camiones de carga, en algunas ocasiones llega a encontrarse con ciclistas o peatones, para estas pruebas se tomó la decisión de solo poner vehículos o camiones y en un caso ciclistas. Lo más relevante en estas pruebas realizadas en este tipo de vías fue la forma de cada una, ya que como se pudo observar en cada pista la forma fue variada y lo más presente fueron las curvas, que en algunos casos son más pronunciadas que otras. Por lo que al aplicar el proceso de detección se obtuvo como resultado una *precision* de 0.95 a 0.8 y un *recall* de 0.89 a 0.8. Al analizar los resultados de la aplicación de este proceso, destaca que su tiempo de realización es de 0.01 segundos, mucho más rápido que en los escenarios de pistas urbanas, a pesar de que el proceso es más rápido la precisión junto a la recuperación disminuye, y estos

principalmente se encuentran afectados por el factor de velocidad en los que se encuentran los obstáculos y el vehículo principal con los sensores. Si bien en este tipo de pistas no hay la interferencia de construcciones urbanas que permiten aprovechar mejor el alcance de los sensores, se presentó aun una dificultad de visualización al pasar por curvas, principalmente muy pronunciadas. A pesar de ello, no fue imposible detectar cada objeto en las simulaciones.

Finalmente podemos decir que las experimentaciones fueron bastante efectivas en cada escenario, logrando obtener la información de cada objeto y una pre-visualización a un entorno real, ya que cada pista utilizada fue basada a un escenario de alguna colonia o carretera que conecte con otra región o estado del país. Cada resultado obtenido en cada pista pudo sustentar el proceso de detección de objetos para desplazamiento vehicular propuesto ya que en la pérdida de información por parte de los sensores es nula y el promedio de *precision* para ambas pistas fue de 0.99 a 0.87 y el *recall* fue de 0.95 a 0.8.

# Capítulo 6

## Conclusiones

### 6.1. Objetivos y Alcances logrados

En la Tabla 6.1, se muestran las actividades en relación a los objetivos planteados para el desarrollo de la tesis.

### 6.2. Resultados del trabajo de investigación

#### Aportaciones

1. Metodología para mejorar el proceso de detección usando dos algoritmos de agrupamiento conjunto a la red neuronal *PointNet*.
2. Simulaciones basadas en pistas de entornos reales del país.
3. Proceso para la obtención de la nube de puntos a través de los sensores LIDAR y RGB-D, con el fin de minimizar el ruido que puede estar presente en el ambiente.

Tabla 6.1: Objetivos Realizados

Número	Objetivo	Comentario
1	Procesar la información sensorial en tiempo real a fin de segmentar los planos del suelo.	La información obtenida por los sensores LIDAR y RGB-D, fueron procesadas a través del algoritmo RANSAC, dividiendo la nube de puntos entre los puntos del suelo y su nivel, permitiendo agrupar estos datos por medio de DBSCAN.
2	Utilizar técnicas de visión por computadora y técnicas convencionales del estado del arte.	Se consultó dentro del estado del arte diferentes técnicas de visualización y procesamiento de señales que permitieron el desarrollo del proyecto por medio de diferentes programas y algoritmos que permitieron cumplir este objetivo.
3	Realizar la caracterización de los sensores a nivel simulación, para que haya una correspondencia con las medidas de los vehículos y objetos detectados.	La calibración de los sensores a nivel simulación, corresponde a una precisión de 0.9, un alcance de 100 m y una elevación de 1.5 m.
4	Implementar el proceso de detección en una simulación basada en un entorno real.	El proceso de detección fue aplicado en un total de 10 escenarios, divididos entre dos experimentaciones. En las experimentaciones de vías urbanas se usaron 5 escenarios y en las experimentaciones en autopistas se usaron otros 5 escenarios, todos basados en algún punto de la República Mexicana.

### 6.2.1. Conclusiones

Al finalizar la implementación y experimentación del proceso de detección en diferentes escenarios virtuales, basados en entornos reales alrededor de la República Mexicana, se permitió que las pruebas realizadas tomarán mayor complejidad al ser pistas con formas irregulares como las carreteras que se conectan a otros estados y con vehículos a mayor velocidad. Mientras que las pistas dentro de un entorno más urbano no solo hay carros particulares, sino también ciclistas y transeúntes. Dando como resultados el apreciar un comportamiento del proceso de detección en escenarios con diferentes tipos de objetos.

Los objetos detectados durante el desplazamiento vehicular, demostraron que pueden identificarse a través de sus características que son, la altura, longitud y ancho de cada obstáculo. Para poder obtener estas características es importante la construcción de un cuadro delimitador, obtenido después del proceso de detección. Por lo que en esta investigación su principal enfoque fue estudiar y comprender cada proceso involucrado en la detección de objetos.

El primer proceso involucrado es la obtención de una nube de puntos, que en un inicio solamente se contemplaba en obtener con el sensor LIDAR, pero después de una investigación y algunas pruebas se encontró algunos problemas que podrían presentarse a causa del ruido en el ambiente y afectan la obtención de datos, por lo que al buscar una solución se optó por combinar el sensor con otro llamado RGB-D, el cual apoya al proceso de obtención de la nube de puntos y disminuye el ruido, sin necesidad de utilizar algún tipo de algoritmo.

Para el segundo proceso en donde se tratara la nube de puntos obtenida, participan dos algoritmos de agrupamiento conocidos como RANSAC y DBSCAN. Si bien ambos son algoritmos básicos en la tarea de agrupación, al trabajar en conjunto con ellos se obtienen muy buenos resultados, ya que en el caso de RANSAC su principal función fue obtener los puntos útiles y remover los puntos de tierra al trabajar con los datos atípicos de la nube de puntos. Mientras que DBSCAN, al recibir una nube de puntos previamente procesada y eliminado los datos atípicos que le dificultan su funcionamiento, puede realizar un agrupamiento más preciso y rápido. Al aplicar en conjuntos estos algoritmos, se demuestra que puedes agrupar cada objeto, mientras se esta en movimientos, sin la intervención de una red neuronal.

Para el último proceso, donde se construye el cuadro delimitador se tomo en cuenta dos simples factores, el área y la cercanía. Estos con el fin de confirmar los puntos agrupados que estarán dentro de cada cuadro, los cuales demostraron en las experimentaciones obtener algunas de las características de los objetos. Adicionalmente se agrego la implementación de una red neuronal conocida como *PointNet*, esta con la única finalidad de etiquetar cada objeto por su nombre. La razón de elegir esta red es su ventaja de trabajar con el mismo proceso anteriormente mencionado y a partir de los datos ya obtenidos colocar el nombre de cada objeto. Aunque la adición es buena, al ser una red neuronal, necesita aún pasar por un proceso de entrenamiento para tener mejores resultados, por lo que esta implementación cuenta con pocas pruebas.

Finalmente, las experimentaciones realizadas, demostraron ser satisfactorias y el proceso de detección logró cumplir con su cometido de obtener las características de cada objeto mientras el vehículo se encontraba en movimiento en un tiempo aproximado de 0,01 segundos. Las métricas aplicadas tanto para la evaluación de los sensores, donde se demostró tener una nula presencia de datos falsos y pérdida de datos, como la obtención de una precisión de 0,9 y una recuperación de 0,89 para los cuadros delimitadores sustentan que el funcionamiento de este proceso de detección es de un 96 % efectivo.

### 6.2.2. Trabajos a futuros

Como actividades futuras para alimentar y continuar con este proyecto se recomienda lo siguiente:

- Llevar el proceso de detección a un entorno físico
- Entrenar la red neuronal *PointNet* para mejorar la obtención de resultados
- Realizar las pruebas en entornos reales

# Bibliografía

- Arya Senna Abdul Rachman, A. (2017). 3d-lidar multi object tracking for autonomous driving: Multi-target detection and tracking under urban road uncertainties.
- Castaño Amorós, J. (2020). Detección basada en datos 3d de objetos urbanos.
- Cavagna, A., Melillo, S., Parisi, L., and Ricci-Tersenghi, F. (2019). Sparta tracking across occlusions via partitioning of 3d clouds of points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Chauhan, N. S. (2020). Dbscan clustering algorithm in machine learning.
- Chávez Cobián, A. L. (2020). Sistema de detección y evasión de obstáculos por medio de un lidar 360° para un sistema aéreo no tripulado.
- Chiu, H.-k., Prioletti, A., Li, J., and Bohg, J. (2020). Probabilistic 3d multi-object tracking for autonomous driving. *arXiv preprint arXiv:2001.05673*.
- Dussan Narvaez, G. A., Mosquera Victoria, C. A., et al. (2020). Detección y seguimiento de múltiples objetos en tiempo real para vehículos autónomos.
- Eduardo Morales, H. J. E. (2018). Clustering.
- Egi, Y. and Otero, C. E. (2019). Machine-learning and 3d point-cloud based signal power path loss model for the deployment of wireless communication systems. *IEEE Access*, 7:42507–42517.
- García-Fernández, Á. F., Rahmathullah, A. S., and Svensson, L. (2020). A metric on the space of finite sets of trajectories for evaluation of multi-target tracking algorithms. *IEEE Transactions on Signal Processing*, 68:3917–3928.
- Gong, Z., Lin, H., Zhang, D., Luo, Z., Zelek, J., Chen, Y., Nurunnabi, A., Wang, C., and Li, J. (2020). A frustum-based probabilistic framework for 3d object detection by fusion of lidar and camera data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 159:90–100.
- Gönültaş, F., ATİK, M. E., and Duran, Z. (2020). Extraction of roof planes from different point clouds using ransac algorithm. *International Journal of Environment and Geoinformatics*, 7(2):165–171.

- Kocur, V. and Ftáčnik, M. (2020). Detection of 3d bounding boxes of vehicles using perspective transformation for accurate speed measurement. *arXiv preprint arXiv:2003.13137*.
- Leng, Z., Li, S., Li, X., and Gao, B. (2020). An improved fast ground segmentation algorithm for 3d point cloud. In *2020 Chinese Control And Decision Conference (CCDC)*, pages 5016–5020. IEEE.
- Li, S.-S. (2020). An improved dbscan algorithm based on the neighbor similarity and fast nearest neighbor query. *IEEE Access*, 8:47468–47476.
- Li, W., Pan, C., Zhang, R., Ren, J., Ma, Y., Fang, J., Yan, F., Geng, Q., Huang, X., Gong, H., et al. (2019). Aads: Augmented autonomous driving simulation using data-driven algorithms. *Science Robotics*, 4(28).
- Li, Y., Díaz, M., Morantes, S., and Dorati, Y. (2018). Vehículos autónomos: Innovación en la logística urbana. *Revista de Iniciación Científica*, 4(1):34–39.
- Liu, S., Li, L., Tang, J., Wu, S., and Gaudiot, J.-L. (2020). Creating autonomous vehicle systems. *Synthesis Lectures on Computer Science*, 8(2):i–216.
- Lopez Torrijos, F. (2021). Random sample consensus (ransac).
- Luna Álvarez, J. A. (2020). *Conducción Autónoma de un Vehículo Simulado mediante un Modelo de Red Neuronal Convolutiva Recurrente*. PhD thesis.
- Manghat, S. K. (2019). *Multi Sensor Multi Object Tracking in Autonomous Vehicles*. PhD thesis, Purdue University Graduate School.
- MATLAB (2020). Track Vehicles Using Lidar: From Point Cloud to Track List - MATLAB & Simulink - MathWorks América Latina.
- McQueen, B. (2019). Unsettled topics concerning the field testing of automated driving systems. Technical report, SAE Technical Paper.
- Medina, V., Jhonas, J., et al. (2020). Desarrollo de un asistente de conducción longitudinal mediante un algoritmo de aprendizaje profundo.
- Michel, O. (2004). Cyberbotics ltd. webots™: professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):5.
- Montiel Marín, S. et al. (2021). Multi-object tracking system based on lidar and radar for intelligent vehicles applications.
- Mu, L., Yao, P., Zheng, Y., Chen, K., Wang, F., and Qi, N. (2020). Research on slam algorithm of mobile robot based on the fusion of 2d lidar and depth camera. *IEEE Access*, 8:157628–157642.

- NHTSA (2019). Vehículos automatizados para la seguridad — NHTSA.
- Park, S.-W., Patil, K., Wilson, W., Corless, M., Choi, G., and Adam, P. (2020). Creating driving scenarios from recorded vehicle data for validating lane centering system in highway traffic. Technical report, SAE Technical Paper.
- Pecho, P., Škvareková, I., Ažaltovič, V., and Bugaj, M. (2019). Uav usage in the process of creating 3d maps by rgb spectrum. *Transportation Research Procedia*, 43:328–333.
- Ponce Herrera, P. D. (2019). Reconstrucción de entorno 3d mediante un sensor lidar. B.S. thesis, Quito: Universidad de las Américas, 2019.
- Prophet, R., Deligiannis, A., Fuentes-Michel, J.-C., Weber, I., and Vossiek, M. (2020). Semantic segmentation on 3d occupancy grids for automotive radar. *IEEE Access*, 8:197917–197930.
- Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660.
- Rendón Castro, Á. A. (2020). Sistema Embebido para Asistencia de Conducción Basado en Lógica Difusa Tipo-2.
- Rozsa, Z. and Sziranyi, T. (2018). Obstacle prediction for automated guided vehicles based on point clouds measured by a tilted lidar sensor. *IEEE Transactions on Intelligent Transportation Systems*, 19(8):2708–2720.
- Schubert, E., Sander, J., Ester, M., Kriegel, H. P., and Xu, X. (2017). Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21.
- Sun, P., Zhao, X., Xu, Z., Wang, R., and Min, H. (2019). A 3d lidar data-based dedicated road boundary detection algorithm for autonomous vehicles. *IEEE Access*, 7:29623–29638.
- Tsai, C.-Y. and Tsai, S.-H. (2018). Simultaneous 3d object recognition and pose estimation based on rgb-d images. *IEEE Access*, 6:28859–28869.
- Wang, C., Ji, M., Wang, J., Wen, W., Li, T., and Sun, Y. (2019). An improved dbscan method for lidar data segmentation with automatic eps estimation. *Sensors*, 19(1):172.
- Weon, I.-S., Lee, S.-G., and Ryu, J.-K. (2020). Object recognition based interpolation with 3d lidar and vision for autonomous driving of an intelligent vehicle. *IEEE Access*, 8:65599–65608.

- Xie, J., Xu, Y., Zheng, Z., Zhu, S.-C., and Wu, Y. N. (2021). Generative pointnet: Deep energy-based learning on unordered point sets for 3d generation, reconstruction and classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14976–14985.
- Yin, H., Wang, Y., Ding, X., Tang, L., Huang, S., and Xiong, R. (2019). 3d lidar-based global localization using siamese neural network. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1380–1392.
- Zhang, C. and Czarnuch, S. (2020). Perspective independent ground plane estimation by 2d and 3d data analysis. *IEEE Access*, 8:82024–82034.
- Zhang, C., Huang, T., and Zhao, Q. (2019). A new model of rgb-d camera calibration based on 3d control field. *Sensors*, 19(23):5082.
- Zhang, X., Xu, W., Dong, C., and Dolan, J. M. (2017). Efficient l-shape fitting for vehicle detection using laser scanners. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 54–59. IEEE.
- Zheng, J., Yang, S., Wang, X., Xia, X., Xiao, Y., and Li, T. (2019). A decision tree based road recognition approach using roadside fixed 3d lidar sensors. *IEEE Access*, 7:53878–53890.

# Apéndice A



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO

## EL TECNOLÓGICO NACIONAL DE MÉXICO A TRAVÉS DEL CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO

OTORGA EL PRESENTE

# RECONOCIMIENTO

A

**ANDREA PALACIOS CERVANTES, DANTE MÚJICA VARGAS**

Por la presentación del artículo "ANÁLISIS DEL USO DE SENSORES LIDAR Y RGB-D PARA LA DETECCIÓN DE OBJETOS DURANTE EL DESPLAZAMIENTO VEHÍCULAR" en la 6a. Jornada de Ciencia y Tecnología Aplicada realizada del 21 al 23 de abril de 2021.

Cuernavaca, Morelos, 21 de abril de 2021.

**DRA. YESICA IMELDA SAAVEDRA BENÍTEZ**  
DIRECTORA

**DR. ARTURO ABÚNDEZ PLIEGO**  
COORDINADOR GENERAL DE LA 6A JCYTA



Sello Digital:

*MjAyMceANjHHgEpDVEEwMDYxMjHHgEFuZHJ1YSBQYWxhY21vcyBDZXJ2YW50ZXMGYW5kIERhbnR1IE3DumpppY2EgVmFyZ2Fz4E2QS4gSk9STkFEQSBERSBDSUVOQ01BIFkgVEVDTk9MT0FDjUEgQVBMSUNBREHHgFRFQ05NL0NFtk1ERVTHgFJFQ09OT0NJTU1FT1Rpx4BFWFBPU01UT1LHgTIwMjEwNA==*

JCTA00612021

<http://constancias.cenidet.edu.mx>



Interior Internado Palmira S/N, Col. Palmira,  
C. P. 62490, Cuernavaca, Morelos  
Tel. (01) 777 3 62 77 70, ext. 4201,  
e-mail: [cyd\\_cenidet@tecnm.mx](mailto:cyd_cenidet@tecnm.mx)

[www.tecnm.mx](http://www.tecnm.mx) | [www.cenidet.tecnm.mx](http://www.cenidet.tecnm.mx)



Figura 6.1: Reconocimiento del artículo presentado en la 6a. Jornada de Ciencia y Tecnología Aplicada.

# Apéndice B



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO

## EL TECNOLÓGICO NACIONAL DE MÉXICO A TRAVÉS DEL CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO

OTORGA EL PRESENTE

### RECONOCIMIENTO

A

**ANDREA PALACIOS CERVANTES, DANTE MÚJICA VARGAS, NOÉ  
ALEJANDRO CASTRO SÁNCHEZ AND NIMROD GONZÁLEZ FRANCO**

POR LA PRESENTACIÓN DEL ARTÍCULO:  
"DETECCIÓN DE OBJETOS DURANTE EL DESPLAZAMIENTO VEHICULAR, APLICANDO  
RANSAC Y DBSCAN"  
EN EL MARCO DE LA 7ª JORNADA DE CIENCIA Y TECNOLOGÍA APLICADA, CELEBRADO  
DEL 17 AL 19 DE NOVIEMBRE DE 2021, EN EL TECN/CENIDET.

CUERNAVACA, MORELOS., 19 DE NOVIEMBRE DE 2021



**DRA. YESICA IMELDA SAAVEDRA BENÍTEZ**  
DIRECTORA

Sello Digital:

gcI9aV67t4bCaVjnOI7EM5OfdJ4wbX8EDRkUiKgxNUqEIr8bYLk/ZWZHHud7W/JxzRSYB6g9emwRY8cyhMTe4P  
j5iI+29zC9jwaK/v/RL0IpJ2h2Gv/qa2HBAr7cVoYviObMz0RiE8D++dK0ToQBeFnnoA1r1VwzMhRQFH56ZxHl  
zaDt7r5JMq7XR4s9E4iA7v7gI6uKvYTBxGi2+Bb1hLF6sZ0DSi81PeUZuIyTwBw6nZNSAlnDANb4Cqc2fuPiKG  
1CUD+mEYs+o/cmNLPREVLFFLNvQ/JC+Jcjuhd4JqqboWaEu+bkF/17grxFjS8zpKgP8s/S8ccrEm14NZX6nw==

JCYTA4821  
<http://constancias.cenidet.tecnm.mx>



Interior Internado Palmira S/N, Col. Palmira,  
C. P. 62490, Cuernavaca, Morelos  
Tel. (01) 777 3 62 77 70, ext. 4201,  
e-mail: [dir\\_cenidet@tecnm.mx](mailto:dir_cenidet@tecnm.mx)

[www.tecnm.mx](http://www.tecnm.mx) | [www.cenidet.tecnm.mx](http://www.cenidet.tecnm.mx)



Figura 6.2: Reconocimiento del artículo presentado en la 7a. Jornada de Ciencia y Tecnología Aplicada.

# Apéndice C

N2020086

	ACEPTACIÓN DE LA PROPUESTA DE TESIS	Código: CENIDET-AC-004-M9
		Revisión: 1
	Referencia a la Norma ISO 9001:2008 7.1, 7.2.1, 7.5.1, 7.6, 8.1, 8.2.4	Página 1 de 1

Cuernavaca, Mor., a 20 de enero de 2021

**DR. JUAN GABRIEL GONZÁLEZ SERNA**  
 Jefe del Departamento de ciencias computacionales  
 Presente

Nos es grato comunicarle que después de haber sometido a revisión académica la propuesta de tesis titulada Detección de obstáculos y planos durante el desplazamiento vehicular, presentada por el estudiante ANDREA PALACIOS CERVANTES, bajo la dirección del C. DR. DANTE MÚJICA VARGAS y habiendo atendido todas las observaciones que se le indicaron, la propuesta ha sido ACEPTADA.

De acuerdo al reglamento en vigor, a partir de esta fecha y hasta la conclusión de su trabajo de tesis, el estudiante deberá reportar 2 veces al semestre, o cuando así lo determine su Comité Tutorial, el avance en el desarrollo de su proyecto de su tesis.

Atentamente



  
 DR. DANTE MÚJICA VARGAS  
 Director de tesis

\_\_\_\_\_  
 Codirector de tesis

  
 DR. NOE ALEJANDRO CASTRO SANCHEZ  
 Revisor 1

  
 DR. NIMROD GONZÁLEZ FRANCO  
 Revisor 2

\_\_\_\_\_  
 Revisor 3

C.c.p. Dpto. servicios Escolares  
 Estudiante/ Expediente

CENIDET-AC-004-M9

Rev. 1