Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

Algoritmo de detección y estimación de trayectorias de obstáculos en desplazamientos vehiculares basado en visión computacional

Tesis presentada para obtener el grado de:
Doctor en Ingeniería del Lenguaje y del Conocimiento

Presenta: Lauro Reyes Cocoletzi
Director de Tesis: Dr. Ivan Olmos Pineda

26 de Abril de 2022

Agradecimientos

Mi agradecimiento al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el soporte proporcionado a través de la beca no. 700546 durante el transcurso del periodo de investigación doctoral. De igual forma agradezco a la Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla por darme la oportunidad de realizar este trabajo de investigación así como a los docentes que proporcionaron su apoyo y conocimiento para culminar de forma satisfactoria este trabajo de investigación.

Mi más sincero agradecimiento al Doctor Ivan Olmos Pineda y al Doctor Arturo Olvera Lopéz por todo el apoyo, soporte, seguimiento, asesoría, experiencia, recomendaciones, profesionalismo y consejos sin los cuales no habría sido posible terminar este trabajo doctoral, siempre les estaré agradecido.

Mi respeto y reconocimiento al comité revisor integrado por: Dr. E. Salvador Ayala Raggi, Dr. Manuel Martín Ortiz y al Dr. José Martinez Carranza por las recomendaciones, observaciones, sugerencias y el tiempo dedicado a revisar el trabajo doctoral para ampliar y mejorar dicha investigación.

A mi familia, es especial a mi mamá Carmen Cocoletzi Mendoza por todo el apoyo recibido no sólo en esta etapa sino en todo el transcurso de mi vida.

Resumen

La navegación autónoma en vehículos terrestres se encuentra en desarrollo por distintas empresas privadas (automotriz, paquetería, agricultura, etc.) así como centros de investigación y desarrollo tecnológico, los cuales están trabajando en distintas técnicas y herramientas para alcanzar el desplazamiento autónomo. La evasión de obstáculos en ambientes reales es una problemática a resolver tomando en consideración que la detección y posterior estimación del recorrido de los objetos evita daños al vehículo además de daños a terceros. La tecnología utilizada actualmente (sensor láser de mapeo 3D) para la detección de obstáculos tiene el inconveniente de elevar los costos de los vehículos y las técnicas implementadas hasta el momento carecen del rendimiento requerido. Este trabajo de investigación propone un algoritmo basado en visión computacional para la estimación de trayectorias de objetos en la ruta de desplazamiento de un vehículo en ambientes de tránsito vehicular para prevenir colisiones.

Abstract

Autonomous navigation in ground vehicles is under development by different private companies (automotive, parcel company, farm company, etc.) also as research and technological development centers, which are working on different techniques and tools to achieve autonomous displacement. The evasion of mobile obstacles in real environments is a problem to be solved taking into consideration that the detection and subsequent estimation of the obstacle route prevents damage to the vehicle besides damages to third parties. The technology currently used (laser 3D mapping sensor) for the detection of obstacles has the disadvantage of raising the costs of vehicles and the techniques implemented until now lack the required performance. This research work proposes an algorithm based on computer vision for the estimation of object trajectories in the travel path of a vehicle in vehicular traffic environments to prevent collisions.

Índice general

1. Introducción **10**
1.1. Antecedentes . . . 10
1.2. Justificación . . . 13
1.3. Problema de investigación . . . 13
1.3.1. Objetivo general . . . 13
1.3.2. Objetivos específicos . . . 14
1.3.3. Preguntas de investigación . . . 14
1.3.4. Hipótesis . . . 14
1.3.5. Alcances y limitaciones . . . 15

2. Marco teórico **16**
2.1. Percepción del ambiente . . . 16
2.1.1. LIDAR . . . 17
2.1.2. Cámaras de video . . . 18
2.1.3. Radar ultrasónico . . . 20
2.2. Visión estereoscópica . . . 21
2.2.1. Modelo de cámara en perspectiva . . . 22
2.2.2. Geometría epipolar . . . 22
2.2.3. Rectificación estéreo . . . 23
2.2.4. Modelo básico del mapeo de disparidad . . . 24
2.3. Detección de objetos . . . 25
2.3.1. Predicción de clase . . . 26
2.3.2. Aprendizaje profundo YOLO . . . 26
2.4. Seguimiento de objetos . . . 29
2.5. Formulación del seguimiento multiobjeto . . . 31
2.6. Marco metodológico: Redes dinámicas bayesianas . . . 32
2.6.1. Inferencia en RDB . . . 34

2.6.1.1. Inferencia por eliminación de variable . . . 35
2.6.1.2. Inferencia por condicionamiento . . . 36
2.6.1.3. Inferencia por árbol de unión . . . 37
2.7. Arquitecturas RBD . . . 37
2.7.1. Red dinámica bayesiana basada en un modelo Markoviano . . . 38
2.7.2. Red bayesiana dinámica híbrida multiagente . . . 39
2.7.3. Redes bayesianas dinámicas multifase . . . 41
2.7.4. Estructura de los RBDM . . . 41
2.7.5. Discretización de las variables de interés . . . 43
2.7.6. Métodos de discretización . . . 43
2.7.7. Complejidad computacional . . . 45
2.8. Parámetros de evaluación . . . 45

3. Análisis del estado del arte **48**
3.1. Trabajos relacionados . . . 49
3.2. Análisis de trabajos relacionados . . . 51
3.3. Métodos basados en la apariencia y movimiento . . . 52
3.4. Métodos basados en aprendizaje profundo . . . 55
3.4.1. Redes neuronales convolucionales . . . 55
3.4.2. Redes neuronales recurrentes . . . 56
3.4.3. Aprendizaje profundo por refuerzo . . . 57
3.5. Comparativa de métodos . . . 59
3.6. Pertinencia de los datos . . . 59
3.7. Discusión . . . 61
3.8. Conclusiones del análisis de la literatura . . . 64

4. Método Propuesto **66**
4.1. Detección ROI . . . 67
4.1.1. Aproximación de la distancia y velocidad . . . 69
4.1.2. Seguimiento de las ROI . . . 73
4.1.3. Topología RDB propuesta . . . 75
4.1.4. Inferencia de parámetros . . . 77
4.1.5. Modelo de interacción . . . 78

5. Experimentos y resultados **80**
5.1. Aproximación de la posición espacial . . . 81

5.2. Resultados experimentales 82
5.3. Estimación de probabilidad de desplazamiento 84
5.3.1. Evaluación del seguimiento de obstáculos 90
5.3.2. Evaluación de la posición estimada 93
5.3.3. Evaluación de estimación de dirección 97
5.4. Características de procesamiento 100
5.5. Discusión 101

6. Conclusiones **103**
6.1. Trabajo a futuro 104
6.2. Publicaciones 105

Bibliografía **106**

Índice de figuras

2.1. Sistemas de escaneo de la información en un sistema lidar, repoducido de (Warren,219). 17
2.2. Sensor lidar para la percepción del ambiente fabricado por la empresa Velodyne tomado de (*https://velodynelidar.com/*). 18
2.3. Reconstrucción del ambiente mediante nube de puntos del escaneo de un sensor lidar, tomado de Waymo, 2021 (*https://waymo.com/intl/es/waymo-driver/)*. 18
2.4. Diagrama del flujo de datos a través de las fases del algoritmo basado en voxels, reproducido de (Sajadi y Ribnick, 2010). 19
2.5. Cámara estereoscópica para la percepción del ambiente tomado de StereoLabs, 2021 (*http:///www.stereolabs.com/zed/*). 20
2.6. Frame original y reconstrucción de la escena en un mapa de disparidades, tomado de StereoLabs, 2021 *(https://www.stereolabs.com/developers/)*. 20
2.7. Parámetros de referencia y puntos de interés eje óptico de acuerdo a la relación geométrica con respecto a los puntos de proyección, reproducido de (Fan y Dahnoum, 2017). 22
2.8. Representación de los planos epipolares izquierdo, derecho y la proyección de los puntos de interés de acuerdo a la referencia de la distancia focal para determinar las coordenadas tridimensionales, reproducido de (Fan, 2018). 23
2.9. Representación de la rectificación estéreo, se observa los planos de la imagen original (recuadro negro) y los planos de la imagen rectificada (recuadro rojo) de acuerdo a la referencia de las lineas epipolares, reproducido de (Fan, Ai y Dahnoun, 2018). 24
2.10. Representación del modelo de visión estéreo básico, se muestran los planos de referencia izquierdo y derecho para la representación de puntos arbitrarios en el sistema de coordenadas 3-D determinados por los parámetros intrinsecos y extrínsecos para la proyección de subplanos en el mapa de disparidad, tomado de (Scharstein y Szeliski, 2002) 24

2.11. Mapeo de las coordenadas para determinar la predicción del posible objeto a detectar en el frame de acuerdo a la referencia superior izquierda, reproducido de (Redmon et al., 2016). 27
2.12. Bloques implementados en la metodología de detección y seguimiento de múltiples objetos a través de una escena, reproducido de (Meng et al., 2020). 30
2.13. Representación de una RDB con 3 variables y 4 cortes, en este caso la estructura base se repite cuatro veces, reproducida de (Song et al., 2009). 33
2.14. Estructura topológica del Modelo Markoviano RDB con enfoque de multiagentes (lado izquierdo) y modelo neuronal convolucional (lado derecho), tomado de (Schulz et al., 2019). 39
2.15. La RBDHM se despliega en dos lapsos de tiempo (izquierda) y se descompone en la dependencia condicional de estado latente (derecha), tomado de (L. Sun et al., 2019). . 40
2.16. Diagrama esquemático de RBDM para la determinación de SIL en dos intervalos consecutivos de interés, reproducido de (Cai et al., 2020). 42

3.1. Bloques generales para la implementación de la detección y estimación de trayectorias de objetos, tomando en consideración los enfoques encontrados en la literatura. . . . 49
3.2. Porcentaje de etapas del problema abordadas por diferentes autores. 63
3.3. Obstáculos presentes en carreteras, a) señal de tráfico en alcantarilla, b) poste de luz en medio de la calle, c) árbol caído que bloquea la calle, d) neumáticos en alcantarilla. 64

4.1. Etapas de procesamiento del algoritmo propuesto para la estimación de trayectorias. . 67
4.2. Configuración estereoscópica de dos cámaras con distancia focal y centros alineados para el cálculo del mapa de disparidad, basado en (Xu et al., 2019). 68
4.3. Análisis del vecindario del centroide con respecto al objeto detectado, la salida corresponde a los vectores resultantes. 70
4.4. Muestras para realizar las mediciones de altura (pixeles) de peatones y vehículos vs la distancia real a la que se encuentran, tanto en ambientes reales como del simulador *city car driving*. 72
4.5. Estimación de distancia de separación del objeto detectado con respecto a la relación de la altura medida por medio del polinomio aproximador calculado. 73
4.6. Gráfica con la información espacial y sentido del movimiento del objeto detectado con respecto a la referencia global correspondiente a los frames de interés. 74

4.7. El modelo de red bayesiana se desarrolla en dos segmentos de tiempo, donde se observa la composición de las dependencias condicionales de estado latente. Las líneas continuas y discontinuas son las dependencias causales y temporales observables, respectivamente. 76

5.1. Ejemplos resultantes del procesamiento para obtener mapas de disparidad. 81
5.2. Conjunto de datos procesados (mapas de disparidad) para obtener los valores aproximados de las distancias a la que se encuentra un obstáculo en el transcurso de los frames. 81
5.3. Representación de la información sobre el desplazamiento de un obstáculo detectado en el transcurso de los frames, ángulo de giro y vectores de desplazamiento. 83
5.4. Secuencia de frames para la inferencia de cambio de dirección de un obstáculo (automóvil) detectado. 84
5.5. Secuencia de frames para la estimación cambio de dirección de un obstáculo (automóvil) detectado. 85
5.6. Secuencia de frames para la estimación de cambio de dirección de un obstáculo (automóvil) detectado en un ambiente virtual. 86
5.7. Recopilación de los valores de la estimación de probabilidad del cambio de dirección para los tres posibles vectores de dirección y su representación gráfica. 86
5.8. Gráfica con los valores de la estimación de probabilidad del cambio de dirección . . . 86
5.9. Probabilidad de cambios de dirección frente a GT (derecha) 87
5.10. Estimación de la probabilidad de cambios de dirección frente al GT (frente). 87
5.11. Probabilidad de cambios de dirección frente a GT (izquierda). 88
5.12. Representación vectores de dirección de múltiples objetos detectados. 88
5.13. Múltiples obstáculos detectados en escena en el entorno de simulación y la representación gráfica de sus parámetros de probabilidad en relación a los vectores de dirección. 89
5.14. Resultados obtenidos para los parámetros MOTA y MOTP para las secuencias de video de kittiVision. 90
5.15. Parámetros obtenidos para las métricas MOTA y MOTP para las secuencias de video capturadas propias. 91
5.16. Resultados de los parámetros MOTA y MOTP al procesar las secuencias de video obtenidas del simulador city car driving. 91
5.17. Probabilidad de movimiento obtenida frente a GT (posición). 93
5.18. Probabilidad de movimiento obtenida frente a GT (ángulo). 94

5.19. Error obtenido de la ubicación espacial de objetos detectados en secuencias de video base de datos KittiVision, propuesta vs trabajos relacionados. 96
5.20. Error obtenido de la ubicación espacial de objetos detectados en secuencias de video base de datos propia, propuesta vs trabajos relacionados. 96
5.21. Error obtenido de la ubicación espacial de objetos detectados en secuencias de video base de datos simulador, propuesta vs trabajos relacionados. 96
5.22. Comparación de la propuesta vs trabajos relacionados. 98
5.23. Comparación de la propuesta vs resultados en trabajos relacionados. 99
5.24. Comparativa resultados de la propuesta vs trabajo relacionado. 99
5.25. Comparativa experimental tiempo de ejecución por módulo. 100

Índice de tablas

2.1. Arquitectura básica de la RNC - YOLO, composición de las capas convolucionales, dimensiones de los filtros utilizados y escalas uniformes de los clusters de salida, reproducida de (Redmon et al., 2016). 27
2.2. Resumen de los distintos métodos de discretización y su complejidad computacional, tomado de (Yang y Webb, 2002). 45

3.1. Comparativa de adquisición de información para la percepción del ambiente. 51
3.2. Comparativa de las técnicas basadas en el aprendizaje y en la correlación. 54
3.3. Descripción comparativa de distintos modelos de aprendizaje. 58
3.4. Comparativa de resultados cuantitativos así como herramientas utilizadas en algunos trabajos relacionados con la detección y seguimiento de objetos. 60
3.5. Etapas del problema abordadas por diversos autores. 63

5.1. Resultados del experimento dada la inferencia de colisión para cada dirección discretizada de trayectoria. 83
5.2. Resultados del experimento dada la inferencia de colisión para cada posible dirección a tomar. 84
5.3. Resultados del experimento dada la inferencia de colisión para cada posible dirección a tomar. 85
5.4. Resultados del experimento dada la inferencia de colisión para cada posible dirección a tomar. 85
5.5. Resultados obtenidos al evaluar el seguimiento de objetos en escena. 90
5.6. Resumen de los resultados obtenidos para las bases de datos con respecto a las métricas de seguimiento. 92
5.7. Resumen comparativo entre trabajos relacionados con las métricas de seguimiento. . 93
5.8. Resultados experimentales (RMSE) entre el GT y la posición espacial y el ángulo de dirección de los objetos en 10 escenas vehiculares. 95
5.9. Resultados obtenidos al evaluar el seguimiento de objetos en escena. 97

5.10. Resultados obtenidos al evaluar el seguimiento de objetos en escena. 99
5.11. Valor promedio de tiempo de ejecución de los bloques correspondientes al proceso de estimación de trayectoria. 101

Capítulo 1

Introducción

En el ámbito industrial existen diversos algoritmos y técnicas desarrolladas para la prevención de colisiones mediante detección de obstáculos, sin embargo debido a los derechos exclusivos y patentes para explotar comercialmente esta tecnología no es de acceso público y la información al respecto es nula o limitada. Adicionalmente a la poca información proporcionada por las armadoras automotrices aún estas no han logrado alcanzar la prevención requerida de colisiones en desplazamientos del vehículo autónomo en ambientes reales. La problemática pendiente por resolver es la implementación de un algoritmo para la detección y estimación de trayectorias de obstáculos con una tasa de detección equiparable contra las capacidades que logra realizar un conductor humano (Leon y Gavrilescu, 2021).

1.1. Antecedentes

La tecnología existente para el desplazamiento de vehículos terrestres autónomos se encuentra en desarrollo y actualmente tiene deficiencias por lo que no puede ser utilizada en vehículos de uso cotidiano, sin embargo la industria automotriz prevé alcanzar el nivel cinco de autonomía a mediano plazo (Fagnant y Kockelman, 2015). La Administración Nacional de Seguridad del Tráfico en Carreteras (NHTSA) define cinco niveles de autonomía para los automóviles, estos niveles varían en función del sistema utilizado así como del control que tienen sobre el vehículo, los sistemas de control se conocen como sistemas avanzados de asistencia al conductor (ADAS). A continuación se mencionan brevemente los cinco niveles de autonomía.

- Nivel 1. El conductor se encuentra en todo momento bajo el control del vehículo (aceleración, frenado, sistema de dirección, etc.).

- Nivel 2. Asistencia al conductor en la automatización específica de funciones, por ejemplo, el sistema ABS para recuperar de forma más rápida el control del vehículo cuando se frena repentinamente.

- Nivel 3. Automatización de al menos dos funciones primarias para relevar al conductor, por ejemplo, el control de velocidad de crucero combinado con el control para mantener la línea del carril.

- Nivel 4. Los vehículos en este nivel de automatización permiten al conductor ceder el control total de todas las funciones críticas únicamente bajo ciertas condiciones de tráfico o ambientales. Se espera que el conductor esté disponible para control del vehículo ocasionalmente, con suficiente tiempo para hacer el relevo del control.

- Nivel 5. En este nivel los vehículos están diseñados para realizar todas las funciones de conducción críticas y monitorear las condiciones de la carretera durante todo un viaje. Dicho diseño anticipa que el conductor proporcionará el destino o la entrada de navegación, pero no se espera que esté disponible para el control en ningún momento durante el viaje. Los vehículos circulan de forma autónoma estén ocupados o desocupados (Rödel, Stadler, Meschtscherjakov, y Tscheligi, 2014).

Uno de los precursores de vehículos eléctricos *TESLA-Motors* fabrica los autos con mayores avances en desplazamiento autónomo disponibles para el público (autonomía nivel 4), su tecnología de manejo *autopilot* (Tesla, 2018), la cual se basa en una serie de sensores acoplados (radares ultrasónicos y cámaras de video) aún requiere mejoras en el desempeño ya que ha tenido percances y dos colisiones fatales hasta el momento.

Por otro lado, la compañía *Daimler-Crysler* en asociación con *WYMO* (sub-empresa de *google research*) prevé vehículos disponibles de competir en esta área. Actualmente cuenta con prototipos de autonomía nivel 4 que no están disponibles al público y tienen la desventaja de utilizar un radar láser que elevan sustancialmente los costos del vehículo (Waymo, 2021).

Mercedes Benz se encuentra en el nivel 3 ya que sus vehículos son capaces sólo de mantener el carril en autopista (Mercedes, 2018), *Audi* y su asistente de frenado de emergencia pertenecen al nivel 2 (Audi, 2018).

En los centros de investigación y desarrollo tecnológico se han realizado diversos trabajos concernientes a desarrollar sistemas para lograr que los vehículos cuenten con autonomía, un ejemplo relevante, es el proyecto *madeingermany* de la Universidad libre de Berlín (Göhring, 2012), el cual propone una arquitectura de control con múltiples sensores (cámaras de video y sensores láser) montados en un vehículo convencional (*vw passat*) para la percepción

del ambiente (obstáculos en especial) en tiempo real. Este trabajo aún no soluciona los problemas de autonomía en tráfico denso.

Existen trabajos de investigación que han propuesto de manera específica el uso de visión computacional para obtener información del ambiente (Fagnant y Kockelman, 2015).

Los resultados que estos trabajos han logrado obtener tienen un alcance limitado como velocidad de desplazamiento, condiciones controladas de iluminación en el ambiente de desempeño, tránsito vehicular y peatonal limitado por lo que existen áreas de mejora. Sin embargo, los algoritmos de visión computacional tiene mayores perspectivas de desarrollo pues los sensores (cámaras estereoscópicas o monoculares) no emiten energía al ambiente por lo que no son invasivas, además de su bajo costo en comparación con los sensores láser (Leon y Gavrilescu, 2021; Velodyne, 2020).

La detección del desplazamiento de los objetos en entornos urbanos es uno de los principales retos pendientes en el desarrollo de vehículos autónomos. Aún esta pendiente el desarrollo de modelos que puedan realizar la estimación de cambios de movimiento de los participantes del tránsito vehicular, es decir, predecir si un vehículo girará a la izquierda, a la derecha o seguirá de frente durante el recorrido.

De forma específica los aspectos fundamentales que tiene áreas de oportunidad pendientes por resolver con respecto a la problemática de estimación de movimiento son el seguimiento de objetos y la predicción de desplazamiento.

El seguimiento implica la identificación individual de los objetos de interés dentro de la escena vehicular a partir de la correlación de secuencias de imágenes y datos de sensores adicionales u observaciones en un momento inicial y hasta que desaparezca del rango de detección. Dicha rastreo debe llevarse a cabo durante el intervalo de tiempo en el cual el objeto se encuentre en escena a una tasa de rendimiento que permita evitar falsas detecciones y detecciones nulas, es decir, perder de vista dentro de la escena los objetos de interés.

La predicción de movimiento es especialmente difícil de procesar porque suele haber múltiples objetos que interactúan en una escena, predecir el desplazamiento también implica evaluar el movimiento futuro de los objetos de interés circundantes para navegar por diversos escenarios de tráfico. Además de la predicción del comportamiento físico de los objetos basada en un conjunto de observaciones pasadas, también es de suma importancia el tener en consideración sus posibles interacciones y dadas las condiciones observables delimitar las más probables a presentarse.

1.2. Justificación

Uno de los problemas por resolver en el desplazamiento autónomo libre de riesgo en ambientes de tráfico reales es la detección y estimación de movimiento de los obstáculos.

La detección y la estimación de las trayectorias de obstáculos en ambientes de tránsito es de mayor importancia para garantizar que un vehículo autónomo se desplace sin colisiones con el propósito de resguardar la integridad del vehículo y de sus ocupantes así como evitar daños a terceros.

El desarrollo de un algoritmo capaz de detectar obstáculos y estimar trayectos de estos obstáculos favorece el alcance del nivel 5 de autonomía, adicionalmente se puede extrapolar el algoritmo con los cambios pertinentes para cualquier vehículo móvil (aéreo, bípedo, acuático).

El uso de visión computacional tiene ventajas con respecto a las técnicas de análisis tridimensional de nubes de puntos (basados en sensores láser) ya que pueden procesar la información de los obstáculos con base en el movimiento o la apariencia, lo cual implica que se pueden utilizar distintos algoritmos e incluso combinaciones de estos para lograr la detección de obstáculos fijos y móviles.

1.3. Problema de investigación

En el ámbito industrial existen diversos algoritmos y técnicas desarrolladas para la prevención de colisiones mediante detección de obstáculos, sin embargo debido a los derechos exclusivos y patentes para explotar comercialmente estas tecnologías no son de acceso público y la información al respecto es nula o limitada (Schwarting, Alonso-Mora, y Rus, 2018).

Adicionalmente a la poca información proporcionada por las armadoras automotrices aún estas no han logrado alcanzar la prevención de colisiones requerida en desplazamientos de vehículos autónomos en ambientes reales.

La problemática pendiente por resolver es la implementación de un algoritmo para la detección y estimación de trayectorias de obstáculos a una tasa de detección equiparable contra las capacidades que logra realizar un conductor humano (Leon y Gavrilescu, 2021).

1.3.1. Objetivo general

El objetivo general en esta propuesta de investigación es realizar un algoritmo para la detección y estimación de trayectorias de obstáculos mediante visión computacional en

ambientes de tránsito vehicular. La base fundamental de la propuesta es el uso de visión computacional debido a que es la forma que se asemeja en mayor medida a como un conductor humano percibe el ambiente que lo rodea y logra manejar un automóvil sin colisionar con obstáculos que se presenten en el recorrido.

1.3.2. Objetivos específicos

Los siguientes objetivos específicos son necesarios para cumplir con el objetivo general.

- Identificar obstáculos (presentes en la escena de video) de acuerdo a la proximidad con el vehículo y su riesgo de colisión.
- Estimar la posición espacial de los obstáculos en la escena de video para determinar el grado de riesgo que representa el ambiente por el cual se transita.
- Estimar la trayectoria de los obstáculos presentes en video para tomar medidas preventivas y evitar colisiones.
- Estimar la velocidad de desplazamiento de los obstáculos para aproximar una predicción en la trayectoria de estos con respecto al vehículo que adquiere la información del ambiente.

1.3.3. Preguntas de investigación

Las preguntas que son necesarias de responder se presentan a continuación:

- ¿Cómo detectar obstáculos en una escena vehicular capturada en video?
- ¿Cuántos obstáculos detectables y con qué riesgo de colisión están en la escena en video?
- ¿Cómo estimar los trayectos de los distintos objetos en la escena en video?
- ¿Cuál es la velocidad de desplazamiento de los obstáculos en la escena capturada en video?

1.3.4. Hipótesis

Aplicar técnicas de visión computacional a la información del ambiente (tránsito vehicular) obtenida a través de cámaras de video permite detectar los obstáculos en escena y

la estimación de las trayectorias de los objetos se puede resolver mediante un modelo de probabilidad.

1.3.5. Alcances y limitaciones

Los alcances de este trabajo de investigación radican en:

1. El algoritmo propuesto procesa video con escenas de ambientes reales del desplazamiento de un vehículo en diversas rutas de tráfico con obstáculos distintos como peatones, autos, animales, arboles, motocicletas, bicicletas, entre otros.

2. El algoritmo detecta los obstáculos presentes en la escena, si aparecen nuevos obstáculos en la escena también se detectan.

3. Se obtienen y presentan los resultados en video, las trayectorias estimadas se visualizan en forma vectorial sobre cada obstáculo, se descartan aquellos obstáculos que no estén en ruta de colisión, con base en este hecho se denotan a los obstáculos en la escena con riesgo de colisión.

4. Se realizan los cálculos (de velocidad, posición y posible trayectoria) para los obstáculos con riesgo de colisión, para los obstáculos sin riesgo de colisión no es necesario realizar estos cálculos.

Las limitaciones de este trabajo de investigación son:

1. La propuesta no considera condiciones limitadas de iluminación, lluvia o neblina en los ambientes vehiculares a procesar.

2. En este trabajo no se contempla el diseño del control mecánico para la modificación de las condiciones dinámicas del ego-vehículo.

3. La propuesta tiene acotada la velocidad de desplazamiento de los obstáculos, el rango viable de procesar corresponde a velocidades ≤ 60 km/hr.

Capítulo 2

Marco teórico

En este capítulo se presentan los conceptos necesarios para el planteamiento y desarrollo de las distintas etapas de la metodología propuesta (algoritmo desarrollado). Por ejemplo, se realiza una comparativa de distintos dispositivos para la percepción del ambiente y con base en esta comparativa se fundamenta el motivo por el cuál se decide utilizar las cámaras digitales de video. También se mencionan las técnicas de visión computacional enfocadas en estereopsis así como los métodos de detección de objetos en escena. Finalmente se presentan los antecedentes matemáticos de las relaciones de probabilidad utilizadas en el enfoque de las Redes Dinámicas Bayesianas (RDB) para la construcción de un modelo de probabilidad para la estimación de los cambios de posición espacial de los objetos de interés en escena y determinar los posibles cambios de dirección con respecto al riesgo de colisión en función del desplazamiento del vehículo con las cámaras a bordo.

2.1. Percepción del ambiente

La interpretación y percepción del ambiente se realiza en niveles, en el primer nivel se tienen a las características como la apariencia, tamaño y disparidad de los objetos. En el siguiente nivel está el seguimiento lo cual corresponde a la asociación de datos, coherencia temporal así como filtrado para rastrear, identificar y medir los parámetros dinámicos para estimar las posiciones de los objetos de interés (Mukhtar, Xia, y Tang, 2015).

En el nivel superior se utilizan un conjunto de características espacio-temporales para aprender, modelar, clasificar y predecir el comportamiento de los obstáculos en la carretera.

Las técnicas que abordan el problema de la detección de obstáculos de entornos en carretera son diferentes entre sí. Existen muchas técnicas de escaneo, tales como algunas relacionadas con el comportamiento mecánico hasta el escaneo de estado sólido, pasan-

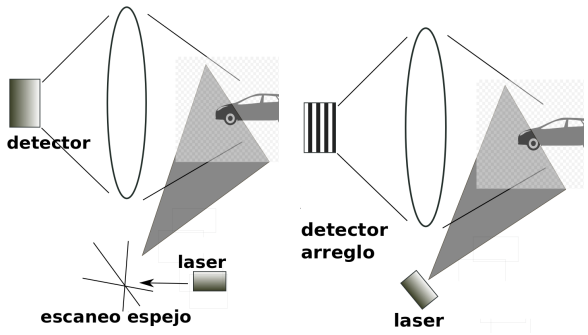
do por los sistemas ópticos por etapas, todas estas técnicas con un costo y complejidad computacional a tomar en consideración (Warren, 2019).

2.1.1. LIDAR

La tecnología lidar (laser imaging detection and ranging) se basa en la construcción 3D espacial de una nube de puntos dentro del campo de visión (field of vision, FOV) mediante la emisión de energía láser.

Existen dos formas principales de proporcionar la información espacial, el lidar de barrido (figura 2.1a) escanea un punto o línea de pulsos de luz a través del FOV. Este tipo de lidar emite un pulso de frecuencia alta durante el escaneo, así la amplitud angular del láser y la velocidad de escaneo determinan la resolución angular del sistema. La luz de retorno se adquiere por un sistema basado en óptica de imagen que cuenta con un detector de puntos para una exploración 2D además de la generación de una matriz lineal de detectores para una exploración de línea.

La segunda categoría son los sistemas flash (figura 2.1b) en los que el láser ilumina todo el FOV de un conjunto de detectores 2D. En este caso, el conjunto de detectores determina completamente la resolución angular del sistema. Este diseño tiene una simplicidad mecánica y óptica, el conjunto de detectores tiene un costo elevado. Una desventaja de los conjuntos de detectores de alta resolución es que la reconstrucción de la información se realiza en dimensiones pequeñas (los sensores recogen menos luz).



(a) lidar de barrido (b) lidar sistema flash

Figura 2.1. Sistemas de escaneo de la información en un sistema lidar, repoducido de (Warren,219).

También existen enfoques híbridos en los que el FOV se divide en zonas iluminadas por separado, combinando algunas características de los sistemas de barrido y de flash.

En la figura 2.2 se observa el sensor lidar HDL-64E de la compañía Velodyne cuyo costo aproximado es de $ 8,000.00 dólares, realiza un mapeo 3D a través de una nube de puntos

para reconstruir el ambiente real en un rango de distancia de hasta 120 mts (Velodyne, 2020).

Figura 2.2. Sensor lidar para la percepción del ambiente fabricado por la empresa Velodyne tomado de (*https://velodynelidar.com/*).

En la figura 2.3 se observa un bloque de datos reconstruidos con un radar lidar con respecto a una escena vehicular, el sensor esta montado en el vehículo de la compañía waymo.

Figura 2.3. Reconstrucción del ambiente mediante nube de puntos del escaneo de un sensor lidar, tomado de Waymo, 2021 (*https://waymo.com/intl/es/waymo-driver/)*.

2.1.2. Cámaras de video

El enfoque de la reconstrucción de la escena mediante el uso de cámaras de video se basa en utilizar múltiples vistas de cámara (point of view, POV) de una escena para detectar y reconstruir las superficies de los objetos en tres dimensiones. La idea fundamental de este planteamiento es que cada POV contiene información única sobre el contenido de la escena.

La problemática de la reconstrucción consiste en combinar la información proyectada de forma robusta para inferir con precisión la estructura de la escena. Las imágenes de entrada pueden obtenerse de un conjunto de cámaras o de una sola cámara en movimiento (Madrid Sánchez, 2018).

En general para la reconstrucción de la información del ambiente se requiere que las cámaras estén calibradas además de conocer sus posiciones y orientaciones relativas. Algunas técnicas utilizan el enfoque multicámaras (reconstrucción mediante voxels) y otras plantean el mapeo de disparidades mediante visión estereoscópica (2 cámaras).

El algoritmo de voxels consta de dos fases principales, en la primera fase, el volumen de reconstrucción se divide en bloques discretos, a los que nos referimos como vóxeles. Se

calcula el haz de luz de incidencia (rayo) desde el centro de cada cámara a cada vóxel del volumen. Para cualquier cámara-vóxel, el valor del píxel en el lugar donde este rayo se cruza con el plano de la imagen de la cámara se trata como puntuación (voto) de la cámara con respecto al contenido del vóxel (Broggi, Cattani, Patander, y Sabbatelli, 2013).

La idea subyacente es que, aunque algunos de los votos en un vóxel determinado contengan información sobre objetos que no existen realmente en ese lugar, muchos de los votos coincidirán en el contenido del vóxel cuando un objeto exista realmente allí. La segunda fase del algoritmo toma estos conjuntos de votos en cada vóxel como entrada y procede a la detección y reconstrucción de objetos en 3D. Dado que se tienen votos de varias cámaras en cada vóxel del espacio, es necesario decidir en qué lugares de este espacio existe realmente una superficie de objeto.

El costo puede interpretarse como una discrepancia, de modo que se intenta minimizar la desigualdad entre los votos en un vóxel determinado. Una vez que este costo se minimiza localmente, se realiza un post-procesamiento para decidir dónde residen las superficies de los objetos. Por último, se puede actualizar el costo y repetir el proceso para reconstruir los objetos ocluidos con mayor precisión (Sadjadi y Ribnick, 2010). La figura 2.4 muestra un diagrama de flujo del algoritmo basado en voxels.

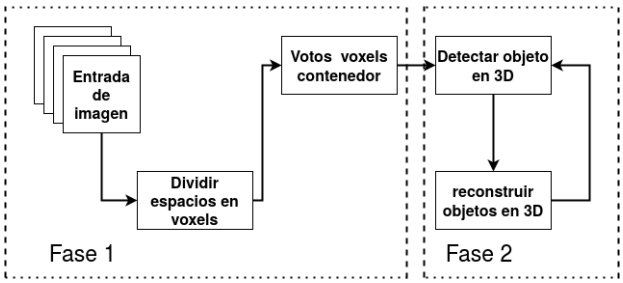


Figura 2.4. Diagrama del flujo de datos a través de las fases del algoritmo basado en voxels, reproducido de (Sajadi y Ribnick, 2010).

Los algoritmos de estimación de la disparidad pueden clasificarse en locales, globales y semiglobales. Los algoritmos locales se limitan a empatar una serie de bloques y seleccionar la correspondencia con el menor costo o la mayor correlación. A diferencia de los algoritmos locales, los algoritmos globales procesan la correspondencia estereoscópica utilizando algunas técnicas de optimización más sofisticadas, por ejemplo la propagación de la creencia (Belief Propagation BP) (Felzenszwalb y Huttenlocher, 2006). Estos algoritmos suelen desarrollarse sobre la base de los campos aleatorios de Markov (Markov Random fields, MRF)(Shabanian y Balasubramanian, 2021), donde la búsqueda de las mejores disparidades se formula como un problema de maximización de la probabilidad. Esto se aborda posteriormente mediante

enfoques de minimización de la energía.

La correspondencia semiglobal (SGM) (Z. Lu, Wang, Li, Chen, y Wu, 2021) se aproxima a la inferencia de los MRF realizando una agregación de costos en todas las direcciones de la imagen, lo que mejora en gran medida la precisión y la eficacia de la correspondencia estereoscópica. Más adelante se hará énfasis en SGM ya que se utiliza como una de las etapas del algoritmo de estimación de trayectorias en este trabajo de investigación. En la figura 2.5 se muestra la cámara estereoscópica ZED de la compañía (StereoLabs, 2021) que permite grabar video a una resolución de 1080 píxeles HD a 30 frames por segundo (fps), mapeo espacial con un rango de profundidad de hasta 20 metros (m) y un costo aproximado de $ 449.00 dólares.

Figura 2.5. Cámara estereoscópica para la percepción del ambiente tomado de StereoLabs, 2021 (*http:///www.stereolabs.com/zed/*).

En la figura 2.6 se muestran los frames izquierdo y derecho, que captura un sistema estereoscopico ZED cámara, así como la reconstrucción en escala de grises del mapa de disparidad procesado por el mencionado sistema.

Figura 2.6. Frame original y reconstrucción de la escena en un mapa de disparidades, tomado de StereoLabs, 2021 *(https://www.stereolabs.com/developers/)*.

2.1.3. Radar ultrasónico

El radar emite una señal de radio que se dispersa en todas las direcciones, el tiempo de reflexión de la señal al radar determina el cálculo de la distancia de los objetos sobre los cuales reflectó la energía. Lo sensores ultrasónicos generalmente se utilizan en conjunto con otros tipos de sensores con el fin de mejorar el rendimiento del sistema. Los métodos donde tiene aplicación la integración de los datos proporcionados por el radar ultrasónico sumados a datos proporcionados por sensores con mayor aporte de información corresponden a la estimación de posición, detección, reconstrucción ambiental y cálculo de riesgos.

A diferencia del lidar o el radar ultrasónico, la detección visual no puede depender de una señal de referencia reflejada; si bien la detección de objetos con cámaras a menudo requiere técnicas de procesamiento más sofisticadas también presenta varias ventajas.

Las imágenes proporcionan una fuente de datos substancial, desde la cual se puede realizar conjeturaras de información y contexto adicionales. Las cámaras proporcionan un amplio campo de visión, lo que permite la detección y el seguimiento a través de múltiples carriles además el costo de equipos lidar es elevado en comparación con cámaras de video.

En este trabajo de investigación dadas las características antes mencionadas se opta por trabajar con captura y procesamiento de video, en la siguiente sección se profundiza con respecto a los conceptos utilizados referentes a visión computacional.

2.2. Visión estereoscópica

El ser humano vive en un mundo tridimensional (3D) sin embargo los ojos de los humanos sólo pueden percibir los objetos en dos dimensiones. La percepción de la profundidad se debe a la capacidad que tiene el cerebro para analizar la diferencia entre dos imágenes bidimensionales (2D) que se proyectan en las retinas de los ojos. En específico, cada par de puntos correspondientes de información en las retinas envía señales a las neuronas binoculares del córtex visual primario y este estima la diferencia posicional relativa entre cada par de puntos correspondientes, dicha diferencia se conoce como disparidad binocular (Capetillo Vázquez, 2021).

Las imágenes digitales capturadas por las cámaras de video son de naturaleza bidimensional, para extrapolar a información tridimensional de una escena determinada, se necesitan múltiples vistas de cámara de la misma escena.

La profundidad del escenario real puede estimarse comparando la diferencia entre las imágenes digitales izquierda y derecha captadas por cada cámara. Este proceso se conoce como estereopsis o visión estereoscópica (Poggio y Poggio, 1984) y es muy similar a la visión binocular humana.

En este trabajo de investigación las imágenes se capturan de forma sincronizada utilizando un sistema estereoscópico que consiste en un par de cámaras digitales de captura de video.

2.2.1. Modelo de cámara en perspectiva

El modelo de cámara en perspectiva (o estenopeica) es el modelo geométrico más utilizado para describir la relación entre un punto tridimensional en el sistema de coordenadas de la cámara y su proyección en el plano (Fan, 2018).

Un ejemplo de la perspectiva de la cámara se muestra en la figura 2.7 donde se presenta el foco de la cámara (0^l), la distancia focal (f), el punto P de la imagen expresado en función de la proyección en el plano de referencia, en el sistema de referencia $\{x, y, z\}$. El eje óptico es el rayo originado desde el foco de la cámara que atraviesa perpendicularmente el plano de la imagen. La relación entre P y su proyección (p')esta dado por la ecuación 2.1 (Fan y Dahnoun, 2017).

$$\hat{p} = \frac{f}{z}P \tag{2.1}$$

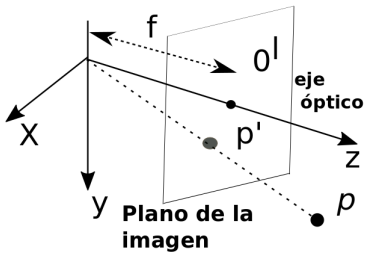


Figura 2.7. Parámetros de referencia y puntos de interés eje óptico de acuerdo a la relación geométrica con respecto a los puntos de proyección, reproducido de (Fan y Dahnoum, 2017).

Adicionalmente, con esta información es necesario obtener los parámetros intrínsecos, la corrección de deformaciones causada por la lente, la distorsión radial (barril o alfiletero) y la distorsión tangencial.

2.2.2. Geometría epipolar

La geometría epipolar es la relación geométrica entre los parámetros de un sistema de visión binocular, los parámetros de interés corresponden a los focos de las cámaras, izquierda y derecha respectivamente, en un sistema principal de coordenadas de referencia en tres dimensiones además de su representación en el sistema de coordenadas de la cámara izquierda (p_l) y derecha (p_r), es decir, $p_l = [X_l, Y_l, Z_l]^T$ y $p_r = [X_r, Y_r, Z_r]^T$ con sus respectivos planos π_l, π_r.

La proyección de un punto P'' sobre el plano lateral π_l, con la distancia focal f_l y el plano epipolar se define para la transformación de un punto en tres dimensiones en el sistema

de coordenadas establecido, los epipolos se denotan como e_r, epipolo derecho y e_l epipolo izquierdo (Fan, 2018). En la figura 2.8 se muestra la representación de la geometría epipolar.

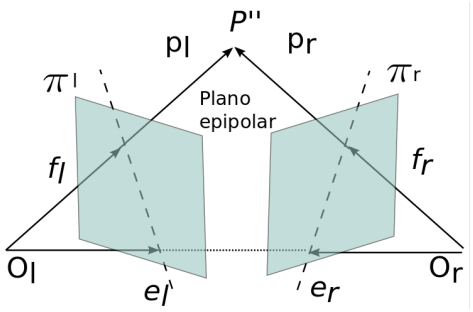


Figura 2.8. Representación de los planos epipolares izquierdo, derecho y la proyección de los puntos de interés de acuerdo a la referencia de la distancia focal para determinar las coordenadas tridimensionales, reproducido de (Fan, 2018).

Con respecto al plano de la cámara izquierda o derecha se puede normalizar para denotar los parámetros intrínsecos. La matriz fundamental ($essential\ matrix$) establece un vínculo entre las restricciones epipolares y los parámetros extrínsecos de un sistema estereoscópico.

En este sentido, la matriz fundamental establece una relación entre par de puntos 3D de correspondencia con los sistemas de coordenadas de referencia izquierdo y derecho, de tal forma que al conocer la matriz intrínseca de cada cámara se establecen los puntos de correspondencia bidimensional. Para un sistema estéreo bien calibrado la matriz homógrafa se utiliza generalmente para distinguir los objetos de una superficie plana.

2.2.3. Rectificación estéreo

Cuando se utiliza un par de cámaras digitales de video para adquirir imágenes de la escena de interés, la tarea principal de la reconstrucción 3D es determinar cada par de puntos correspondientes entre las imágenes izquierda y derecha (Lin, Li, Xu, y Cao, 2017).

Para un sistema de visión estereoscópica no calibrado, la búsqueda de los pares de correspondencia suele implicar una búsqueda bidimensional, que es una tarea computacionalmente intensiva. Por lo tanto, siempre se realiza un proceso de transformación de la imagen conocido como rectificación estereoscópica para reducir la dimensión de la búsqueda de correspondencias. Cada par de líneas epipolares conjugadas se vuelve colineal y paralela al eje horizontal de la imagen, como se muestra en la figura 2.9 (Fan y Dahnoun, 2017).

Donde π_r y π_l son los planos de la imagen original y π'_l y π'_r son los planos de la imagen rectificada. Tras el proceso de rectificación, las imágenes de la izquierda y la derecha aparecen como si se hubieran tomado con un par de cámaras paralelas (Fan, Ai, y Dahnoun,

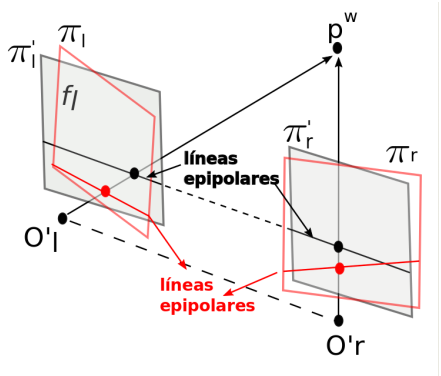


Figura 2.9. Representación de la rectificación estéreo, se observa los planos de la imagen original (recuadro negro) y los planos de la imagen rectificada (recuadro rojo) de acuerdo a la referencia de las lineas epipolares, reproducido de (Fan, Ai y Dahnoun, 2018).

2018). Por lo tanto, la búsqueda de los pares de correspondencia se simplifica a un proceso unidimensional (1-D).

2.2.4. Modelo básico del mapeo de disparidad

Un sistema calibrado de forma correcta puede representarse en la figura 2.10, puede ser considerado como una especialización de la geometría epipolar, dado que se considera que la lente de cada cámara esta alineada en forma paralela una con respecto a la otra y los sistemas de referencia son colineales (ejes $X_l^{\mathscr{C}}$ y $X_r^{\mathscr{C}}$).

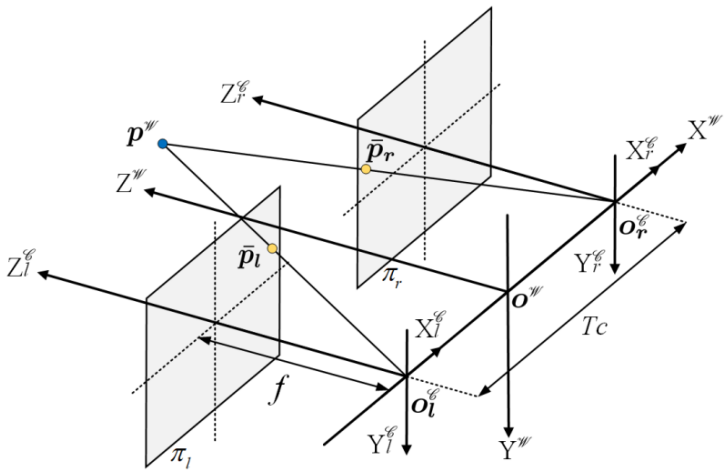


Figura 2.10. Representación del modelo de visión estéreo básico, se muestran los planos de referencia izquierdo y derecho para la representación de puntos arbitrarios en el sistema de coordenadas 3-D determinados por los parámetros intrinsecos y extrínsecos para la proyección de subplanos en el mapa de disparidad, tomado de (Scharstein y Szeliski, 2002)

Se supone que el equipo estereoscópico tiene dos cámaras coplanares (izquierda y derecha) con los mismos parámetros intrínsecos y su coaxial horizontal paralelo al plano horizontal, donde el ángulo de inclinación respecto al plano horizontal es $O_l^{\varphi}\ O^{w}\ O_r^{\varphi}$. Los objetos y las superficies horizontales en el sistema de coordenadas se consideran como una colección de pequeños planos y representan las características de proyección de varios planos primarios en el mapa de disparidad (Scharstein y Szeliski, 2002) .

Para un punto tridimensional arbitrario $p^{'W} = [X^{'W}, Y^{'W}, Z^{'W}]$ situado en una superficie plana, sus proyecciones en las imágenes izquierda y derecha se pueden relacionar con una matriz homógrafa normal a la superficie plana.

Para un sistema de visión estereoscópica bien calibrado, la búsqueda de los pares de correspondencia sólo implica una búsqueda 1-D y por tanto el mapa de disparidad puede obtenerse simplemente calculando la distancia horizontal entre cada par de correspondencias del sistema de coordenadas del mundo real (SCM) $\overline{p}_l$ y $\overline{p}_r$. Este proceso se denomina generalmente estimación de la disparidad o coincidencia estéreo (Fan, 2018).

2.3. Detección de objetos

En lo que respecta a la detección de objetos a partir de imágenes, en el campo del aprendizaje profundo las redes neuronales artificiales han experimentado un incremento constante de popularidad en los últimos años, especialmente las redes neuronales convolucionales (RNC).

Las redes neuronales artificiales tienen el mérito de ser capaces de aprender características importantes y relevantes dados los datos de entrenamiento en cantidad suficiente. La ventaja de las RNC sobre los clasificadores convencionales se encuentra en las capas convolucionales ya que se obtienen durante el entrenamiento varios filtros y mapas de características fundamentales para su alto desempeño (Leon y Gavrilescu, 2021).

Las RNC son capaces de aprender las características de los objetos mediante múltiples y complejas operaciones de optimización. La elección adecuada de los parámetros y la arquitectura de la red puede garantizar que estas características contengan las correlaciones necesarias para la identificación robusta de los objetos, aunque esta elección suele ser un proceso empírico.

En la literatura relacionada con respecto a la detección de objetos en ambientes vehículares existe una amplia variedad de configuraciones de RNC, algunos ejemplos concretos son aquellas propuestas por (Cui et al., 2019; S. Li, Ma, Chang, Shan, y Chen, 2018; Ristani y Tomasi, 2018).

2.3.1. Predicción de clase

Las RNC además de detectar objetos de interés también realizan una predicción de pertenencia con respecto a un número de clases posibles a identificar. Esta predicción se indica a través de un cuadro envolvente del objeto detectado en escena así como el porcentaje de pertenencia de clase utilizando una clasificación multietiqueta. Durante el entrenamiento se utiliza la pérdida de entropía cruzada binaria para las predicciones de clase por medio de clasificadores logísticos independientes (Cui et al., 2019).

Por ejemplo, la RNC-YOLO predice cajas a 3 escalas diferentes para extraer cualidades distintivas de esas escalas utilizando un concepto similar al de las redes de pirámides de características (Tan, Pang, y Le, 2020).

A partir de la extracción de características base se añaden capas convolucionales para extraer aquellas cualidades relevantes, en específico la última capa predice un tensor tridimensional que codifica la caja delimitadora y las predicciones de clase. En el siguiente paso se toma el mapa de características de las 2 capas anteriores y se incrementa en dos, a continuación, se utiliza un mapa de características de una capa anterior de la red y se fusiona con las particularidades del muestreo ascendente utilizando la concatenación para predecir un tensor similar, aunque ahora del doble de tamaño.

Este método permite obtener información semántica más significativa de las cualidades distintivas muestreadas e información más detallada del mapa de características anterior.

Por último se realiza la adición de mas capas convolucionales bajo el modelo adición + concatenación para predecir cajas para la escala final. De este modo, las predicciones para la tercera escala se benefician de todo el cálculo previo, así como de las características de grano fino de las primeras fases de la red. En este punto y con los datos obtenidos se eligen 9 clusters, 3 escalas arbitrariamente y luego se dividen los clusters uniformemente a través de las escalas.

La red planteada utiliza un número de capas convolucionales sucesivas de $3x3$ y $1x1$ conexiones de acceso directo, se presenta de forma resumida esta arquitectura en la tabla 2.1.

2.3.2. Aprendizaje profundo YOLO

Este sistema predice cuadros delimitadores utilizando grupos de dimensiones como recuadros de anclaje. La red predice 4 coordenadas (t_x, t_y, t_w, t_h) para cada cuadro envolvente (posible objeto detectado) y si la celda esta desplazada con respecto a la esquina superior izquierda de la imagen (c_x, c_y), además el cuadro envolvente calculado previamente tiene una anchura y altura (p_w, p_h) específica, con margen de error σ, por lo que entonces la predicción

Tabla 2.1. Arquitectura básica de la RNC - YOLO, composición de las capas convolucionales, dimensiones de los filtros utilizados y escalas uniformes de los clusters de salida, reproducida de (Redmon et al., 2016).

	Tipo	Filtros	Tamaño	Salida
	Convolucional	32	3x3	256x256
	Convolucional	64	3x3/2	128x128
	Convolucional	32	1x1	
1x	Convolucional	64	3x3	
	Residual			128x128
	Convolucional	128	3x3/2	64x64
	Convolucional	64	1x1	
2x	Convolucional	128	3x3	
	Residual			64x64
	Convolucional	256	3x3/2	32x32
	Convolucional	128	1x1	
8x	Convolucional	256	3x3	
	Residual			32x32
	Convolucional	512	3x3/2	16x16
	Convolucional	246	1x1	
8x	Convolucional	512	3x3	
	Residual			16x16
	Convolucional	1024	3x3/2	8x8
	Convolucional	512	1x1	
4x	Convolucional	1024	3x3	
	Residual			8x8
	Promedio		global	
	conectadas		1000	

es confirmada. La predicción se puede determinar con la ecuación 2.2 (Redmon, Divvala, Girshick, y Farhadi, 2016).

$$b_x = \sigma(t_x) + c_x, \quad b_y = \sigma(t_y) + c_y, \quad b_w = p_w e^{t_w}, \quad b_h = p_h e^{t_h} \tag{2.2}$$

En la figura 2.11 se ilustra la búsqueda de la información en los bloques de pixeles de una imagen para determinar la predicción y el cuadro envolvente.

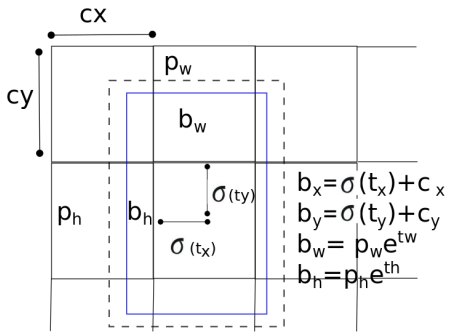


Figura 2.11. Mapeo de las coordenadas para determinar la predicción del posible objeto a detectar en el frame de acuerdo a la referencia superior izquierda, reproducido de (Redmon et al., 2016).

Durante la etapa de entrenamiento se utiliza el error cuadrático medio, dado el *ground truth* $\hat{t}_*$ para la predicción de coordenadas (t_x, t_y, t_w, t_h) , el gradiente corresponde a la diferencia con respecto a la predicción $\hat{t}_* - t_*$.

Las RNC-YOLO utilizan características de toda la imagen para detectar y predecir los cuadros delimitadores de cada objeto detectado en escena así como predecir todas las clases de una imagen simultáneamente.

RNC-YOLO cuenta con un sistema de detección que divide la imagen de entrada en una malla de SxS celdas, cada celda de la malla es la responsable de detectar objetos si el centro de un objeto cae en una celda. Cada celda predice B cuadros delimitadores y puntuaciones de confianza para dichos cuadros, estas puntuaciones reflejan el grado de precisión que tiene el modelo al detectar objetos.

La confiabilidad se define como $Pr(Objecto) * IOU_{pred}^{truth}$, si no existe objeto en la celda. Las puntuaciones de confianza deben ser cero, en caso contrario, la puntuación de confianza puede ser igual a la intersección sobre la unión (IOU) entre la casilla predicha y el *ground truth* de la imagen. Es importante mencionar además que cada celda de la cuadrícula también predice probabilidades de clase condicionales (C), es decir $Pr(Clase_i|Objeto)$.

Como se ha mencionado las probabilidades están condicionadas a que la celda de la cuadrícula contenga un objeto, para obtener las probabilidades de clase se establece la ecuación 2.3 (Redmon et al., 2016).

$$Pr(Clase_i|Objeto) * Pr(Objeto) * IOU_{pred}^{truth} = Pr(Clase_i) * IOU_{pred}^{truth} \tag{2.3}$$

Donde se multiplican las probabilidades de clase condicionales y las predicciones de confianza de cada casilla, lo que proporciona puntuaciones de confianza específicas de cada clase para cada casilla. Estas puntuaciones codifican tanto la probabilidad de que esa clase aparezca en la caja como el grado de adecuación de la caja predicha al objeto.

El algoritmo YOLO asigna un puntaje objetivo para cada cuadro envolvente al utilizar una regresión logística, esta puntuación debe ser 1 si el cuadro delimitador previo se traslapa con un objeto correcto en mayor medida que cualquier otro cuadro delimitador previo. El umbral establecido corresponde a 0.5, si la predicción del cuadro delimitador no cumple con el parámetro objetivo pero se traslapa con un objeto real, aún así se ignora la predicción. Paras las capas de la RNC se utiliza la función 2.4 de activación lineal rectificada con fugas y para la capa final se utiliza una función de activación lineal (Redmon et al., 2016).

$$\phi(x) = \begin{cases} x & si\ x > 0 \\ 0{,}1x & en\ otro\ caso \end{cases} \tag{2.4}$$

Se optimiza el error de la suma de cuadrados en la salida para facilitar la optimización del modelo, sin embargo no se ajusta perfectamente a la maximización de la precisión media ya que pondera por igual el error de localización y el de clasificación, algo que no es ideal.

Además, en cada imagen muchas celdas de la cuadrícula no contienen ningún objeto, esto implica parámetros de confianza en las celdas, con frecuencia se supera el gradiente de las celdas que sí contienen objetos. Esto puede conducir a la inestabilidad del modelo, provocando que al inicio el entrenamiento se desvíe (de las características de interés).

Para evitar las desviaciones del entrenamiento, se incrementa la pérdida de las predicciones de las coordenadas del cuadro delimitador y se reduce la pérdida en las predicciones de confianza para las casillas que no contienen objetos. Se usan para este fin 2 parámetros λ_{coord} y λ_{noobj} con los siguientes valores $\lambda_{noobj} = 0.5$, $\lambda_{coord} = 5$.

Durante el entrenamiento se optimiza la función 2.5 de perdida multiparte, donde 1_i^{obj} denota si el objeto aparece en la celda i y 1_{ij}^{obj} denota que el $j-esimo$ cuadro envolvente predictor en la celda i corresponde a la predicción (Redmon et al., 2016).

$$\begin{aligned}
\lambda_{coord} &= \sum_{i=0}^{s^2}\sum_{j=0}^{B} 1_{ij}^{obj}\left[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2\right] \\
&+\sum_{i=0}^{s^2}\sum_{j=0}^{B} 1_{ij}^{obj}\left[\left(\sqrt{w_i}-\sqrt{\hat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right] \\
&+\sum_{i=0}^{s^2}\sum_{j=0}^{B} 1_{ij}^{obj}(C_i-\hat{C_i}^2) \\
&+\lambda_{noobj}\sum_{i=0}^{s^2}\sum_{j=0}^{B} 1_{ij}^{obj}(C_i-\hat{C_i}^2)
\end{aligned} \tag{2.5}$$

La función de pérdida (ecuación 2.5) sólo penaliza el error de clasificación si un objeto está presente en esa celda de la cuadrícula, es decir, la probabilidad de clase condicional. También se penaliza el error de coordenadas de la caja delimitadora es decir, tiene el mayor intersección sobre unión (IOU) de cualquier predictor en esa celda de la cuadrícula.

2.4. Seguimiento de objetos

El enfoque general de seguimiento en video se basa en seguimiento de múltiples objetos (multiple object tracking, MOT) en dos dimensiones, este enfoque se basa en una sucesión de pasos de detección y seguimiento. Las detecciones consecutivas que se clasifican de forma similar se unen para determinar las trayectorias (Cao et al., 2020).

El problema de seguimiento en video puede resumirse como la asignación de un identificador para todos los objetos de interés (OI) detectados en un frame y posteriormente intentar hacer coincidir los identificadores en los frames siguientes. Esto suele ser una tarea compleja, teniendo en cuenta que los objetos rastreados pueden entrar y salir del frame en diferentes momentos también es posible que el entorno los ocluya o que se ocluyan entre sí.

Otros problemas pueden ser causados por defectos en las imágenes adquiridas: ruido, artefactos de muestreo o compresión, aliasing o errores de adquisición (Shabanian y Balasubramanian, 2021).

Tener en cuenta las variaciones de movimiento conlleva una serie de problemas adicionales, como cuando los objetos se ven afectados por transformaciones de rotación o de escala, o cuando la velocidad de movimiento de los objetos es alta en relación con la velocidad de captura de la imagen.

El cálculo de características robustas es un aspecto importante de la detección de objetos; las características son representativas de una amplia gama de propiedades del objeto: color, frecuencia y distribución, forma geometría, contornos o correlaciones dentro de los objetos segmentados. Un modelo de movimiento analiza la curva de desplazamiento del objetivo en los frames históricos y predice la posición del objetivo en el frame actual.

La mayoría de los rastreadores de un solo objeto no utilizan un modelo de movimiento sin embargo el modelo de movimiento ha demostrado ser útil en el seguimiento de múltiples objetos ya que puede ayudar a localizar los objetivos y realizar la correspondencia de las etiquetas de estos en diferentes frames.

En la mayoría de las aplicaciones MOT se utiliza un modelo de movimiento lineal simple para estimar el estado del objetivo. Tales modelos de movimiento pueden causar una pérdida de seguimiento cuando el objetivo gira rápidamente, se detiene de repente o se mueve en reversa. En la figura 2.12 (Meng, Wang, Wang, Shao, y Fu, 2020) se presenta la metodología general mencionada con respecto al seguimiento multiobjetos.

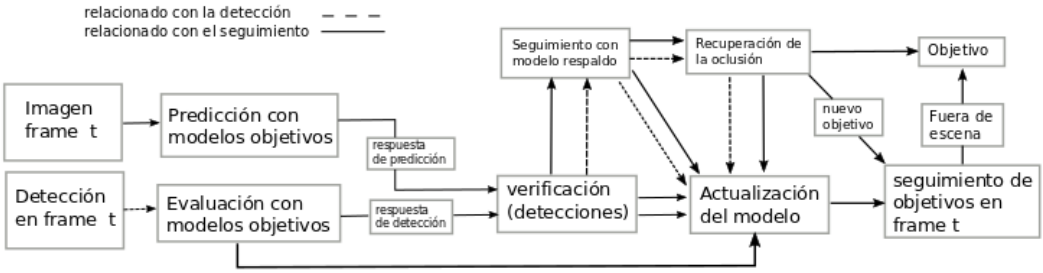


Figura 2.12. Bloques implementados en la metodología de detección y seguimiento de múltiples objetos a través de una escena, reproducido de (Meng et al., 2020).

Actualmente, los métodos de detección de características implican un aprendizaje supervisado, estos enfoques requieren datos de entrenamiento adecuados y una cuidadosa selección de hiperparámetros a menudo mediante el método de ensayo y error. Sin embargo, muchos resultados muestran que los métodos de clasificación y regresión supervisada ofrecen los mejores resultados tanto en términos de precisión y robustez frente a las transformaciones afines, la oclusión y el ruido (Chu, Fan, Tan, y Ling, 2019).

2.5. Formulación del seguimiento multiobjeto

Los modelos de rastreo se utilizan para predecir las ubicaciones de los objetivos de forma independiente y estimar el acierto en la detección por medio de puntuaciones. El problema del seguimiento puede formularse como un problema de optimización, donde en un frame, el conjunto de objetivos N^t con localización espacial $\hat{X}^t = \{\hat{x}_i^t\}_{i=1}^{N^t}$ en una imagen actual I^t son elegidos de M^t candidatos del conjunto global $O^t = \{\hat{x}_j^t\}_{j=1}^{M^t}$ para maximizar el resultado como en las ecuaciones 2.6 y 2.7 (Cao et al., 2020).

$$\hat{X}^t = argmaxf(I^t, X^t; a^t, W^{t-1}) \tag{2.6}$$

$$\sum_i a_{ij}^t \leq 1, a_{ij}^t \in \{0,1\} \tag{2.7}$$

El parámetro $a^t = \{a_{ij}^t \in \{0,1\}\}$ indica la asociación entre el $i-esim$ objetivo rastreado en $\hat{X}^{t-1}$ en el frame $t-1$ y la $j-esima$ localización en $\hat{X}^t$ en el frame en el tiempo t (función 2.8).

$$a_{ij}^t = \begin{cases} 1 & si\ \hat{x}_i^{t-1}\ es\ asociado\ con\ x_j^t \\ 0 & en\ otro\ caso \end{cases} \tag{2.8}$$

Cada candidato solo puede asignarse como máximo a un objetivo rastreado, donde $W^t = \{w_i^t\}_{i=1}^{N^t}$ es el conjunto de parámetros para modelar cada objetivo, que generalmente se aprende a través de un procedimiento de entrenamiento utilizando la información de apariencia o ubicación del objetivo (Cao et al., 2020).

La función objetivo f mide la calidad general de los resultados del seguimiento para todos los objetivos en el marco t, definido como:

$$f(I^t, X^t) = \sum_{ij} a_{ij}^t g_i(I^t, x_j^t; w_i^{t-1}) \tag{2.9}$$

El conjunto de funciones g_i puedes ser interpretadas como la función objetivo para el segui-

miento de un solo objetivo de modo que $g_i(I^t, x_j^t; w_i^{t-1})$ asigna un puntaje al $j-esimo$ candidato de localización x^t en I^t de acuerdo a el $i-esimo$ parámetro del modelo $w_i^{t-1} \in W^{t-1}$.

Los parámetros del modelo deben determinarse mediante imágenes anteriores y ubicaciones destino hasta el frame $t-1$. Resolver el problema de MOT, por lo tanto, es resolver a^t y g_i para cada frame.

2.6. Marco metodológico: Redes dinámicas bayesianas

A continuación se presentan algunas consideraciones técnicas que denotan las capacidades de las Redes Bayesianas (RB) y por consiguiente su extensión en un modelo dinámico (Redes Dinámicas Bayesianas RDB).

Una Red Bayesiana se asocia con un conjunto de variables aleatorias $X = (X_1, X_2, ..., X_N)$ mediante el par $B = (G, \theta)$, donde G es una estructura de RB, es decir, un grafo directo acíclico (GDA) cuyos nodos corresponden a las variables $X_i \ \epsilon \ X$, cuyos ejes representan su dependencia condicional y θ representa el conjunto de parámetros que codifican las probabilidades condicionales de cada variable de nodo dados sus padres.

Las distribuciones están representadas por una tabla de probabilidad condicional (TPC) cuando un nodo y sus padres representan variables discretas o mediante una distribución de probabilidad condicional (DPC) cuando un nodo representa una variable continua (Likforman-Sulem y Sigelle, 2008).

Cada DPC generalmente sigue una función de densidad de probabilidad gaussiana. Una propiedad clave de las RB es que los factores de distribución de probabilidad conjunta están dados por:

$$P(X_1, X_2, ..., X_N) = \prod_{i=1}^{N} P(X_i | Pa(X_i)) \tag{2.10}$$

donde $Pa(X_i)$ denota los padres de X_i; esta propiedad es central en el desarrollo de algoritmos de inferencia rápida (Sucar, 2015).

Las redes bayesianas suelen representar el estado de ciertos fenómenos en un instante de tiempo, sin embargo en muchas aplicaciones se requiere representar la evolución temporal de un determinado proceso, es decir, como se modifican las distintas variables a lo largo del tiempo.

Existen dos tipos básicos de modelos de redes bayesianas para procesos dinámicos los basados en estados y los basados en eventos. Los modelos basados en estados representan el estado de cada variable en intervalos de tiempo discretos, de modo que las redes consisten en una serie de cortes de tiempo, donde cada corte de tiempo indica el valor de cada variable

en el momento t, estos modelos se denominan redes dinámicas bayesianas.

Las RDB son una extensión de las redes bayesianas para modelar procesos dinámicos. Una RDB consiste en una serie de cortes temporales que representan el estado de todas las variables en un momento determinado t, similar a una captura instantánea del proceso temporal en evolución.

Para cada corte temporal se define una estructura de dependencia entre las variables en ese momento, denominada red base. Generalmente se implica que esta estructura está duplicada para todos los cortes temporales excepto el primer corte que puede ser diferente.

Además existen aristas entre variables de diferentes cortes, con sus direcciones en relación con el tiempo, definiendo así la red de transición.

En general, las RDB tiene enlaces dirigidos entre cortes temporales consecutivos, lo que se conoce como un modelo de Markov de primer orden. En la figura 2.13 se muestra un ejemplo de RDB con 3 variables y 4 cortes temporales (L. Song, Kolar, y Xing, 2009).

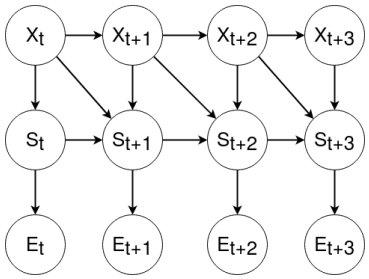


Figura 2.13. Representación de una RDB con 3 variables y 4 cortes, en este caso la estructura base se repite cuatro veces, reproducida de (Song et al., 2009).

La mayoría de las RDB satisfacen dos condiciones particulares, la primera condición implica que las variables de estado en el momento t sólo dependen de las variables de estado en el momento $t-1$ (y de otras variables en el momento t), a esto se le llama modelo de primer orden de Markov. La segunda condición indica que la estructura y los parámetros del modelo no cambia con el tiempo, es decir, la condición del proceso estacionario.

Las RDB pueden considerarse una generalización de las cadenas de Markov y de los modelos ocultos de Markov (HMM). Una cadena de Markov es la RDB más sencilla en la que sólo hay una variable Xt por cada tramo de tiempo, influida directamente sólo por la variable en el tiempo anterior. En este caso, la distribución conjunta puede escribirse como en la ecuación 2.11 (Sucar, 2015).

$$P(X_1, X_2, X_3, ..., X_T) = P(X_1)P(X_2|X_1)...P(X_t|X_{T-1}) \tag{2.11}$$

Por su parte los modelos ocultos de Markov tienen dos variables por etapa en el tiempo,

es decir, la variable de estado S y la variable observable Y. Generalmente se considera que S_t depende sólo de S_{t-1} de la misma forma que Y_t sólo depende de S_t. Por tanto, la probabilidad conjunta puede ser factorizada como en la ecuación 2.12 (Sucar, 2015).

$$P\{(S_{1:T}, Y_{1:T})\} = P(S_1)P(Y_1)|S_1) \prod_{t=2}^{T} P(S_t|S_{t-1})P(Y_y|S_t) \tag{2.12}$$

Cabe mencionar que las cadenas de Markov y los HMMS son casos particulares de RDB, en general pueden tener N variables por fase de tiempo, con cualquier estructura de base y de transición. Otra variante particular de las RDB son los filtros de Kalman ya que tienen una variable de estado y otra de observación además ambas variables son continuas. El filtro de Kalman básico considera distribuciones gausianas y funciones lineales para los modelos de transición y observación.

2.6.1. Inferencia en RDB

La inferencia probabilística consiste en propagar los efectos de ciertas pruebas en una red bayesiana para estimar su efecto sobre las variables desconocidas. Es decir, conociendo los valores de algún subconjunto de variables en el modelo, se obtienen las probabilidades posteriores de las demás variables. El subconjunto de variables desconocidas puede estar vacío, en este caso se obtienen las probabilidades a priori de todas las variables (Sucar, 2015).

Existen varias formas de procesar la inferencia en las RDB, a continuación se hará un mención en general de las principales formas. Los conceptos que se deben recordar son las variables ocultas (o no observables) y las variables observables, en este caso se denotan por $\mathbf{X}$ y $\mathbf{Y}$ respectivamente.

1. Inferencia por filtrado. Predice el siguiente estado basado en las observaciones previas, $P(\mathbf{X}_{t+1}|\mathbf{Y}_{1:t})$.

2. Inferencia por predicción. Predice estados siguientes basado en observaciones pasadas, $P(\mathbf{X}_{t+n}|\mathbf{Y}_{1:t})$.

3. Inferencia suavizada (Smoothing). Estima el estado actual basado en observaciones pasadas y futuras, $P(\mathbf{X}_t|\mathbf{Y}_{1:T})$.

4. Inferencia decodificada (Decoding). Encuentra la secuencia más probable de las variables ocultas dadas las observaciones, $ArgMax(\mathbf{X}_{1:T})P(\mathbf{X}_{1:T}|\mathbf{Y}_{1:T})$ (Murphy, 2002).

Sin embargo, existen algoritmos eficientes (polinómicos) para ciertos tipos de estructuras (grafos uniconexos) mientras que para otras estructuras depende de la conectividad del grafo. En muchas aplicaciones los grafos son dispersos, por lo que en este caso existen algoritmos de inferencia que son muy eficientes (Likforman-Sulem y Sigelle, 2008).

Existen varias clases de técnicas para la inferencia de probabilidad en redes bayesianas multiconectadas, las principales son por medio de eliminación de variables, condicionamiento y árbol de unión.

2.6.1.1. Inferencia por eliminación de variable

La técnica de eliminación de variables se basa en la idea de calcular la probabilidad marginando la distribución conjunta. Sin embargo, a diferencia del enfoque ingenuo, este aprovecha las condiciones de independencia de la red bayesiana y las propiedades asociativas y distributivas de la suma y la multiplicación para realizar los cálculos de manera eficiente.

Se asume una representación de la DPC de $\mathbf{X} = \{X_1, X_2, ..., X_n\}$ donde se requiere calcular la probabilidad posterior de una cierta variable o subconjunto de variables X_H, se considera X_E como un subconjunto de evidencias de las variable (Sucar, 2015). Las variables restantes se denotan por X_R, por lo que $\mathbf{X} = \{X_H \cup X_E \cup X_R\}$.

La probabilidad posterior de X_H dada la evidencia esta dada por:

$$P(X_H|X_E) = P()X_H, X_E)/P(X_E) \tag{2.13}$$

Además se puede obtener ambos términos por medio de la umbralización de la distribución conjunta.

$$\begin{aligned} P(X_H, X_E) &= \sum_{X_R} P(\mathbf{X}) \\ P(X_E) &= \sum_{X_H} P(X_H, X_E) \end{aligned} \tag{2.14}$$

Un caso particular de interés es obtener la probabilidad marginal de las variables cuando no hay pruebas, en este caso $X_E = \emptyset$. Otro cálculo de interés es obtener la probabilidad de la evidencia, ésta viene dada por la ecuación 2.14. El objetivo de la técnica de eliminación de variables es realizar estos cálculos de forma eficiente, por este motivo se representa primero la distribución conjunta como un producto de probabilidades locales según la estructura de la red. A continuación, el sumatorio pueden realizarse con el subconjunto de términos que son una función de las variables que se están normalizando.

Este enfoque aprovecha las propiedades de la suma y la multiplicación, lo que permite reducir el número de operaciones necesarias.

El aspecto crítico del algoritmo de eliminación de variables es seleccionar el orden apropiado para eliminar cada variable, ya que esto tiene un efecto importante en el número de operaciones requeridas. Los diferentes términos que se generan durante los cálculos se conocen como factores, que son funciones sobre un subconjunto de variables, que asignan cada instanciación de estas a un número no negativo (estos números no son necesariamente probabilidades), en general, un factor puede representarse como $f(x_1, X_2, ...X_m)$.

Una desventaja de la eliminación de variables es que sólo obtiene la probabilidad posterior de una variable (o subconjunto de variables) así que para obtener la probabilidad posterior de cada una de las variables no iniciadas en una RDB hay que repetir los cálculos para cada variable.

2.6.1.2. Inferencia por condicionamiento

El método de condicionamiento se basa en el hecho de que una variable instanciada bloquea la propagación de la evidencia en un grafo bayesiano. Este método puede interrumpir u omitir el grafo en una variable instanciada por lo que el grafo multiconectado se transforma en un multiárbol, al que se le puede aplicar el algoritmo de propagación de probabilidad.

En general, se puede instanciar un subconjunto de variables para transformar un grafo multiconectado en un grafo conectado simple. Si las variables no son conocidas, se les puede asignar valores para posteriormente hacer la propagación de la probabilidad para cada valor.

Con cada propagación se obtiene una probabilidad para cada variable desconocida, posteriormente los valores finales de probabilidad se obtienen como una combinación ponderada de estas probabilidades. En primer lugar se desarrolla el algoritmo de condicionamiento suponiendo que sólo se necesita particionar una única variable para expandir la operación a múltiples variables. El planteamiento se define para obtener la probabilidad de cualquier variable B dada la evidencia E, condicionando en la variable A. Por la regla de la probabilidad total (Murphy, 2002) se postula la ecuación 2.15.

$$P(B|E) = \sum_i P(B|E, a_i)P(a_i|E) \tag{2.15}$$

Donde $P(B|E, a_i)$ es la probabilidad posterior de B la cual es obtenida por propagación para cada posible valor de A y $P(a_i|E)$ es un valor con peso específico dada la evidencia de comportamiento de la topología, es decir, dada la regla de bayes se obtiene por:

$$P(a_i|E) = \alpha\, p(a_i)P(E|a_i) \tag{2.16}$$

El término $P(a_i)$ se obtiene por propagación sin evidencia, $P(E|a_i)$ se calcula propagando dada la asignación de $A = a_i$ para obtener la probabilidad de la evidencia de las variables, finalmente α es una constante normalizada.

En general, para transformar una red multiconectada en un poliárbol se necesita instanciar m variables, por lo tanto, la propagación debe realizarse para todas las combinaciones de valores (producto cruz) de las variables instanciadas. Si cada variable tiene k valores, el número de propagaciones es de k^m.

2.6.1.3. Inferencia por árbol de unión

El método del árbol de unión se basa en una transformación de la red bayesiana en un árbol de unión, donde cada nodo de este árbol es un grupo o cluster de variables de la red original y la inferencia probabilística se realiza sobre esta nueva representación. La intuición detrás del método del árbol de unión se basa en una transformación de una red (que es un grafo dirigido) a un grafo de Markov (grafo no dirigido), es decir, un agrupamiento de las variables lo que implica como resultado que el grafo obtenido esté conectado individualmente.

2.7. Arquitecturas RBD

La simulación progresiva de estos modelos para diferentes rutas posibles de todos los objetos de interés permite realizar predicciones de escenas multi-modales y conscientes de la interacción en diseños de rutas arbitrarias.

Existen diferentes topologías de referencia para construir una RDB de acuerdo al planteamiento del modelo y la respectiva predicción de las observaciones de un proceso de interacción estacionario.

El modelo dinámico se plantea de acuerdo a los nodos raíz y las relaciones causales y temporales consecutivos. Cada intervalo de muestreo puede contener variables observables y variables no observables, que pueden estar vinculadas. Por lo tanto, las topologías de las RDB son capaces de modelar secuencias de interacción con estructura temporal y espacial a con múltiples resoluciones dependiendo de las condiciones específicas del planteamiento del problema (Sucar, 2015).

A continuación se presentan algunas arquitecturas de referencia, a tomar en consideración, estas consisten en el planteamiento de un modelo de RDB para la estimación de la intención de movimiento.

2.7.1. Red dinámica bayesiana basada en un modelo Markoviano

Una escena de tráfico consiste en un conjunto de objetos $V = \{V^0, V^1, ..., V^k\}$, con $K \epsilon N_0$, en un entorno estático (mapa) en tiempo discreto, estado continuo y espacio de acción continua. El mapa consta de una red de carreteras con información topológica, geométrica y de infraestructura (líneas de rendimiento, señales de tráfico, etc.), así como las normas de tráfico vigentes (Schulz, Hubmann, Morin, Löchner, y Burschka, 2019).

En el paso de tiempo t cada uno de los objetos está representado por sus intenciones de ruta r_t^i y su estado cinemático $\mathbf{x}_t^i = [x_t^i, y_t^i, \psi_t^i, v_t^i]$ que comprende la posición cartesiana, rumbo y velocidad absoluta. Las longitudes y anchos de los objetos se consideran dados, pero por razones de brevedad, no están incluidos en x^i.

La intención de la ruta r_t^i define un camino a través de la red de carreteras que el objeto desea seguir. En cada intervalo de tiempo, cada agente ejecuta una acción $\mathbf{a}_t^i = [a_t^i, \delta_t^i]$ que comprende la aceleración longitudinal y el ángulo de dirección. Esta acción depende de la intención de ruta del objeto, el mapa y los estados cinemáticos de todos los elementos, transformando la cinemática del estado actual x_t^i al nuevo estado x_{t+1}^i.

Las mediciones de ruido $z_t^i = [z_{x,t}^i, z_{y,t}^i, z_{\theta,t}^i, z_{v,t}^i]$ se utilizan para actualizar la creencia del estado del ente.

El objetivo de (Schulz et al., 2019) es derivar un modelo de acción preciso dado por $p(a^i | r^i, x^0, ..., x^K, mapa)$ que permita predecir el próximo estado cinemático de un objeto lo más cerca posible dado su estado cinemático actual y su intención de ruta.

Este modelo está destinado a integrarse como un modelo de transición probabilístico en algoritmos basados en muestreo, la entrada al modelo es determinista (una muestra de la creencia), mientras que la salida es una distribución de probabilidad sobre las acciones, de la cual se puede extraer muestras nuevamente si se requiere.

La acción se modela para distribuir normalmente dada una intención de ruta específica y el contexto de la situación actual:

$$p(a^i | r^i, x^0, ..., x^K, mapa) = \left(\begin{bmatrix} \mu_a \\ \mu_\delta \end{bmatrix}, \begin{bmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_\delta^2 \end{bmatrix} \right)$$

Se modela el desarrollo de una situación de tráfico como un proceso de Markov que consta de múltiples elementos que interactúan. La acción de un ente en un intervalo de tiempo específico t se modela para ser independiente de las acciones de los otros entes en el mismo paso de tiempo dados sus estados cinemáticos actuales.

Esto se muestra en la parte izquierda de la figura 2.14 donde se presenta un extracto de la

RDB que muestra las dependencias de la acción de un agente en su intención de ruta y los estados de los agentes circundantes.

Este modelo de acción Markoviano utiliza una red neuronal (figura 2.14 lado derecho), que define el mapeo probabilístico de las características a una distribución de acción gaussiana que comprende la aceleración y el ángulo de dirección δ. Así la acción a^i del agente V^i solo depende de su intención de ruta r^i y de los estados cinemáticos $[x^0,...,x^K]$ de todos los agentes, pero no en las acciones de otros agentes. Por lo tanto, todos los agentes se pueden predecir independientemente de t a $t+1$ (Schulz et al., 2019).

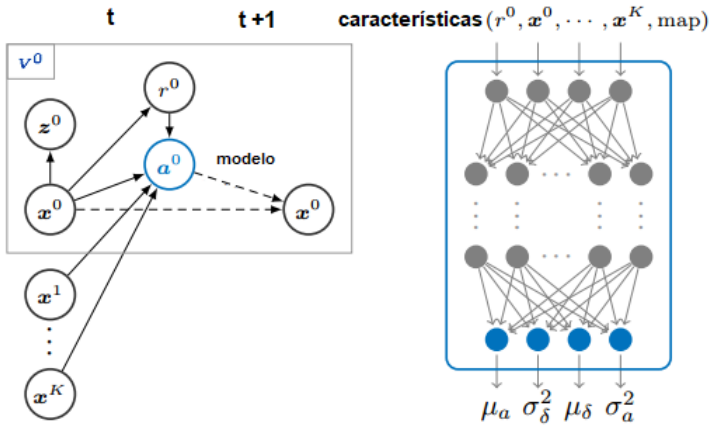


Figura 2.14. Estructura topológica del Modelo Markoviano RDB con enfoque de multiagentes (lado izquierdo) y modelo neuronal convolucional (lado derecho), tomado de (Schulz et al., 2019).

A medida que el contexto se actualiza en cada intervalo con el tiempo surge una interdependencia entre las trayectorias de múltiples agentes. A diferencia de los modelos de acción basados en reglas existentes, se establece aprender el mapeo probabilístico del contexto y la intención de ruta a la distribución de la acción con redes neuronales profundas.

2.7.2. Red bayesiana dinámica híbrida multiagente

La predicción interactiva con múltiples participantes del tráfico en escenarios altamente dinámicos es extremadamente difícil para la conducción autónoma, especialmente cuando están involucrados agentes heterogéneos como vehículos y peatones. Los métodos de predicción existentes encuentran problemas de interpretabilidad y generalización para abordar una tarea tan complicada.

(L. Sun, Zhan, Wang, y Tomizuka, 2019) proponen un método de Red Bayesiana Dinámica Híbrida Multi-agente (RBDHM), que puede modelar los cambios de estado de múltiples agentes heterogéneos en una variedad de escenarios. Se incorporan conocimientos previos,

como información de mapas y reglas de tráfico, en la estructura del gráfo además se utiliza el filtro de partículas (FP) para rastrear, predecir las intenciones y trayectorias de los agentes.

El ambiente seleccionado involucra información de movimiento con interacciones entre peatones - vehículos de una intersección de cuatro paradas en el mundo real.

La escena del tráfico formulada incluye varios tipos de participantes en el tráfico con diferentes características. En el paso de tiempo k cada agente $i = 0, 1, 2, ...N$ en interacción tiene un estado continuo $X_i(k)$,un estado latente discreto $Z_i(k)$ y acción $A_i(k)$.

El estado continuo describe las características de comportamiento, que incluyen estados cinemáticos y muestras, es decir, características observables.

Las variables en el espacio de estado latente discreto se definen para facilitar el análisis de toma de decisiones y acciones, diseñadas para diferentes tipos de agentes según las reglas de tráfico. La acción describe el movimiento en el mapa en un lapso de tiempo específico, que revela la transición cinemática estimada de cada agente bajo interacción.

Una observación $Y_i(k)$ se define como la medida con ruido de los estados observables $X_i(k)$. Las trayectorias se representan en un Frenet frame con posiciones longitudinales y laterales con respecto a la ruta de referencia.

El conocimiento previo se incluye dentro de la estructura de diferentes maneras, la información del mapa y las reglas de tráfico se usan para diseñar estados latentes de diferentes tipos de agentes, mientras que las dependencias causales se usan para diseñar dependencias condicionales.

En la figura 2.15 se muestra la topología de la red propuesta por (L. Sun et al., 2019) donde los nodos discretos y continuos son rectangulares y circulares respectivamente, las líneas continuas, las líneas discontinuas y las líneas punteadas son dependencias causales, temporales y de observación respectivamente.

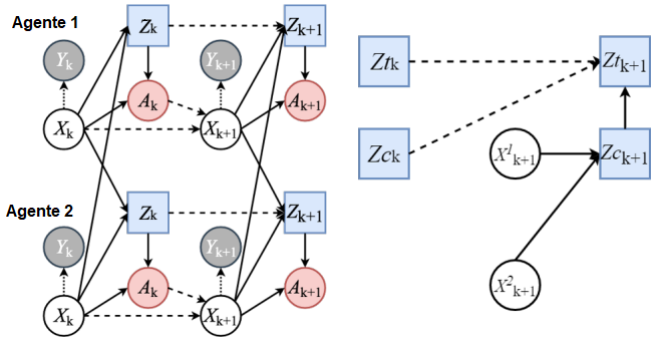


Figura 2.15. La RBDHM se despliega en dos lapsos de tiempo (izquierda) y se descompone en la dependencia condicional de estado latente (derecha), tomado de (L. Sun et al., 2019).

Los modelos dinámicos de agente se utilizan para diseñar el espacio de estado de inten-

ción y las variables de acción. Se debe notar que la existencia de variables discretas latentes $Z_i(k)$ diseñadas para incluir las intenciones de cada agente. La distribución de $Z_i(k)$ depende de todos los estados observables $X_1(k), X_2(k), ..., X_N k)$ pero no de los estados latentes de otros agentes $Z_j(k)$, este diseño evita los componentes acíclicos en la red.

2.7.3. Redes bayesianas dinámicas multifase

La metodología de determinación de niveles de integridad de seguridad (SIL) se basa en redes bayesianas dinámicas multifase (RBDM) para sistemas de seguridad instrumentados.

La fase de entrenamiento y la fase de prueba se modela por separado utilizando redes dinámicas bayesianas para fusionarlas y formar los RBDM (Cai, Liu, Liu, Chang, y Jiang, 2020). Es necesario construir los modelos de estructura unificada de RBDM, denominadas arquitecturas $k\ out\ of\ n$, para desarrollar procedimientos automáticos y así crear las tablas de probabilidad condicional del sistema de estudio planteado.

Las medidas de falla objetivo, es decir, la probabilidad de falla bajo demanda, la probabilidad promedio de falla bajo demanda, la probabilidad de falla segura, la probabilidad promedio de falla segura y el SIL de los sistemas de seguridad instrumentados operan en un modo de baja demanda y se evalúan utilizando la propuesta RBDM.

Esta en investigación aún los efectos del intervalo de tiempo de los RBDM, el peso de causa común, la prueba imperfecta y la precisión del modelo.

2.7.4. Estructura de los RBDM

En los sistemas instrumentados de seguridad (SIS) se requiere pruebas para verificar que el SIL especificado se cumpla y se mantenga durante un intervalo de tiempo. Las pruebas puede detectar fallas ocultas, potencialmente peligrosas, que no se pueden detectar a través de los métodos de monitoreo tradicional.

El intervalo de testeo es el período de tiempo entre pruebas adyacentes. Además, el intervalo de prueba varía para cada SIS y depende de la tecnología, la arquitectura del sistema y el SIL objetivo. Por lo tanto, el modelo RBDM propuesto para la determinación de SIL se compone de dos fases, es decir, la fase del intervalo de prueba y la fase de prueba.

El autodiagnóstico, la falla y la reparación en la fase del intervalo de prueba se pueden modelar utilizando RDB porque el intervalo de prueba generalmente es un período de tiempo largo. Del mismo modo, la operación en la fase de prueba también se puede modelar utilizando RDB de acuerdo con las condiciones reales.

Los RDB para la fase de intervalo de diagnostico y los RDB para la fase de prueba se integran para formar los RBDM, como se muestra en la figura 2.16.

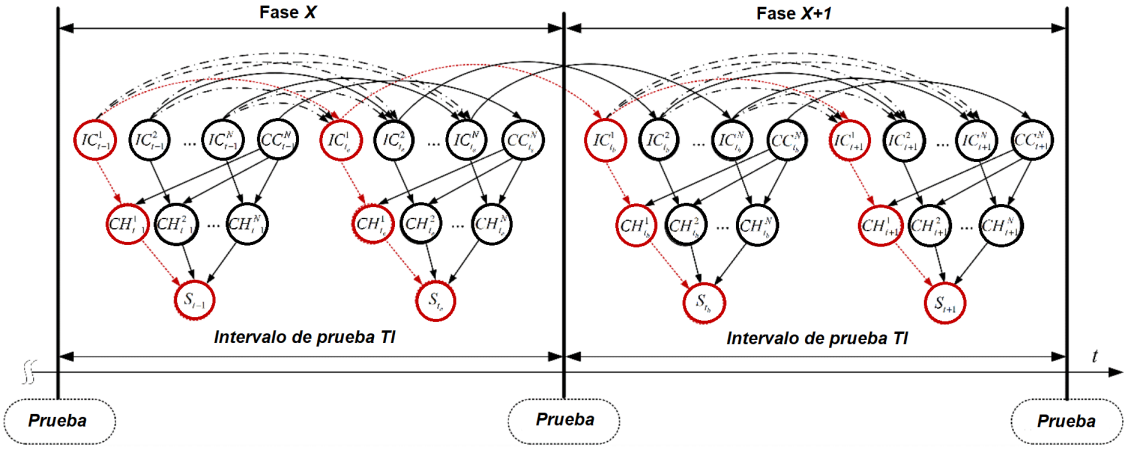


Figura 2.16. Diagrama esquemático de RBDM para la determinación de SIL en dos intervalos consecutivos de interés, reproducido de (Cai et al., 2020).

El intervalo de prueba es un período a largo plazo; las RDB incluyen muchos segmentos de tiempo, denotados por indices $t-1$ y t. Se construye un modelo de estructura RBD unificado para la fase de intervalo de prueba(figura 2.16).

Los RDB se divide en tres capas de arriba hacia abajo, es decir, causa de falla, estado del canal y capas de estado del sistema sucesivamente. El nodo IC en la capa de causa de falla denota la falla de causa independiente para un solo canal y el nodo CC denota la falla de causa común para múltiples canales.

La falla de causa común se puede denominar falla de modo común y falla dependiente, además, la falla de causa común es el resultado de un evento. Estos eventos con dependencias causan una coincidencia de estados de falla de los componentes en dos o más canales separados de un sistema de redundancia, lo que lleva a la falla de los sistemas definidos para realizar su función prevista.

El nodo CH en la capa de estado de un solo canal denota el estado de un solo canal afectado simultáneamente por un fallo de causa independiente y un fallo de causa común. Por lo tanto, los nodos IC están conectados a los nodos CH correspondientes y el nodo CC está conectado a todos los nodos CH a través de arcos.

El nodo S en la capa de estado del sistema denota el estado de un sistema completo que consta de todos los canales. Los nodos IC, CC y CH tienen cinco estados, es decir, estado normal (NS), falla segura detectada (SD), falla segura no detectada (SU), falla peligrosa detectada (DD) y falla peligrosa no detectada (DU).

Para el nodo S, los estados SD y SU se combinan para formar un estado de seguridad (SS). Por lo tanto, el nodo S tiene cuatro estados, es decir, NS, SS, DD y DU. En segmentos de tiempo adyacentes, las relaciones causales de los nodos IC se ilustran conectando los nodos correspondientes a través de arcos entre cortes (línea continua negra o línea de trazos rojos) y otros nodos a través de arcos auxiliares entre cortes (línea de trazos de puntos negros) (Cai et al., 2020).

Aplicar RBD en el caso específico del problema de predecir trayectorias es un desafío, ya que es necesario tener en cuenta múltiples hipótesis e interacciones a largo plazo entre múltiples objetos en movimiento (Wen, Du, Li, Bian, y Lyu, 2019).

2.7.5. Discretización de las variables de interés

El problema de la discretización de variables es esencialmente el encontrar para cada variable continua, un conjunto de valores de umbral que dividen un valor continuo en un número finito de intervalos.

Las RDB son un método cada vez más popular para modelar sistemas en ambientes del mundo real, a menudo se requiere la discretización de variables continuas para poder implementar arquitecturas de RDB novedosas.

Hay tres métodos principales de discretización; manual, supervisado y no supervisado. Los resultados demuestran que los métodos de discretización supervisados producen redes bayesianas con enfoque predictivo promedio de alrededor del 73.8 %, seguidos de la discretización manual (69.2 %) y la discretización no supervisada (64.8 %). Sin embargo, cada método tiene ventajas específicas que pueden hacerlos más adecuados para aplicaciones particulares (Beuzen, Marshall, y Splinter, 2018).

Los métodos manuales pueden producir redes bayesianas para entornos físicos significativos, lo que es favorable en la modelización ambiental. Los métodos supervisados pueden discretizar variables de manera autónoma y óptima por lo que pueden ser preferidos cuando la habilidad predictiva es una prioridad de modelado. Los métodos no supervisados son computacionalmente simples y versátiles.

El esquema de discretización óptimo debe considerar tanto el rendimiento como la practicidad del método.

2.7.6. Métodos de discretización

Al discretizar un atributo numérico Υ, se supone que hay n instancias de entrenamiento para las que se conoce el valor de Υ_i, el valor mínimo y máximo respectivamente, cada

método de discretización primero clasifica los valores en orden ascendente.

A continuación se mencionan los métodos usados con mayor frecuencia en aplicaciones con RDB.

El método denominado EWD (Equal Width Discretization) divide las instancias entre el valor mínimo y máximo en κ-intervalos de igual longitud.

Por otro lado EFD (Equal Frequency Discretization) divide los valores de las instancias n ordenadas en κ intervalos, tal que, cada intervalo contiene $\frac{n}{\kappa}$ valores adyacentes, el valor κ es un parámetro predefinido por quien analiza el fenómeno o sistema.

EMD (Entropy Minimization Discretization) evalúa como posible punto de corte el punto medio entre cada par sucesivo de valores ordenados. Para evaluar cada candidato a punto de corte, los datos se discretizan en dos intervalos y se calcula la entropía de información de clase resultante. La discretización binaria se determina seleccionando el punto de corte para el que la entropía es mínima entre todos los candidatos. La discretización binaria se aplica de forma recursiva, siempre seleccionando el mejor punto de corte. Se aplica un criterio de longitud de descripción mínima (MDL) para decidir cuándo detener la discretización.

PKID (Proportional k-Interval Discretization) ajusta el sesgo de discretización y la varianza ajustando el tamaño y el número del intervalo. La idea detrás de PKID es que el sesgo y la varianza de discretización se relacionan con el tamaño y el número del intervalo. Cuanto mayor sea el tamaño del intervalo (cuanto menor sea el número de intervalo), menor será la varianza pero cuanto mayor sea el sesgo. A la inversa, cuanto menor sea el tamaño del intervalo (cuanto mayor sea el número de intervalo), menor será el sesgo pero mayor la varianza. Se puede lograr un error de aprendizaje más bajo ajustando el tamaño y el número del intervalo para encontrar una buena compensación entre el sesgo y la varianza (Nojavan, Qian, y Stow, 2017).

Es de igual forma importante hacer notar el número de intervalos a elegir al discretizar variables continuas para modelos basados en redes bayesianas es recomendable de tres a cinco, este es un rango realista considerando las restricciones comúnmente impuestas por la disponibilidad de datos y la complejidad de los modelos.

Puede parecer que más intervalos representarían mejor datos continuos; sin embargo, el tamaño de la tabla de probabilidad condicional para cada nodo, calculado como el producto del número de intervalos de ese nodo y el número de intervalos de cada nodo padre, también aumenta. Incluso en una red simple de tres nodos, la diferencia entre tres y cinco intervalos para cada variable da como resultado el cálculo de 27 (3^3) frente a 125 (5^3) probabilidades condicionales (Nojavan et al., 2017).

2.7.7. Complejidad computacional

Las tareas de inferencia bayesiana a menudo involucran atributos numéricos. Para los algoritmos bayesianos, los atributos numéricos a menudo se procesan previamente mediante discretización, ya qué el rendimiento de la inferencia tiende a ser mejor cuando los atributos numéricos se discretizan que cuando se supone que siguen una distribución normal.

Dependiendo del algoritmo a utilizar y la cantidad de datos el costo computacional puede incrementar las necesidades del procesamiento del equipo. Tomando en consideración este parámetro a continuación se presenta la complejidad computacional de varios algoritmos de discretización de variable contínua utilizados en métodos bayesianos.

La tabla 2.2 (Yang y Webb, 2002) muestra el método y la complejidad asociado al mismo. Donde n es el número de instancias, m el número de clases a discretizar y l es el número de instancias de prueba.

Tabla 2.2. Resumen de los distintos métodos de discretización y su complejidad computacional, tomado de (Yang y Webb, 2002).

Método	Complejidad
EWD (equal width discretization) EFD (equal frecuency discretization) PKID (proportional k-interval discretization) PKID (proportional k-interval discretization) WPKID (weighted proportional k-interval discretization) NDD (nondisjoint discretization)	O(n log n)
EMD(entropy minimization discretization)	O(mn log n)
ID (iterative discretization)	O(n)
MDL (minimum descriptive length)	O(n^2)
LD (lazy discretization)	O(nl)

2.8. Parámetros de evaluación

En la literatura relacionada, se han propuesto distintas métricas para la evaluación cuantitativa del seguimiento de múltiples objetos, la cuestión es elegir las métricas de evaluación adecuadas a la información y resultados obtenidos. Por un lado, es preferible resumir el rendimiento en un sólo número para permitir una comparación directa, por otro lado, no se debe omitir la información respecto a los errores parciales cometidos al implementar los algoritmos. Por lo tanto se debe proporcionar la estimación del rendimiento general de la metodología a evaluar (Ristani, Solera, Zou, Cucchiara, y Tomasi, 2016).

Existen dos métricas muy utilizadas en la literatura para la evaluación del seguimiento de objetos en movimiento *Multiple Object Tracking Accuracy* (MOTA) y *Multiple Object Tracking*

Precision (MOTP) (Y. Xu, Ban, Alameda-Pineda, y Horaud, 2019; Hu et al., 2019; Weng, Wang, Held, y Kitani, 2020; Wu, Han, Wen, Li, y Wang, 2021; Meng et al., 2020).

MOTA y MOTP tienen en cuenta los errores cometidos por el detector de objetos, los falsos positivos, los falsos negativos y los desajustes en los frames analizados. Ofrecen una medida intuitiva del rendimiento en el seguimiento de objetos y en el mantenimiento de sus trayectorias, independientemente de la precisión con la que se estiman las posiciones de los objetos. A continuación se entra en detalle con respecto a estas métricas de evaluación.

MOTA es una métrica utilizada ampliamente para evaluar el rendimiento de un rastreador, la razón principal es que esta métrica combina tres fuentes de errores definidos: falso positivo *FP*, falso negativo *FN* y la equivalencia en cambio de identidad *IDSW* (equivalently an identity switch).

El parámetro MOTA puede ser negativo en casos donde el número de errores cometidos por el rastreador excede el número de todos los objetos en la escena *gt* (ground truth), cabe señalar que *t* se considera como el índice del número consecutivo de cuadros a analizar en el video, la expresión matemática se representa por la ecuación 2.17.

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t gt_t} \tag{2.17}$$

Como se mencionó MOTA puede considerarse derivado de 3 relaciones de error. En primer lugar, la proporción de fallos en la secuencia, calculada sobre el número total de objetos presentes en todos los frames analizados (ecuación 2.18).

$$\overline{FN}_t = \frac{\sum_t FN_t}{\sum_t gt_t} \tag{2.18}$$

En segundo lugar la relación de falsos positivos (ecuación 2.19) y por último la relación de desajustes (ecuación 2.20).

$$\overline{FP}_t = \frac{\sum_t FP_t}{\sum_t gt_t} \tag{2.19}$$

$$\overline{IDSW}_t = \frac{\sum_t IDSW_t}{\sum_t gt_t} \tag{2.20}$$

Al sumar los distintos porcentajes de error se obtiene la tasa de *Error total*, por tanto, *1-Error total* es la precisión de seguimiento resultante.

Aunque el parámetro MOTA obtiene una buena aproximación del rendimiento general del sistema, es discutible si sólo este número puede servir como una medida de rendimiento única, por lo que otro parámetro de evaluación usado de forma recurrente es MOTP (Bernardin

y Stiefelhagen, 2008).

MOTP es la diferencia promedio entre todos los verdaderos positivos y sus correspondientes objetivos *gt*, para la superposición del cuadro delimitador, esto se calcula con la expresión 2.21.

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t} \tag{2.21}$$

Donde c_t denota el número de coincidencias en el cuadro t y $d_{t,i}$ es la superposición del cuadro delimitador del objetivo i con el objeto gt asignado.

MOTP da la superposición promedio entre todas las hipótesis correctamente coincidentes y sus respectivos objetos. Es importante señalar que MOTP es una medida de precisión de localización, que no debe confundirse con el valor predictivo positivo o la relevancia en el contexto de las curvas de precisión/recuerdo utilizadas en la detección de objetos (Meng et al., 2020).

Además MOTP muestra la capacidad del rastreador para estimar posiciones precisas de los objetos, independientemente de su habilidad para reconocer configuraciones de objetos, mantener trayectorias consistentes, etcétera.

Con estos parámetros concluye la sección del marco teórico, en el siguiente capítulo se presenta y se analiza el estado del arte relacionado con el tema de investigación.

Capítulo 3

Análisis del estado del arte

Las técnicas encontradas en la literatura que abordan el problema de la detección de objetos y estimación de desplazamiento en entornos de tráfico vehicular mediante visión computacional son diferentes entre sí.

La interpretación y percepción del ambiente en video se realiza por niveles, en el primer nivel se tienen a las características como apariencia, tamaño y disparidad de los objetos. En el siguiente nivel está el seguimiento el cual corresponde a la asociación de datos, coherencia temporal así como filtrado el cual permite rastrear, identificar y medir los parámetros dinámicos para estimar las posiciones de los objetos. En el nivel superior se utilizan un conjunto de características espacio-temporales para aprender, modelar, clasificar y predecir el comportamiento de los objetos en la carretera.

Se pueden definir un conjunto de etapas generales necesarias para la implementación de la tarea de detección de objetos e inferencia dinámica del entorno vehicular para estimar trayectorias de desplazamiento.

Las etapas identificadas en la literatura son: detección de objetos de interés (OI) presentes es regiones específicas de la escena, estimación de la distancia a la que se encuentran los OI y la dinámica de estos con respecto a la posición del ego-vehículo (es decir, el vehículo sobre el cual se encuentra montado el sensor que captura la información ambiental) y finalmente la estimación de la trayectoria de los obstáculos en el entorno (Chandra, Bhattacharya, Bera, y Manocha, 2019; Kampker, Sefati, Rachman, Kreisköther, y Campoy, 2018; Prabhakar, Kailath, Natarajan, y Kumar, 2017).

Los bloques necesarios del planteamiento metodológico general, encontrado en la literatura, para llevar a cabo la tarea de detección de OI y el cálculo de la trayectoria se muestran en el diagrama de la figura 4.7.

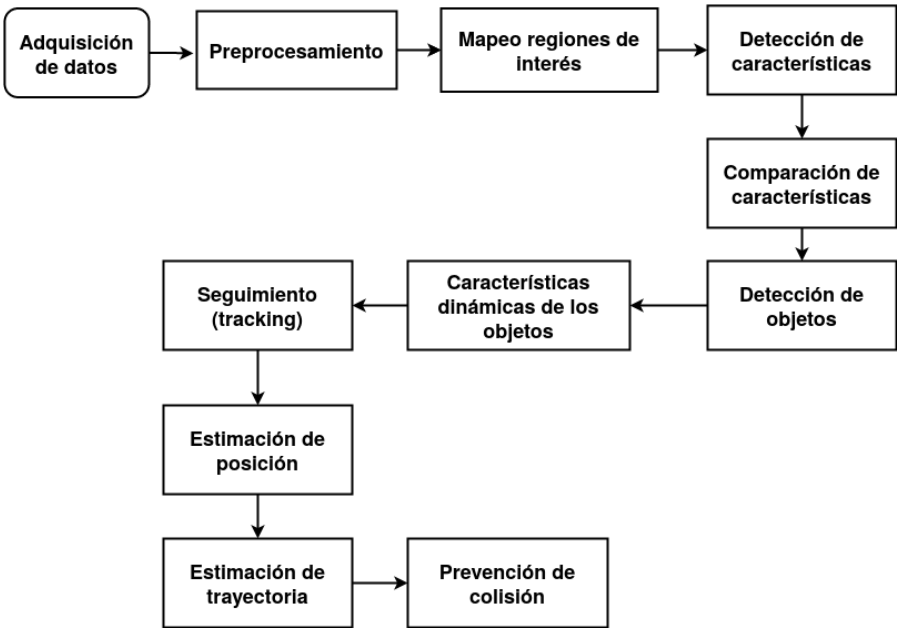


Figura 3.1. Bloques generales para la implementación de la detección y estimación de trayectorias de objetos, tomando en consideración los enfoques encontrados en la literatura.

3.1. Trabajos relacionados

El problema de la estimación de la trayectoria en entornos vehiculares se ha abordado ampliamente, por ejemplo, (Duan, Li, Guan, Sun, y Cheng, 2020) presentan un método de aprendizaje por refuerzo jerárquico para la toma de decisiones en vehículos autónomos de tal forma que no dependa de una gran cantidad de datos etiquetados para llevar a cabo la conducción.

Este enfoque modela y aprende el cambio de dirección como un proceso de decisión markoviano, por lo que en cada intervalo de tiempo de interés, el vehículo observa un estado, realiza una acción, recibe una señal escalar y finalmente alcanza el estado siguiente.

Sin embargo, este modelo presenta variación debido al proceso de arranque aleatorio y al número de veces que el vehículo autónomo llega al destino en cada época de acuerdo al modelo de aprendizaje planteado. En comparación, la velocidad de aprendizaje y el rendimiento tiene rango de mejora según el autor.

En el artículo publicado por (C. Xu, Zhao, y Wang, 2019), se desarrolla un esquema de normalización de características y se establece una estrategia para construir modelos de regresión de procesos gaussianos tridimensionales a partir de patrones de trayectoria bidimensionales para capturar las características espacio-temporales de situaciones de tráfico. Sin embargo, dado que el entorno del tráfico es un sistema dinámico e incierto, la

acción posterior obtenida por el modelo no es óptima al ejecutar una secuencia de decisiones, ya que la velocidad en este proceso se considera invariante, lo que no ocurre en situaciones en entornos reales.

Por otro lado, (Schulz et al., 2019), modelan el proceso en una RDB que permite la especificación de las relaciones entre los objetos, así como las dependencias causales y temporales para manejar la incertidumbre de las mediciones. El proceso planteado de toma de decisiones para cada obstáculo visible se compone de tres capas jerárquicas: intención de ruta, intención de maniobra y acción continua.

La estructura de la red se adapta en tiempo de ejecución (creando y eliminando hipótesis de ruta - maniobra), sin embargo el autor sólo ha probado este método en 3 escenarios de tráfico diferentes.

Como se mencionó en los trabajos anteriores, la problemática por resolver consiste en determinar el conjunto de posibles rutas y maniobras dada la información percibida del ambiente y en algunos casos la información proporcionada a través de un mapa topológico. De manera que la acción continua de cada objeto detectado se especifica mediante modelos de comportamiento dependientes del contexto, sus intenciones de ruta y maniobra (González, Garzón, Dibangoye, y Laugier, 2019; Hou, Chen, y Chen, 2019; P. Sun et al., 2020; Tran y Firl, 2014; Kampker et al., 2018; W. Wang y Neumann, 2018).

Es importante señalar que la actualización recurrente de la creencia con observaciones de posición y velocidades de un grupo de OI permite inferir las intenciones de que exista o no un cambio de ruta. Con esta información, se genera una predicción probabilística de la trayectoria al simular - procesar directamente la creencia actual. De este modo, las trayectorias previstas incorporan explícitamente la mayor probabilidad de cambio de trayectoria y consideran interdependencias entre múltiples objetos (Asljung, Westlund, y Fredriksson, 2019; J. Liu et al., 2020).

En resumen, los trabajos relacionados proponen integrar un sistema de transición probabilístico en algoritmos basados en el muestreo de la entrada hacia el modelo determinista (una muestra de la creencia), mientras que la salida es una distribución de probabilidad sobre las acciones, de las que se pueden extraer muestras recursivas si es necesario. La acción se plantea para distribuir normalmente dada una intención de trayectoria potencial y el marco de referencia de la situación en el instante actual.

3.2. Análisis de trabajos relacionados

La primera etapa para la detección de objetos en ambientes vehiculares es la adquisición de la información, esta es una forma inicial de separar a los trabajos existentes con base en el sensor utilizado.

Las técnicas y algoritmos utilizados para detectar obstáculos varían con respecto al sensor utilizado en función de los datos obtenidos; para los tipo radar y lidar la segmentación del ambiente es espacial de acuerdo a las mediciones de distancia-tiempo.

Para el caso de las técnicas que utilizan captura de imagen a través de cámaras de video, la detección de obstáculos se fundamenta en la apariencia y el movimiento.

La tabla 3.1 muestra una comparativa de los sensores utilizados en los distintos trabajos de investigación[1] para la detección de objetos en ambientes con tránsito vehicular. De la tabla 3.1 se observa que el sensor con mayor porcentaje de aplicación es el de tipo óptico (a través de cámaras de video monoculares o estereoscópicas), esto debido a su menor costo en comparación con los sensores laser tipo lidar.

Tabla 3.1. Comparativa de adquisición de información para la percepción del ambiente.

Adquisición de datos	**Energía percibida**	**Parámetro de medición**	**Unidades**	**Reconocimiento de objetos**	**Porcentaje trabajos que usan el tipo de sensor**
Óptico (cámara)	Luz visible (ambiente)	Intensidad de luz	pixeles	Vía apariencia, movimiento	75 %
lidar	Señal láser de long. de onda 600-1000 nm (emitida)	Distancia	metros	Vía segmentación espacial, movimiento	18 %
Radar Ultrasónico	Señal de radio frecuencia milimétrica (emitida)	Distancia	metros	Vía seguimiento	6 %

Debido a la cantidad de información (color, textura, resolución, posición espacial, etc.) obtenida de una imagen digital, las técnicas de visión computacional se asemejan a la percepción humana del entorno, por ejemplo, cuando un conductor visualiza y detecta objetos alrededor de un vehículo que se mueve en tres dimensiones.

Por esta razón, se decide trabajar con la información capturada en video para alcanzar las tasas de detección de trabajos recientes y aprovechar las ventajas que presentan las técnicas de visión computacional contra otras técnicas que utilizan otros tipos de sensores, en específico el radar láser.

En la literatura se reportan múltiples estrategias para la detección y seguimiento de objetos a través de visión computacional, a continuación se mencionan brevemente algunos enfoques encontrados.

[1] Se realizó una revisión sistemática de ochenta trabajos relacionados con el tema de interés

Las técnicas basadas en visión estéreo producen un mapa de disparidad denso que se utiliza como entrada para los algoritmos de disparidad adaptativos U-V para el mapeo de las regiones de interés (ROI) y la consiguiente localización espacial de los obstáculos (Roggeman et al., 2017a).

Los algoritmos evolutivos (EA) detectan múltiples obstáculos en el video mediante el uso de un algoritmo de regresión genética para prevenir la detección falsa a través de la percepción del movimiento. La base del método es el procesamiento de la función de aptitud física con los parámetros de aptitud heurística, lo que significa la elección de los parámetros de aptitud óptimos utilizando EA y el método de mínimos cuadrados (Du, Yan, Chen, y Hua, 2020).

El último enfoque a mencionar se trata de las técnicas basadas en aprendizaje, las cuales generan resultados de segmentación mediante redes neuronales, por ejemplo, las redes neuronales convolucionales (RNC) (W. Wang y Neumann, 2018) o Redes Neuronales Completas (RNC) (Zhang, Qiu, Yao, Liu, y Mei, 2018). Estas técnicas suelen transferir los marcos de referencia de un dominio espacial a otro desde la perspectiva de la apariencia visual (color RGB a la segmentación de escenas en profundidad).

Los algoritmos basados en aprendizaje profundo pueden proporcionar resultados de segmentación con precisión, pero el costo computacional es elevado (Prabhakar et al., 2017) incluso en simulaciones (Mo, Xing, y Lv, 2020). Además, la mayoría de estos algoritmos necesitan un conjunto extenso de datos lo cual lleva a invertir mucho tiempo en recopilar y etiquetar las imágenes en profundidad.

Los métodos para la detección de objetos y seguimiento se pueden abordar por medio de una descripción general, los conceptos relevantes se encuentran en los métodos basados en la apariencia y los métodos basados en el movimiento, estos se describen a continuación.

3.3. Métodos basados en la apariencia y movimiento

Textura y color. La textura de los objetos varía con respecto a su entorno (fondo), se puede mencionar por ejemplo la superficie de la carretera, los edificios o la vegetación. Las técnicas utilizadas incluyen tanto el análisis estadístico sobre entropía como la matriz de concurrencia para segmentar las regiones de interés (regions of interest ROI) en las imágenes del ambiente especificado (Cao et al., 2020; Liang, Jiang, Murphy, Yu, y Hauptmann, 2020).

Enfoque basado en estéreo. La geometría multidimensional hace posible la medición directa de información en 3D, lo que permite comprender la escena, las características de movimiento y las mediciones físicas. La variación en las imágenes izquierda y derecha entre

los pixeles correspondientes de una escena se denomina disparidad.

Con la concordancia estéreo de las disparidades de todos los puntos de la imagen se calcula el mapa de disparidad. Si se conocen los parámetros de la plataforma estéreo, el mapa de disparidad se puede transformar en una vista 3D de la escena observada.

La disparidad en vertical (d-v) se ha utilizado ampliamente para modelar la superficie del suelo, a fin de identificar objetos que se encuentran sobre el terreno. La disparidad en d-v forma un histograma de valores de disparidad para ubicaciones de pixeles con la misma coordenada de imagen vertical (Mukherjee, Adarsh, y Ramachandran, 2021),(Y. Xie, Zeng, Zhang, y Chen, 2017).

Múltiples características. Es posible combinar múltiples parámetros para la detección de OI, por ejemplo: entropía, características de simetría, cuadrículas de ocupación de objetos, múltiples capas de anotación de información semántica. De este modo, se establecen las regiones de la imagen que potencialmente pertenecen a la información semántica, posteriormente, se eliminan todas las filas con baja entropía en la región detectada y se verifica la intensidad horizontal de simetría para determinar si pertenece o no a un OI.

Las clases semánticas y las características de integración dinámica de la escena se pueden combinar para identificar los objetos en diversas condiciones de iluminación y rastrear con una referencia (filtro de partículas PF). Esta fusión da como resultado un error promedio de desplazamiento (EPD) aproximado (Salzmann, Ivanovic, Chakravarty, y Pavone, 2020), (Weng et al., 2020), (Wu et al., 2021), (Hu et al., 2019).

Característica aprendidas. Sistemáticamente las características convolucionales para la tarea de detección y seguimiento son más útiles que otras calculadas explícitamente como por ejemplo: Haar, Fusión de Histograma de Gradientes Orientados (FHGO), etiquetado de colores. Además este sentido se trata el seguimiento de múltiples objetos utilizando combinaciones de valores de capas convolucionales situadas en múltiples niveles. El método se basa en la noción de que las capas convolucionales representan de mejor forma la imagen de entrada y, por tanto, contienen más detalles para la identificación de objetos (Chen, Ai, Shang, Zhuang, y Bai, 2017),(Q. Liu, Lu, He, Zhang, y Chen, 2017),(C. Wang, Galoogahi, Lin, y Lucey, 2018), (Y. Xu, Ban, et al., 2019).

Enfoque basado en probabilidad. En general, los métodos basados en Filtros de Kalman se utilizan para un seguimiento más sencillo, sobre todo en escenarios en los que el bloque de rastreo sólo accede a la información de un número limitado de frames a la vez, posiblemente sólo los actuales y los anteriores (C. Sun, Karlsson, Wu, Tenenbaum, y Murphy, 2019). En el trabajo presentado por (Deo, Rangesh, y Trivedi, 2018) se propone un proceso que incluye el seguimiento multivista, la proyección del plano del suelo, el reconocimiento de maniobras

y la predicción de trayectorias. El método implica una serie de enfoques, entre los que se incluyen los modelos ocultos de Markov así como los modelos de mezcla variable gaussiana (Variational Gaussian mixture).

Flujo óptico. El enfoque basado en el flujo óptico es uno de los pocos métodos fundamentados únicamente en el movimiento, implica el cálculo de píxeles coincidentes o puntos característicos entre cuadros consecutivos. El movimiento aparente de los objetos, superficies y bordes en una escena son parametrizados de acuerdo al movimiento relativo entre el observador y la escena. La tasa de detección de este enfoque se encuentra en 84.83 % (MOTA) para una velocidad relativa de desplazamiento de 5 km/hr - 25 km/hr (Luiten, Fischer, y Leibe, 2020).

Actualmente la fusión de técnicas han logrado obtener resultados significativos en cuanto a la tasa de detección de objetos y seguimiento, por ejemplo: los métodos basados en la correlación son aquellos que utilizan comparación de plantillas (template matching), los métodos basados en el aprendizaje son aquellos que usan clasificadores de características, como lo pueden ser las redes neuronales artificiales basadas en LSTM (long short term memory) o incluso la metodología de máquinas de vector de soporte (MVS). En la tabla 3.2 se muestra un sumario de estos métodos y el desempeño mostrado.

Tabla 3.2. Comparativa de las técnicas basadas en el aprendizaje y en la correlación.

Tipo	Técnica	Métodos	Resultados cuantitativos
Basados en la correlación	Comparación de plantillas	Entropía de imagen y análisis de sombra	Tiempo de ejecución 30 cps, tasa de detección 75 % (Wan, Li, y Yau, 2017)
		Gráfo, Unidades recurrentes controladas, mezcla gaussiana	Errores de predicción 10 % lapsos de 4 seg (N. Lee et al., 2017)
		Bordes verticales y características de sombras para localización de ROI	Tasa de detección 90.37 % a una velocidad de procesamiento de 6 cps (Lan, Jiang, y Yu, 2015)
		Filtro de partículas, estimación del estilo de conducción, simulación Monte Carlo	Horizonte de predicción 0.5 a 1 s (Hoermann, Stumper, y Dietmayer, 2017a)
Basados en aprendizaje	Clasificadores de características	Red convolucional basada en LSTM	Tasa de clasificación 88 % (Kampker y Sefati, 2018)
		Seguimiento mediante características convolucionales a través de filtros de correlación	Tasa de detección MOTA 87.5 % enfocado en peatones ambientes controlados (Ristani y Tomasi, 2018)
		Algoritmo adaBoost con características Haar	Tasa de detección 71 %, vel. de procesamiento a 39 cps (Y.-K. Lai, Chou, y Schumann, 2017)
		Entrenamiento con característica de Gabor en un clasificador MVS	90.8 % tasa de detección, vel. de procesamiento a 30 cps (Ponz et al., 2015)
		Redes neuronales convolucionales	67.8 % tasa de detección MOTA MOTP = 85.2 % (Ren, Lu, Wang, Tian, y Zhou, 2018)
		Softmax para conjunto de trayectorias discretas	Error de desplazamiento promedio 0.64 (Phan-Minh y Grigore, 2020)

Para incrementar la tasa de detección de obstáculos y posteriormente estimar trayectorias

los métodos analizados hasta el momento son insuficientes, por lo que se requieren técnicas complementarias para alcanzar una tasa de detección mayor.

3.4. Métodos basados en aprendizaje profundo

En la última década se ha progresado de forma acelerada con respecto a los avances en el área del aprendizaje profundo (deep learning DL) y la inteligencia artificial en la aplicación de modelos para la estimación de desplazamiento. La aplicación de las metodologías de aprendizaje en la estimación de desplazamiento implica: la percepción de la escena de conducción, planificación de la trayectoria, arbitraje del comportamiento y control del movimiento (Bojarski et al., 2017; H. Xu, Gao, Yu, y Darrell, 2017; Eraqi, Moustafa, y Honer, 2017; Hecker, Dai, y Van Gool, 2018).

Las redes neuronales convolucionales, redes neuronales recurrentes y el aprendizaje profundo por refuerzo, son las metodologías de aprendizaje más comunes aplicadas a la estimación de dirección de desplazamiento.

3.4.1. Redes neuronales convolucionales

Las redes neuronales convolucionales (RNC) son capaces de aprender automáticamente una representación del espacio de características codificado en el conjunto de entrenamiento. Una RNC está parametrizada por su vector de pesos $q = [W, b]$, donde W es el conjunto de pesos que rigen las conexiones interneuronales y b es el conjunto de valores de sesgo de las neuronas. El conjunto de pesos W se organiza como filtros de imagen, con coeficientes aprendidos durante el entrenamiento. Las capas convolucionales de una RNC explotan las correlaciones espaciales locales de los píxeles de la imagen para aprender los filtros de convolución invariantes de la traslación, que capturan las características discriminantes de la imagen (Grigorescu, Trasnea, Cocias, y Macesanu, 2020).

Por ejemplo, para una RNC se considera una representación de señal multicanal M_k en la capa k, que es una integración por canales de las representaciones de la señal $M_{k,c}$ donde $c \in N$. Se puede generar una representación de la señal generada en la capa $k+1$ considerando la formula 3.1.

$$M_{k+1,l} = \phi(M_k * w_{k,l} + b_{k,l}) \tag{3.1}$$

Donde $w_{k,l} \in W$ es un filtro convolucional con el mismo número de canales y $M_k, b_{k;l} \in b$ representa el sesgo, l es un índice de canal y $*$ denota la operación de convolución, ϕ es una

función de activación aplicada a cada pixel de la señal de entrada. Normalmente, la unidad lineal rectificada (ReLU) es la función de activación más utilizada en aplicaciones de visión por computadora. La última capa de una RNC suele ser una capa totalmente conectada que actúa como discriminador de objetos en una representación abstracta de alto nivel (Zhu, Yuen, Mihaylova, y Leung, 2017).

De forma supervisada, la respuesta de una RNC puede entrenarse utilizando una base de datos, los parámetros óptimos de la red pueden calcularse mediante la estimación de máxima verosimilitud (MLE).

Se toma como ejemplo la función de error por mínimos cuadrados simple, que puede utilizarse para dirigir el proceso de MLE cuando se entrenan estimadores de regresión. Para la clasificación, el error por mínimos cuadrados suele sustituirse por la entropía cruzada o las funciones de pérdida de logaritmo negativo. El problema de optimización se resuelve normalmente con el gradiente descendente estocástico y el algoritmo de retropropagación para la estimación del gradiente (Rausch et al., 2017; Bechtel, McEllhiney, Kim, y Yun, 2018; Sallab, Abdou, Perot, y Yogamani, 2017).

3.4.2. Redes neuronales recurrentes

Las Redes Neuronales Recurrentes (RNR) son especialmente aptas para procesar datos de secuencias temporales como lo es el flujo de vídeo. A diferencia de las redes neuronales convencionales, una RNN contiene un bucle de retroalimentación dependiente del tiempo en su célula de memoria. Dada una secuencia de entrada dependiente del tiempo y una secuencia de salida, una RNR puede ser desplegada n veces para generar una arquitectura de red sin bucles que coincida con la longitud de la entrada (Grigorescu et al., 2020).

Una RNR tiene $n+1$ capas idénticas, es decir, cada capa comparte los mismos pesos aprendidos, una vez desplegada puede entrenarse mediante el algoritmo de retropropagación en el tiempo. En comparación con una red neuronal convencional, la única diferencia es que los pesos aprendidos en cada copia desdoblada de la red se promedian, lo que permite que la red comparta los mismos pesos a lo largo del tiempo.

Las redes de memoria a corto plazo (Long Short-Term Memory LSTM) son aproximadores de funciones no lineales para estimar dependencias temporales en datos secuenciales. A diferencia de las redes neuronales recurrentes tradicionales, las LSTM resuelven el problema del gradiente de fuga incorporando tres puertas, que controlan la entrada, la salida y el estado de la memoria (Bresson, Alsayed, Yu, y Glaser, 2017).

El principal problema al utilizar las RNR básicas es el gradiente desvanecido que se

encuentra durante el entrenamiento, la señal del gradiente puede acabar multiplicándose un gran número de veces tantas como el número de pasos de tiempo. Por lo tanto, una RNR tradicional no es adecuada para capturar las dependencias a largo plazo en datos secuenciales (Marina et al., 2019).

Si una red es muy profunda o procesa secuencias largas, el gradiente de la salida de la red tendría dificultades para propagarse hacia atrás y afectar a los pesos de las capas anteriores. En caso de desaparición del gradiente, los pesos de la red no se actualizarán de forma efectiva, terminando con valores de peso pequeños.

3.4.3. Aprendizaje profundo por refuerzo

El concepto de Aprendizaje Profundo por Refuerzo (APR) para la tarea de estimación de trayectoria, utiliza el Proceso de Decisión de Markov Parcialmente Observable (PDMPO). En un POMDP, un agente (vehículo autónomo) percibe el entorno con la observación I^t, realiza una acción a^t en el estado s^t, interactúa con su entorno a través de una recompensa recibida R^{t+1} y cambia al siguiente estado s^{t+1} siguiendo una función de transición Ts^{t+1}.

En la conducción autónoma basada en aprendizaje por refuerzo, la tarea consiste en aprender una política de conducción óptima para navegar desde el estado inicial a un estado de destino, dada una observación del entorno I^t en el tiempo t y el estado del sistema s^t, donde k es el número de pasos de tiempo necesarios para alcanzar el estado de destino s^{t+k}.

El objetivo en APR es encontrar la política de trayectoria deseada que maximice la recompensa futura acumulada asociada, se define la función óptima acción-valor Q^* que estima la máxima recompensa futura cuando se inicia en el estado s^t y se realizan acciones $[a^t, ..., a^{t+k}]$ dado la ecuación 3.2.

$$Q^*(s,a) = max_\pi E[R^t | s^t = s, a^{\hat{t}} = a, \pi] \tag{3.2}$$

El parámetro π es una referencia de acción, vista como una función de densidad de probabilidad sobre un conjunto de posibles acciones que pueden tener lugar en un estado determinado. La función de valor de acción óptima Q^* asigna un estado determinado a la política de acción óptima del agente en cualquier estado (Grigorescu et al., 2020).

Sin embargo, el método estándar de aprendizaje por refuerzo descrito anteriormente no es factible en espacios de estado de alta dimensión. En las aplicaciones de conducción autónoma, el espacio de observación se compone principalmente de información sensorial compuesta por imágenes, radares e información de sensores lidar.

APR ha sido desarrollado para operar en espacios de acción discreta, en el caso de estima-

ción de movimiento, las acciones discretas se traducirían en órdenes discretas, como girar izquierda, girar a la derecha, acelerar o frenar.

En la tabla 3.3 se muestra una descripción de los modelos encontrados en la literatura referentes al aprendizaje.

Tabla 3.3. Descripción comparativa de distintos modelos de aprendizaje.

Referencia	Arquitectura red neuronal	Sensor de entrada	Tarea a resolver	Descripción
Bojarski et al. 2017	Nvidia PilotNet CNN	Imágenes duras de cámara	Manejo autónomo en situaciones de tráfico real	El sistema aprende automáticamente representaciones internas de los pasos de procesamiento necesarios, como la detección de características útiles de la carretera con el apoyo de dirección humana como señal de entrenamiento.
Xu et al. 2017	FastRC-LSTM	Datos de video a gran escala	Predicción de ego-movimiento	Se obtiene un modelo genérico de movimiento del vehículo a partir de datos de vídeo a gran escala procedentes de la multitud, al tiempo que se desarrolla una arquitectura entrenable de extremo a extremo (FCN-LSTM) para predecir una distribución de datos de movimiento del ego del vehículo en el futuro.
Eraqui et al. 2018	C-LSTM	Imágenes de la cámara ángulo del volante	Control del ángulo de dirección	C-LSTM es entrenable de principio a fin, aprendiendo tanto las dependencias temporales dinámicas de la conducción. El problema de regresión del ángulo de dirección se considera la clasificación mientras se impone una relación espacial entre las neuronas de la capa de salida.
Hecker et al. 2018	RNC+LSTM	Cámaras de visión envolvente, lector de bus CAN	Control del ángulo de dirección y velocidad	La configuración de los sensores proporciona datos para una visión de 360 grados de la zona que rodea al vehículo. Se recoge un nuevo conjunto de datos de conducción que abarca diversos escenarios. Se desarrolla un nuevo modelo de conducción integrando las cámaras de visión envolvente con el planificador de rutas.
Rausch et al. 2017	RNC+FC	Imágenes de cámara	Control del ángulo de dirección	La red neuronal entrenada asigna directamente los datos de los píxeles de una cámara frontal a las órdenes de dirección y no requiere ningún otro sensor. Se compara el rendimiento del controlador con el comportamiento de la dirección de un conductor humano.
Bechtel et al. 2018	RNC	Imágenes de cámara	Control del ángulo de dirección	DeepPicar es una réplica en miniatura de un coche real de autoconducción llamado DAVE-2 de NVIDIA. Utiliza la misma arquitectura de red y puede conducirse en tiempo real utilizando una cámara web y una Raspberry Pi 3.
Sallab et al. 2017	RNR+RNC	Imágenes de simulador	Mantener el carril y evitar obstáculos	Incorpora redes neuronales recurrentes para la integración de la información lo que permite al coche manejar escenarios parcialmente observables. También reduce la complejidad computacional para su despliegue en hardware integrado.
Pan et al. 2018	RNC Agile-Driving	Imágenes duras de cámara	Ángulo de dirección y control de velocidad para una conducción agresiva	La RNC denominada aprendiz, se entrena con ejemplos de trayectorias óptimas proporcionados en el momento del entrenamiento por un controlador de modelo predictivo (MPC). El MPC actúa como un experto, codificando la dinámica de la escena en las capas de la red neuronal.
Jaritz et al. 2018	RNC+LSTM encoder	WRC6 Racing Game	Manejo en un video juego	Se utiliza un marco ActorCritic asíncrono (A3C) para aprender el control del auto en un juego de rally física y gráficamente realista, con los agentes evolucionando simultáneamente en diferentes pistas.

Los paradigmas de aprendizaje profundo más representativos para la planificación de trayectorias son: el Aprendizaje por Imitación (AI) y la planificación basada en el Aprendizaje Profundo por Refuerzo (L. Sun, Peng, Zhan, y Tomizuka, 2018; Pan et al., 2017; Jaritz, De Charette, Toromanoff, Perot, y Nashashibi, 2018).

El objetivo del aprendizaje por imitación es aprender el comportamiento de un conductor humano a partir de experiencias de conducción grabadas, dicha estrategia implica un proceso

de enseñanza del sistema a partir de la demostración humana, por lo que distintos autores emplean las RNC para aprender la planificación a partir de la imitación (Schwarting et al., 2018).

El control de aprendizaje se define como un mapeo directo de los datos sensoriales a los comandos de control. Las entradas suelen proceder de un espacio de características de alta dimensión (por ejemplo, imágenes o nubes de puntos).

En los últimos años, los avances tecnológicos en el hardware computacional han facilitado el uso de modelos de aprendizaje. El algoritmo de retropropagación para la estimación del gradiente en las redes profundas se implementa ahora de forma eficiente en unidades de procesamiento gráfico (GPU). Este tipo de procesamiento permite el entrenamiento de arquitecturas de redes grandes y complejas, que a su vez requieren grandes cantidades de muestras de entrenamiento.

3.5. Comparativa de métodos

En la tabla 3.4 se presentan algunos trabajos de referencia en el área, se informan los métodos implementados para resolver el problema de detección y seguimiento de objetos, así como los conjuntos de pruebas, el software - hardware utilizado y los resultados obtenidos, tales como: tasa de detección (detection rate DR), precisión promedio (mean average precision mAP), cuadros por segundo y precisión de seguimiento de múltiples objetos (MOTA).

Además la tabla 3.4 también muestra los parámetros importantes de cada técnica, se puede observar que la tasa de detección en algunos casos esta alrededor del 80 % sin embargo, el número total de obstáculos detectados (en cada técnica) debe tomarse en consideración y la clase o tipo (peatones, vehículo, animales, cualquier objeto en el escenario), ya que en muchos casos sólo detectan un solo tipo de objeto o clases limitadas.

Con respecto al seguimiento de obstáculos, el parámetro MOTA está por debajo del 90 %, ya que el error, con respecto a la posición o movimiento en el video, incrementa debido a la complejidad y la cantidad de datos a procesar, la velocidad de procesamiento del algoritmo, la velocidad de desplazamiento y el número de objetos presentes en la escena, entre otros.

3.6. Pertinencia de los datos

Para finalizar cabe mencionar que los trabajos desarrollados en el ambito industrial estan muy avanzados y su propuesta metodológica implica inversión de recursos y adquisición de un gran cúmulo de información. En este sentido tanto Tesla como Waymo están intentando

Tabla 3.4. Comparativa de resultados cuantitativos así como herramientas utilizadas en algunos trabajos relacionados con la detección y seguimiento de objetos.

Referencia	Método	Técnicas (detección / seguimiento)	Resultados cuantitativos	Conjunto de prueba	Software Hardware
(Roggeman et al., 2017b)	Mapas estéreo	-Mapas de disparidad -Comparación estéreo -Patrones locales binarios (LBP) -Histograma de gradiente	-MOTP = 82.5 % -MOTA = 79.5 -cps = 14	-KITTI Benchmark (Vision, 2019) -HCI EISAT	-CPU xeon 2.67 GHz -Tesla K40 GPU
(W. Song y Yang, 2018)		- Campo aleatorio condicional -Modelo de trayectoria	-MOTA = 75.9 % -mAP = 69.2 % -MOTP = 88 %	-Entrenamiento: 490 cuadros 1,578 peatones etiquetados resolución 640 x 480 pixeles	-nVidia GeForce 8800
(Scheidegger, 2018)	CNN	-K-medias -Clasificadores -Patrones locales de confianza	(Condiciones óptimas de iluminación) -MOTP = 77 % -Tasa de alarma de fallo (FA) FA = 2.25 % (Condiciones de iluminación variable) -DR = 50.8 %, FA = 1 % -mPA = 63.4 % , fps = 30	-KITTI Benchmark -Daimler dataset: 15,560 muestras de peatones -Imágenes con autos peatones y ciclistas etiquetados	-Intel core7 -Tesla V100 GPU
(Sualeh,2019)		-Clasificación de objetos -Box Fitting	-Regiones verdaderas de 95 % -Riesgo de colisión estimada mayor que 50 % - Distancia estimada de 2.6 m. a una velocidad de 50 km / hr	-50 rutas vehiculares -Se dividen en tres clases diferentes. Inicio: consistente en 20 rutas de aprox. 1 km Detenido: consistente en 50 escenarios, 10 km En marcha: consistente en 30 km de muestreo	-Multilidar -VLP-16 Velodyne -IONIQ Platform
(K. Lu, An, Li, y He, 2017)	Aprendizaje profundo	-Capas progresivas convolucionales -Etapa de decodificación	-DR =81.5 a 90.6 % dependiendo de la clase -mAP = 78.6 % -Runtime 110 ms en promedio	-PASCAL VOC 2007 -40 mil iteraciones -COCO benchmark	-GPU TitanX -Caffe
(Sangineto y Nabi, 2019)		-Detectores deconvolucionales	-MOTA = 71.8 % -fps = 20	-DETRAC 60 secuencias de entrenamiento. -40 secuencias de prueba.	-AlexNet -Caffe -Titan Black GPU -iiRAV
(Nguyen y Nguyen, 2013)	Algoritmos evolutivos	-Algoritmos genéticos -Función fitness -Generación de hipótesis	-Valor fitness = 0.1216 a 0.4315 -Tiempo de procesamiento = 48 ms -DR = 86.08 %	-63025 imágenes capturadas en un conjunto controlado y otro real a una vel. de 20 a 70 km / hr	-FPGA xilinx

recopilar y procesar suficientes datos para crear un vehículo que capaz de predecir con mayor eficacia las rutas de desplazamiento de los participantes del tráfico. además de abordar la problemática de maneras distintas.

Tesla está aprovechando los cientos de miles de coches que tiene en la carretera recopilando datos del mundo real sobre cómo funcionan esos vehículos (y cómo podrían funcionar) con Autopilot, su actual sistema semiautónomo. Waymo, que comenzó como el proyecto de coches de autoconducción de Google, utiliza potentes simulaciones por computadora y alimenta lo que aprende de ellas en una flota más pequeña del mundo real.

Es difícil establecer con exactitud cuántos kilómetros de datos ha obtenido Tesla pues no es dominio público esa información, sin embargo, en 2016 la directiva de Tesla comentó que se habían registrado 780 millones de millas de datos, según IEEE Spectrum citando al dueño de Tesla, se mencionó que se estaban recopilando alrededor de 3 millones de millas de datos por día. A medida que Tesla vende más autos, la cantidad de datos que se pueden recoger aumenta exponencialmente (Leon y Gavrilescu, 2021).

Los vehículos de Tesla pueden registrar los casos en los que el software de Autopilot habría realizado una acción, y esos datos acaban cargándose en Tesla. Este modo de recolección de datos significa que Tesla podría estar simulando datos completos a través de muchos de los

miles de millones de kilómetros que se conducen.

La única otra empresa que trabaja con cantidades similares de datos es Waymo, que mencionó que ha simulado 8,000 millones de kilómetros de conducción autónoma. La compañía también dijo que ha anotado 5 millones de millas de conducción autónoma en carreteras públicas. Eso es más que prácticamente todas las demás empresas que prueban vehículos de autoconducción juntas (Schwarting et al., 2018).

Cabe resaltar que las técnicas que mayor necesidad tienen de datos son los modelos con base en sistemas de aprendizaje. Por lo tanto lo robusto del modelo conlleva a la necesidad de información de aprendizaje suficiente para tener un espectro amplio de todas las circunstancias que se pueden presentar en el mundo real. De allí la pertinencia al momento de seleccionar una técnica con perspectivas altas de desempeño pero con necesidad de una gran cantidad de datos entrenamiento, como claro ejemplo se pueden señalar a los modelos de aprendizaje profundo mencionados en la sección 3.4.

3.7. Discusión

Los métodos propuestos en la literatura tiene aspectos positivos, pero también factores limitantes. A continuación se mencionan algunos puntos y áreas de oportunidad encontradas en trabajos relacionados.

Con respecto a los modelos de mapeo ROI se tiene que:

- Sólo se toman en consideración dos tipos de obstáculos: vehículos y peatones (Schwarting et al., 2018; Mane y Vhanale, 2016; Roggeman et al., 2017b; W. Song y Yang, 2018; Y. Xu, Zhao, Baker, Zhao, y Wu, 2019; Schulz, Hubmann, Löchner, y Burschka, 2018).
- Alta tasa de tiempo de procesamiento y baja tasa de detección de obstáculos (Hoermann, Stumper, y Dietmayer, 2017b; Suraj, Grimmett, Platinskỳ, y Ondrúŝka, 2018).
- Mapeo monocular de ROI, en algunos casos adquisición de datos mediante cámaras estáticas en ambientes controlados o poca variabilidad (Messaoud, Deo, Trivedi, y Nashashibi, 2020; Fernando, Denman, Sridharan, y Fookes, 2018).
- El problema de la predicción de la trayectoria tiene áreas de oportunidad debido a fallas de detección, apariencias similares entre objetos, oclusiones, variaciones de iluminación y distintos puntos de visualización (Mane y Vhanale, 2016; W. Song y Yang, 2018; Zou et al., 2020).

Los modelos de probabilidad tienen las siguientes características: los algoritmos de filtros Kalman y los filtros de partículas utilizados para rastrear obstáculos tienen inconvenientes debido a los problemas de deriva causados por cambios en la apariencia (Schulz et al., 2018).

Los métodos de estimación de trayectoria con alta incertidumbre tienen problemas de complejidad (con muchas muestras) o baja precisión (con un número insuficiente de muestras) (W.-C. Lai et al., 2020; Schulz et al., 2018; Gupta, Johnson, Fei-Fei, Savarese, y Alahi, 2018).

Para el caso de los modelos de aprendizaje profundo consideran sólo características como el color o profundidad, lo que implica una limitación para obtener un mayor nivel de abstracción de las características representativas de los obstáculos (Mangalam et al., 2020; Sangineto y Nabi, 2019; Sangineto, Nabi, Culibrk, y Sebe, 2019; J.-G. Wang et al., 2016).

Los métodos de seguimiento basados en el aprendizaje por medio de clasificadores en línea sufren el problema de la acumulación de errores durante el proceso de auto-aprendizaje (Sangineto et al., 2019; J.-G. Wang et al., 2016; Redmon et al., 2016).

Algunos algoritmos basados en redes neuronales convolucionales se aproximan a captar las invarianzas de la traslación y no capturan la invariancia rotacional o la rotación fuera de plano, lo que las hace susceptibles de error al clasificar e identificar obstáculos (Gidaris y Komodakis, 2015; Mandal, Biswas, Balas, Shaw, y Ghosh, 2020; Redmon et al., 2016; D. Lee, Gu, Hoang, y Marchetti-Bowick, 2019).

Los modelos basados en agentes inteligentes tienden a fallar en la clasificación de los objetos debido a la similitud de las formas, por lo que confunden diferentes tipos de obstáculos, lo que implica una detección errónea del tipo de obstáculo presente en la escena, por ejemplo, detectar un automóvil en escena cuando en realidad no está (J. Li, Yang, Tomizuka, y Choi, 2020; C. Sun et al., 2019).

De los trabajos relacionados se obtienen las etapas generales (tabla 3.5) que debe seguir un algoritmo para resolver el problema de detección, clasificación, seguimiento de obstáculos y predicción de la trayectoria.

A partir de estas etapas en la revisión de los trabajos relacionados, se puede obtener una aproximación (porcentaje) de las fortalezas del trabajo realizado hasta el momento así como el trabajo que falta por realizar. En la tabla 3.5 se puede observar la distribución de las investigaciones realizadas y los puntos (etapas) que tienen menor progreso o desarrollo.

La figura 3.2 muestra una gráfica de la distribución porcentual de las etapas abordadas por los investigadores, donde los bloques que tienen el porcentaje más bajo son las áreas de oportunidad de investigación.

Con base en la información recopilada a través del análisis de los diferentes trabajos que

Tabla 3.5. Etapas del problema abordadas por diversos autores.

#	Etapa		Artículo de referencia																		
			Bochinski et al.,2017	Mandal et al., 2020	Hu et al., 2020	Braga et al.,2016	Tedrake et al.,2015	Prabhakar et al.,2017	Hu et al.,2020	Litman, 2017	Wen et al.,2019	Li et al., 2020	Naiel et al.,2017	Roggeman et al.,2017	Cao et al.,2020	Sangineto y Nabi, 2019	Gupta et al.,2018	Jaritz et al., 2018	Schulz et al.,2019	Xu et al.,2017	Sun et al.,2019
I	Selección ROI			✓	✓	✓		✓	✓		✓	✓		✓	✓			✓		✓	✓
II	Estimación de la posición espacial en la escena		✓	✓	✓		✓		✓		✓			✓	✓	✓	✓		✓	✓	
III	Detección de objetos			✓	✓	✓	✓	✓	✓				✓		✓	✓	✓		✓		✓
IV	Extracción de características	Entrenamiento								✓			✓	✓			✓			✓	
		Mapeo del vecindario			✓		✓	✓			✓	✓			✓						
V	Clasificación	Calse específica											✓		✓		✓			✓	
		Obstáculo No obstáculo	✓	✓	✓			✓	✓	✓	✓	✓		✓					✓		
VI	Ranqueo (nivel de peligro)				✓		✓	✓	✓		✓			✓			✓				✓
VII	Seguimiento	Múltiple obstáculo				✓									✓				✓		
		Obstáculo único	✓				✓				✓		✓	✓							✓
VIII	Estimación de trayectoria				✓	✓					✓		✓				✓		✓		✓

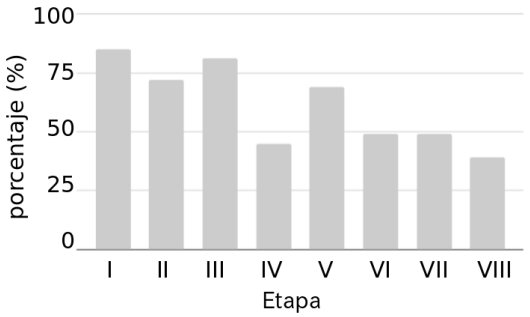


Figura 3.2. Porcentaje de etapas del problema abordadas por diferentes autores.

abordan el problema de interés, se puede decir que las tareas pendientes por resolver son el seguimiento y la estimación de la trayectoria.

La revisión de la literatura indica que la etapa de detección de obstáculos es la que presenta mayor avance y desarrollo, las tasas de detección reportadas son cercanas al 90 %, sin embargo, existe un tarea pendiente ya que debe clasificar más de un par de clases (peatones y vehículos), esto debido al hecho de que se pueden encontrar obstáculos peculiares en las carreteras.

En la figura 3.3 se pueden observar los obstáculos que son difíciles de identificar por los modelos actuales de reconocimiento / detección de obstáculos, pero estos son comunes en las calles de algunos países (por ejemplo en la región de América Latina).

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

Figura 3.3. Obstáculos presentes en carreteras, a) señal de tráfico en alcantarilla, b) poste de luz en medio de la calle, c) árbol caído que bloquea la calle, d) neumáticos en alcantarilla.

En los últimos años se han logrado avances significativos en el seguimiento de objetos, se han desarrollado varios algoritmos robustos que pueden rastrear objetos en tiempo real en escenarios simples. Sin embargo, está claro que las conjeturas utilizadas para hacer que el problema de seguimiento sea manejable (por ejemplo, la suavidad del movimiento, la mínima cantidad de oclusión, la constancia de la iluminación, el alto contraste con respecto al fondo, etc.), infringe muchos escenarios realistas y por lo tanto, limita la utilidad de los seguidores (trackers) en aplicaciones de navegación de vehículos. Por lo tanto, el seguimiento y los problemas asociados de selección de características, representación de objetos, forma dinámica y estimación de movimiento son áreas activas de investigación y continuamente se proponen nuevas soluciones.

3.8. Conclusiones del análisis de la literatura

El enfoque común, de las distintas técnicas analizadas, consiste en calcular el movimiento futuro del objeto propagando su estado a lo largo del tiempo basándose en suposiciones del sistema físico subyacente y utilizando técnicas como filtros de Kalman, RDB o modelos de aprendizaje. Si bien este enfoque funciona bien para las predicciones a corto plazo, su rendimiento se deteriora para horizontes más largos, ya que el modelo ignora el contexto circundante, por ejemplo, carreteras, multiobjetos, normas de tráfico entre otros. (Cui et al., 2019).

Para resolver este problema, es necesario además utilizar información cartográfica como restricción para calcular la posición futura del vehículo a largo plazo. Por lo que el algoritmo o modelo propuesto debe asociar en primera instancia cada vehículo detectado con uno o más de carriles en un mapa del entorno. A continuación, se deberían generar todas las trayectorias posibles para cada par (objeto - carril asociado) basándose en la topología del mapa, la conectividad del carril y la estimación del estado actual del vehículo. Este método de predicción proporciona aproximaciones razonables en los casos más comunes, pero es

sensible a los errores en la asociación de vehículos y carriles.

Por lo que para tener un sistema con alto rendimiento en la detección de objetos y la estimación de sus trayectorias a largo plazo es necesario conjuntar alto desempeño y fiabilidad para obtener un método robusto cuya reacción sea similar en velocidad y respuesta a la lograda por un un conductor humano.

Como alternativa a los enfoques existentes abordados para resolver la problemática planteada, el método propuesto en este trabajo de tesis implicó realizar la implementación de un algoritmo que conjunta múltiples técnicas computacionales para obtener un desempeño robusto y capacidad de generalización para su aplicación en entornos vehiculares con características dinámicas variables.

La propuesta de este trabajo de investigación realiza aportaciones en cuanto a la detección de múltiples objetos y la predicción de su cambio de dirección. Con respecto a la detección y seguimiento de objetos se implementó la fusión de información procesada y obtenida en primer lugar por medio de una red neuronal convolucional que identifica la posición espacial de los objetos de interés. Dicha posición espacial se proporciona a una etapa del algoritmo que procesa mapas de disparidades de la escena para determinar la distancia aproximada a la que se encuentra el objeto detectado con respecto al sensor (cámaras) que adquiere la imagen del ambiente.

Además el método propuesto tambien también calcula la velocidad de desplazamiento aproximada de múltiples objetos tomando como referencia la velocidad de desplazamiento del vehículo sobre el cual se montan el sensor (GPS) así como el posicionamiento espacial de los objetos proporcionado por los datos de las imágenes capturadas en el recorrido.

Finalmente, los cambios de dirección de los objetos se obtienen según los bloques de detección y el suministro de información relevante a la etapa de estimación (RDB) respecto a la distribución de probabilidad en función del riesgo de colisión.

Capítulo 4

Método Propuesto

En la sección del análisis de la literatura se han presentado algunos escenarios de estimación de trayectoria, los métodos previamente revisados pueden resolver esta tarea a través de la percepción, planificación de trayectorias y control de movimiento en simuladores o en ambientes con situaciones u obstáculos específicos del mundo real, sin embargo los escenarios restantes son casos en los que los métodos analizados no presentan información o no se consideran por tener un mayor grado de dificultad.

En este sentido, el trabajo propuesto en esta investigación realiza aportaciones en cuanto a la detección de múltiples OI y la predicción de su cambio de dirección en ambientes del mundo real así como una comparativa de desempeño vs trabajos relacionados. Los cambios de dirección de los objetos en esta investigación se obtienen por medio de una RDB con respecto a la distribución de probabilidad en función del riesgo de colisión. En lo que resta de la sección se describe a detalle la metodología implementada.

Para este trabajo, los obstáculos en una escena de tráfico consisten en un conjunto de objetos que participan en el tráfico en un espacio variable en el tiempo y que representan un riesgo de colisión para el vehículo que captura el vídeo (ego-vehículo). Los objetos a detectar pueden ser coches, peatones, ciclistas, animales entre otros.

El enfoque propuesto estima la trayectoria que debe seguir un objeto detectado, el flujo y el cálculo de la información se realiza a través de los módulos representados en la figura 4.1.

En primer lugar, se captura la información de la carretera con un par de cámaras de vídeo (para emular la visión estereoscópica) para determinar las ROI.

El módulo de detección de ROI consta de dos submódulos que cooperan entre sí: una red neuronal convolucional multicapa (submódulo RNC) que señala los objetos detectados en el frame en un cuadro delimitador y el submódulo que procesa mapas de disparidad (MD) de la escena para estimar la distancia aproximada de los objetos con respecto al ego-vehículo

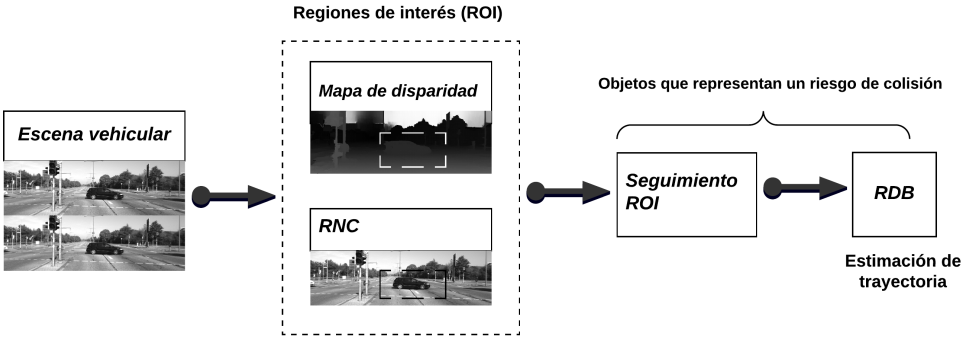


Figura 4.1. Etapas de procesamiento del algoritmo propuesto para la estimación de trayectorias.

(información de la imagen izquierda-derecha correspondiente).

Estas ROI son rastreadas con respecto a los datos consecutivos que se adquieren en el tiempo, lo permite obtener características físicas sobre el movimiento de los objetos e información relacionada. Por último, la información obtenida, de los módulos previos, se proporciona a un modelo probabilístico RDB que, dado un conjunto de datos de obstáculos detectados y observados, estima las posibles trayectorias de cada uno de estos y su correspondiente porcentaje de riesgo de colisión.

4.1. Detección ROI

El submódulo RNC considerado en este trabajo se basa en la referencia de aprendizaje profundo Yolo-V5 y la arquitectura de RNC estándar, esta red predice 4 coordenadas para cada caja delimitadora (Redmon et al., 2016).

La entrada del submódulo RNC es la imagen original de la escena del tráfico para cada frame, con respecto a la arquitectura de la red se tiene el mapa de características F_m de la *m-ésima* capa convolucional para procesar, la información del video, mediante la función:

$$F_m = f(F_{m-1} \otimes W_m + b_m)(1 \leq m \leq 5) \tag{4.1}$$

donde W_m representa el vector de pesos de la *m-ésima* capa, b_m es el vector de desplazamiento de la *m-ésima* capa, $\otimes$ significa la convolución relacionada con el mapa de características de la $m-1$ capa y el mapa de características F_m de la capa m es obtenida por la función f (en el capítulo 2 se mencionó a detalle la información de este submódulo).

El proceso del submódulo-RNC finaliza con la detección de todas los obstáculos y la obtención de las coordenadas bidimensionales de la caja delimitadora. Posteriormente, con las coordenadas de la caja delimitadora, se obtienen los cambios de orientación y traslación, además de los cambios en las dimensiones de altura y anchura del objeto. El siguiente paso consiste en proporcionar esta información al submódulo MD.

El cálculo de la distancia del obstáculo con respecto al ego-vehículo se lleva a cabo en el submódulo MD, un dispositivo estéreo (dos cámaras ligeramente desplazadas horizontalmente montadas en el ego-vehículo) se utiliza para proporcionar la información para calcular el mapa de disparidad.

El marco de referencia se muestra en la figura 4.2, la disparidad $disp$ es procesada dada la imagen izquierda (F_l) con respecto a la imagen derecha (F_r), esta información es captada por un par de cámaras con la misma distancia focal (Φ).

Los valores de los puntos φ_l y φ_r se triangulan dada la distancia focal en ambas cámaras y la distancia entre ellas (B). El valor negativo obtenido φ_r se debe a que pasa por el plano de la imagen con respecto a la línea central de la cámara (flecha sólida con respecto a Fr). Por otra parte, φ_l es un número positivo ya que el plano central de la cámara de la izquierda (flecha sólida con respecto a Fl) se toma como referencia (C. Xu et al., 2019).

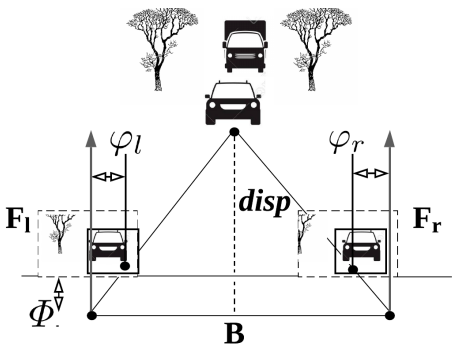


Figura 4.2. Configuración estereoscópica de dos cámaras con distancia focal y centros alineados para el cálculo del mapa de disparidad, basado en (Xu et al., 2019).

La formula 4.2 se usa para estimar la disparidad con respecto a las imágenes de referencia por medio de todos los valores de los frames izquierdo y derecho.

$$disp = \frac{\Phi B}{\varphi_r - \varphi_l} \tag{4.2}$$

En la siguiente subsección se establecen las fórmulas complementarias para obtener la distancia aproximada con la ayuda de los valores de disparidad.

4.1.1. Aproximación de la distancia y velocidad

La descripción de la ubicación del ego-vehículo con respecto del tiempo sirve para determinar la velocidad aproximada de desplazamiento de los obstáculos detectados.

Por lo que es importante obtener la información de la posición de los OI, la información requerida se obtiene a partir de la diferencia de posición dada en coordenadas cartesianas obtenidas por medio del dispositivo GPS (instalado en el ego-vehículo) y el resultado del procesamiento de los MD con respecto a los frames capturados por las cámaras de video en intervalos de tiempo definidos.

La relación del vector resultante con la información obtenida de la estimación de la velocidad aproximada del ego-vehículo sirve como punto de referencia para aproximar las velocidades relativas de los objetos detectados en escena.

Se calcula la relación de cambio de posición para todos los pixeles de una región determinada (disparidades de la ROI) de la imagen para aproximar los conjuntos de pixeles que pertenecen a una misma región utilizando polinomios cuadráticos en dos frames consecutivos, la información se expresa en un sistema de coordenadas locales $\xi = disp(x_t^i, y_t^i)$ planteado a través de la función 4.3.

$$f(\xi) \sim \xi^T A\xi + b^T \xi + c \tag{4.3}$$

Donde A es una matriz simétrica a obtener con coeficientes relacionados al cambio de movimiento, b es un vector de coeficientes relativos a las posiciones espaciales previas y c un escalar de ajuste con respecto al error inherente debido a la aproximación de la dinámica del objeto seguido. Los coeficientes se aproximan por el criterio de mínimos cuadrados ponderados para todos los pixeles dentro de un entorno de vecindad.

La ponderación consta de la certeza que está asociada a los valores de los pixeles en la vecindad, además, fuera de la imagen la certeza es cero, de esta forma estos pixeles no tienen impacto en el coeficiente de estimación del flujo.

Otro elemento de la ponderación es la aplicabilidad, la cual determina el peso relativo de los pixeles en la vecindad basada en suposición dentro de la misma, de esta forma se consigue dar más peso al pixel central y hacer que disminuya radialmente desde dicho pixel a medida que se aleja del central.

Además,el tamaño de la ventana de aplicabilidad determina la escala de las estructuras que se quieran capturar por el coeficiente de expansión.

Suponiendo una vecindad que se ajuste a la ecuación 4.4 se puede calcular un desplazamiento global d obteniendo la nueva señal f_2 tal que:

$$f_2(\xi) = f(\xi - d)^T A(\xi - d)^T + b^T(\xi - d) + c$$

$$f_2(\xi) = \xi^T A\xi + (b - 2Ad)^T \xi + d^T Ad - b^T d + c$$

$$f_2(\xi) = \xi^T A_2 \xi + b_2^T \xi + c_2 \tag{4.4}$$

igualando los coeficientes con f_ξ se obtiene:

$$A_2 = A$$

$$b_2 = b - 2Ad$$

$$c_2 = d^T Ad - b^T d + c \tag{4.5}$$

Así, con la ecuación 4.5 se puede resolver el desplazamiento d como se sigue:

$$2Ad = -(b_2 - b)$$

$$d = -\frac{1}{2} A^{-1}(b_2 - b_1) \tag{4.6}$$

De esta forma se puede obtener un vector d para cada píxel con la estimación de la dirección y magnitud del movimiento en dicho pixel.

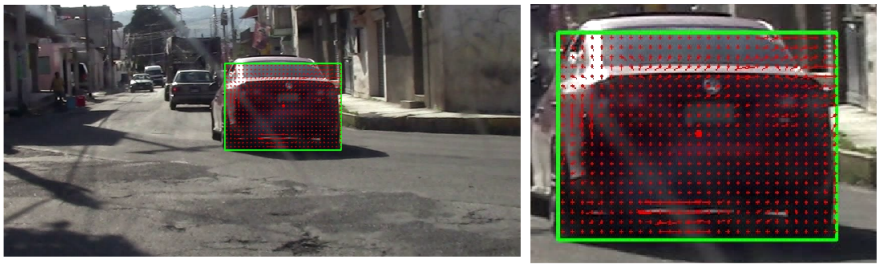


Figura 4.3. Análisis del vecindario del centroide con respecto al objeto detectado, la salida corresponde a los vectores resultantes.

Dado el concepto anterior, se aproximan vectores de dirección para una porción de la superficie del objeto, el número de vectores obtenidos, como lo remarca el análisis planteado depende del entorno de vecindad a analizar. Sea la región de interés marcada en la figura 4.3 (el recuadro verde), correspondiente a la marcación del objeto detectado.

Se crea una malla para dividir la información a analizar en la región de interés. Cada contenedor (bin) tiene una dimensión cuadrada de tamaño (en pixeles) seleccionable, los

vectores obtenidos en cada contenedor se muestran en la figura 4.3.

Con los vectores resultantes parciales se puede realizar una suma vectorial para obtener el vector resultante total y de esta forma el desplazamiento en frames consecutivos.

También se pueden observar los cambios de dirección en frames consecutivos y con esa información construir un vector resultante (velocidad de desplazamiento) con un ángulo de orientación.

La información obtenida del vector resultante se apoya de datos obtenidos a la par con el dispositivo GPS (velocidad absoluta), con lo cual se determina la velocidad relativa a través de la razón de cambio de las coordenadas de georeferencia en el tiempo.

La relación entre la velocidad absoluta ($\dot{x}_A$)y la velocidad de los objetos en escena ($\dot{x}_0$) corresponden a la velocidad relativa (ecuación 4.7).

$$\dot{x}_{0A} = \dot{x}_0 - \dot{x}_A \tag{4.7}$$

El movimiento realizado con respecto a los vectores de posición (d_0) en un tiempo de muestreo (Δt) proporciona la relación de la velocidad media (ecuación 4.10).

$$\dot{x}_0 = \frac{\Delta d_0}{\Delta t} \tag{4.8}$$

$$d_{0A} = d_0 - d_A \tag{4.9}$$

$$\dot{x}_{0A} = \frac{\Delta d_{0A}}{\Delta t} = \frac{\Delta d_0}{\Delta t} - \frac{\Delta d_A}{\Delta t} \tag{4.10}$$

Dadas las ecuaciones 4.6 y 4.10 la posición de los obstáculos detectados se consideran dentro de un rango observable de 20 metros (m) en la dirección lateral (con referencia central del frame esto se expresaría como -10 a +10 m) y de 1 a 45 m aproximadamente en la dirección longitudinal. La velocidad media detectable se encuentra en el intervalo de 0-60 km/hr aproximadamente. Estos rangos se determinan teniendo en cuenta el rango de detección válido de los sensores y el entorno de la carretera donde se capturan los datos.

La aproximación de la velocidad de desplazamiento planteada hasta este punto corresponde a la información obtenida mediante la captura de información de trayectos vehiculares de entornos reales. Sin embargo dado que se realizaron pruebas experimentales en simulador las particularidades de los datos del entorno virtual requieren una caracterización previas a la entrada al bloque de estimación de trayectoria.

En este sentido, en lo que respecta al entorno de simulación *city car driving* (CityCar, 2022) al no proporcionar imágenes estereoscópicas se realiza el cálculo de la distancia de los objetos mediante una aproximación-interpolación de la altura de los objetos detectados en

perspectivas tomando como referencia mediciones de la altura de objetos en el mundo real.

En la figura 4.4 se pueden observar algunas muestras tomadas del mundo real en relación a la distancia medida en metros contra la altura del peatón o vehículo.

Figura 4.4. Muestras para realizar las mediciones de altura (pixeles) de peatones y vehículos vs la distancia real a la que se encuentran, tanto en ambientes reales como del simulador *city car driving*.

Con relación a la altura observada de 20,000 objetos en escenas del simulador y en función de los datos, altura vs distancia medida, se realiza una aproximación de la distancia respecto a la referencia (punto de vista de la cámara) por medio de un polinomio de quinto orden. De entre los 20,000 datos obtenidos existen valores atípicos los cuales se descartan mediante el rango intercuartil calculado (parámetro = 169 en rango de pixeles).

El ajuste de la curva con respecto a los datos medidos permite aproximar la representación de los datos al asignar una función de ajuste (curva) a lo largo del rango de muestreo. En el mejor de los casos, se capta la tendencia de los datos lo que permite hacer predicciones sobre el comportamiento de la distancia vs la altura de los objetos en escena.

Los resultados de las mediciones del mundo real y la gráfica del polinomio aproximador (ec. 4.11) obtenido se muestra en la figura 4.5.

$$-1.131^{-5}Altura^3 + 8.183Altura^2 - 1.979Altura + 165.26 \tag{4.11}$$

Como información final en lo que respecta a los parámetros del entorno virtual cabe señalar que el rango de la distancia de detección es de 1 m a 45 m, ya que es el umbral de

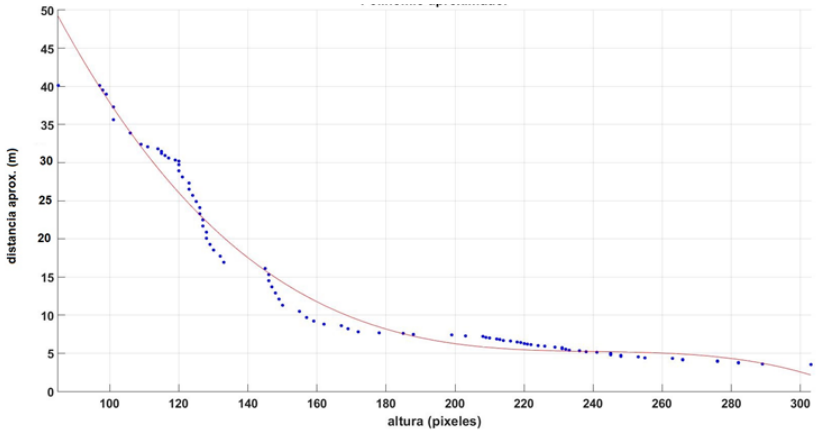


Figura 4.5. Estimación de distancia de separación del objeto detectado con respecto a la relación de la altura medida por medio del polinomio aproximador calculado.

detección correspondiente al submódulo detección de objetos (RNC).

Además, la razón de cambio de posición esta dada por la velocidad de procesamiento del frame de la escena virtual la cual corresponde a 30 fps, con estos datos se puede utilizar como en el caso de entornos reales la aproximación mediante las ecuaciones 4.6 y 4.10.

4.1.2. Seguimiento de las ROI

A lo largo de este análisis se modela el ambiente con objetos que se mueven en un plano horizontal. La dimensionalidad de cada objeto en el plano corresponde a tres dimensiones, dos de las cuales sirven para determinar la posición en el plano de referencia además de la orientación a lo largo del eje vertical, que es ortogonal al plano y la última dimensión corresponde a la temporalidad.

Para determinar la posición de los objetos en el plano bidimensional, se establece una relación entre el marco de referencia local del vehículo que captura las imágenes con respecto a los obstáculos presentes en escena.

Cabe recordar que de los módulos (del algoritmo) anteriores, la información sobre el obstáculo se obtiene mediante el procesamiento de los datos procedentes de un par de cámaras de vídeo y de un dispositivo del sistema de posicionamiento global (GPS) montado en el ego-vehículo. Las condiciones de movimiento del obstáculo (velocidad relativa) se obtienen indirectamente con el uso del sensor GPS.

La posición del *i-ésimo* obstáculo detectado en la escena se especifica por sus coordenadas (x^i, y^i) desde el centro del cuadro envolvente. Se debe notar que x y y indican la ubicación con respecto al ego-vehículo en las direcciones longitudinal y lateral respectiva-

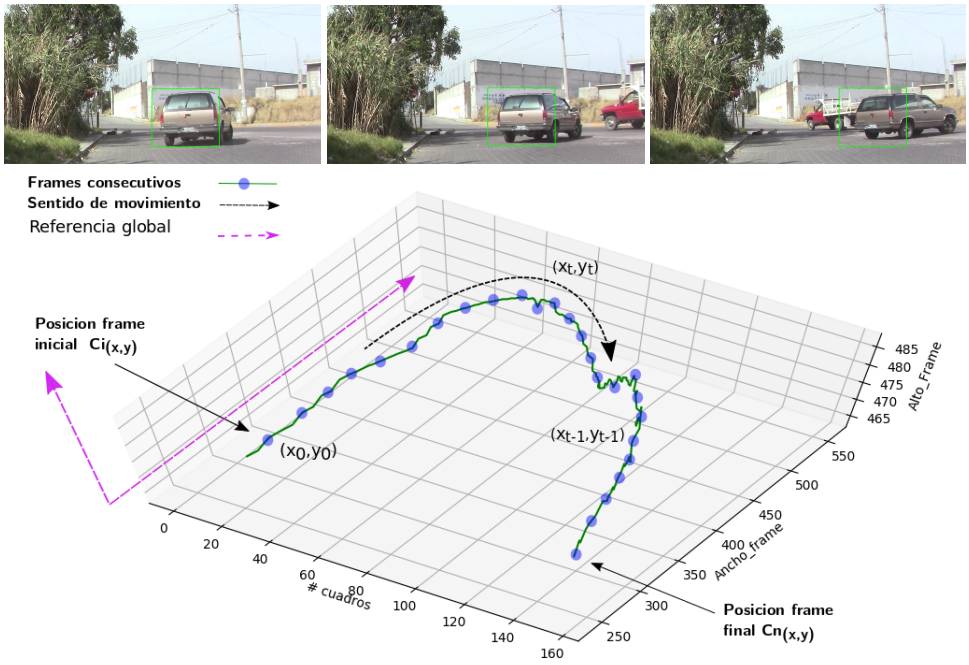


Figura 4.6. Gráfica con la información espacial y sentido del movimiento del objeto detectado con respecto a la referencia global correspondiente a los frames de interés.

mente. El sistema de referencia, coordenadas globales $O = (0,0)$, corresponde a la ubicación del ego-vehículo, la posición de los obstáculos circundantes en el tiempo t se denota por (x_t^i, y_t^i) y la velocidad relativa del *i-ésimo* obstáculo se representa por medio de $(\dot{x}_t^i, \dot{y}_t^i)$.

La trayectoria del *i-ésimo* obstáculo se determina por la secuencia de posiciones $(x_{t-1}^i, y_{t-1}^i), (x_t^i, y_t^i), (x_{t+1}^i, y_{t+1}^i)$ definen el vector de movimiento del cuadro delimitador para calcular la traslación, además, la variación de posición en frames consecutivos proporciona los cambios de orientación (rotación ψ_t^i) con respecto a la referencia global O.

Las coordenadas de posición se obtienen cada Δt segundos; el entorno dinámico de la escena cambia con el tiempo, por lo que los intervalos de cambio pueden representarse como $t_i = t_{i-1} + \Delta t$. Un cambio temporal corresponde al número de frames consecutivos considerados en el intervalo Δt, por ejemplo, puede corresponder a 30 fps o un rango menor. Además, la captura de datos del sensor GPS están sincronizados con la captura en video de la escena. En la figura 4.6 se muestra una representación de la información pertinente para la etapa de seguimiento.

4.1.3. Topología RDB propuesta

La topología RDB propuesta incorpora un análisis previo del estado-posición de los obstáculos para rastrear y estimar sus trayectorias en la escena de tráfico para evitar colisiones. La formulación general del problema consiste en definir una escena de tráfico que incluya varios tipos de objetos participantes con diferentes características de movimiento.

Las variables del espacio de estado discreto se definen para facilitar el análisis del problema, estas variables son: velocidad relativa de desplazamiento ($v_t^j = (\dot{x}_t^i,\ \dot{y}_t^i)$), posición espacial ($p_t^j = (x_t^i, y_t^i)$), distancia con respecto al ego-vehículo ($d_t^j = \frac{\Phi B}{\varphi_r - \varphi_l}$) y rotación o giro ($\psi_t^j$) donde $j = 1, 2, 3, ..., n$ corresponden a los obstáculos detectados en escena.

La información del entorno y la interacción de algunos obstáculos (vehículos, peatones, ciclistas, etc.) se utilizan para diseñar los estados latentes, mientras que las dependencias causales se emplean para diseñar las dependencias condicionales en el tráfico. Los modelos dinámicos implican el diseñar el espacio de estados en función del movimiento y las variables de acción-reacción (C. Wang et al., 2018).

La captura de vídeo considera la existencia de variables latentes discretas Tr^j diseñadas para incluir las intenciones de cada participante en la escena, es decir, la trayectoria del *j-ésimo* obstáculo.

La distribución de probabilidad Tr^j depende de todos los estados observables Tr^1, Tr^2 ,..., Tr^n pero no depende de los estados latentes de otros participantes en la escena. Con esto se obtiene una distribución discreta condicional para el *j-ésimo* obstáculo y también la probabilidad de colisión (c_t^j) dada la ruta seguida (inferencia dada por las variables espaciales).

Dependiendo de la estructura de la red, se puede obtener una distribución de probabilidad conjunta entre dos intervalos de tiempo. La figura 4.7 muestra tanto la estructura de la topología RDB como la interacción en la red entre las variables dadas.

El proceso de inferencia de la RDB se refiere al cálculo de la probabilidad de una determinada intención de maniobra, basándose en la red establecida y en los parámetros aprendidos previamente, se utilizan todos los estados observables en dos cortes de tiempo continuos como evidencia para la inferencia (Cappé y Moulines, 2009).

El resultado de la inferencia es la probabilidad de la intención en el tiempo $t+1$. La intención que tiene la máxima probabilidad posterior se elige como resultado de la predicción, el proceso computacional específico se describe a continuación.

Para cada conjunto de objetos de la escena, las variables de interés son: las velocidades de los objetos en el frame a analizar $V_t = (v_t^1, v_t^2, ..., v_t^j)$, la posición espacial para el grupo de objetos descrito como $\Gamma_t = (p_t^1, p_t^2, ..., p_t^j)$, distancia como $D_t = (d_t^1, d_t^2, ..., d_t^j)$ y orientación dada por $\Psi_t = (\psi_t^1, \psi_t^2, ..., \psi_t^j)$. Finalmente, los trayectos observados $TR_t = (Tr_t^1, Tr_t^2, ..., Tr_t^j)$

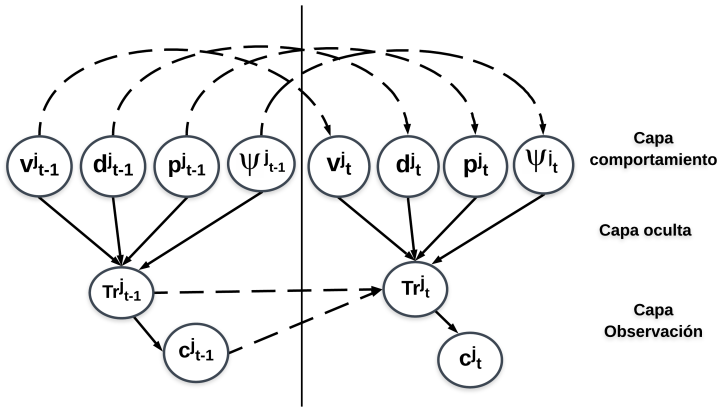


Figura 4.7. El modelo de red bayesiana se desarrolla en dos segmentos de tiempo, donde se observa la composición de las dependencias condicionales de estado latente. Las líneas continuas y discontinuas son las dependencias causales y temporales observables, respectivamente.

y la probabilidad de colisión asociada $C_t = (c_t^1, c_t^2, ..., c_t^j)$.

Además, se considera un conjunto de objetos con parámetros dados $Z_t = [V_t, D_t, \Gamma_t, \Psi_t]$ detectados en la escena y muestreados en el transcurso de los frames analizados a partir de t, bajo condiciones iniciales se puede implicar que la condición del conjunto Z_0 puede ser dado como una aproximación ($\sim$) de la probabilidad de los datos de entrada, es decir, $Z_0 \sim P(V_0, D_0, \Gamma_0, \Psi_0 \mid C_0)$. Entonces, cada cambio de dirección se predice según la probabilidad de transición $Z_{t+1} \sim P(Z_{t+1} \mid Z_t)$.

La RDB se define para propagar las relaciones de dependencia condicional entre las variables de interés y observar su efecto en el intervalo de tiempo a analizar. Por lo tanto, dado el intervalo $t = 1, 2, 3, ..., T$, con respecto a la topología propuesta y las variables de la capa de comportamiento, la distribución de probabilidad conjunta se expresa por medio de la ecuación 4.12.

$$P(TR_t \mid Z_{t+1}) = \prod_{t=1}^{T} Z_{t-1} \times \prod_{t=1}^{T} P(Z_t \mid P(Z_{t+1})) \tag{4.12}$$

La probabilidad condicional puede escribirse como $P(C_t|TR_t)$ cuando TR_t puede ser alguna combinación para las variables espaciales latentes, lo que da lugar a una matriz de varianza. De tal forma que se definen variables latentes discretas para ayudar a estimar las intenciones además de utilizarlas como indicadores para el sistema dinámico de conmutación y finalmente incorporar el conocimiento previo de la interacción del tráfico dentro de la RDB.

La predicción en la propuesta implica que el modelo se desarrolla sin la observación de todos los estados; un punto crítico es mostrar si el modelo es capaz de capturar las posibles

interacciones entre los objetos sin observar la información de un estado particular.

4.1.4. Inferencia de parámetros

Una vez determinadas las variables de interés y la estructura de la red, el siguiente paso es estimar la distribución de probabilidad condicional entre los nodos principales y secundarios.

Los resultados de la estimación del comportamiento se obtienen en función de la información previamente observada, que puede expresarse dada la relación 4.13.

$$Z_t^* = max_{Z_t} P(Z_t \mid C_{1:t}) \tag{4.13}$$

Donde Z_t es el comportamiento de los objetos y la dinámica en el momento de la detección t y Z_t^* representa la inferencia resultante más probable basada en la información previamente observada (G. Xie, Gao, Huang, Qian, y Wang, 2018).

Las variables de la estructura de la red implican la estimación de los parámetros de las distribuciones de probabilidad condicional basadas en la estructura de la RDB. Dado que hay nodos ocultos en la red, este problema se considera parcialmente observable. En la topología propuesta, se definen los parámetros de la capa oculta $H = (TR_t)$, así como los parámetros de la capa observable $Op = (Z_t, C_t)$.

Por lo tanto, en el caso parcialmente observable, la probabilidad logarítmica se define por la ecuación (4.14).

$$L = \sum_{nodos} log(\sum_{h} P(H = h, Op = o_{nodos})) \tag{4.14}$$

Donde o_{nodos} son los nodos observables, h son nodos ocultos en la topología desarrollada.

El algoritmo de maximización expectación (Expected Maximization, EM) (Cappé y Moulines, 2009) itera entre un paso de expectativa y un paso de maximización para encontrar los parámetros de máxima verosimilitud (maximum likelihood, ML), en este caso, los posibles caminos en relación con la información proporcionada. Estos dos pasos iteran hasta que se alcanza una convergencia, es decir, dado Z_t converge en probabilidad a K (trayectoria aproximada obtenida afín a la trayectoria real) con respecto al tamaño de los datos aumenta $Z_t \longrightarrow^P K$ (Cappé y Moulines, 2009).

Para optimizar el caso parcialmente observable, se utiliza la teoría de estimación ML para ajustar el modelo propuesto y estimar los posibles parámetros más cercanos al valor del riesgo de colisión en función de las distribuciones de probabilidad condicional con respecto a los parámetros de las variables observables de múltiples objetos en la escena.

El algoritmo EM utiliza la desigualdad de Jensens (Cover, 1999) para maximizar iterativa-

mente, por lo tanto la formula (4.14) puede ser reescrita como (4.15).

$$L \geq \sum_{nodes} \sum_{h} f(h \mid o_{nodes}) log(P(H,O)) - \sum_{nodes} \sum_{h} f(h \mid o_{nodes}) log(f(h \mid o_{nodes})) \qquad (4.15)$$

Donde f es la función de verosimilitud que ajusta el modelo y estima sus parámetros óptimos dado que satisface $\sum_h f(h \mid o_{nodos})$ alrededor de 1 (que es el verdadero valor del parámetro) y además cumplir con las especificaciones de $0 \leq f(h \mid o_{nodos}) \leq 1$.

4.1.5. Modelo de interacción

El modelo de interacción propuesto representa el comportamiento de interacción de múltiples participantes en el tráfico con una categoría arbitraria. Por ejemplo, es posible definir los espacios de estado de un entorno de tráfico típico para el caso de los obstáculos detectados. Una primera inferencia para el desarrollo del algoritmo propuesto para la estimación de la trayectoria se basa en vectores de estado latentes globales (Z_t), es decir, estados previos específicos de los participantes del tráfico.

Algoritmo 1 Modelo inferencia red dinámica bayesiana

Require: Z_t, TR_{t-1}
Ensure: TR_{t+1}, C_{t+1}
1: Información precedente $Z_0 \leftarrow P(Z_0 \mid Z_{t-1})$
2: Actualización y bucle de control (frames a tener en cuenta)
3: **for** $t = 1$ to T **do**
4: cálculo de la probabilidad de los tres movimientos posibles
5: **for** $u = 1$ to 3 **do**
6: $TR_t \leftarrow argmax(P(TR_u \mid Z_t, Z_0))$
7: **end for**
8: $Z_0 \leftarrow Z_t$
9: $TR_{t+1} \leftarrow P(TR_t | Z_t, TR_{t-1})$
10: **end for**
11: $C_{t+1} \leftarrow P(TR_{t+1} | Z_t)$
12: **return** TR_{t+1}, c_{t+1}

En el Algoritmo 1, se proporcionan como entrada el vector de estado actual y el estado de inferencia de colisión C_t de los participantes en el tráfico de vídeo.

En este caso el algoritmo se generaliza para simplificar el proceso iterativo en relación a las variables de interés con respecto a los distintos tipos de obstáculos detectados (peatón , vehículo, camión, ciclista, motociclista, etcetera). El resultado esperado es la aproximación del vector de estado latente en el tiempo $t+1$.

Se observa que se produce un estado condicional tal que $P(TR_t|Z_{t+1})$, es decir, las condiciones de trayectoria implican que hay un cambio de dirección en los desplazamientos de los participantes ($j - participantes$), esto se denota por $P(TR_{t-1}|Z_{1:T}, Z_0)$ durante el transcurso del vídeo (frames previos $t-1$).

Por último se definen tres vectores de dirección discretos (u) para identificar el cambio de dirección los cuales son: mantener del desplazamiento frontal, cambiar el sentido de desplazamiento hacia la izquierda o cambiar el sentido de desplazamiento hacia la derecha.

Capítulo 5

Experimentos y resultados

En este capítulo se muestran las pruebas realizadas para la inferencia de trayectoría de los obstáculos, se señalan los resultados cuantitativos y características cualitativas del procesamiento de las escenas vehiculares en cada una de los bloques de la metodología propuesta del algoritmo de estimación de trayectorias.

Para la ejecución de las pruebas en los diferentes módulos del algoritmo propuesto y el posterior análisis de resultados obtenidos se utilizaron tres conjuntos de datos: kittiVision Benchmark (KITTI Benchmark, 2019), capturas propias y videos de simulador.

El conjunto de datos kittiVision,correspondiente a imágenes izquierda y derecha de la escena vehicular, consta de 25 vídeos con una duración aproximada de 29 seg. cada uno con una tasa de muestreo de 30 fps y una resolución de 1242 x 374 píxeles. La base de datos de capturas propias corresponden a 26 videos en calles con tráfico vehicular y peatonal con una duración de 8 a 9 minutos con una tasa de captura de 30 fps y una resolución de 1920 x 1080 pixeles. Cabe mencionar que se capturan videos correspondientes al lado izquierdo y derecho de la escena. Con respecto a entornos de simulación, se obtuvieron 3 videos a 30 fps con duración total de 30 minutos en el entorno *city car driving* con múltiples objetos en escena a una resolución de 1912 x 1024 pixeles.

El total de videos - escenas se dividieron en secuencias de duración más corta para recabar datos y parámetros para la evaluación del desempeño del algoritmo propuesto a través de las métricas de interés así como de la comparación contra el ground truth.

Así, la cantidad de información procesada corresponde a aproximadamente 650,920 frames en entornos reales más 54,000 frames en el entorno simulado.

5.1. Aproximación de la posición espacial

EL cálculo de la estimación de la distancia a la que se encuentran los obstáculos corresponde en primer lugar a la detección de estos a partir de la información obtenida en la adquisición de datos. La posición espacial en el frame analizado se obtiene por medio del cuadro delimitador, proporcionado por la RNC, además las ROI se proporcionan al modulo de mapeo de disparidades para aplicar la ec. 4.3 dada en el capítulo 4.

En la figura 5.1 se muestran ejemplos de los resultados obtenidos al procesar la imagen izquierda y derecha de la escena para obtener el mapa de disparidad correspondiente de autos o peatones. Generalmente existe una referencia de las coordenadas de la ROI del cuadro envolvente del mapa de disparidad con respecto a la imagen original (recuadro verde.)

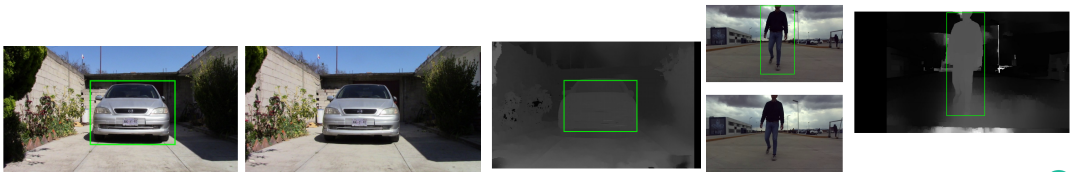


Figura 5.1. Ejemplos resultantes del procesamiento para obtener mapas de disparidad.

Al realizar las operaciones de ubicación y mapeo con una secuencia de video se obtienen los datos necesarios para calcular la distancia aproximada, en metros (m), a la que se encuentra un obstáculo. En la figura 5.2 se muestra un conjunto de frames procesados y en el gráfico se muestran los datos obtenidos al calcular la distancia aproximada de un obstáculo detectado en la secuencia del video, dicha información también se obtiene cuando se detecta más de un obstáculo en la escena.

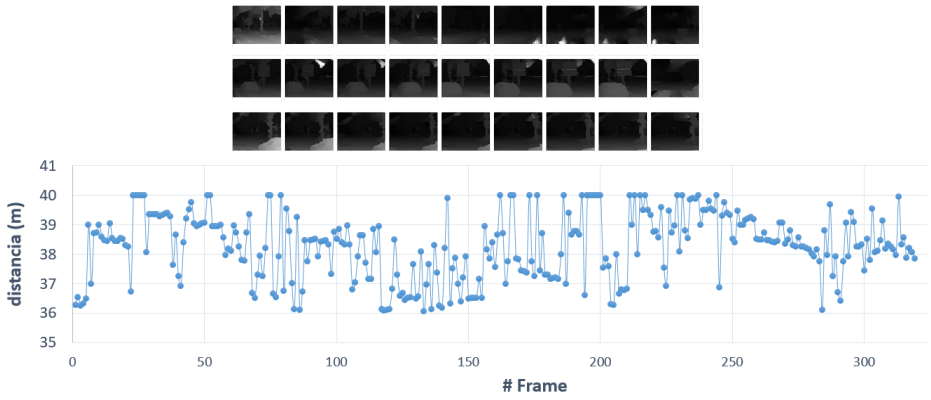


Figura 5.2. Conjunto de datos procesados (mapas de disparidad) para obtener los valores aproximados de las distancias a la que se encuentra un obstáculo en el transcurso de los frames.

Por lo tanto, con la información del desplazamiento (distancia de separación con respecto al ego-vehículo) en el transcurso de los frames, se puede obtener mediante la aplicación de la ec. 4.10 la velocidad relativa estimada de los obstáculos detectados en escena.

5.2. Resultados experimentales

Con respecto a la topología descrita (sección 4.1.3), se realizan experimentos para determinar los resultados de la propuesta, dadas las condiciones de las variables se puede realizar la consulta e inferir la probabilidad de colisión.

Las relaciones de las variables y las distribuciones de probabilidad conjuntas (DPC) son necesarias para procesar las consultas en la topología RDB para obtener el conjunto de inferencias probables con respecto a la probabilidad de colisión normalizada.

El vector cambio de dirección del obstáculo se determina al analizar la posición lateral del objeto y la ubicación anterior donde se encontraba dicho obstáculo, por lo tanto, las características previas al cambio de dirección del obstáculo deben determinarse utilizando datos estadísticos de la velocidad relativa lateral, el ángulo de dirección, la distancia estimada al obstáculo y la posición espacial, de la misma forma también se debe tomar en consideración los parámetros espaciales de los demás obstáculos presentes en la escena.

Los resultados obtenidos en este trabajo de investigación son: probabilidad de colisión según el cambio de vector de dirección, la estimación de la posición espacial y la comparación de la trayectoria a seguir frente a la verdad fundamental (ground truth GT). Se muestra la información en casos particulares, es decir, primero se presentan y analizan las condiciones de los resultados de los experimentos realizados con respecto a un obstáculo detectado para posteriormente incluir los resultados generales con respecto a múltiples obstáculos detectados.

Por ejemplo, las características de los parámetros obtenidos para las variables de interés en una escena vehicular procesada, en este caso para un único obstáculo detectado, son:

- Velocidad relativa media aproximada (v_t) desde la detección en el rango de 20 a 50 km/h.
- Distancia (d_t) de la detección en un rango inicial de 10 m y final de 40 m en el desplazamiento en dirección frontal.
- Ángulo inicial de orientación (ψ_t) (frontal aproximado(p_t)) y al final de la trayectoria (movimiento hacia la izquierda (p_{t+1})).

La información mencionada sobre los parámetros durante el recorrido se representa en un esquema espacial, de igual forma se representa los cambios de dirección vectorial durante el trayecto (figura 5.3) así como frames de interés de la escena ilustrada. Los cambios de dirección corresponden a los vectores dirección frente, derecha e izquierda, según su magnitud es su preponderancia en el posible cambio de dirección.

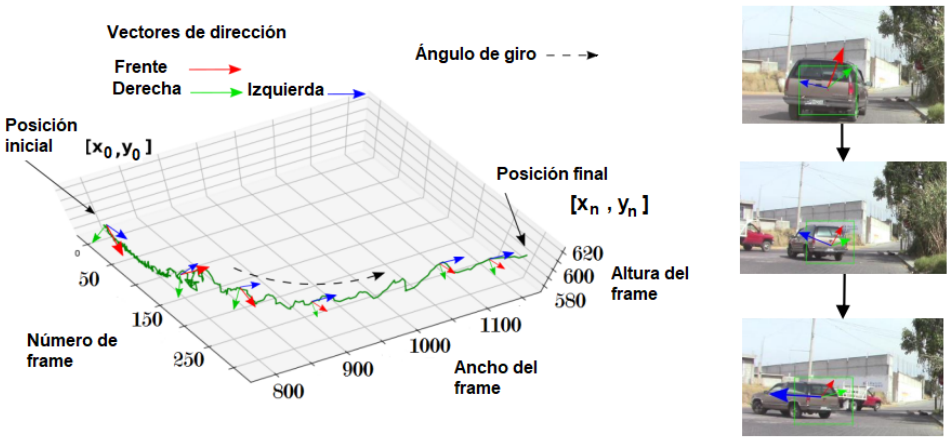


Figura 5.3. Representación de la información sobre el desplazamiento de un obstáculo detectado en el transcurso de los frames, ángulo de giro y vectores de desplazamiento.

Para ejemplificar el funcionamiento del algoritmo propuesto se procesa la información de una escena de video donde sólo se encuentra un obstáculo en la escena vehicular. Se obtienen mediante las etapas de detección y seguimiento de obstáculos las variables de interés necesarias para procesar la estimación del vector de dirección en la RDB. Los resultados obtenidos se muestran en una distribución de probabilidad relacionada con el riesgo de colisión (si lo hay o no) con respecto a mantener una trayectoria determinada.

La tabla 5.1 muestra los resultados obtenidos del procesamiento de la información de la escena vehicular dada en la inferencia del vector cambio de dirección por la RDB.

Tabla 5.1. Resultados del experimento dada la inferencia de colisión para cada dirección discretizada de trayectoria.

	Dirección	Probabilidad (normalizada)
Colisión (si)	izquierda	0.02
Colisión (si)	derecha	0.01
Colisión (si)	frente	0.03
Colisión (no)	izquierda	0.75
Colisión (no)	derecha	0.09
Colisión (no)	frente	0.10

Por lo tanto, el evento que tiene la mayor probabilidad de ocurrir, en este caso específico,

es: no hay probabilidad de colisión del obstáculo detectado ya que se está alejando del lado izquierdo de la referencia de la carretera con respecto a la posición del ego-vehículo (probabilidad *colisión(no)_izquierda = 0.75*).

5.3. Estimación de probabilidad de desplazamiento

Los experimentos con respecto a la estimación del vector cambio de dirección se realizan con tres bases de datos: la primera corresponde a la información del repositorio de kittiVision, la segunda corresponde a secuencias de video capturas propias y la tercera corresponde a videos tomados de un entorno de simulación de ambientes vehiculares, las características de estos datos se mencionan al inicio del capítulo. A continuación se muestran ejemplos específicos procesados con la información de las mencionadas bases de datos.

En el ejemplo siguiente la información corresponde a una escena capturada propia, se detecta un vehículo y se determina el vector cambio de dirección a seguir en frames posteriores (figura 5.4 vector coloreado en amarillo). Se muestra la tabla 5.2 con los resultados de la topología RDB propuesta respecto a la probabilidad de colisión. La probabilidad de cambio de dirección obtenida indica que no hay riesgo de colisión dado el movimiento hacia el lado derecho.

Tabla 5.2. Resultados del experimento dada la inferencia de colisión para cada posible dirección a tomar.

	Dirección	Probabilidad (normalizada)
Colisión (si)	izquierda	0.01
Colisión (si)	derecha	0.01
Colisión (si)	frente	0.02
Colisión (no)	izquierda	0.10
Colisión (no)	derecha	0.81
Colisión (no)	frente	0.05

Figura 5.4. Secuencia de frames para la inferencia de cambio de dirección de un obstáculo (automóvil) detectado.

Con respecto a escenas de ambientes vehiculares de la base de datos kittiVision se tiene

la detección de un obstáculo (vehículo) moviéndose de manera frontal en el intervalo de muestreo (figura 5.5 vector verde). La respuesta de la propuesta RDB arroja los valores con respecto a la estimación de colisión dado el desplazamiento observado (tabla 5.3).

Tabla 5.3. Resultados del experimento dada la inferencia de colisión para cada posible dirección a tomar.

	Dirección	Probabilidad (normalizada)
Colisión (si)	izquierda	0.02
Colisión (si)	derecha	0.02
Colisión (si)	frente	0.06
Colisión (no)	izquierda	0.88
Colisión (no)	derecha	0.01
Colisión (no)	frente	0.01

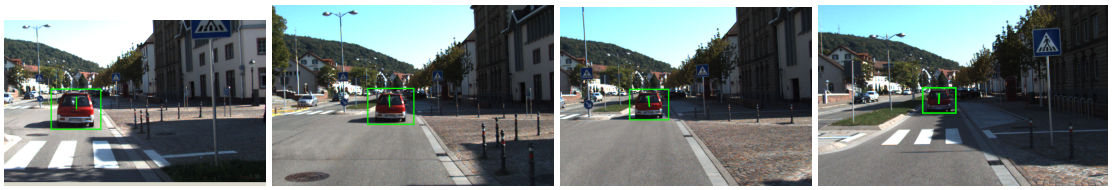


Figura 5.5. Secuencia de frames para la estimación cambio de dirección de un obstáculo (automóvil) detectado.

El ejemplo mostrado en la figura 5.6 corresponde a la estimación de desplazamiento en un ambiente vehicular del video tomado del simulador city car driving. Se observa la estimación de movimiento lateral izquierdo (vector cambio de dirrección rojo) dado que en la tabla 5.4 de resultados el mayor valor corresponde a la probabilidad *colisión(no)_izquierda = 0.80*.

Tabla 5.4. Resultados del experimento dada la inferencia de colisión para cada posible dirección a tomar.

	Dirección	Probabilidad (normalizada)
Colisión (si)	izquierda	0.03
Colisión (si)	derecha	0.05
Colisión (si)	frente	0.04
Colisión (no)	izquierda	0.80
Colisión (no)	derecha	0.03
Colisión (no)	frente	0.05

De la misma forma como se obtiene la probabilidad de cambio del vector de dirección para el conjunto de frames analizados en los ejemplos anteriores, también se puede obtener

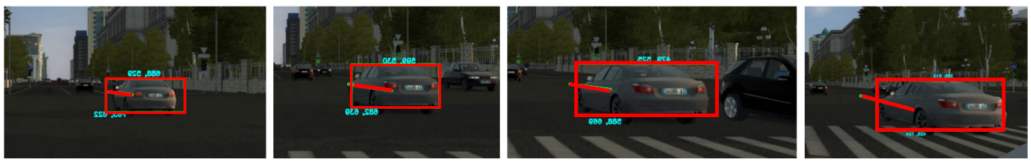


Figura 5.6. Secuencia de frames para la estimación de cambio de dirección de un obstáculo (automóvil) detectado en un ambiente virtual.

para un intervalo de interés del video mientras se detecte el obstáculo. Para indicar la información de la inferencia de probabilidad del vector de dirección a través de un intervalo de muestreo (frames donde se detecta el obstáculo) se pueden representar dichos resultados de forma gráfica, es decir, recopilar la información de los resultados de las tablas de probabilidad. Además, en la gráfica se coloca de forma representativa la dirección GT tomada por el vehículo (figura 5.7).

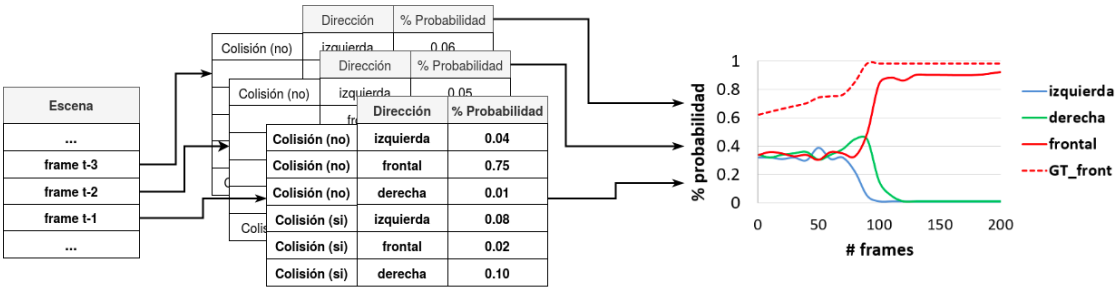


Figura 5.7. Recopilación de los valores de la estimación de probabilidad del cambio de dirección para los tres posibles vectores de dirección y su representación gráfica.

Por ejemplo, en la figura 5.8, la interpretación de la gráfica, muestra la probabilidad de cambio de dirección (en línea continua) y el GT (en línea punteada) del intervalo de frames en donde se detectó un obstáculo.

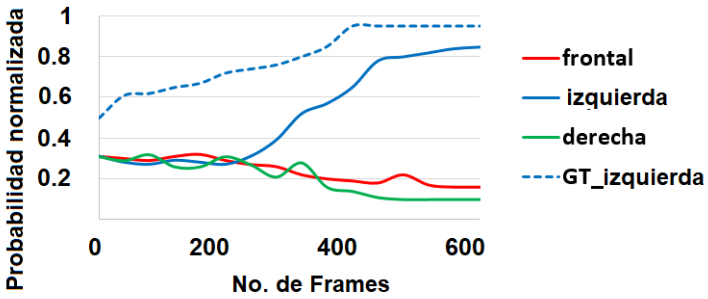


Figura 5.8. Gráfica con los valores de la estimación de probabilidad del cambio de dirección

De igual forma, la línea GT_izquierda indica la dirección tomada por el obstáculo en ese intervalo (el verde implica el cambio de dirección a la derecha, el azul el cambio de dirección a la izquierda y el rojo implica mantener la dirección frontal).

De forma específica, la estimación de los resultados de las trayectorias puede compararse con respecto al GT trazando los valores de las probabilidades obtenidas del vector de dirección en el transcurso del vídeo (# frames determinados). Para mostrar el comportamiento (cambios) del vector de dirección obtenido durante el intervalo de interés, la información se muestra en forma gráfica.

Por ejemplo, en la gráfica de la figura 5.9 se muestra que el GT de la trayectoria del objeto en escena corresponde al movimiento en dirección lateral derecha y el comportamiento de la respuesta obtenida, por la topología RDB implementada, distribuye la probabilidad de desplazamiento inicialmente en proporciones similares (en los 3 vectores de dirección) sin embargo al obtener más información (transcurso de los frames) el cambio de dirección a la derecha incrementa su probabilidad, lo que refuerza la inferencia del movimiento en dirección derecha (valor cercano a 1). El análisis mencionado se puede replicar para las gráficas de las figuras 5.10, 5.11.

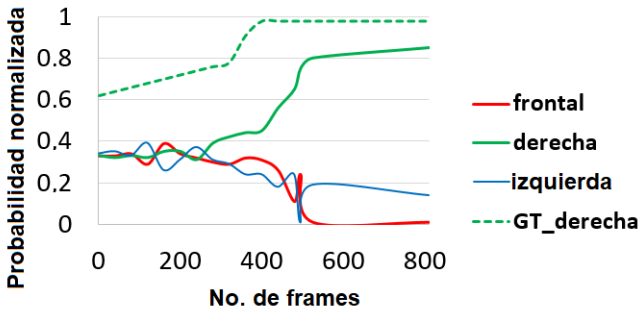


Figura 5.9. Probabilidad de cambios de dirección frente a GT (derecha)

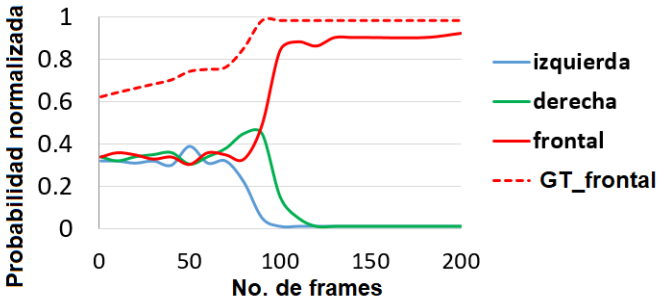


Figura 5.10. Estimación de la probabilidad de cambios de dirección frente al GT (frente).

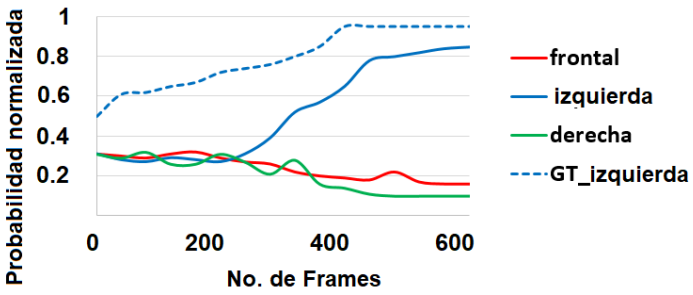


Figura 5.11. Probabilidad de cambios de dirección frente a GT (izquierda).

En las figuras anteriores cada gráfica respectiva, corresponde al seguimiento de un solo objeto y a la estimación de la probabilidad porcentual normalizada obtenida para cada vector de dirección, el cambio de dirección más probable es el que tiene el valor más alto.

En el caso de la estimación del cambio de dirección de múltiples objetos en la escena, se lleva a cabo el mismo proceso de forma iterativa, lo que implica que, el planteamiento de la dinámica de la escena (de acuerdo al sección 4.1.3) queda de la siguiente forma: los objetos detectados con información Z_t relativa a la posición temporal ..., $(V_{t-1}, \Gamma_{t-1}, D_{t-1}, \Psi_{t-1}), (V_{t+1}), (V_t, \Gamma_t, D_t^i, \Psi_t)$, $(V_{t+1}, \Gamma_{t+1}, D_{t+1}, \Psi_{t+1})$... ya que el riesgo de colisión se consulta dada la estimación de la trayectoria $P(TR_{t+1}|Z_t)$.

La figura 5.12 muestra los vectores de dirección inferidos para múltiples objetos en la escena. Se muestra (de arriba a abajo) una serie de frames consecutivos con los objetos

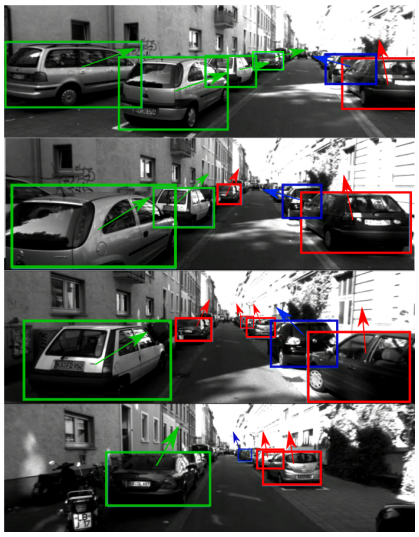


Figura 5.12. Representación vectores de dirección de múltiples objetos detectados.

detectados, por lo que, se colorea el vector de dirección predominante obtenido (para re-

saltar el frame envolvente también se colorea del mismo color) en este caso el color verde corresponde al vector de dirección derecho, el azul al vector de dirección izquierdo y el rojo al vector de dirección frontal.

La información de la probabilidad de cambio de dirección se mostró en forma gráfica para sólo un obstáculo en las figuras 5.9, 5.10,5.11. En el caso de múltiples obstáculos detectados también se puede desplegar las estimaciones de probabilidad en el transcurso de los frames a partir del frame en el cual fueron detectados hasta donde desaparecen de escena.

En la representación gráfica de la figura 5.13, se muestra la información de probabilidades de los respectivos vectores de dirección de múltiples obstáculos detectados, como se mencionó si el objeto desaparece de la escena o deja de detectarse la estimación culmina. Por ejemplo, los datos mostrados en la línea azul marino (car_2b) de la gráfica terminan aproximadamente después del frame 100, esto quiere decir que el objeto salió de escena.

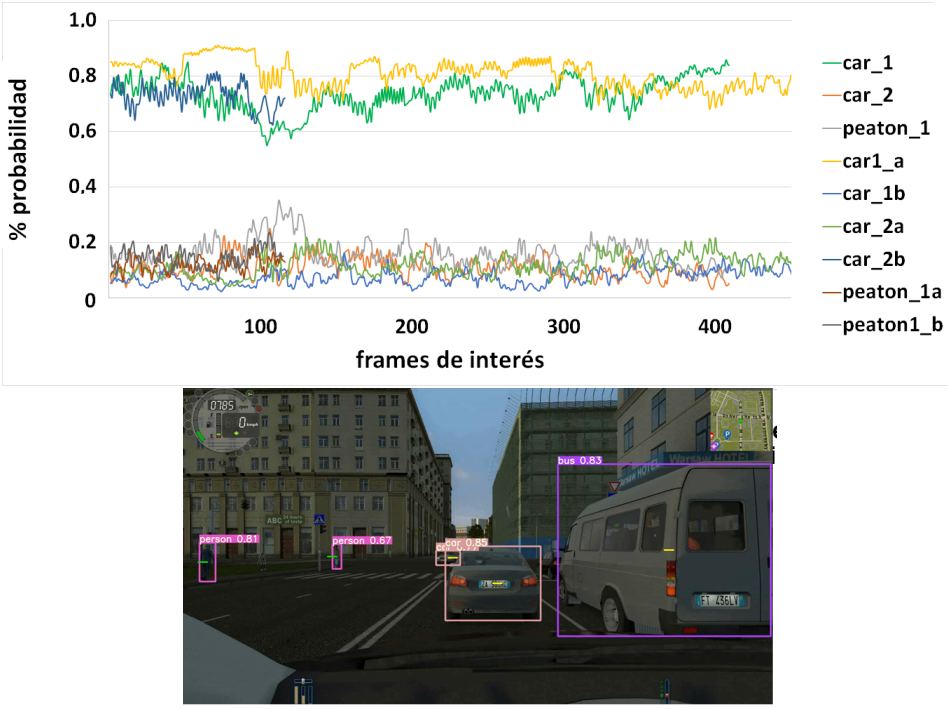


Figura 5.13. Múltiples obstáculos detectados en escena en el entorno de simulación y la representación gráfica de sus parámetros de probabilidad en relación a los vectores de dirección.

Por cuestiones prácticas la información tiene una mejor representación e interpretación analizando las gráficas de forma individual para cada obstáculo detectado.

5.3.1. Evaluación del seguimiento de obstáculos

Para la evaluación de las métricas de detección se toman en consideración los resultados obtenidos al procesar secuencias de videos de las tres bases de datos utilizadas. Con respecto a la evaluación del algoritmo propuesto concerniente al módulo de seguimiento se utilizan como base de medición los parámetros descritos en el marco teórico (sección 2.8).

En la tabla 5.5 se muestran un conjunto de resultados de las métricas de evaluación al realizar experimentos referentes al seguimiento de objetos. Los datos procesados se obtienen en este caso de secuencias de video de la base de datos kittiVision.

Tabla 5.5. Resultados obtenidos al evaluar el seguimiento de objetos en escena.

Evaluación del seguimiento (parámetros)					
Secuencias kittiVision	Duración (segundos)	MOTA (%)	MOTP(%)	FN	FP
S1	107.7	80.3	83.8	41	35
S2	273.9	84.2	86.8	105	88
S3	342.0	76.8	83.7	16	13
S4	248.6	81.3	84.0	96	80
S5	179.6	85.9	89.0	69	58
S6	279.0	84.1	87.7	30	25
S7	165.0	72.2	75.9	25	21
S8	279.5	80.1	85.5	107	90
S9	11.8	83.9	87.8	36	24
S10	218.0	78.4	81.3	84	70

Para visualizar el total de las secuencias de la información correspondiente a la base de datos kittiVision, los videos capturados propios y los videos del simulador se presentan los resultados en forma gráfica. En la gráfica de la figura 5.14 se muestran los valores obtenidos de las métricas MOTA y MOTP para las secuencias de video ($S1, S2, ...Sn$) de la base de datos kittiVision.

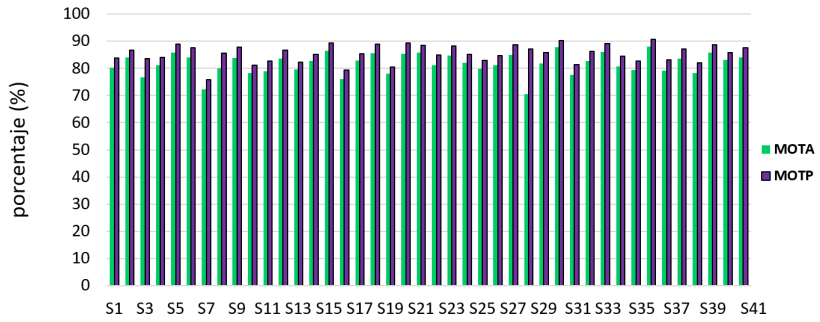


Figura 5.14. Resultados obtenidos para los parámetros MOTA y MOTP para las secuencias de video de kittiVision.

De la misma forma la figura 5.15 muestra la gráfica con los parámetros MOTA y MOTP obtenidos para las secuencias de ambientes vehiculares capturados propios.

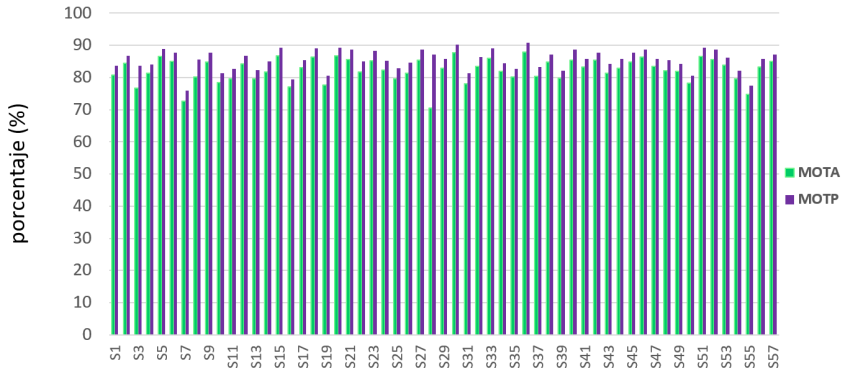


Figura 5.15. Parámetros obtenidos para las métricas MOTA y MOTP para las secuencias de video capturadas propias.

En la gráfica de la figura 5.16 se observan los datos correspondientes a MOTP y MOTA obtenidos al procesar las secuencias de video obtenidas del simulador.

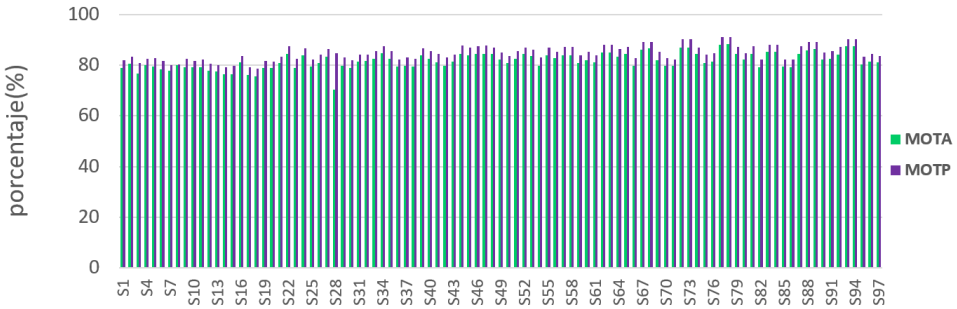


Figura 5.16. Resultados de los parámetros MOTA y MOTP al procesar las secuencias de video obtenidas del simulador city car driving.

Para describir el comportamiento en general de los resultados obtenidos referentes a las métricas de seguimiento se muestra el resumen de la información en la tabla 5.6, en esta se observan las medidas de tendencia central (promedio y desviación estándar μ).

En resumen, la métrica MOTA obtenida se encuentra en el intervalo entre 81.8 % a 82.3 % con una desviación estandar de alrededor de $\mu = 3$, para MOTP el rango se encuentra entre 84.9 % y 85.5 % con $\mu = 3$ aproximadamente, lo cual muestra una consistencia en el funcionamiento del seguimiento de los obstáculos, sin embargo también se puede hacer mención de los parámetros obtenidos en el mejor de los casos y el peor de los casos.

Las secuencias de video en el mejor de los casos implica un ambiente vehicular con oclusiones nulas y pocos obstáculos en escena, tomando en consideración los videos con

Tabla 5.6. Resumen de los resultados obtenidos para las bases de datos con respecto a las métricas de seguimiento.

Evaluación de seguimiento de la propuesta						
Base de datos	MOTA promedio (%)	MOTP promedio (%)	μ MOTA	μ MOTP	FN promedio	FP promedio
KittiVision	81.8	85.5	3.86	3.23	56	49
Datos propios	81.9	84.9	3.11	2.90	78	61
Videos simulador	82.3	85.4	3.70	3.20	85	71

estas características se puede realizar un submuestreo de las secuencias a procesar. Los videos correspondientes al mejor de los casos corresponden al 29 % de la información total.

Los resultados bajo las mejores condiciones de submuestreo obtenidos por la propuesta corresponden a base de datos kittiVision MOTA = 88.0 % y MOTP = 90.7 %, base de datos propia MOTA = 87.8 % y MOTP = 90.7 % y finalmente base de videos del simulador MOTA = 89.7 % y MOTP = 91.1 %.

De la misma forma considerando el submuestreo en el peor de los casos, es decir, con oclusiones de los obstáculos de forma continua a la vez que en la escena se encuentran gran cantidad de estos, los valores obtenidos para las métricas definidas corresponden a: base de datos kittiVision MOTA = 74.6 % y MOTP = 79.5 %, base de datos propia MOTA = 73.8 % y MOTP = 77.5 % y finalmente base de videos del simulador MOTA = 73.7 % y MOTP = 79.8 %.

La información correspondiente a condiciones consideradas como peores casos corresponden al 19 % de la información total procesada.

En lo que concierne al marco de referencia, los trabajos relacionados establecen valores obtenidos de MOTA entre 75.7 y 88.9 % en la tasa de rendimiento, para MOTP el valor deseado se encuentra en el rango de 73.4 % y 85.6 % (Wu et al., 2021). Sin embargo no se mencionan las condiciones del submuestreo o los parámetros obtenidos en el mejor y el peor de los casos ya que sólo se tiene referencia de la base de datos utilizada (kittiVision).

Debido a esto, no se puede hacer el análisis comparativo del muestreo o submuestreo por lo que de acuerdo a la variabilidad de escenarios vehiculares de la base de datos kittiVision los esquemas de respuesta y rendimiento sólo se pueden realizar de forma cuantitativa comparando los resultados obtenidos vs los datos duros proporcionados por los autores. La tabla 5.7 muestra la comparativa resultante.

Como se logra observar de la comparativa cuantitativa realizada, de forma general la propuesta tiene resultados competitivos en comparación con los trabajos relacionados cabe señalar que los resultados reportados de la propuesta corresponden al caso promedio. El valor promedio obtenido por la propuesta en la métrica MOTA esta alrededor de los mejores

Tabla 5.7. Resumen comparativo entre trabajos relacionados con las métricas de seguimiento.

Evaluación de seguimiento (base de datos kittiVision)					
Modelo	Autores	MOTA (%)	MOTP (%)	FP	FN
FANTrack	Baser et al.,2019	75.70	78.46	163	55
PMBM	Scheidegger et al.,2018	76.15	73.42	57	73
autoTrack	Burnett et al.,2019	77.72	82.33	127	62
AB3DMOT	Weng et al.,2019	80.39	81.26	100	56
3DT	Hu et al.,2019	82.25	80.52	104	40
mmMOT	Zhang et al.,2019	83.84	85.24	105	44
MTPCPCG	HU et al.,2021	88.88	85.64	70	42
Propuesta	Reyes et al.,2022	81.80	85.5	72	49

rendimientos y el parámetro MOTP por otro lado es muy cercano al mayor valor.

Sin embargo si se considera el submuestreo con las mejores condiciones, la propuesta tiene un rendimiento superior en el parámetro MOTP y prácticamente igual a la métrica MOTA con respecto al estado del arte.

5.3.2. Evaluación de la posición estimada

La localización espacial de los obstáculos detectados en escena se determina comparando puntos en las ROI de los frames de vídeo consecutivos de interés dado el frame de referencia (cuadro envolvente). El objetivo es encontrar ubicaciones coincidentes o no coincidentes entre el transcurso del video.

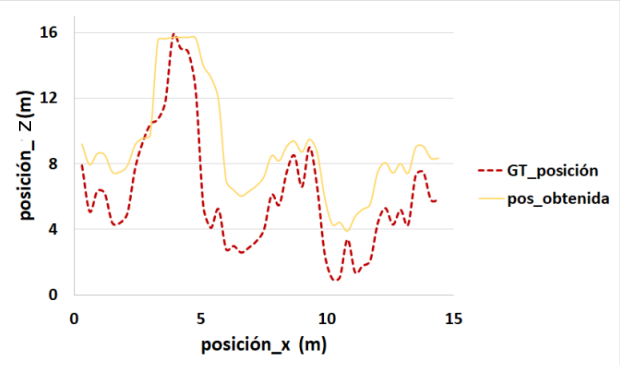


Figura 5.17. Probabilidad de movimiento obtenida frente a GT (posición).

Por ejemplo, en la figura 5.17 se muestran los cambios de posición que sigue un obstáculo. La coordenada x se refiere a la distancia aproximada (metros) a la que se encuentra el obstáculo con respecto a la referencia de la perspectiva central (carretera o entorno vehicular capturado en imagen). La coordenada z se refiere a la distancia aproximada (metros) de separación de la que se encuentra el objeto con respecto al ego-vehículo.

Los valores de *posición_z* y *posición_x* corresponden a la estimación de ubicación dentro del intervalo de seguimiento del obstáculo dentro de los frames de interés (línea continua amarilla), mientras que la ubicación real esta representada por la línea punteada ($GT\_posicion$).

Además de obtener cada par de puntos (x, z) aproximados a partir de la posición espacial en intervalos de tiempo de interés, también es importante conocer la diferencia entre el cambio de dirección real, a través del ángulo de orientación, contra el valor estimado ya que con esta información se puede describir el vector cambio de dirección de los obstáculos durante el transcurso de la escena.

En la figura 5.18 se muestra la comparativa del ángulo de orientación del obstáculo mencionado en el ejemplo de la figura 5.17. La línea sólida amarilla representa el ángulo aproximado obtenido, la línea punteada roja representa el GT del ángulo de orientación del obstáculo través de frames consecutivos de interés. Como era de esperar, existe una diferencia entre los parámetros obtenidos en la estimación del vector de dirección de los obstáculos detectados con respecto al GT.

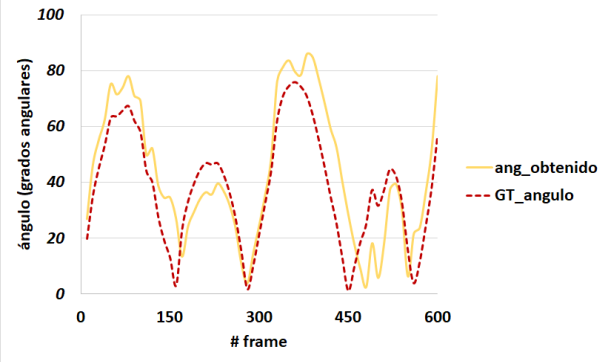


Figura 5.18. Probabilidad de movimiento obtenida frente a GT (ángulo).

Los resultados de la propuesta metodológica se evalúan cuantificando el vector de dirección del movimiento estimado, el resultado a evaluar es el error obtenido de la diferencia entre la estimación del movimiento (Z_t) y la trayectoria real $(Z_{(GT)t})$ en el intervalo (T) donde se detecta el obstáculo en escena. Para cuantificar el desempeño de los resultados obtenidos del cálculo de la estimación del vector de dirección, se utiliza la raiz del error cuadrático medio (RMSE) ecuación (5.1) tanto para la posición espacial como para el ángulo de orientación.

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=0}^{T} ||Z_t - Z_{(GT)t}||^2} \tag{5.1}$$

La tasa de detección de la información corresponde a un intervalo de 5 frames, es decir, el vector de dirección se recalcula cada 0.166 segundos, por lo que las secuencias corresponden al análisis del error con respecto al desplazamiento ($Ruta\_n$) a través de 650 a 1250 frames procesados aproximadamente por escena.

La tabla 5.8 muestra algunos ejemplos de los resultados del error obtenido respecto a un conjunto de secuencias dadas al modelo propuesto para evaluar las condiciones de movimiento de los objetos en la escena con respecto al GT.

Tabla 5.8. Resultados experimentales (RMSE) entre el GT y la posición espacial y el ángulo de dirección de los objetos en 10 escenas vehiculares.

	RMSE									
	Ruta_1	Ruta_2	Ruta_3	Ruta_4	Ruta_5	Ruta_6	Ruta_7	Ruta_8	Ruta_9	Ruta_10
estimación espacial vs GT (m)	1.8	2.3	3.5	2.1	2.3	1.2	1.5	1.8	2.1	1.7
ángulo estimado vs GT (° angulares)	8.5	7.9	10.4	11.4	14.3	12.1	5.8	6.5	6.4	9.8

Para validar los resultados obtenidos con respecto a la tasa de error, se realiza una comparación de la propuesta contra los trabajos relacionados en el estado del arte. El modelo propuesto se compara con el modelo RBDHM (Schulz et al., 2019) y con el modelo ERI (L. Sun et al., 2019), dicha comparación incluye datos de las variables (velocidad, posición, orientación y distancia de separación de los objetos) de interés presentes y procesados en los modelos mencionados.

Las bases de datos comparadas corresponden a kittiVision, la base de datos propia y los videos obtenidos del simulador.

Cabe hacer mención que sólo se realiza la comparativa contra el modelo RDBHM con la base de datos kittiVision ya que para la estimación de posición, además de las variables mencionadas, utiliza información proveniente de sensores lidar y en el caso de la base de datos propia y los videos del simulador no se cuenta con ese tipo de datos. Para el caso del modelo ERI no hay inconveniente y se realiza la comparativa con las tres bases de datos.

En la gráfica de la figura 5.19 se muestran los resultados obtenidos a partir de 41 secuencias de vídeo, de la base de datos kittiVision, con respecto al error RMSE de la posición espacial obtenido por cada método con respecto al GT . Esto permite una evaluación cualitativa de los resultados obtenidos.

En el mismo sentido se realizan las comparaciones con respecto a los datos capturados propios (figura 5.20) y con datos de los videos del simulador (figura 5.21).

La comparación se realiza con respecto a muestreos de distintos ambientes vehiculares de la base de datos kittiVision. Las condiciones de la muestra implican procesar ambientes vehiculares con alta carga de objetos en escena, es decir, 10 o más obstáculos en escena, además, existen oclusiones totales o parciales durante el intervalo de análisis, velocidad de

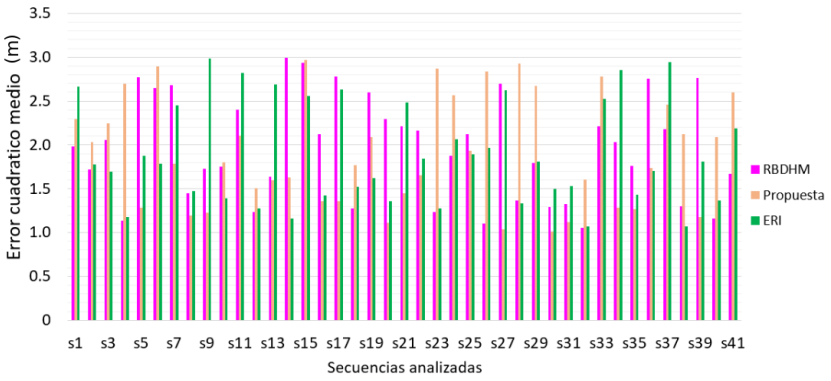


Figura 5.19. Error obtenido de la ubicación espacial de objetos detectados en secuencias de video base de datos KittiVision, propuesta vs trabajos relacionados.

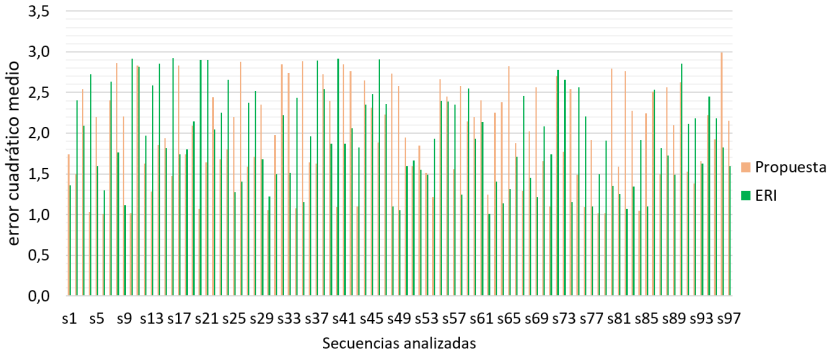


Figura 5.20. Error obtenido de la ubicación espacial de objetos detectados en secuencias de video base de datos propia, propuesta vs trabajos relacionados.

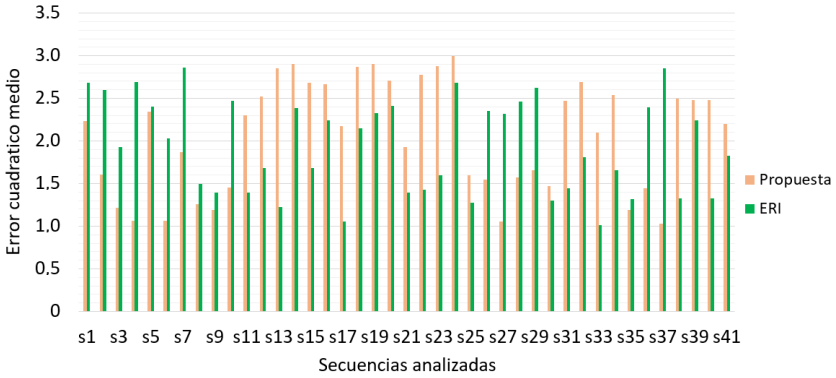


Figura 5.21. Error obtenido de la ubicación espacial de objetos detectados en secuencias de video base de datos simulador, propuesta vs trabajos relacionados.

desplazamiento alrededor de 50 km/hr, entre otras dificultades para llevar a cabo de forma correcta el seguimiento y posterior estimación de cambio de dirección.

Las condiciones antes mencionadas implican submuestreo de las condiciones del am-

biente que se identifican como los peores casos. Por otro lado, el mejor de los casos implica submuestreo de las escenas vehiculares donde la tasa de vehículos es menor o igual a 3 obstáculos en escena, la oclusión de estos mínima o es nula, la velocidad de desplazamiento se encuentra alrededor de 20 km/hr.

Para el caso promedio se consideran aquellas muestras donde la velocidad de desplazamiento esta entre 20 - 40 km/hr, existen condiciones de oclusión viables de procesar así como las demás variables de interés se encuentran en condiciones de procesar sin mayores complicaciones.

El resumen de la comparación de los datos obtenidos se presenta en la tabla 5.9.

Tabla 5.9. Resultados obtenidos al evaluar el seguimiento de objetos en escena.

Evaluación posición espacial RMSE (m)					
Base de datos	Modelo	RMSE (peor caso)	RMSE (mejor caso)	RMSE (caso promedio)	μ
kittiVision	RBDHM	1.8	1.1	1.5	0.7
	ERI	2.3	1.3	2.1	0.9
	Propuesta	2.0	1.1	1.6	0.5
Simulador (city car)	ERI	1.9	1.1	1.7	0.6
	Propuesta	1.7	1.0	1.5	0.4
Datos propios	ERI	1.7	1.2	1.6	0.5
	Propuesta	1.6	0.9	1.4	0.6

El siguiente parámetro a comparar es la estimación del vector de desplazamiento. Como se ha mencionado, el vector de dirección estimado es el que, dados los parámetros de las variables obtenidas, tiene más probabilidades de suceder.

5.3.3. Evaluación de estimación de dirección

La trayectoria de un obstáculo es una secuencia de estados espaciales en un intervalo de muestreo entre frames, el cambio potencial de dirección de la trayectoria implica evaluar y conocer la dinámica previa del desplazamiento. En esta sección se muestra la evaluación cuantitativa y cualitativa de la estimación del cambio de dirección en comparación con las metodologías RBDHM y ERI mencionadas en la subsección 5.3.2.

El análisis y posterior comparativa de la estimación del trayecto implica evaluar el desempeño en cuanto a la estimación de probabilidad de la detección–cambio del vector de dirección de los obstáculos detectados, así como también el GT de la dirección del obstáculo.

Los resultados obtenidos se presentan mediante una gráfica compuesta por el eje vertical correspondiente a la estimación de probabilidad de cada método y el eje horizontal corresponde al intervalo de muestreo. También se representa con una linea punteada la dirección del GT del obstáculo detectado, como se mencionó anteriormente la dirección se identifica con movimientos relacionados a la izquierda, derecha o seguir de manera frontal.

En la figura 5.22 se muestra el resultado (porcentaje de probabilidad normalizado) de la estimación de la dirección obtenida por el método propuesto, para un obstáculo en una escena vehicular de 800 frames, contra los resultados obtenidos de trabajos relacionados con respecto a la misma escena y finalmente el GT (en este caso vector *dirección frontal*) de la trayectoria seguida por el obstáculo.

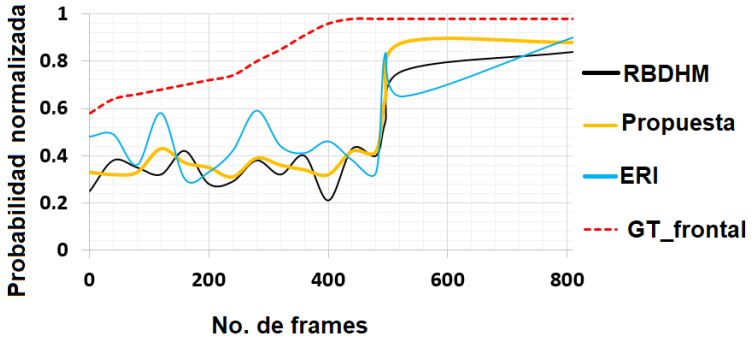


Figura 5.22. Comparación de la propuesta vs trabajos relacionados.

Con el porcentaje de probabilidad obtenido a través de los frames de interés se obtiene el error de cada método respecto al GT. El error se obtiene de la diferencia entre la probabilidad estimada de movimiento (Z_p) con respecto al GT del desplazamiento que realiza el objeto (Z_{GT}) en el intervalo de frames fr analizados por medio de la ecuación 5.2.

$$RMSE_p = \sqrt{\frac{1}{fr}\sum_{i_t=0}^{fr} ||Z_{p(fr)} - Z_{GT(fr)}||^2} \tag{5.2}$$

El error estimado para la inferencia de la trayectoria frente al GT en el intervalo de interés corresponde a RBDHM de 0.12, a ERI de 0.16 y la propuesta obtiene un error de 0.14. El análisis se realiza de forma similar para la información presentada en las figuras 5.23 y 5.24.

Para el intervalo de interés en la escena con GT *dirección derecha*, para RBDHM el error estimado corresponde a 0.29, la propuesta tiene 0.25 y ERI tiene 0.31 (figura 5.23).

Para la escena con GT *dirección izquierda* los errores obtenidos por RBDHM, ERI y la propuesta son 0.33, 0.35 y 0.26 respectivamente (figura 5.24).

La información presentada en las gráficas anteriores corresponden a el resultado de escenas particulares. La comparativa general de los resultados del método propuesto contra los trabajos relacionados se observa en la tabla 5.10.

Se debe notar qué como en el caso de los parámetros evaluados para el modelo RBDH en lo que comprende a la estimación de probabilidad sólo se realiza la comparativa con la base de datos kittiVision por las razones mencionadas en la subsección 5.3.2. En el caso del

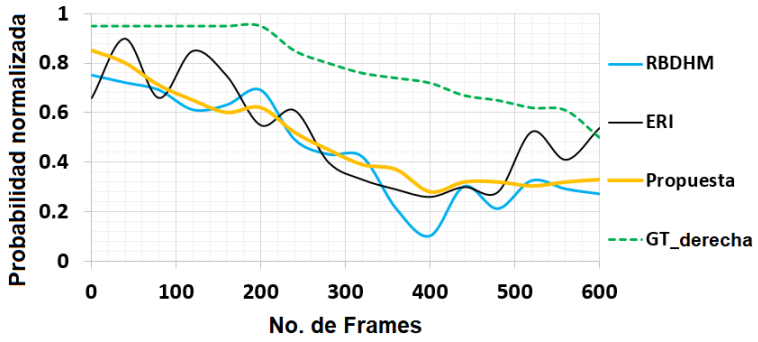


Figura 5.23. Comparación de la propuesta vs resultados en trabajos relacionados.

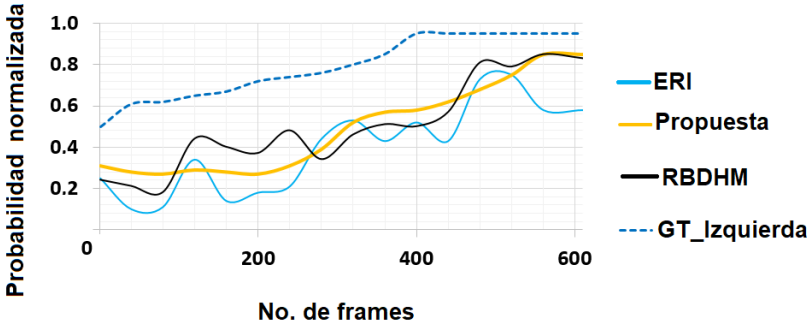


Figura 5.24. Comparativa resultados de la propuesta vs trabajo relacionado.

Tabla 5.10. Resultados obtenidos al evaluar el seguimiento de objetos en escena.

Evaluación estimación de dirección RMSE					
Base de datos	Modelo	RMSE (peor caso)	RMSE (mejor caso)	RMSE (caso promedio)	μ
kittiVision	RBDHM	0.286	0.126	0.176	0.073
	ERI	0.321	0.161	0.181	0.049
	Propuesta	0.274	0.125	0.177	0.052
Simulador (city car)	ERI	0.295	0.141	0.155	0.039
	Propuesta	0.284	0.124	0.161	0.054
Datos propios	ERI	0.272	0.131	0.181	0.051
	Propuesta	0.265	0.125	0.169	0.056

modelo ERI no hay inconveniente y se realiza la comparativa con las tres bases de datos.

El método RBDHM tiene un enfoque cuantitativo cercano al GT, la propuesta y el método ERI tienen resultados similares, pero como muestra el gráfico, en las figuras 5.22,5.23,5.24, hay oscilación en el transcurso (del intervalo) de la escena. Esta oscilación no es deseable ya que implica cambios bruscos de dirección.

5.4. Características de procesamiento

Como punto final a tratar es importante mencionar las características de procesamiento en cuanto a los tiempos de ejecución de los módulos que comprenden al algoritmo propuesto.

Dado que las perspectivas de implementación de la propuesta del algoritmo implican su operación en tiempo real es necesario definir en que términos de funcionamiento (temporal) se encuentra trabajando actualmente bajo las condiciones del hardware utilizado.

Cabe mencionar que el algoritmo propuesto se ejecutó en un equipo con las características siguientes: procesador core i7 a 2,5 GHZ, con una memoria ram disponible de 8 GB, un procesador gráfico intel HD Graphics 520 (skylake GT2) y S.O. ubuntu 18. Dadas las limitantes del hardware para realizar las operaciones correspondientes de la estimación de trayectorias en un tiempo competitivo se procede a realizar un análisis experimental de los rangos de ejecución para cada bloque-módulo procesado en el algoritmo propuesto.

Los experimentos para recabar información comprenden escenas de las tres bases de datos utilizadas a lo largo de esta investigación. En la gráfica de la figura 5.25 se observa los módulos que comprenden al algoritmo propuesto y el sumario del tiempo de procesamiento utilizado para ejecutar las múltiples operaciones sobre los datos recibidos.

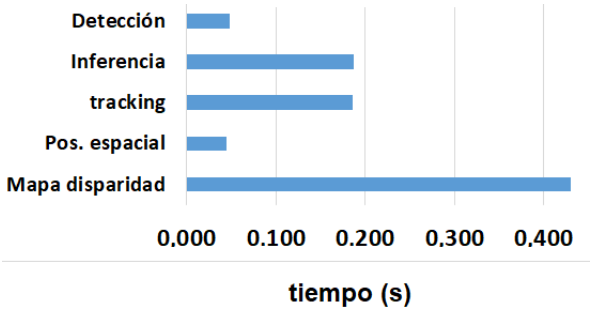


Figura 5.25. Comparativa experimental tiempo de ejecución por módulo.

En la tabla 5.11 se puede observar a detalle los valores respectivos del tiempo de procesamiento experimental obtenido al realizar las ejecuciones en las distintas escenas vehiculares de las bases de datos disponibles. También se presenta la tasa de variabilidad expresada a través de la desviación estándar.

Cabe recordar que la tasa de procesamiento de video es de 30 fps, el intervalo de muestreo corresponde a un intervalo de 5 frames, lo que equivale a toma de datos aproximadamente cada 0.166 segundos. De la tabla 5.11 puede observarse que el proceso completo le toma 0.897 s al algoritmo para completar las operaciones y arrojar una respuesta de salida. Por lo qué al hacer una relación proporcional del tiempo total que le toma realizar un ciclo de

Tabla 5.11. Valor promedio de tiempo de ejecución de los bloques correspondientes al proceso de estimación de trayectoria.

Bloque	Tiempo promedio (s)	μ
Mapa disparidad	0.431	0.028
Cálculo pos. espacial	0.044	0.014
Seguimiento	0.186	0.007
Detección	0.049	0.015
Estimación trayectoria	0.187	0.012
total (suma)	0.897	

operación al algoritmo contra el tiempo de muestreo resulta que se debe mejorar la tasa de procesamiento en aproximadamente 5 veces ($\frac{0.897}{0.166} = 5.4$).

Mejorar el tiempo de procesamiento es posible con la ayuda de hardware más potente y considerando trasladar los bloques, que necesitan mayor tiempo de procesamiento, en un enfoque paralelo o a través de tensores, esta mejora se plantea realizar en el trabajo a futuro.

5.5. Discusión

En esta sección se entra a detalle en cuanto a la comparativa cuantitativa y cualitativa de los resultados obtenidos a través de los experimentos realizados.

Con respecto a las métricas de la evaluación de seguimiento, los resultados de la comparativa muestran en primer lugar valores dentro de los parámetros encontrados en la literatura. Como se mencionó en la sección 5.3.1 para realizar la comparativa se analizaron escenarios correspondientes a los mejores casos, peores casos y el caso promedio, actividad que no presentan los modelos de los trabajos relacionados. En este sentido, cuantitativamente el rendimiento de la métrica MOTA del caso promedio obtenido por la propuesta sólo esta por debajo de 3 modelos y por arriba de 4 modelos analizados, en el caso de MOTP esta a 0.14 % del mejor rendimiento. Si se considera el resultado de la propuesta considerando el submuestreo del mejor de los casos la métrica MOTA se encuentra a 0.88 % del mejor resultado y la variable MOTP se encuentra por encima de cualquier trabajo por 5.06 %.

En el caso de la evaluación de la posición espacial, como también se mencionó en la sección 5.3.2, el modelo RBDHM solamente se pudo comparar con la base de datos kittiVision ya que dentro de su estructura topológica requiere de datos provenientes de un sensor lidar lo que es un inconveniente ya que los datos capturados propios y los datos del simulador no proporcionan dicha información.

La evaluación de la propuesta de forma cuantitativa para los trabajos que utilizan la base

de datos kittiVision arrojan las siguientes observaciones: se puede identificar que los resultados de la propuesta están a 0.1 unidades del mejor rendimiento, pero cabe mencionar que se tiene menor variabilidad ya que la desviación estándar es menor ($\mu_{RBDHM} > \mu_{propuesta}$).

En el caso de la evaluación que se realiza con respecto al modelo ERI, la propuesta tiene mejor rendimiento al procesar la información en las tres bases de datos utilizadas, sólo cabe mencionar que ERI tiene un valor menor (menor variabilidad) en μ en los escenarios de la base de datos capturados propios, dicha diferencia es de una décima.

En lo que respecta a la evaluación de la posición espacial, el rendimiento en el mejor de los casos es similar al comparar el algoritmo propuesto vs RBDHM.

En lo referente a la estimación de dirección, el modelo RBDHM como en la evaluación anterior solamente se compara con respecto a la base kittiVision, el modelo ERI se compara con las tres bases de datos. Los resultados de la evaluación cuantitativa (sección 5.3.3) muestran valores obtenidos para la estimación de movimiento prácticamente iguales del modelo RDBHM vs la propuesta, aunque la propuesta tiene menor variabilidad pues μ es menor, en el caso de la comparación con ERI la propuesta es mejor en el rendimiento pero tiene variabilidad en comparación con ERI ya que μ es mayor ($\mu_{ERI} = 0.049$, $\mu_{propuesta} = 0.052$), sin embargo dicha diferencia es considerablemente baja.

Para la comparación de la propuesta en el mejor de los casos con respecto a los datos de los videos en el simulador, el rendimiento de la propuesta es mejor por 0.06 unidades con una variabilidad mayor de alrededor de 0.015 unidades. En el caso del análisis de los datos propios ERI vs la propuesta, ERI tiene menor rendimiento contra la propuesta por 0.12 unidades y su variabilidad es similar (la diferencia es de 0.005 unidades).

Por último, como se mencionó las condiciones de trabajo en cuanto al tiempo de procesamiento están limitadas por las características del hardware utilizado sin embargo es posible hacer eficiente el procesamiento de la información a través de una distribución paralela al lanzar conjuntos de datos con la ayuda de $cuda - cores$ en una tarjeta gráfica. De esta manera quedaría subsanado el inconveniente del tiempo de procesamiento.

Capítulo 6

Conclusiones

En este trabajo se presenta una topología RDB distinta para inferir las probabilidades de cambio de trayectoria con respecto a la información obtenida en vídeo mediante el modelado de las características espacio-temporales del movimiento de los objetos detectados.

Se realizó una comparativa con trabajos relacionados para el análisis del desempeño del funcionamiento global del algoritmo así como del desempeño por etapas con bases de datos de acceso libre. Se logró adquirir información propia de ambientes vehiculares con características desafiantes para realizar las pruebas y obtener resultados para evaluar el algoritmo propuesto.

La métricas de evaluación MOTA y MOTP corresponden a la etapa de seguimiento que si bien son importantes no deberían tomarse como la métrica fundamental para avalar el rendimiento de la estimación de trayectorias. En este sentido se realiza la comparativa de la propuesta vs algoritmos de aprendizaje profundo por considerarse que actualmente las redes neuronales convolucionales tienen alto impacto y aún mejores perspectivas de desarrollo a futuro.

Los experimentos realizados proporcionaron datos relevantes sobre el rendimiento de la implementación del algoritmo propuesto, así como características importantes de la topología RDB diseñada. En concreto, se puede mencionar que cualitativamente la RDB es capaz de determinar los cambios de trayectoria, cuantitativamente la RDB obtiene parámetros de probabilidad de colisión normalizados que diferencian los objetos con riesgo de colisión de aquellos que no.

La percepción del ambiente con sistemas estéreo y aplicando técnicas en este mismo sentido demuestran rendimiento adecuado e integración con técnicas de otra índole para llevara a cabo la tarea de seguimiento y estimación de desplazamiento.

Se propuso el submuestreo de la información de las bases de datos para analizar las carac-

terísticas a mejorar en las condiciones de las relaciones causales para mejorar el rendimiento de la estimación de dirección. El comportamiento del modelo del algoritmo propuesto no tiene una caída abrupta en el rendimiento dadas las condiciones del caso promedio, el peor caso o el mejor caso. Se puede valorar que la propuesta tiene consistencia de funcionamiento.

La propuesta presenta cualitativamente resultados con variación sútil y consistente cercano a lo que infiere un conductor humano (cambios ascendentes o descendentes no abruptos). De forma cuantitativa se determinan las probabilidades de cambio de dirección con un valor límite superior de 84 % de probabilidad, esto se debe al hecho de que la propuesta divide las relaciones causales de la inferencia de la trayectoria no sólo en los vectores de dirección, sino que también tiene en cuenta el riesgo de colisión asociado a cada cambio de dirección. Por lo tanto, el método propuesto determina una probabilidad $\geq 85\,\%$, implica un resultado similar al intervalo 80 - 90 % encontrado en la literatura.

Los módulos-etapas que conforman al algoritmo propuesto tienen una integración novedosa ya que en la literatura no se encontró el uso e implementación como lo presentado en este trabajo de investigación.

La propuesta de la topología de la RDB si bien requiere de información preliminar para construir las relaciones causales del modelo de estimación trayectorias tiene la ventaja de no requerir tanta información como las técnicas basadas en el aprendizaje profundo.

En los trabajos relacionados se presentan estimación de trayectorias con respecto al vector de desplazamiento con información obtenida de sensores lidar, en el caso de la propuesta la percepción del ambiente se realiza únicamente por medio de cámaras de video y un dispositivo GPS.

Lo propuesto es mejorable, ya que a pesar de no realizar cambios tan bruscos en los vectores de dirección, la estimación de la siguiente posición respecto al GT puede seguir siendo suavizada, por lo que es importante analizar y en su caso ampliar las relaciones causales de las variables en la topología de la RDB.

6.1. Trabajo a futuro

Como trabajo futuro, se propone analizar y complementar el enfoque presentado respecto a la inferencia de trayectorias en un entorno vehicular obteniendo una mejor tasa de error respecto a la localización espacial (distancia de los objetos respecto al ego-vehículo), así como aumentar el número de vectores de dirección a estimar (> 3).

Evaluar la posibilidad de incrementar las variables relacionadas con la percepción del ambiente para aumentar las relaciones causales y por lo tanto hacer más robusta la topología

de la RDB para verificar si se puede mejorar el rendimiento de las estimaciones realizadas.

De igual forma se plantea utilizar un enfoque híbrido entre modelos de RDB y modelos de aprendizaje profundo para aprovechar las ventajas de las técnicas de las redes neuronales convolucionales y verificar si el enfoque híbrido puede tener mejor rendimiento contra los resultados obtenidos en el trabajo desarrollado en esta investigación.

Finalmente, como trabajo futuro complementario, se plantea la mejora del tiempo de ejecución con hardware más potente que el que se está utilizando. En este sentido, se debe experimentar el enfoque propuesto en casos de tiempo real, es decir con la ayuda de hardware con mejores capacidades y metodologías en enfoque paralelo o de tensores se pueda replantear la etapa de mapeo de disparidad asi como la etapa de inferencia de trayectorias ya que presentan menor rendimiento en el tiempo.

6.2. Publicaciones

Los artículos publicados que se mencionan a continuación corresponden al producto del trabajo de investigación realizado durante el periodo de estancia en el programa doctoral.

- Reyes-Cocoletzi L., Olmos-Pineda I., Olvera-López J.A. (2019) *Detección de obstáculos móviles aplicada a la conducción autónoma vehicular*. Komputer Sapiens, Vol 2 (11), pp. 62-66.
- Reyes-Cocoletzi L., Olmos-Pineda I., Olvera-López J.A. (2019) *Obstacle Detection and Trajectory Estimation in Vehicular Displacements based on Computational Vision*. Research in Computing Science Vol. 148(9), pp. 57-70.
- Reyes-Cocoletzi L., Olmos-Pineda I., Olvera-López J.A. (2021) *Estimación de Trayectorias Vehiculares por Medio de Inferencias en Redes Dinámicas Bayesianas*. United Academic Journal, AILCIHM Vol. (1), pp. 5-14.
- Reyes-Cocoletzi L., Olmos-Pineda I., Olvera-López J.A. (2022) *Motion Estimation in Vehicular Environments based on Bayesian Dynamic Networks*. Journal of Intelligent & Fuzzy Systems, pre-press, pp. 1-12, 2021.

Bibliografía

Asljung, D., Westlund, M., y Fredriksson, J. (2019). A probabilistic framework for collision probability estimation and an analysis of the discretization precision. En *2019 IEEE intelligent vehicles symposium (iv)* (pp. 52–57).

Audi. (2018). *Audi automotive product information.* url http://www.audi.com.

Bechtel, M. G., McEllhiney, E., Kim, M., y Yun, H. (2018). Deeppicar: A low-cost deep neural network-based autonomous car. En *2018 ieee 24th international conference on embedded and real-time computing systems and applications (rtcsa)* (pp. 11–21).

Bernardin, K., y Stiefelhagen, R. (2008). Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, *2008*, 1–10.

Beuzen, T., Marshall, L., y Splinter, K. D. (2018). A comparison of methods for discretizing continuous variables in bayesian networks. *Environmental Modelling & Software*, *108*, 61–66.

Bojarski, M., Yeres, P., Choromanska, A., Choromanski, K., Firner, B., Jackel, L., y Muller, U. (2017). Explaining how a deep neural network trained with end-to-end learning steers a car. *arXiv e-prints*, arXiv–1704.

Bresson, G., Alsayed, Z., Yu, L., y Glaser, S. (2017). Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, *2*(3), 194–220.

Broggi, A., Cattani, S., Patander, M., y Sabbatelli, M. (2013). A full-3d voxel-based dynamic obstacle detection for urban scenario using stereo vision. En *Intelligent transportation systems-(itsc), 2013 16th international IEEE conference on* (pp. 71–76).

Cai, B., Liu, Y., Liu, Z., Chang, Y., y Jiang, L. (2020). A multiphase dynamic bayesian network methodology for the determination of safety integrity levels. En *Bayesian networks for reliability engineering* (pp. 217–237). Springer.

Cao, J., Song, C., Peng, S., Song, S., Zhang, X., y Xiao, F. (2020). Trajectory tracking control algorithm for autonomous vehicle considering cornering characteristics. *IEEE Access*, *8*, 59470–59484.

Capetillo Vázquez, C. A. (2021). *Correlación del espectro de absorción de filtros de radiación azul con la agudeza visual, discriminación al color, sensibilidad al contraste y estereopsis*. Master Thesis: Universidad Autónoma de Aguascalientes.

Cappé, O., y Moulines, E. (2009). On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *71*(3), 593–613.

Chandra, R., Bhattacharya, U., Bera, A., y Manocha, D. (2019). Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions. En *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8483–8492).

Chen, L., Ai, H., Shang, C., Zhuang, Z., y Bai, B. (2017). Online multi-object tracking with convolutional neural networks. En *2017 IEEE international conference on image processing (icip)* (pp. 645–649).

Chu, P., Fan, H., Tan, C. C., y Ling, H. (2019). Online multi-object tracking with instance-aware tracker and dynamic model refreshment. En *2019 IEEE winter conference on applications of computer vision (wacv)* (pp. 161–170).

CityCar. (2022). *City car driving, automotive product information: simulator*. url https://citycardriving.com/ .

Cover, T. M. (1999). *Elements of information theory*. John Wiley & Sons.

Cui, H., Radosavljevic, V., Chou, F.-C., Lin, T.-H., Nguyen, T., Huang, T.-K., ... Djuric, N. (2019). Multimodal trajectory predictions for autonomous driving using deep convolutional networks. En *2019 international conference on robotics and automation (icra)* (pp. 2090–2096).

Deo, N., Rangesh, A., y Trivedi, M. M. (2018). How would surround vehicles move? a unified framework for maneuver classification and motion prediction. *IEEE Transactions on Intelligent Vehicles*, *3*(2), 129–140.

Du, Y., Yan, Y., Chen, S., y Hua, Y. (2020). Object-adaptive lstm network for real-time visual tracking with adversarial data augmentation. *Neurocomputing*, *384*, 67–83.

Duan, J., Li, S. E., Guan, Y., Sun, Q., y Cheng, B. (2020). Hierarchical reinforcement learning for self-driving decision-making without reliance on labelled driving data. *IET Intelligent Transport Systems*, *14*(5), 297–305.

Eraqi, H. M., Moustafa, M. N., y Honer, J. (2017). End-to-end deep learning for steering autonomous vehicles considering temporal dependencies. *arXiv e-prints*, arXiv–1710.

Fagnant, D. J., y Kockelman, K. (2015). Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, *77*, 167–181.

Fan, R. (2018). *Real-time computer stereo vision for automotive applications* (Tesis Doctoral no publicada). University of Bristol.

Fan, R., Ai, X., y Dahnoun, N. (2018). Road surface 3d reconstruction based on dense subpixel disparity map estimation. *IEEE Transactions on Image Processing*, *27*(6), 3025–3035.

Fan, R., y Dahnoun, N. (2017). Real-time implementation of stereo vision based on optimised normalised cross-correlation and propagated search range on a gpu. En *2017 IEEE international conference on imaging systems and techniques (ist)* (pp. 1–6).

Felzenszwalb, P. F., y Huttenlocher, D. P. (2006). Efficient belief propagation for early vision. *International journal of computer vision*, *70*(1), 41–54.

Fernando, T., Denman, S., Sridharan, S., y Fookes, C. (2018). Gd-gan: Generative adversarial networks for trajectory prediction and group detection in crowds. En *Asian conference on computer vision* (pp. 314–330).

Gidaris, S., y Komodakis, N. (2015). Object detection via a multi-region and semantic segmentation-aware cnn model. En *Proceedings of the IEEE international conference on computer vision* (pp. 1134–1142).

Göhring, D. (2012). Controller architecture for the autonomous cars: Madeingermany and e-instein.

González, D. S., Garzón, M., Dibangoye, J. S., y Laugier, C. (2019). Human-like decision-making for automated driving in highways. En *2019 IEEE intelligent transportation systems conference (itsc)* (pp. 2087–2094).

Grigorescu, S., Trasnea, B., Cocias, T., y Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, *37*(3), 362–386.

Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., y Alahi, A. (2018). Social gan: Socially acceptable trajectories with generative adversarial networks. En *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2255–2264).

Hecker, S., Dai, D., y Van Gool, L. (2018). End-to-end learning of driving models with surround-view cameras and route planners. En *Proceedings of the european conference on computer vision (eccv)* (pp. 435–453).

Hoermann, S., Stumper, D., y Dietmayer, K. (2017a). Probabilistic long-term prediction for autonomous vehicles. En *2017 IEEE intelligent vehicles symposium (iv)* (pp. 237–243).

Hoermann, S., Stumper, D., y Dietmayer, K. (2017b). Probabilistic long-term prediction for autonomous vehicles. En *2017 IEEE intelligent vehicles symposium (iv)* (pp. 237–243).

Hou, G., Chen, S., y Chen, F. (2019). Framework of simulation-based vehicle safety performance assessment of highway system under hazardous driving conditions. *Transportation Research Part C: Emerging Technologies*, *105*, 23–36.

Hu, H.-N., Cai, Q.-Z., Wang, D., Lin, J., Sun, M., Krahenbuhl, P., ... Yu, F. (2019). Joint monocular 3d vehicle detection and tracking. En *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 5390–5399).

Jaritz, M., De Charette, R., Toromanoff, M., Perot, E., y Nashashibi, F. (2018). End-to-end race driving with deep reinforcement learning. En *2018 IEEE international conference on robotics and automation (icra)* (pp. 2070–2075).

Kampker, A., y Sefati, M. (2018). Towards multi-object detection and tracking in urban scenario under uncertainties. En *Vehits* (pp. 156–167).

Kampker, A., Sefati, M., Rachman, A. S. A., Kreisköther, K., y Campoy, P. (2018). Towards multi-object detection and tracking in urban scenario under uncertainties. En *Vehits* (pp. 156–167).

KITTI Benchmark. (2019). *www.cvlibs.net*.

Lai, W.-C., Xia, Z.-X., Lin, H.-S., Hsu, L.-F., Shuai, H.-H., Jhuo, I.-H., y Cheng, W.-H. (2020). Trajectory prediction in heterogeneous environment via attended ecology embedding. En *Proceedings of the 28th acm international conference on multimedia* (pp. 202–210).

Lai, Y.-K., Chou, Y.-H., y Schumann, T. (2017). Vehicle detection for forward collision warning system based on a cascade classifier using adaboost algorithm. En *Consumer electronics-berlin (icce-berlin), 2017 IEEE 7th international conference on* (pp. 47–48).

Lan, J., Jiang, Y., y Yu, D. (2015). A new automatic obstacle detection method based on selective updating of gaussian mixture model. En *Transportation information and safety (ictis), 2015 international conference on* (pp. 21–25).

Lee, D., Gu, Y., Hoang, J., y Marchetti-Bowick, M. (2019). Joint interaction and trajectory prediction for autonomous driving using graph neural networks. *arXiv e-prints*, arXiv–1912.

Lee, N., Choi, W., Vernaza, P., Choy, C. B., Torr, P. H., y Chandraker, M. (2017). Desire: Distant future prediction in dynamic scenes with interacting agents. En *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 336–345).

Leon, F., y Gavrilescu, M. (2021). A review of tracking and trajectory prediction methods for autonomous driving. *Mathematics*, *9*(6), 660.

Li, J., Yang, F., Tomizuka, M., y Choi, C. (2020). Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning. *arXiv e-prints*, arXiv–2003.

Li, S., Ma, B., Chang, H., Shan, S., y Chen, X. (2018). Continuity-discrimination convolutional neural network for visual object tracking. En *2018 IEEE international conference on multimedia and expo (icme)* (pp. 1–6).

Liang, J., Jiang, L., Murphy, K., Yu, T., y Hauptmann, A. (2020). The garden of forking paths:

Towards multi-future trajectory prediction. En *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10508–10518).

Likforman-Sulem, L., y Sigelle, M. (2008). Recognition of degraded characters using dynamic bayesian networks. *Pattern Recognition*, *41*(10), 3092–3103.

Lin, C., Li, Y., Xu, G., y Cao, Y. (2017). Optimizing zncc calculation in binocular stereo matching. *Signal Processing: Image Communication*, *52*, 64–73.

Liu, J., Xiong, H., Wang, T., Huang, H., Zhong, Z., y Luo, Y. (2020). Probabilistic vehicle trajectory prediction via driver characteristic and intention estimation model under uncertainty. *Industrial Robot: the international journal of robotics research and application*.

Liu, Q., Lu, X., He, Z., Zhang, C., y Chen, W.-S. (2017). Deep convolutional neural networks for thermal infrared object tracking. *Knowledge-Based Systems*, *134*, 189–198.

Lu, K., An, X., Li, J., y He, H. (2017). Efficient deep network for vision-based object detection in robotic applications. *Neurocomputing*, *245*, 31–45.

Lu, Z., Wang, J., Li, Z., Chen, S., y Wu, F. (2021). A resource-efficient pipelined architecture for real-time semi-global stereo matching. *IEEE Transactions on Circuits and Systems for Video Technology*.

Luiten, J., Fischer, T., y Leibe, B. (2020). Track to reconstruct and reconstruct to track. *IEEE Robotics and Automation Letters*, *5*(2), 1803–1810.

Madrid Sánchez, A. (2018). *Desarrollo e implementación óptica y computacional para la generación de hologramas sintéticos de imágenes 3d reales* (Tesis Doctoral no publicada). Universidad EAFIT.

Mandal, S., Biswas, S., Balas, V. E., Shaw, R. N., y Ghosh, A. (2020). Motion prediction for autonomous vehicles from lyft dataset using deep learning. En *2020 IEEE 5th international conference on computing communication and automation (iccca)* (pp. 768–773).

Mane, S. B., y Vhanale, S. (2016). Real time obstacle detection for mobile robot navigation using stereo vision. En *2016 international conference on computing, analytics and security trends (cast)* (pp. 637–642).

Mangalam, K., Girase, H., Agarwal, S., Lee, K.-H., Adeli, E., Malik, J., y Gaidon, A. (2020). It is not the journey but the destination: Endpoint conditioned trajectory prediction. En *European conference on computer vision* (pp. 759–776).

Marina, L. A., Trasnea, B., Cocias, T., Vasilcoi, A., Moldoveanu, F., y Grigorescu, S. M. (2019). Deep grid net (dgn): A deep learning system for real-time driving context understanding. En *2019 third IEEE international conference on robotic computing (irc)* (pp. 399–402).

Meng, F., Wang, X., Wang, D., Shao, F., y Fu, L. (2020). Spatial–semantic and temporal attention mechanism-based online multi-object tracking. *Sensors*, *20*(6), 1653.

Mercedes. (2018). *Mercedes benz automotive product information.* url https://www.mercedes-benz.com.du.

Messaoud, K., Deo, N., Trivedi, M. M., y Nashashibi, F. (2020). Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation. *arXiv preprint arXiv:2005.02545*.

Mo, X., Xing, Y., y Lv, C. (2020). Interaction-aware trajectory prediction of connected vehicles using cnn-lstm networks. En *Iecon 2020 the 46th annual conference of the IEEE industrial electronics society* (pp. 5057–5062).

Mukherjee, A., Adarsh, S., y Ramachandran, K. (2021). Ros-based pedestrian detection and distance estimation algorithm using stereo vision, leddar and cnn. En *Intelligent system design* (pp. 117–127). Springer.

Mukhtar, A., Xia, L., y Tang, T. B. (2015). Vehicle detection techniques for collision avoidance systems: A review. *IEEE Trans. Intelligent Transportation Systems*, *16*(5), 2318–2338.

Murphy, K. P. (2002). *Dynamic bayesian networks: representation, inference and learning*. University of California, Berkeley.

Nguyen, V. D., y Nguyen, T. (2013). A fast evolutionary algorithm for real-time vehicle detection. *IEEE Transactions on Vehicular Technology*, *62*(6), 2453–2468.

Nojavan, F., Qian, S. S., y Stow, C. A. (2017). Comparative analysis of discretization methods in bayesian networks. *Environmental Modelling & Software*, *87*, 64–71.

Pan, Y., Cheng, C.-A., Saigol, K., Lee, K., Yan, X., Theodorou, E., y Boots, B. (2017). Agile autonomous driving using end-to-end deep imitation learning. *arXiv e-prints*, arXiv–1709.

Phan-Minh, T., y Grigore, E. C. (2020). Covernet: Multimodal behavior prediction using trajectory sets. En *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14074–14083).

Poggio, G. F., y Poggio, T. (1984). The analysis of stereopsis. *Annual review of neuroscience*, *7*(1), 379–412.

Ponz, A., Rodríguez-Garavito, C., García, F., Lenz, P., Stiller, C., y Armingol, J. (2015). Laser scanner and camera fusion for automatic obstacle classification in adas application. En *Smart cities, green technologies, and intelligent transport systems* (pp. 237–249). Springer.

Prabhakar, G., Kailath, B., Natarajan, S., y Kumar, R. (2017). Obstacle detection and classification using deep learning for tracking in high-speed autonomous driving. En *IEEE*

region 10 symposium (tensymp), 2017 (pp. 1–6).

Rausch, V., Hansen, A., Solowjow, E., Liu, C., Kreuzer, E., y Hedrick, J. K. (2017). Learning a deep neural net policy for end-to-end control of autonomous vehicles. En *2017 american control conference (acc)* (pp. 4914–4919).

Redmon, J., Divvala, S., Girshick, R., y Farhadi, A. (2016). You only look once: Unified, real-time object detection. En *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779–788).

Ren, L., Lu, J., Wang, Z., Tian, Q., y Zhou, J. (2018). Collaborative deep reinforcement learning for multi-object tracking. En *Proceedings of the european conference on computer vision (eccv)* (pp. 586–602).

Ristani, E., Solera, F., Zou, R., Cucchiara, R., y Tomasi, C. (2016). Performance measures and a data set for multi-target, multi-camera tracking. En *European conference on computer vision* (pp. 17–35).

Ristani, E., y Tomasi, C. (2018). Features for multi-target multi-camera tracking and re-identification. En *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6036–6046).

Rödel, C., Stadler, S., Meschtscherjakov, A., y Tscheligi, M. (2014). Towards autonomous cars: the effect of autonomy levels on acceptance and user experience. En *Proceedings of the 6th international conference on automotive user interfaces and interactive vehicular applications* (pp. 1–8).

Roggeman, H., Marzat, J., Derome, M., Sanfourche, M., Eudes, A., y Le Besnerais, G. (2017a). Detection, estimation and avoidance of mobile objects using stereo-vision and model predictive control. En *Iccv workshop uavision2017*.

Roggeman, H., Marzat, J., Derome, M., Sanfourche, M., Eudes, A., y Le Besnerais, G. (2017b). Detection, estimation and avoidance of mobile objects using stereo-vision and model predictive control. En *Proceedings of the IEEE international conference on computer vision* (pp. 2090–2099).

Sadjadi, F., y Ribnick, E. (2010). Passive 3d sensing, and reconstruction using multi-view imaging. En *2010 IEEE computer society conference on computer vision and pattern recognition-workshops* (pp. 68–74).

Sallab, A. E., Abdou, M., Perot, E., y Yogamani, S. (2017). Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, *2017*(19), 70–76.

Salzmann, T., Ivanovic, B., Chakravarty, P., y Pavone, M. (2020). Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. En *Computer vision–eccv 2020: 16th european conference, glasgow, uk, august 23–28, 2020, proceedings, part xviii 16* (pp.

683–700).

Sangineto, E., y Nabi, M. (2019). Self paced deep learning for weakly supervised object detection. *IEEE transactions on pattern analysis and machine intelligence, 41*(3), 712–725.

Sangineto, E., Nabi, M., Culibrk, D., y Sebe, N. (2019). Self paced deep learning for weakly supervised object detection. *IEEE transactions on pattern analysis and machine intelligence*, *41*(3), 712–725.

Scharstein, D., y Szeliski, R. (2002). *Middlebury stereo vision page*.

Schulz, J., Hubmann, C., Löchner, J., y Burschka, D. (2018). Multiple model unscented kalman filtering in dynamic bayesian networks for intention estimation and trajectory prediction. En *2018 21st international conference on intelligent transportation systems (itsc)* (pp. 1467–1474).

Schulz, J., Hubmann, C., Morin, N., Löchner, J., y Burschka, D. (2019). Learning interaction-aware probabilistic driver behavior models from urban scenarios. En *2019 IEEE intelligent vehicles symposium (iv)* (pp. 1326–1333).

Schwarting, W., Alonso-Mora, J., y Rus, D. (2018). Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, *1*, 187–210.

Shabanian, H., y Balasubramanian, M. (2021). A novel factor graph-based optimization technique for stereo correspondence estimation. *arXiv e-prints*, arXiv–2109.

Song, L., Kolar, M., y Xing, E. (2009). Time-varying dynamic bayesian networks. *Advances in neural information processing systems*, *22*, 1732–1740.

Song, W., y Yang, Y. (2018). Real-time obstacles detection and status classification for collision warning in a vehicle active safety system. *IEEE transactions on intelligent transportation systems*, *19*(3), 758–773.

StereoLabs. (2021). *zed camera features and specifications*. url http:///www.stereolabs.com/zed/ .

Sucar, L. E. (2015). Probabilistic graphical models. *Advances in Computer Vision and Pattern Recognition. London: Springer London. doi*, *10*(978), 1.

Sun, C., Karlsson, P., Wu, J., Tenenbaum, J. B., y Murphy, K. (2019). Stochastic prediction of multi-agent interactions from partial observations. *arXiv e-prints*, arXiv–1902.

Sun, L., Peng, C., Zhan, W., y Tomizuka, M. (2018). A fast integrated planning and control framework for autonomous driving via imitation learning. En *Dynamic systems and control conference* (Vol. 51913, p. V003T37A012).

Sun, L., Zhan, W., Wang, D., y Tomizuka, M. (2019). Interactive prediction for multiple,

heterogeneous traffic participants with multi-agent hybrid dynamic bayesian network. En *2019 IEEE intelligent transportation systems conference (itsc)* (pp. 1025–1031).

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., ... others (2020). Scalability in perception for autonomous driving: Waymo open dataset. En *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2446–2454).

Suraj, M., Grimmett, H., Platinskỳ, L., y Ondrúŝka, P. (2018). Predicting trajectories of vehicles using large-scale motion priors. En *2018 IEEE intelligent vehicles symposium (iv)* (pp. 1639–1644).

Tan, M., Pang, R., y Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. En *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 10781–10790).

Tesla. (2018). *Tesla motors, automotive product information: autopilot model 3s.* url https://www.tesla.com/es_MX/model3 .

Tran, Q., y Firl, J. (2014). Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression. En *2014 IEEE intelligent vehicles symposium proceedings* (pp. 918–923).

Velodyne. (2020). *Velodyne features and specifications.* url https://velodynelidar.com/ .

Vision, K. (2019). *Karlsruche institute of technology vision benchmark.* url http://www.cvlibs.net/datasets/kitti/ .

Wan, K.-W., Li, Z., y Yau, W.-Y. (2017). Learning visual odometry for unmanned aerial vehicles. En *Signal and image processing (icsip), 2017 IEEE 2nd international conference on* (pp. 316–320).

Wang, C., Galoogahi, H. K., Lin, C.-H., y Lucey, S. (2018). Deep-lk for efficient adaptive object tracking. En *2018 IEEE international conference on robotics and automation (icra)* (pp. 627–634).

Wang, J.-G., Zhou, L., Pan, Y., Lee, S., Song, Z., Han, B. S., y Saputra, V. B. (2016). Appearance-based brake-lights recognition using deep learning and vehicle detection. En *2016 IEEE intelligent vehicles symposium (iv)* (pp. 815–820).

Wang, W., y Neumann, U. (2018). Depth-aware cnn for rgb-d segmentation. En *Proceedings of the european conference on computer vision (eccv)* (pp. 135–150).

Warren, M. E. (2019). Automotive lidar technology. En *2019 symposium on vlsi circuits* (pp. C254–C255).

Waymo. (2021). *Waymo, driver, lidar.* url https://waymo.com/intl/es/waymo-driver/ .

Wen, L., Du, D., Li, S., Bian, X., y Lyu, S. (2019). Learning non-uniform hypergraph for multi-object tracking. En *Proceedings of the aaai conference on artificial intelligence* (Vol. 33,

pp. 8981–8988).

Weng, X., Wang, J., Held, D., y Kitani, K. (2020). 3d multi-object tracking: A baseline and new evaluation metrics. En *2020 IEEE/RSJ international conference on intelligent robots and systems (iros)* (pp. 10359–10366).

Wu, H., Han, W., Wen, C., Li, X., y Wang, C. (2021). 3d multi-object tracking in point clouds based on prediction confidence-guided data association. *IEEE Transactions on Intelligent Transportation Systems*.

Xie, G., Gao, H., Huang, B., Qian, L., y Wang, J. (2018). A driving behavior awareness model based on a dynamic bayesian network and distributed genetic algorithm. *International Journal of Computational Intelligence Systems*, *11*(1), 469–482.

Xie, Y., Zeng, S., Zhang, Y., y Chen, L. (2017). A cascaded framework for robust traversable region estimation using stereo vision. En *2017 chinese automation congress (cac)* (pp. 3075–3080).

Xu, C., Zhao, W., y Wang, C. (2019). An integrated threat assessment algorithm for decision-making of autonomous driving vehicles. *IEEE transactions on intelligent transportation systems*, *21*(6), 2510–2521.

Xu, H., Gao, Y., Yu, F., y Darrell, T. (2017). End-to-end learning of driving models from large-scale video datasets. En *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2174–2182).

Xu, Y., Ban, Y., Alameda-Pineda, X., y Horaud, R. (2019). Deepmot: a differentiable framework for training multiple object trackers. *e-print arXiv:1906*.

Xu, Y., Zhao, T., Baker, C., Zhao, Y., y Wu, Y. N. (2019). Learning trajectory prediction with continuous inverse optimal control via langevin sampling of energy-based models. *e-print arXiv:1904*.

Yang, Y., y Webb, G. I. (2002). A comparative study of discretization methods for naive-bayes classifiers. En *Proceedings of pkaw* (Vol. 2002).

Zhang, Y., Qiu, Z., Yao, T., Liu, D., y Mei, T. (2018). Fully convolutional adaptation networks for semantic segmentation. En *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6810–6818).

Zhu, H., Yuen, K.-V., Mihaylova, L., y Leung, H. (2017). Overview of environment perception for intelligent vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *18*(10), 2584–2601.

Zou, X., Sun, B., Zhao, D., Zhu, Z., Zhao, J., y He, Y. (2020). Multi-modal pedestrian trajectory prediction for edge agents based on spatial-temporal graph. *IEEE Access*, *8*, 83321–83332.