



TESIS DOCTORAL

Sistema Avanzado de Asistencia a la Conducción para Entornos Interurbanos

Autor:

Cesar Hernán Rodríguez Garavito

Directores:

Arturo de La Escalera Hueso

José María Armingol

Tutor:

Arturo de La Escalera Hueso

**DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y
AUTOMÁTICA**

Leganes, Febrero de 2017

TESIS DOCTORAL

Sistema Avanzado de Asistencia a la Conducción para Entornos Interurbanos

Autor: *Cesar Hernán Rodríguez Garavito*

Directores: **Arturo de La Escalera Hueso**
José María Armingol

Firma del Tribunal Calificador:

Firma

Presidente:

Vocal:

Secretario:

Calificación:

Leganés, de de

¡Callado abuelo! También tú viviste y amaste. Por eso moras junto a tus hijos como inmortal. Y a veces, la vida parece venir de ti, como del Éter silencioso, descendiendo sobre esta casa, varón apacible, que ve crecer, madurar y dignificarse año tras año esta felicidad modesta que tú plantaste lleno de esperanza.

Juan Cristóbal Federico Hölderlin .

A la memoria de mi Padre Mateo.

1941-∞

Agradecimientos

Estas líneas son el reconocimiento a todos aquellos que prepararon y sembraron el terreno durante estos últimos años para que este trabajo pudiese dar fruto. Ellos fueron el motor que dio impulso a esta investigación, a través de la motivación y el apoyo que me brindaron, por lo tanto, son coautores fundamentales de este sueño hecho realidad.

Ante todo quisiera dar gracias a Dios, porque siempre he reconocido su existencia, presencia e inspiración para orientar mi vida. Agradezco a mi Padre Mateo, que ya partió a su encuentro, por el ejemplo de vida que nos brindó, muchas mañanas cuando no había motivo para seguir, su recuerdo cumpliendo el deber durante cada día, me animó a seguir. Agradezco a mi Madre, Rosita, porque siempre me ha apoyado en todos los proyectos que he emprendido con un entusiasmo fundado en el amor. Agradezco también a mis hermanas, Pily y Nana, porque con la razón y el corazón, han sido mis consejeras de vida. Agradezco a mis sobrinas Sofy, Alejita y Vale, porque con una palabra o una imagen suya, fueron mi más dulce hacedero de esperanza y recuerdo claro de para que hice un doctorado. Agradezco a mis directores, Arturo y José María, por haber dado un voto de confianza en alguien desconocido que un día tocó a sus puertas. Ellos han sido una guía luminosa que con una opinión certera señalaron el camino. Agradezco a mi compañera de vida, Hasbleidy, quien con su pasión por el trabajo y su ternura, trajeron a mi vida la inspiración necesaria para realizar la tarea más difícil de todas, terminar la *Tesis*. Finalmente, agradezco a mis compañeros de laboratorio *LSI*, a mis amigos en Madrid y en Bogotá y donde quiera que se encuentren, por haber compartido un momento de sus vidas y por haber dejado una parte de lo mejor de sí mismos en mí. A todos los estudiantes que he tenido y que vendrán, les agradezco por permitir dar razón a mi paso por esta bella realidad.

Resumen

Los Sistemas de Asistencia a la Conducción, conocidos como ADAS (*Advanced Driver Assistant Systems*), desde su aparición en el año 2000, han permitido que las cifras de muertes en accidentes de tráfico disminuyan hasta 8.7 por cada 100.000 habitantes, según la OMS (*Organización Mundial de la Salud*), en su informe anual 2013. Esta tendencia se ha dado en economías de alto ingreso a pesar del crecimiento sostenido en las últimas dos décadas del parque automotor a nivel mundial. Algunos de los dispositivos ADAS más conocidos como la alerta por salida de carril, LDW (*Lane Departure Warning*), el asistente para mantenimiento de carril, LKA (*Lane Keeping Assistance*) o el asistente para cambio de carril, LCA (*Lane Change Assistance*), han hecho de la conducción una tarea cada vez más segura.

La presente investigación doctoral propone un sistema ADAS diseñado para entornos de carretera tipo autopista interurbana, que clasifica carriles según la norma Española IC 8.2. La información visual se obtiene por medio de una cámara estéreo, con la cual es posible auto-calibrar los parámetros extrínsecos en tiempo de aplicación, para cambiar la perspectiva de las imágenes a vista superior o *bird view*, alcanzando una precisión de décimas de grado en orientación y milésimas de metro en posición vertical. El funcionamiento del ADAS propuesto para la plataforma IV-VI 2.0 (*Intelligent Vehicle Based on Visual Information*) del laboratorio de Sistemas Inteligentes de la Universidad Carlos III, se compone de cuatro fases. En la primera fase se simplifica la imagen filtrando los objetos que cumplen las características propias de las marcas viales, restringiendo la búsqueda a la zona de la imagen coincidente con el plano tierra. En la segunda fase, el procesamiento continúa con la detección de los elementos constitutivos de la carretera, las líneas de carril y los correspondientes carriles que se forman con ellas. Posteriormente, en la tercera fase se realiza la detección parcial o total de la carretera, que corresponde con la búsqueda de carriles adyacentes en el espacio. En la última fase, con las detecciones parciales o totales de carretera, formadas por una cadena de carriles, se aplica un proceso de

casamiento carril a carril, para actualizar en el tiempo una cadena final completa, que tendrá la cualidad de rechazar oclusiones y errores por sombras y cambios de iluminación.

Finalmente, se aborda el estudio de cómo aplicar aprendizaje de máquina en la estructura ADAS desarrollada, logrando buenos resultados en la segmentación de marcas viales sobre carreteras donde las líneas se han degradado visualmente, en esta fase, sobre una secuencia de referencia de 75 imágenes, se observó cómo con el uso de un clasificador basado en máquinas de soporte vectorial, SVM (*Support Vector Machine*), logró la segmentación de marcas viales con un índice de sensibilidad de 70 %, 44 % por encima del siguiente mejor método analizado. Así mismo, al integrar la estrategia SVM para detección de marcas viales en el algoritmo general ADAS propuesto, se obtuvo una mejora en la detección de líneas viales hasta de un 83 % en relación al 77 % obtenido en la versión desarrollada inicialmente sin aprendizaje de máquina.

Abstract

The Advanced Driver Assistant Systems, *ADAS*, have allowed to reduce deaths on the road up to 8.7 per 100.000 inhabitants since they were presented at the first time in the year 2000, according to World Health Organization, *WHO*, in its Global status report on road safety 2013. This is a trend that is characteristic of the high-income economies beside the continuous automotive fleet growing in the last two decades. Some of the most well-known *ADAS* applications, such as *LDW* (*Lane Departure Warning*), *LKA* (*Lane Keeping Assistance*) or *LCA* (*Lane Change Assistance*), have made driving a safer task.

This research proposes an *ADAS* application designed for interurban highways, classifying lanes according to the Spain standard IC 8.2. The visual information is captured from stereo camera, this camera retrieves 3D information from the visual pattern. This makes it possible for the application to be self-tuning by automatically calculating its extrinsic parameters up to tenths of a degree accuracy for orientation and millimeters for camera's height. This process allows to change the perspective of the images to bird view in application time. The proposed *ADAS* was designed for the platform *IVVI 2.0* (*Intelligent Vehicle Based on Visual Information*) from Intelligent Systems Lab at Carlos III University, is carried out in four stages. In the first one, the image is simplified by filtering objects that meet the road markings features, limiting the search area to the ground plane. In the second stage, the detection of the basic elements of the road is made, i.e. the road lines and the lanes formed by them. Subsequently, the partial or total road detection is performed by the search of adjacent lanes. In the last stage, a matching process is applied lane by lane, updating in real time a composed final chain, the complete lane chain has the ability to discard errors caused by occlusions and mismatches due to shadows and lighting changes.

Finally, feasibility study is presented for machine learning usability in the *ADAS* application developed, achieving good results in the segmentation of road markings

on highways where lines have been deteriorated and in the classification of lane's lines. The road mark segmentation results, obtained over a sequence of 75 images and applying SVM (*Support Vector Machine*) classifier, exhibit a sensibility rate of 70 %, 44 % above the next better method analyzed. Furthermore, by integrating the SVM strategy to detection of road markings in the proposed general algorithm, an improvement in the road lines detection was obtained, reaching to 83 %, considerable higher than the 77 % obtained by the algorithm without Machine Learning.

Índice general

Índice de figuras	XV
Índice de cuadros	XXII
1. Introducción	1
1.1. Motivación de la Tesis	1
1.2. Objetivos	5
1.2.1. Objetivo Principal	5
1.2.2. Objetivos Específicos	5
1.3. Esquema del Documento	6
2. Estado del Arte	9
2.1. Seguridad en la Conducción	9
2.2. Conducción Autónoma	11
2.3. Esquema General de un Sistema para Detección de Carril/Carretera basado en Visión	12
2.3.1. Extracción de características	12
2.3.2. Ajuste de modelo de carretera	15
2.3.3. Integración temporal	19
2.4. Hitos del Desarrollo ADAS	20
2.4.1. Trabajos Clásicos	20
2.4.2. Trabajos Recientes ADAS	22
2.5. Conclusiones	29
3. Reconstrucción Tridimensional de la Carretera	33
3.1. Visión Estéreo	33
3.2. Auto-calibración de parámetros extrínsecos	36
3.2.1. Estimación del plano de la carretera	37

3.2.2.	Extracción de extrínsecos	43
3.2.3.	Análisis de estabilidad de extrínsecos	45
4.	Modelado y Segmentación de la Carretera	61
4.1.	Máscara de Marcas Viales	62
4.1.1.	Detección de marcas viales en el espacio del gradiente	63
4.1.2.	Medición cuantitativa de la segmentación para marcas viales	67
4.2.	Máscara de Espacio libre	73
4.3.	Detección de Líneas y Carriles Viales	77
4.3.1.	Detección del ángulo de la carretera.	81
4.3.2.	Identificación de Conjuntos Adyacentes de Carriles	85
4.4.	Integración Temporal	87
4.5.	Rendimiento del Algoritmo para Detección de Carreteras	94
5.	Aprendizaje de Máquina para Clasificación de Carriles	103
5.1.	Fundamentos del aprendizaje de máquina	104
5.2.	Generación máscara de marcas viales	107
5.2.1.	Descriptor de marcas viales	108
5.2.2.	Mapeado en espacio de características	108
5.2.3.	Detección de marcas viales basada en clasificación	109
5.2.4.	Detección de Carreteras usando Máquinas de soporte vectorial	119
5.3.	Clasificación de líneas	121
5.3.1.	Descriptor de líneas viales	122
5.3.2.	Mapeado en espacio de características	122
5.3.3.	Clasificador de líneas viales y sintonización de parámetros	125
6.	Conclusiones y Trabajo Futuro	129
6.1.	Reconstrucción Tridimensional de la Carretera	129
6.2.	Modelado y Detección de la Carretera	130
6.3.	Aprendizaje de máquina en Clasificación de Carriles	131
6.4.	Trabajo Futuro	132
A.	Cambio de Perspectiva a Vista de Pájaro	136
A.1.	Introducción	136
A.2.	Homografía por Composición Matricial	137
B.	Elementos del Estándar IC 8.2	140

C. Clasificadores	143
C.1. Clasificador Bayesiano	144
C.2. Clasificador K-Vecinos más Próximos	145
C.3. Redes Neuronales	145
C.4. Máquinas de Soporte Vectorial	148
 Bibliografía	 150

Índice de figuras

1.1. Serie histórica del parque automotor Europeo. fuente: European Commission - Directorate general for energy and transport - CARE database	2
1.2. Estadísticas globales de población mundial, muertes en accidentes viales y vehículos motorizados registrados según el nivel de ingreso de los países. Fuente [77].	3
1.3. Serie histórica de personas fallecidas en accidentes de tránsito.	5
2.1. Algoritmo detección de marcas viales, Collado [18].	13
2.2. Diagrama general de la arquitectura VioLET. Fuente [68].	22
2.3. Diagrama general de la arquitectura Talos. Fuente [42].	25
2.4. Núcleo del filtro direccional propuesto por Huang et al, [42].	26
2.5. Algoritmo para la detección de marcas viales Huang et al [42].	27
2.6. Esquema general de una CNN. Fuente [2].	28
3.1. Cámara Estéreo XB3.	34
3.2. Reconstrucción tridimensional de la escena.	35
3.3. Escena de referencia para reducción 3D.	38
3.4. Reducción por muestreo uniforme.	38
3.5. Reducción por regularización.	39
3.6. Representación gráfica de los parámetros extraídos de la forma hesiana del plano de la carretera.	44
3.7. Secuencia estática capturada desde IVVI 2.0.	46
3.8. Comparación de métodos para estimación del plano de la carretera, en función de la resolución de muestreo.	47
3.9. Comparación de métodos para estimación del plano de la carretera, en función del tamaño de voxel utilizado en la regularización del entorno.	48
3.10. Secuencia de referencia Raw:Road:20112609_0015, tomada del KITTI.	49

3.11. Comparación de métodos para estimación del plano de la carretera con respecto a valores de referencia KITTI.	50
3.12. Comparación entre extrínsecos medidos por el método RANSAC @muestreo uniforme de 2000 puntos y valores de referencia KITTI, para la secuencia Road 2011_09_26_drive_0015. DTR (Desviación Típica respecto Referencia.)	52
3.13. Reducción del ruido presente en los extrínsecos, en una escena dinámica.	55
3.14. Sintonización del Filtro de Kalman para diferentes valores de desviación estándar en el ruido de proceso, σ_p y para diferentes valores de desviación estándar en el ruido de medida, σ_m	58
3.15. Reducción del ruido presente en los extrínsecos, en una escena dinámica usando Filtro de Kalman.	59
4.1. Algoritmo de Segmentación de carreteras según el estándar IC 8.2. . .	61
4.2. Fases del algoritmo para detección de marcas viales. Detección de bordes.	64
4.3. Algoritmo para detección de marcas viales. Restricciones que definen una marca vial. (a) Detalle gradiente de la imagen sobre los bordes detectados en una marca vial. Los puntos negros representan los píxeles detectados como bordes en la primera fase del algoritmo. (b) Descripción gráfica de los elementos que conforman las restricciones de una marca vial. $L_{j,i}$ y $\theta_{j,i}$, representan la magnitud y la dirección del gradiente en el punto $p_{j,i}$	65
4.4. Algoritmo detección de marcas viales. (a) Superposición de la máscara de marcas viales sobre la imagen de partida. (b) Máscara de marcas viales, \mathcal{RM} . (c) Centro de marcas viales.	66
4.6. Secuencia de referencia 1, “Ground Truth”. (a) Coloreado manual de las marcas viales en secuencia de referencia. (b) Máscara de marcas viales en secuencia de referencia.	68
4.5. Algunas imágenes de la secuencia de referencia (a) 1 y (b) 2, para la detección de marcas viales.	68
4.7. Comparación de índice de Sensibilidad entre variaciones del algoritmo para detección de marcas viales, sobre la secuencia 1 de referencia fig. 4.5a.	69

4.8. Comparación de índice de Sensibilidad entre variaciones del algoritmo para detección de marcas viales, sobre la secuencia 2 de referencia fig. 4.5b.	70
4.9. Comparación de índice de Precisión entre variaciones del algoritmo para detección de marcas viales, sobre la secuencia 1 de referencia fig. 4.5a.	70
4.10. Comparación de índice de Precisión entre variaciones del algoritmo para detección de marcas viales, sobre la secuencia 2 de referencia fig. 4.5b.	71
4.11. Comparación de índice de coste computacional entre variaciones del algoritmo para detección de marcas viales, sobre la secuencia 1 de referencia fig. 4.5a.	71
4.12. Comparación de índice de coste computacional entre variaciones del algoritmo para detección de marcas viales, sobre la secuencia 2 de referencia fig. 4.5b.	72
4.13. Comparación: \mathcal{FSM} resultantes.	76
4.14. Comparación del índice de Sensibilidad aplicando la máscara de espacio libre obtenida por 3 diferentes métodos, sobre la secuencia 2 de referencia fig. 4.5b.	77
4.15. Comparación del índice de Precisión aplicando la máscara de espacio libre obtenida por 3 diferentes métodos, sobre la secuencia 2 de referencia fig. 4.5b.	78
4.16. Detección de líneas de la carretera, basada en la transformada de Hough.	79
4.17. Detección de carriles viales de la carretera, basada en la transformada de Hough.	80
4.18. Direcciones en carriles detectados.	81
4.19. Detección del ángulo de la carretera a través de conjuntos soporte. . .	83
4.20. Detección del ángulo de la carretera usando agrupamiento Mean Shift. .	84
4.21. Algunas imágenes de la secuencia de referencia (a) 1 y (b) 2, para la detección de ángulo de carretera.	84
4.22. Comparación de los métodos para detección de ángulo de carretera sobre la secuencia de la fig. 4.21a.	85
4.23. Comparación de los métodos para detección de ángulo de carretera sobre la secuencia de la fig. 4.21b.	86

4.24. (a) Carriles detectados según la ec. (4.8). (b) Carriles detectados en la dirección próxima a ϕ_C . (c) Resultado de la búsqueda de carriles adyacentes y no solapados en dirección ϕ_C . En verde se representa el carril inicial de la búsqueda lan_{i^*} , en azul y rojo, aparecen los carriles obtenidos de la búsqueda en direcciones opuestas partiendo del carril inicial, \mathcal{R}_{perc}^{d1} y \mathcal{R}_{perc}^{d2}	87
4.25. Relación entre estructuras de carriles para generar integración temporal.	89
4.26. Integración temporal a lo largo de 27 detecciones consecutivas de carriles en una carretera interurbana de 4 carriles. La imagen superior de cada cuadro muestra el contenido de \mathcal{R}_{mem} en color verde, mientras que la imagen inferior, muestra el contenido de \mathcal{R}_{perc} en color azul. En color rojo se marca las actualizaciones de carriles desde \mathcal{R}_{perc} a \mathcal{R}_{mem}	93
4.27. Secuencia 1 para la detección de carreteras, tamaño 1000 imágenes.	94
4.28. Secuencia 2 para la detección de carreteras, tamaño 500 imágenes.	94
4.29. Secuencia 3 para la detección de carreteras, Raw:Road:Drive_0015, tomada del KITTI [32], tamaño 297 imágenes.	95
4.30. Secuencia 4 para la detección de carreteras, Raw:Road:Drive_0027, tomada del KITTI [32], tamaño 188 imágenes.	95
4.31. Aplicación para etiquetado de líneas de carretera.	96
4.32. Rendimiento del algoritmo Detección de Líneas Viales Secuencia 1.	97
4.33. Rendimiento del algoritmo Detección de Líneas Viales Secuencia 2.	97
4.34. Rendimiento del algoritmo Detección de Líneas Viales Secuencia 3 del KITTI Drive_0015.	98
4.35. Rendimiento del algoritmo Detección de Líneas Viales Secuencia 4 del KITTI Drive_0027.	98
4.36. Rendimiento del algoritmo Detección de Líneas Viales Secuencia 5. fig. 4.5 (a)	99
5.1. Diagrama de bloques general para el sistema avanzado de asistencia a la conducción presentado en el capítulo 4.	104
5.2. Metodología para la implementación de un clasificador por aprendizaje de máquina: (a) Descripción característica. (b) Mapeado en espacio de características. (c) Selección del clasificador. (d) Ajuste parámetros del clasificador.	105
5.3. Funcionamiento del clasificador.	107

5.4. Descriptor de marcas viales. ROIs en escala de grises y su correspondiente representacion HOG.	108
5.5. Experimento de comparación entre las representaciones del conjunto de entrenamiento para distintos filtros aplicados a la imagen de entrada y para los descriptores propuestos: ROI escala de grises y HOG con diferentes resoluciones, 18, 36 y 72 ángulos.	110
5.6. Índice de Sensibilidad medido sobre secuencia de referencia 1 para clasificadores clásicos y SVM.	111
5.7. Índice de Precisión medido sobre secuencia de referencia 1 para clasificadores clásicos y SVM.	112
5.8. Experimento de variación de los parámetros: Gama (radio de kernel) y C (parámetro de regularización).	112
5.9. Comparación de la sensibilidad entre algoritmos para segmentación de marcas viales vrs SVM.	113
5.10. Comparación de la precisión entre algoritmos para segmentación de marcas viales vrs SVM.	114
5.11. Comparación cualitativa entre algoritmos para segmentación de marcas viales presentados en la Sec. 4.1.2 (a) Seg. Kernel, (b) Seg. Grad, (c) Seg. Grad. SNM y (d) Seg. Grad. Canny. vrs (e) SVM.	114
5.12. Secuencia de referencia 2 con líneas degradadas ó poco visibles para la detección de marcas viales.	115
5.13. Comparación de la sensibilidad entre algoritmos para segmentación de marcas viales degradadas vrs SVM.	116
5.14. Comparación de la precisión entre algoritmos para segmentación de marcas viales degradadas vrs SVM.	117
5.15. Comparación cualitativa entre algoritmos para segmentación de marcas viales presentados en la Sec. 4.1.2 (a) Seg. Kernel, (b) Seg. Grad, (c) Seg. Grad. SNM y (d) Seg. Grad. Canny. vrs (e) SVM, aplicados sobre marcas viales degradadas.	117
5.16. Segmentación SVM sobre imagen de referencia para marcas viales degradadas. (a) Vista original. (b) Vista de pájaro. (c) Máscara marcas viales ground truth. (d) Máscara de marcas viales SVM.	118
5.17. Rendimiento del algoritmo Detección de Líneas Viales usando SVM. fig. 4.5 (a).	119
5.18. Extracción de perfiles de línea vial.	121

5.19. Descriptor de líneas viales.	122
5.20. Representación en espacio de características de un conjunto de líneas, etiquetadas manualmente.	123
5.21. Representación PCA del conjunto de líneas extraído de la secuencia mostrada en la fig. 4.27.	124
5.22. Comportamiento del Clasificador SVM para la detección de líneas viales para un descriptor de 2 características: Valor medio de perfil de línea y longitud de línea en metros.	125
5.23. Experimento de variación de los parámetros Gama, radio de kernel, y C, parámetro de regularización.	126
A.1. Transformación homográfica desde la perspectiva real de la cámara a la perspectiva en <i>vista de pájaro</i> , $ROI_{vp}(x_{bv}, y_{bv}, z_{bv})$	138
A.2. Representación gráfica de la transformación homográfica desde la perspectiva <i>real</i> de la cámara a la perspectiva en <i>vista de pájaro</i> . . .	139
C.1. Esquema básico perceptrón multicapa. Imagen obtenida de [37]. . . .	146
C.2. Resumen gráfico del flujo de señales para el algoritmo de aprendizaje: “Back-Propagation”. Imagen obtenida de [37].	147

Índice de cuadros

3.1. Especificaciones cámara estéreo	37
3.2. Rendimiento de los estimadores usados en el experimento de análisis de estabilidad en una secuencia estática con reducción por muestreo uniforme de 2000 puntos sobre la secuencia de la fig.3.7	46
3.3. Rendimiento del estimador RANSAC con reducción por muestreo uniforme de 2000 puntos sobre la secuencia de la fig.3.10	51
3.4. Coeficientes del filtro FIR diseñado para mitigación de ruido en parámetros extrínsecos.	54
3.5. Algoritmo recursivo Filtro de Kalman.	57
3.6. valores de varianza para ruido de proceso y ruido de medida, usados en el experimento de la fig. 3.14.	57
3.7. valores óptimos de varianza para ruido de proceso y ruido de medida, seleccionados por parámetro extrínseco	58
4.1. Parámetros de configuración para la transformada Hough.	78
4.2. Resultados generales del método propuesto para detección de carreteras Sec. 4.3 sin I. T. (Integración temporal) y con I.T Sec. 4.4. . . .	100
5.1. Resultados generales del método propuesto para detección de carreteras Sec. 4.3 sin I. T. (Integración temporal) y con I.T Sec. 4.4. Comparando el efecto de usar para la detección de marcas viales el algoritmo basado en gradiente versus el algoritmo basado en clasificación SVM	120
5.2. Matriz de confusión del entrenamiento SVM para clasificación de líneas de carretera para valores óptimos de ajuste.	127

6.1. Desviación típica respecto a referencia, eliminando el error sistemático de medida, de los parámetros extrínsecos medidos en la secuencia de referencia, Raw:Road:20112609_0015, tomada del KITTI.	130
B.1. Función de los tipos de marcas viales del estándar IC 8.2., referidos en la tabla B.2.	141
B.2. Tipos de marcas viales según el estándar IC 8.2 medidas en m. y VM en Km/h.	142

Capítulo 1

Introducción

1.1. Motivación de la Tesis

El desarrollo económico mundial ha traído consigo un aumento en el consumo de bienes que suplen necesidades básicas tales como el transporte. Desde la producción del vehículo en una cadena de montaje por Henry Ford en 1908, el mundo ha experimentado un crecimiento sostenido en el número de vehículos que transitan por vía terrestre. Este hecho puede ser observado en los datos reportados por la Oficina de Estadística de la Comisión Europea, Eurostat, encargada de consolidar datos macro económicos para la Unión Europea, UE, en áreas como el transporte, ver fig. 1.1.

En toda la zona Europea, el crecimiento en número de vehículos ha sido correspondiente con el crecimiento de la población y su desarrollo económico, esta tendencia se ha mantenido en los últimos 22 años. En Alemania por ejemplo, la relación número de vehículos por cada mil habitantes es de 540, mientras que en España la relación es de 471 vehículos por cada mil habitantes para el año 2012. Así mismo, una cota de saturación se observa en los Estados Unidos, país con la mayor relación de vehículos por cada mil habitantes del mundo, con un índice de 816 para el año 2014, datos tomados del departamento de transporte Estadounidense, Administración Federal de Autopistas. Este comportamiento en el crecimiento de propietarios de vehículos se describe en [23], allí se confirma un nivel de saturación cercano a 800 vehículos por cada mil habitantes en países con un ingreso per capita superior a los \$20.000 dolares. Este modelo, basado en datos históricos desde 1960 hasta 2002 sobre 45 países que corresponden al 75 % de la población mundial, estima que el inventario de automóviles se incrementará de 800 millones en 2002 a cerca de 2 mil millones de unidades para el año 2030.

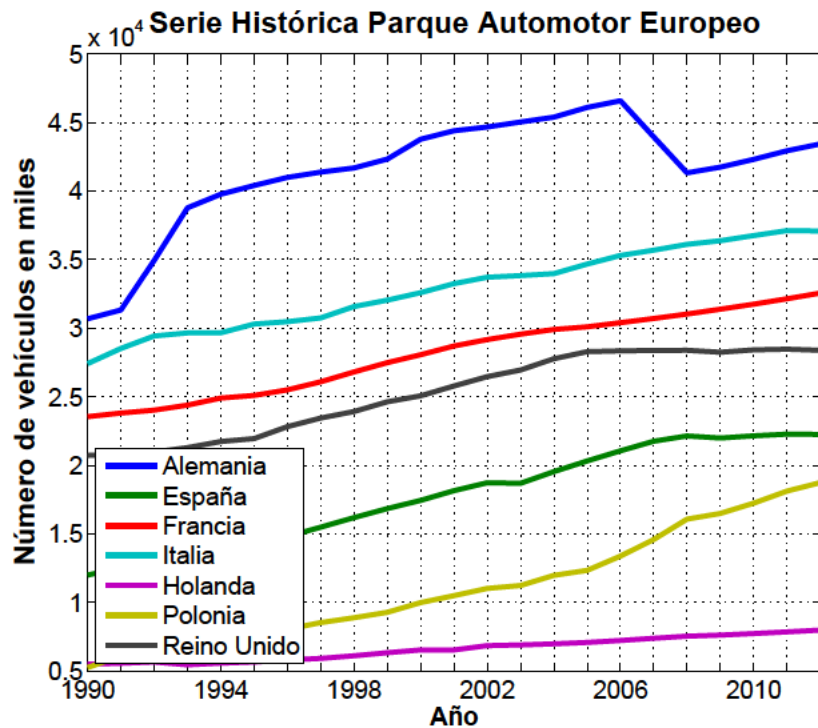


Figura 1.1: Serie histórica del parque automotor Europeo. fuente: European Commission - Directorate general for energy and transport - CARE database

Este crecimiento en el número de vehículos en circulación por las carreteras del mundo trae consigo una consecuente accidentalidad debido a múltiples factores tanto humanos como técnicos. Según el informe [19], en la UE se han reportado 29.2 miles de personas muertas y 1.4 millones de personas heridas en 2012 sobre vías terrestres.

A nivel global, según cifras de la Organización Mundial de la Salud, en su informe de 2013 [77], se reportan cerca de 1.24 millones de muertes en accidentes de tráfico, sin variaciones significativas en el periodo 2007-2013, a pesar del incremento reportado de 15 % en el número de vehículos en circulación.

Analizando las estadísticas mundiales según las categorías de ingreso definidas por el Banco Mundial para las economías: ingreso bajo (IB), menor a \$1.046 dólares, ingreso medio (IM), entre \$1046 dólares a \$12763 dólares e ingreso alto (IA), superior a \$12763 dólares. Economías IA como la UE, que representan el 16 % de la población mundial, poseen el 47 % del parque automotriz y generaron en 2013 el 8 % de las muertes en accidentes de tráfico. Mientras que las economías IM que representan al 72 % de la población mundial poseen el 52 % del parque automotor y generaron en 2013 el 80 % de las muertes en accidentes de tránsito, tal y como se observa en

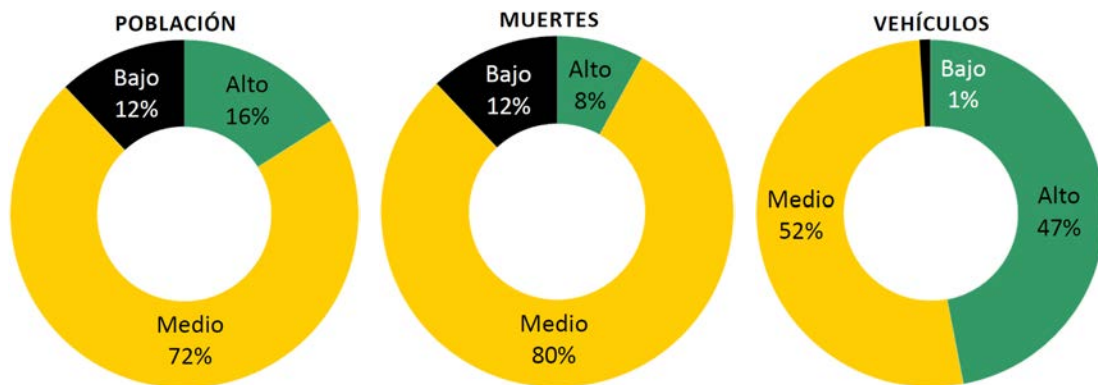


Figura 1.2: Estadísticas globales de población mundial, muertes en accidentes viales y vehículos motorizados registrados según el nivel de ingreso de los países. Fuente [77].

la Fig. 1.2. Las estadísticas anteriores muestran dos tendencias diferenciadas, una en las economías desarrolladas, donde la siniestralidad vial se ha visto reducida, y otra, en economías en vía de desarrollo, donde el número de accidentes de tráfico va en aumento con 20,1 fallecidos por cada 100.000 habitantes en relación a los 8,7 fallecidos por cada 100.000 habitantes en economías IA.

En España, según datos del último reporte de 2014, generado por la secretaría general de tráfico, [25], se produjeron el 6.6 % de las víctimas fatales en accidentes de tránsito de la UE, y de ellos, los factores causantes de los siniestros se pueden asociar a:

- Exceso de velocidad: presente en un 17 % de accidentes con víctimas en vías interurbanas y en el 21 % de los accidentes mortales en vías generales.
- Presencia de sustancias psicoactivas en conductores: según el Instituto Nacional de Toxicología y Ciencias Forenses (INTCF), en el año 2014 de 614 conductores fallecidos en accidentes viales, el 39,1 % presentaron alguna sustancia psicoactiva, 240 casos.
- Distracción: aparece como factor de accidentalidad en un 30 % de los accidentes con víctimas. En vías interurbanas la distracción se presenta en un 36 % de los accidentes y en vías urbanas en un 26 %.

En Estados Unidos, según [89], un estimado de 16.5 % de los accidentes viales con víctimas estuvieron igualmente relacionados con un conductor somnoliento.

Iniciativas globales para desarrollar tecnologías que ayuden a la reducción de factores de riesgo asociados al error humano en la conducción, se han abordado desde

hace más de 30 años. Uno de los programas pioneros fue DRIVE [13], promovido por la Unión Europea en 1993, dedicado al desarrollo de la infraestructura vial para seguridad vehicular, y dentro del cual se encuentran proyectos como *GIDS*, Generic Intelligent Driver Support por sus siglas en inglés, cuyo objetivo era determinar los requerimientos y estándares de diseño para sistemas inteligentes de soporte a la conducción, tecnología que en años posteriores se convertiría en *ADAS*, Advanced Driver Assistant Systems, ubicada en el tiempo desde el año 2000 hasta el presente en [96].

La incursión de esta nueva tecnología coincide con la reducción de víctimas fatales en accidentes viales en economías de alto ingreso como la UE, ver Fig. 1.3. Sin embargo, las cifras aun son alarmantes para el año 2014, y en una mirada a nivel mundial, según [77], se estima que para el año 2030 las muertes en accidentes viales pasarán del puesto ocho al puesto quinto en el ranking global de causas de muerte.

Estas proyecciones plantean un escenario donde es importante continuar desarrollando la tecnología *ADAS*, logrando sistemas integrados cada vez más accesibles a todos los niveles de ingreso, con el objetivo de lograr que la tendencia mostrada en reducción de siniestralidad para economías IA se transfiera al resto del mundo.

La presente investigación doctoral aborda el desarrollo de un Sistema Avanzado de Asistencia a la Conducción basado en visión por computador que sea robusto y fácilmente transferible a un bajo costo. La solución *ADAS* presentada se centra en la detección de carriles, por cuanto esta funcionalidad es el núcleo de la mayoría de los módulos para asistencia a la conducción modernos tales como: alerta por salida de carril, *LDW* (Lane Departure Warning), asistente para mantenimiento de carril, *LKA* (Lane Keeping Assistance), asistente para cambio de carril, *LCA* (Lane Change Assistance), entre otros.

El aporte de esta tesis está en el uso de estrategias flexibles que permiten incorporar invarianza en cuanto a factores de iluminación, inclusión de múltiples estándares para demarcación de carriles y detección de líneas en la calzada bien definidas y degradadas, aspecto importante que en la actualidad no se ha resuelto por completo. Así mismo, la auto-calibración de parámetros extrínsecos a partir de la reconstrucción estéreo del entorno, permite la adaptación del sistema de captura de imágenes a bordo en cualquier ubicación sobre el vehículo con línea de vista sobre la carretera, haciendo al sistema de fácil instalación.

Esta investigación hace parte del desarrollo del *IVVI 2.0*, Vehículo Inteligente basado en Información Visual, plataforma de pruebas del laboratorio de Sistemas

Inteligentes, LSI, de la Universidad Carlos III de Madrid.

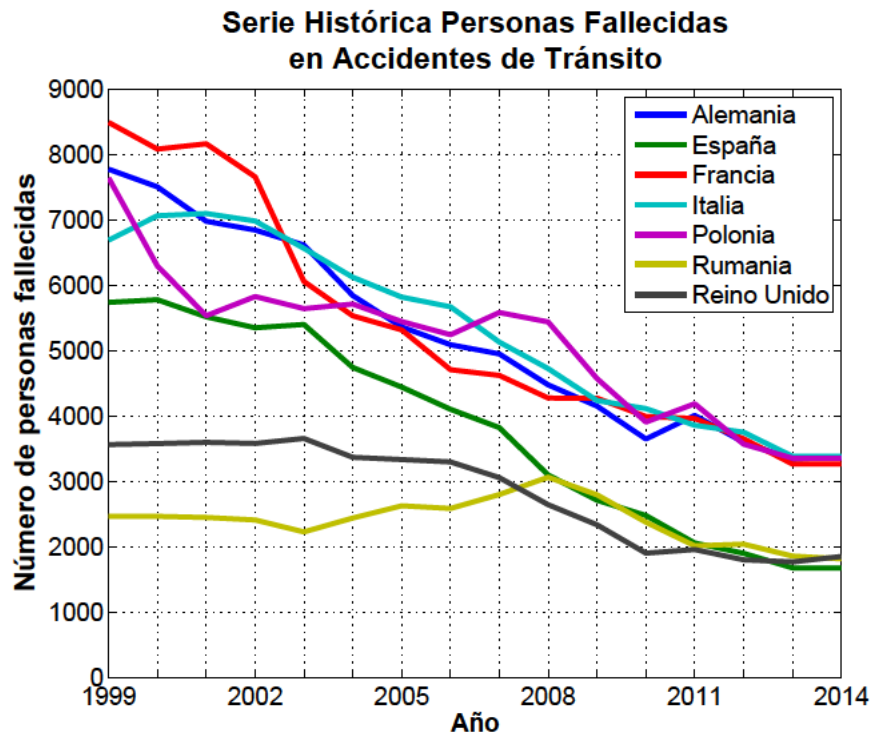


Figura 1.3: Serie histórica de personas fallecidas en accidentes de tránsito.

1.2. Objetivos

1.2.1. Objetivo Principal

Diseñar e implementar un sistema avanzado de asistencia a la conducción para entornos inter-urbanos utilizando imágenes estéreo que pueda ser integrado al vehículo inteligente IVVI 2.0.

1.2.2. Objetivos Específicos

- Desarrollar y evaluar un algoritmo de auto-calibración de parámetros extrínsecos, que permita a un ADAS basado en visión, ubicar indistintamente su sensor tipo cámara.
- Segmentar carreteras inter-urbanas conforme a la norma I.C 8.2.

- Clasificar las líneas que separan los carriles de carreteras iter-urbanas conforme a la norma I.C 8.2.
- Integrar temporalmente la segmentación de carreteras iter-urbanas para mitigar el efecto de oclusiones y falsos positivos en las detecciones estáticas.
- Aplicar técnicas de aprendizaje de máquina para mejorar el Sistema Avanzado de Asistencia a la conducción propuesto.

1.3. Esquema del Documento

La estructura del documento consta de 4 capítulos principales y un capítulo de conclusiones.

En el capítulo 2 se presenta el estado de la técnica relacionado con los ADAS, explorando desde sus orígenes, principales hitos que impulsaron su desarrollo, iniciativas mundiales tanto gubernamentales como privadas, hasta el desarrollo de la investigación en algoritmos de visión por computador enfocados en la percepción de entornos tipo carretera.

Posteriormente, en el capítulo 3 se aborda la reconstrucción tridimensional de la carretera, y se presenta un algoritmo para la auto-calibración de parámetros extrínsecos a partir de la detección del plano dominante en el espacio. Adicionalmente se realiza un análisis de estabilidad de los parámetros extrínsecos obtenidos, junto con diferentes propuestas para reducir el error de medida. El análisis comparativo se efectúa tanto sobre secuencias propias del laboratorio LSI, Universidad Carlos III de Madrid, como sobre secuencias de prueba para percepción del KITTI, Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago [32].

A continuación, en el capítulo 4 se presenta el algoritmo general para modelado y detección de la carretera en cinco fases: primero se describe el procedimiento para cambio de perspectiva a vista de pájaro; segundo, se realiza la simplificación de la imagen a partir de las máscaras de marcas viales y espacio libre; tercero se detectan los elementos constitutivos de la carretera como líneas y carriles; cuarto, se detecta parcial o totalmente la carretera, definiendo el ángulo de la carretera y los carriles coherentes a él, para realizar una búsqueda de carriles adyacentes ordenados espacialmente; quinto y por último, se integra temporalmente las detecciones de carretera, interpolando en el tiempo la información parcial de cada captura y adicionando restricciones según la norma Española para demarcación de carreteras I.C 8.2.

Finalmente, el capítulo 5 se realiza una descripción sobre el aprendizaje de máquina en ADAS. Primero se plantea una metodología general para solucionar tareas a través de la clasificación de patrones, para luego, en la secuencia de procesos del ADAS propuesto, mostrar en dos tareas puntuales: generación de máscara de marcas viales y clasificación de líneas de la carretera, la aplicación de la metodología planteada inicialmente.

El documento concluye con las ideas que sintetizan el trabajo, producto del análisis de resultados sobre los experimentos realizados, además, se indica posibles trabajos futuros que continúen la investigación planteada.

Capítulo 2

Estado del Arte

2.1. Seguridad en la Conducción

Las tecnologías para la seguridad en la conducción han mostrado avances significativos en dos líneas complementarias: investigación en algoritmos y desarrollo de módulos comerciales. Los dispositivos de asistencia introducidos en la producción de vehículos en serie, han sido el resultado visible de un campo investigación muy dinámico soportado por la industria automotriz, entidades gubernamentales y los centros de investigación alrededor del mundo.

En un estudio realizado por [96], este avance comercial en las tecnologías para la seguridad en la conducción pueden agruparse en 2 etapas. La primera etapa se ubica desde el año 1950 hasta el año 2000, periodo en donde aparecen los primeros sistemas automáticos como el control de velocidad (CC, Cruise Control) en 1958, el sistema de bolsa de aire (Airbag) en 1973 o el control de estabilidad electrónica en 1987. Posteriormente, se inicia un segundo periodo tecnológico conocido como el desarrollo de los ADAS (Advanced Driver Assistance Systems) desde el año 2000 al presente. En este periodo se encuentra la aparición de módulos comerciales como: LDW (Lane Departure Warning), alerta por salida de carril en 2005, IPA (Intelligent Parking Assist), sistema de asistencia para aparcado inteligente en 2003, AVM (Around View Monitor System), sistema de visión periférica en 2007, CAS (Collision Avoidance System) sistema para evasión de colisión frontal en 2008, AA (Attention Assist), alerta por somnolencia en 2010, LKA (Lane Keeping Assist), asistente para mantenimiento de carril en 2014, entre otros.

Simultáneamente, la investigación en seguridad para la conducción ha evolucionado con mayor rapidez, siendo la fuente de donde se han derivado todos los sistemas

de asistencia a la conducción. Algunos trabajos agrupados por módulos funcionales ADAS se presentan a continuación:

- LDW (Lane Departure Warning), funcionalidad que supervisa, a través de algoritmos de visión por computador, la posición del vehículo respecto a su carril de circulación, alertando al conductor en caso de detectarse una trayectoria de cruce inminente a través de las líneas viales, separadoras de carril. [27, 84, 87, 91].
- ACC (Adaptative Cruice Control), control adaptativo de velocidad, módulo que ajusta la velocidad del vehículo para mantener una distancia segura con el vehículo que se encuentre adelante. [31, 102].
- LKA (Lane Keeping Assist) es la extensión funcional del LDW, una vez se ha detectado el cruce de las marcas viales, se modifica la dirección de movimiento del vehículo para mantener la trayectoria alineada con el centro del carril de circulación. [10, 65, 95].
- LCA (Lane Change Assistance), asistente para cambio de carril, se define según la norma ISO 17387, como un sistema capaz de evaluar la situación de cambio de carril por petición del conductor e indicar si ésta maniobra es segura o no. [41, 98].

La convergencia de estas capacidades en un vehículo dotado con sistemas de planificación para navegación terrestre, hará factible la conducción semi-autónoma y completamente autónoma. Objetivo que aún no ha sido alcanzado a bajo costo y con la confiabilidad suficiente.

Ahora bien, mientras las tecnologías ADAS convergen en los vehículos autónomos, la seguridad total en la conducción es un objetivo alcanzable combinando niveles de asistencia. En esta línea, en [40] se propone una metodología para garantizar la cooperación entre el conductor real de un vehículo y el sistema ADAS. Este modelo representa un flujo de toma de decisiones sobre el control de trayectoria. Una vez el conductor falla en la operación segura del vehículo, el sistema de alertas ADAS proporciona la información de anomalías detectadas al conductor, posteriormente, la desatención por parte del conductor o falla en el sistema de alerta, activa el sistema de asistencia para evasión de colisión de forma autónoma, quitando temporalmente el control al conductor. Finalmente, una falla en este sistema de evasión produce la activación de los sistemas pasivos de emergencia.

2.2. Conducción Autónoma

El uso de la infraestructura actual por los vehículos autónomos, es una restricción que impone el incalculable coste de reemplazar otros tipos de demarcadores en carreteras y autopistas, diferentes a las marcas viales actuales [42]. Es así, cómo múltiples esfuerzos se han hecho en esta dirección, para probar el grado de madurez de las tecnologías relacionadas con la conducción autónoma en la infraestructura actual.

Algunas demostraciones alrededor del mundo han presentado vehículos transitando de forma autónoma por carreteras urbanas e inter-urbanas en largos recorridos. Por ejemplo, en el concurso the Urban Challenge en 2006, auspiciado por DARPA (the Defense Advanced Research Projects Agency), los vehículos ganadores, Boss [90] de la universidad Carnegie Mellon, y Junior [72] de la universidad de Stanford, circularon de manera autónoma en el evento final a lo largo de 85 Km, cumpliendo con las leyes de tráfico de California, Estados Unidos, y desplazándose simultáneamente con vehículos tripulados y otros vehículos autónomos. Sin embargo, tanto Boss como Junior, usaron algoritmos de percepción basados en posicionamiento global GPS de alta precisión, láser y radares, tecnologías de alto costo que no pueden ser transferidas al público. Así mismo, el comportamiento de los robots terrestres no mostró una confiabilidad del 100 %, el vehículo ganador Boss cometió en 2 ocasiones errores en el plan de ruta, realizando giros de 180 grados innecesarios y perdiendo 3.2 Km en su recorrido. Junior por su parte realizó ciclos repetitivos en una sección de su trayectoria. Estos y algunos otros problemas de calibración de instrumentos e interferencias en las comunicaciones, dejan claro los desafíos para llevar la conducción autónoma a un nivel de desarrollo comercial.

Otras demostraciones como the VisLab Intercontinental Autonomous Challenge [6], desafío llevado a cabo por el centro de investigación VisLab en 2010, donde 4 vehículos autónomos condujeron desde Parma Italia, hasta Shanghai China, a lo largo de 13000 km, mayoritariamente a través de zonas donde no se contaba con mapas digitales ni tampoco con información detallada del ambiente, probó que para enfrentar condiciones de gran diversidad en escenarios y no información a priori, el uso de visión por computador es una alternativa aplicable en conjunto con el posicionamiento GPS y la reconstrucción tridimensional del entorno.

2.3. Esquema General de un Sistema para Detección de Carril/Carretera basado en Visión

Con el ánimo de crear un marco de referencia general para el campo de la detección de carriles y/o carreteras, basado en visión por computador, se analizaron los siguientes trabajos [18, 38, 68, 85]. Allí se observan una serie de módulos comunes con los cuales se propone el siguiente esquema general.

- 1 Extracción de características, métodos ascendentes
- 2 Ajuste de modelo de carretera, métodos descendentes
- 3 Integración temporal

A continuación se clasifican los trabajos mas relevantes en el estado de la técnica, de acuerdo con el esquema taxonómico propuesto.

2.3.1. Extracción de características

La selección de características en cualquier tipo de imagen depende del objetivo que se requiera perseguir, en este caso, el objetivo es la detección de los límites de un carril estructurado, o la extracción de frontera para una zona transitable tipo carretera.

2.3.1.1. Detección de Carriles Estructurados

Para abordar la detección de los límites de un carril estructurado a partir del análisis de imágenes, se debe tener en cuenta, cómo las líneas viales sufren degradación visual por efecto del contacto con en tráfico circulante, además de las condiciones variables de iluminación e interferencia por sombras u oclusiones con otros objetos sobre la vía, a que son sometidas. Es por ello, que se deben utilizar múltiples características para definir su posición en la imagen tales como color, variación de intensidad, forma, orientación entre otros.

Algunas estrategias para la segmentación de marcas viales hacen uso del espacio del gradiente, una transformación que facilita la detección, por cuanto explota el contraste que exhibe su forma sobre la calzada. Trabajos como [18, 42, 73], hacen uso del gradiente de la imagen de intensidad para detectar los flancos que confinan una marca vial. Dentro de esta clase de algoritmos es usual encontrar enfoques donde

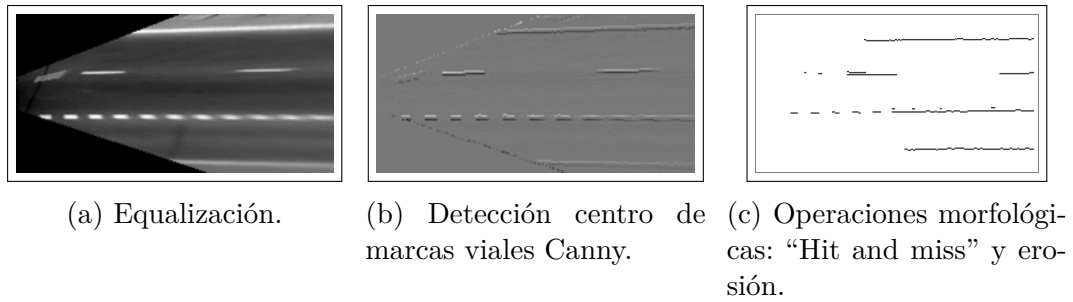


Figura 2.1: Algoritmo detección de marcas viales, Collado [18].

se ajusta un patrón de marca vial a través de núcleos de convolución fijos, tal es el caso de los filtros direccionales (steerable filter) [68] que puede detectar además de las líneas viales, elementos del tipo reflector circular; filtro horizontal-vertical [42]; filtro morfológico Top Hat [80] o el filtro de escalón (step filter) para encontrar parejas de respuesta adyacentes con signos opuestos.

Un aspecto que parte en dos tendencias, los métodos de extracción, tiene que ver con la perspectiva de la imagen sobre la que actúan, o bien, perspectiva normal tomada abordo del vehículo [42], o bien, su transformación a vista superior (top view), también llamada vista de pájaro (bird view) o perspectiva inversa (inverse perspective, IP) [12, 18, 43, 44, 62, 68, 73].

Un ejemplo del análisis de marcas viales en vista superior es el trabajo desarrollado por [18], donde se presenta un algoritmo de detección de marcas viales consistente en 3 fases. Primero se realiza la ecualización de la imagen en perspectiva inversa. Segundo, se aplica un filtro gaussiano para eliminación de ruido, a continuación, se calcula el gradiente en dirección horizontal usando las máscaras de Sobel, junto con supresión de no máximos para realzar bordes verticales de ancho 1 pixel, una vez extraídos los bordes verticales es posible realizar la búsqueda de parejas de pixeles distantes un ancho de marca vial para fijar la posición del pixel central. Finalmente, el ruido remanente es eliminado a través de operaciones morfológicas como “Hit and miss”. En la fig. 2.1 se muestra el proceso descrito anteriormente.

Existen otras características que pueden ayudar a la identificación con mayor confiabilidad de las marcas viales, por ejemplo en [62], se utiliza la diferencia entre los dos picos más altos del histograma calculado para una ventana deslizante de 8×8 pixeles. Igualmente, se mide la direccionalidad de los pixeles que pertenecen al pico del histograma con mayor soporte, para diferentes ángulos, utilizando el cociente de varianza entre componentes en x e y.

En [5, 16] se utiliza filtros de color específicos, blanco, amarillo y rojo, correspondientes a la mayoría de marcas viales existentes. En [16] se eliminan posteriormente las regiones que puedan pertenecer a vehículos con colores similares a través del análisis de movimiento de regiones respecto al movimiento de la carretera.

La detección de marcas viales como un problema de clasificación se propone en [50], allí se realiza un análisis comparativo de los principales clasificadores clásicos como modernos.

2.3.1.2. Detección de Zona Transitabile

La detección de los límites de la zona transitabile es una tarea que aporta robustez y seguridad a los algoritmos de seguimiento de carriles o evasión de obstáculos, ya que solo pueden permitirse trayectorias en esta región. En este sentido, muchos trabajos abordan esta tarea de forma única o en apoyo a la verificación de las marcas viales.

Los límites de la calzada no siempre poseen una demarcación bien definida con líneas visibles pintadas sobre la calzada o cualquier otro indicativo hecho por el hombre, en algunos casos se requiere tener en cuenta descriptores tales como información 3D: depresiones, barreras o acera; apariencia: color o textura, además de las marcas viales.

Segmentación basada en información 3D estéreo es usada en [42], allí se analiza el entorno en dirección radial al vehículo en un barrido de 360 grados, los límites de carretera son detectados por el primer cambio drástico en la continuidad de cada trayecto. [79] utiliza estimación de planos a través de votación sobre las normales de los puntos 3D para definir escalones límite de carretera. [22] detecta la acera como los límites de calzada, marcando los puntos cercanos con cambios considerables de altura sobre la nube de puntos (Point Cloud, PC) que representa la escena. [97] realiza estimación de movimiento de la cámara y estimación del plano de la carretera basado en imágenes monoculares, para calcular la homografía entre imágenes consecutivas, con ella, proyecta la imagen de la captura anterior al instante actual y por diferencia de imágenes, usando suma de diferencias absolutas SAD por sus siglas en inglés, obtiene la segmentación de la carretera en entorno urbano.

Otros algoritmos para segmentación de la carretera se basan en apariencia, trabajos relacionados como [4] utilizan la característica de color a partir del crecimiento de regiones en un espacio de color invariante a la iluminación. En [48] se utiliza un descriptor compuesto por información proveniente de 4 imágenes de baja resolución, sobre la zona cercana al vehículo, en espacios de color y textura derivados de la com-

binación de componentes HSI , YC_bC_r y $LABColor$, junto con densidad de bordes, con ellos se construye sus respectivos histogramas y de allí se caracteriza las regiones transitables.

En entornos no estructurados la detección de la zona transitable es posible fundamentalmente a través del análisis de textura, algunos trabajos hacen uso del filtro de Gabor como en [51], allí se utiliza este filtro para detectar la dirección dominante de la textura y con ella, utilizando un esquema de votación basado en una medida de confianza para cada pixel, se estima un punto de fuga. Posteriormente se calcula, como límites de carretera, el par de líneas más externas con mayor soporte de dirección que corten el punto de fuga hallado.

La segmentación de la carretera basada en la textura, frecuentemente es abordada a partir del entrenamiento de clasificadores, en esta línea, [35] trata el problema a través de una estrategia de clasificación robusta con el ánimo de no depender de modelos a priori. Allí se hace uso de descriptores jerárquicos extraídos de la aplicación de filtros tipo Haar-like sobre una colección de ROIs circulares centrados en cada pixel de la imagen, con radio creciente y sobre información visual derivada de la imagen original, tal como la intensidad de la imagen, su mapa de bordes y filtros de Gabor que capturan la textura, con el objetivo de aplicar un clasificador Adaboost que combine los clasificadores débiles tipo Haar en un clasificador fiable.

Así mismo, en [1] se presenta un método basado en clasificación AdaBoost para un conjunto de clasificadores derivados de características como la respuesta de filtros de convolución con núcleo Walsh-Hadamard y Momentos de energía. También en [54], se utiliza una versión similar de clasificación (GentleBoost) para abordar la diferenciación entre zonas pertenecientes a la carretera y fuera de ella, con características SFA (Slow Feature Analysis) en conjunto con color y Walsh-Hadamard.

2.3.2. Ajuste de modelo de carretera

El modelo de la carretera se define como el conjunto de parámetros que describe su forma y ubicación en la imagen. Existen diversos modelos paramétricos y no paramétricos que se ajustan al tipo de carretera analizada. Así por ejemplo, el modelo más apropiado en entornos de autopistas es un par de líneas convergentes en un punto del horizonte (punto de fuga) o incluso un par de aproximaciones a circunferencias o parábolas que tengan en cuenta pequeñas curvaturas del perfil de la vía. Otros entornos menos predecibles son los caminos a campo abierto en zonas desérticas, que requieren el ajuste de curvas más flexibles definidas a partir de puntos de control,

como en el caso de las curvas tipo spline.

La definición de un modelo de carretera, ya sea a través de sus parámetros o puntos de ajuste, es importante en la mayoría de trabajos relacionados con aplicaciones ADAS, por cuanto la información visual registrada por una cámara a bordo de un vehículo, sufre pérdidas por interferencias o ruido, razón por la cual, normalmente la confiabilidad de las aplicaciones requieren de algoritmos de seguimiento que actúan sobre el espacio de parámetros o puntos definidos como representación de la realidad, en este caso, la carretera.

Dentro de los modelos encontrados en la literatura se encuentran:

- Carriles formados por líneas paralelas

Este modelo aplica tanto para entornos no estructurados como para carreteras con líneas pintadas sobre la calzada.

Muchos trabajos relacionados con la navegación en escenarios con límites de carril con poca o sin definición, se desarrollaron alrededor del Darpa Grand Challenge de 2004 y 2005, algunos como [1] presentan un algoritmo para hallar los límites de la carretera como líneas paralelas, basándose en la idea de estimar el cabeceo y la rotación de la cámara de manera adaptativa a través del flujo óptico y un modelo simple de movimiento, para luego transformar la imagen a vista superior y allí, calcular la dirección de la textura de cada pixel usando filtros de Walsh-Hadamard. Luego se proyecta las componentes de dirección de textura sobre el eje horizontal con el objetivo de detectar picos de máxima concentración, de allí se puede estimar la posición de las líneas. Un enfoque similar [51], obtiene los límites de carretera basándose en el análisis de la imagen en perspectiva normal, calcula la orientación de la textura de la vía a través de filtros de Gabor, con lo cual obtiene el punto de fuga desde donde se busca las líneas cuya dirección sea coincidente con la dirección de textura.

En [7, 12, 80] se aborda escenarios estructurados, donde se trata la carretera como líneas paralelas sujetas a restricciones espaciales en perspectiva inversa. En [100] se obtiene el modelo de líneas convergentes fusionando la segmentación de carretera basada en un grafo de similitud con las líneas y el punto de fuga obtenido de la transformada de Hough. [12] estima el modelo usando el estimador RANSAC para eliminar puntos extraños resultantes de la extracción de características. Otros trabajos como [47], analiza la imagen dividiéndola en N segmentos horizontales. En cada sub-imagen se etiqueta los pixeles de acuer-

do a su ángulo de gradiente, en una de 8 regiones distribuidas uniformemente entre 0 y 2π . Posteriormente, se realiza agrupación de píxeles, según un algoritmo de componentes conectadas, para así, formar grupos de soporte de línea, sobre los cuales, se calcula la línea que mejor los representa. La definición de cada línea se realiza a través de la pendiente del eje mayor de la dispersión del grupo soporte y un punto, correspondiente a su centro de masa. Una vez se han hecho las definiciones de líneas por segmento horizontal, se aplica un proceso de búsqueda, para encontrar todos los caminos, construidos por N líneas, una por cada segmento horizontal. El camino debe minimizar una cierta función de energía que depende del paralelismo entre líneas conectadas.

- Carriles formados por parábolas y polinomios.

Dentro de los trabajos que usan el modelo parabólico de la carretera se pueden mencionar a [18, 42, 68, 83, 95, 101]. En [42] se aproximan segmentos de puntos conectados con gradiente común a través de arcos de parábola cuya desviación no supere un cierto umbral. Por su parte en [83] se utiliza la Transformada de Hough para estimar dos de los cuatro parámetros del modelo parabólico y en conjunto con un algoritmo genético se ajustan los parámetros restantes al conjunto de observaciones como un problema de optimización. Labayrade [56] ajusta el modelo hiperbólico en perspectiva original.

En [101], se tratan imágenes aéreas a través de un algoritmo de clasificación no supervisado para segmentar las carreteras en espacio de color, una vez la calzada ha sido identificada, se realiza un análisis basado en la matriz hessiana de intensidad, para encontrar el inicio de segmentos correspondientes a marcas viales junto con su dirección, para de esta manera, modelar las líneas como polinomios de segundo orden, a continuación se realiza la verificación y ajuste de parámetros como un problema de mínimos cuadrados. El proceso se repite para las dos imágenes estéreo. A continuación se representa la estructura de marcas viales detectadas a través un grafo, donde los nodos representan las líneas, y las conexiones entre nodos, una función estadística de compatibilidad. El casamiento de líneas como un problema de búsqueda en el grafo, permite verificar la segmentación de una marca vial.

- Carriles formados por splines.

Las curvas semi-paramétricas Splines son definidas por una serie de puntos llamados puntos de control, que en dependencia de la clase de Spline, pueden

estar adentro (Spline cúbica) o afuera de la curva (B-Spline).

En [50] una curva Spline cúbica modela independientemente cada límite de carretera. Aunque el ajuste es rápido, pueden generarse curvas irregulares cuando el espacio entre puntos de control no es suficiente. Para solucionar la inestabilidad de posibles realizaciones, los autores utilizan el algoritmo de estimación RANSAC, para escoger la mejor curva con el número apropiado de puntos de control.

Un caso especial de las curvas de tipo Snake o contornos activos, es su aproximación por curvas B-Spline. Esta descripción consiste en funciones polinomiales que aproximan un contorno a trozos de acuerdo a un equilibrio de fuerzas virtuales que actúan sobre ella, usando un conjunto de puntos de control para su ajuste. En [93], se modela los carriles como una curva B-Snake aproximada, central a los bordes de carril. Para ello primero se subdivide la imagen en 5 secciones horizontales, en cada una se aplica detección de bordes usando el filtro Canny y posteriormente, se detectan las líneas presentes a través de la transformada Hough. A continuación, se realiza la búsqueda de la línea de fuga (vanish line) y el punto de fuga (vanish point), por votación de pares de línea en cada sección. Esta información es utilizada en la definición del conjunto de puntos de control con que se inicializa la curva B-Spline que modela el carril de circulación (ego lane). Finalmente, se aplica el método de error cuadrático medio mínimo para encontrar los puntos de control dados por las fuerzas globales de la imagen definidas en ambos lados del carril.

Existen otros tipos de Spline como el planteado en [92], donde se utiliza la descripción de curvas Catmull-Rom Spline, para la descripción de límites de la carretera. El tratamiento de imágenes que permiten detectar los bordes de un carril se realiza de la siguiente manera, primero se asocian los píxeles que poseen orientación de gradiente similar, con ello se definen grupos de píxeles, donde una línea puede ser determinada a través de una pendiente extraída de los eigenvalores asociados a la matriz de covarianza del grupo y un punto, que corresponde con el centro de masa del mismo. Posteriormente se aplica el filtro de Canny para marcar píxeles que pertenecen al modelo de carril, luego, se define una función de energía inducida por el campo potencial de borde o función de distancia a la magnitud del gradiente de la imagen. La función de energía incluye la dirección entre el gradiente de un píxel y la orientación del borde más cercano, con lo cual se logra obtener una imagen donde se resalta

con mayor intensidad aquellos píxeles, que no solo cumplen con la posición del modelo, sino también con la orientación de borde.

En [42] la selección de puntos de control para ajustar una Spline, se realiza sobre 100 semillas, en zona cercana al vehículo y en vista superior, candidatos a ser marcas viales, luego se efectúa una búsqueda voraz punto a punto, seleccionando un candidato a la vez que se encuentre distante, fuera de un radio de cercanía de 50 píxeles.

2.3.3. Integración temporal

La integración temporal es una etapa de refinamiento en los algoritmos de detección de carril/carretera, se emplea para corregir errores que hayan permanecido después de la etapa de detección de características y ajuste de modelo, como consecuencia de la violación de suposiciones o restricciones impuestas a los algoritmos utilizados, como por ejemplo, carretera plana, textura uniforme, contornos demarcados u oclusiones de corta duración entre otros.

El uso de la estimación de un modelo, es decir, la inferencia de su comportamiento en tiempo futuro, permite compensar los vacíos en información, presentes por cambios drásticos en las condiciones con que se percibe la información visual de la carretera. El proceso de integración temporal asume una dinámica conocida que relacione los parámetros del modelo en el tiempo y una distribución de probabilidad asociada a la incertidumbre presente en las variables y mediciones observables del modelo. Con esta información es factible hacer la mejor aseveración posible sobre el comportamiento futuro del modelo de carretera dado que se tiene un histórico de observaciones con las cuales actualizar el sistema de inferencia.

Ahora bien, existen dos algoritmos generalizados para seguimiento de parámetros que son aplicados como etapa de integración temporal para los parámetros que representan un modelo de carretera, estos son el filtro de Kalman [12, 50, 56, 68] y el filtro de partículas [22, 43, 50, 90]. El filtro de partículas, a diferencia del clásico filtro de Kalman, permite realizar seguimiento a múltiples hipótesis de modelo simultáneamente.

2.4. Hitos del Desarrollo ADAS

En la revisión de la literatura relacionada con el desarrollo de los ADAS, se han encontrado trabajos de gran trascendencia, dada su repercusión en trabajos posteriores. A continuación se hará una breve referencia a los trabajos más relevantes que pueden ser considerados como hitos en la historia de la visión por computador aplicada a la percepción de carreteras.

2.4.1. Trabajos Clásicos

2.4.1.1. ALVIN

Uno de los trabajos pioneros en el campo de la navegación autónoma fue ALVIN [78], se trata de una red neuronal de 3 capas con topología back-propagation. Las neuronas de entrada se encuentran conectadas a 3 fuentes de información: una matriz bidimensional de 30X32 píxeles que detecta el nivel de intensidad del canal azul en una imagen RGB de vídeo, cuya función es capturar la carretera; la segunda fuente de información es una matriz bi-dimensional de 8X32 puntos provenientes de un sensor láser que mide la distancia entre punto y fuente; finalmente, la tercera fuente de información es la re-alimentación de una neurona en la capa de salida hacia la capa de entrada, ésta salida es entrenada con información de si la zona, correspondiente en la escena a la región de no carretera, es más clara o más oscura que la zona tipo carretera. Esta mezcla de fuentes de información es uno de los primeros intentos reportados de fusión sensorial.

2.4.1.2. GOLD

El algoritmo GOLD (Generic Obstacle an Lane Detection) [7], para detección simultánea de obstáculos y carriles fue desarrollado en el marco del proyecto Prometheus y probado en el vehículo de pruebas MOB-LAB. Se basa en el procesamiento paralelo de información visual estéreo y el análisis de la carretera en perspectiva inversa.

Para detectar los carriles presentes en la imagen en perspectiva inversa, primero se realiza un realzado de las marcas viales, y luego, se aplica un filtro morfológico denominado dilatación geodésica para lograr contornos de línea continuos y bien definidos. Una vez detectados los contornos de carril, se identifica la carretera según la posición de su línea central y del ancho predominante, obtenidos a través de un

histograma construido del análisis línea a línea de la imagen.

La detección de obstáculos, por su parte, se efectúa en modo estéreo sobre las imágenes calibradas y en perspectiva inversa. Una vez, las proyecciones de la carretera vistas por cada cámara, izquierda y derecha, son alineadas en orientación y desplazamiento, se obtiene la misma información de la escena. Entonces, al restar ambas imágenes, las diferencias que aparecen, se deben a las zonas que no son observadas por ambas cámaras a la vez, efecto de la oclusión resultante, consecuencia de capturas desde diferentes lugares físicos.

Por otro lado, basados en el principio de que dos líneas perpendiculares al plano de la carretera se transforman en dos líneas convergentes en perspectiva inversa, la parte trasera de los vehículos que se puede observar en la zona cercana de la carretera, cumplen el principio anterior, y por lo tanto, los obstáculos observados se transforman en un par de triángulos.

Para estandarizar el patrón de búsqueda en la imagen, los triángulos se unifican en un solo triángulo envolvente con vértice inferior en el promedio de los vértices individuales de cada triángulo. Ahora el paso a seguir es caracterizar la presencia de triángulos tipo obstáculo a través de un histograma polar o firma del objeto.

Un obstáculo en su representación de firma normalizada, se presenta como una señal de dos picos, la forma, magnitud y posición depende de múltiples factores como punto de vista, posición y condiciones de iluminación entre otros.

2.4.1.3. LANA

El Algoritmo LANA (Lane-finding in ANother domAin, por sus siglas en inglés), presentado en [52], es un detector de carriles que utiliza como característica de marcas viales, información del dominio de la frecuencia. Esta representación junto con un patrón de carril permite calcular una inferencia de tipo Bayesiano. Donde el modelo de carril sirve como información a priori, mientras que del conjunto de características se extrae la probabilidad de ajustar el modelo correctamente. La detección de carril consiste en encontrar un máximo global en la función de probabilidad a posteriori, a través de un algoritmo de búsqueda.

El algoritmo comienza por calcular las características en el dominio de la frecuencia que corresponden a la sumatoria de un sub-grupo de descomposiciones ortogonales del tipo coseno discreto, DTC, para un bloque de tamaño 8×8 .

El modelo utilizado es una parábola cuya definición se deduce a partir de la descomposición de la ecuación de una circunferencia por series de Taylor, proyectando

sus primeros términos sobre el plano de la imagen.

El artículo propone aplicar la regla de Bayes para obtener el modelo de carretera, H (la hipótesis del razonamiento), que mejor explique las características extraídas de la imagen de la carretera, E (la evidencia u observación del modelo). El modelo se extrae de maximizar su probabilidad a posteriori $P(H | E)$ en un espacio paramétrico.

2.4.2. Trabajos Recientes ADAS

2.4.2.1. VioLET

Sistema de asistencia a la conducción, centrado en la tarea de alerta por salida de carril (LDW por sus siglas en inglés), se basa en la estimación y seguimiento del carril de conducción a través de información visual [68]. Un diagrama general puede ser observado en 2.2.

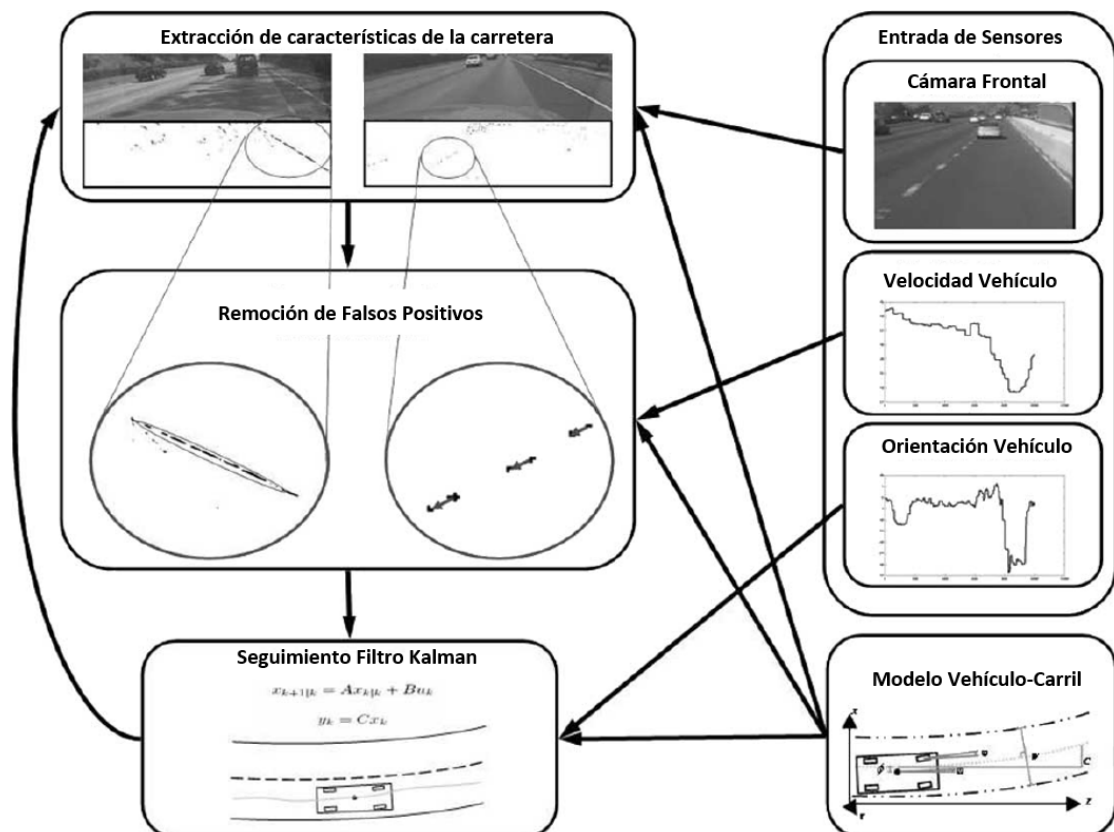


Figura 2.2: Diagrama general de la arquitectura VioLET. Fuente [68].

VioLET comienza con la definición del modelo de la carretera parabólico, de-

ducido a partir de la aproximación por serie de Taylor de una carretera circular, lo cual permite conservar el parámetro de curvatura C , ver ecuación (2.1).

$$\begin{aligned} X_{borde}(Z_s) &= \phi + \theta Z_s + CZ_s^2 + \frac{lW}{2(\theta + CZ_s)^2 + 2} \\ Z_{borde}(Z_s) &= Z_s - \frac{lW(\theta + CZ_s)}{2(\theta + CZ_s)^2 + 2} \end{aligned} \quad (2.1)$$

Donde Z_e es la distancia hacia el frente del vehículo, ϕ es el desplazamiento lateral, θ representa el ángulo del carril, C es la curvatura del carril, W representa el ancho de carril y l toma valores de 1 o -1 , según se trate de referir el carril izquierdo o derecho.

La extracción de características en este trabajo se basa en la aplicación de filtros dirigidos (steerable filter). Un filtro dirigido es de la forma mostrada en la ecuación (2.2).

$$G2^\theta(x, y) = G_{xx}\cos^2\theta + G_{yy}\sin^2\theta + G_{xy}\cos\theta\sin\theta \quad (2.2)$$

G_{xx} , G_{xy} y G_{yy} , son la convolución de la imagen con 3 kernel que se corresponden con las segundas derivadas parciales de una función campana de gauss bidimensional G .

La selección del ángulo del filtro θ se efectúa en la dirección de su máxima y mínima respuesta, componentes que se obtienen de los puntos de inflexión de la función presentada en la ecuación (2.2). La clasificación de cada pixel en la imagen $I(x, y)$, en las posibles clases de marca vial, depende de la relación obtenida entre las magnitudes del filtro en la dirección de máxima y mínima respuesta. Si la razón es cercana a 1, se trata de un reflector circular, si es muy grande, entonces se trata de una marca vial tipo línea.

El paso siguiente es la estimación de curvatura de la carretera, procedimiento necesario por cuanto las marcas viales circulares no son visibles después de 20m. Inicialmente se asume un perfil de borde de carretera obtenido con las marcas viales conocidas. La estimación de nuevos puntos de borde se realiza en perspectiva inversa casando lateralmente una plantilla adaptable de la carretera en un punto Z_s del perfil de borde con la carretera en un punto posterior Z_{s+k} . La posición de borde encontrada sirve para actualizar el modelo del perfil de borde hasta Z_{s+k} . La plantilla que se compara en cada nuevo punto Z_{s+k} se obtiene utilizando un promedio ponderado de los valores de intensidad de la plantilla anterior en Z_{s-k} con los valores de intensidad del área de carril en Z_s . La comparación se realiza por ajuste de mínimos cuadrados.

Una vez se ha estimado el perfil curvatura de los bordes de carril, es posible eliminar marcas viales en la siguiente imagen capturada, por la cercanía a dicho perfil. Para el caso de reflectores circulares, la remoción de falsos positivos se implementa a través del análisis de sus momentos estadísticos, además, se descartan aquellas marcas que no se encuentren en la posición estimada por el movimiento del plano de la carretera.

Finalmente, se usa un filtro de Kalman, útil cuando se trata de estimar variables inciertas relacionadas por un sistema dinámico conocido, sujeto igualmente a incertidumbre. En este caso, se plantean dos sistemas dinámicos en serie, el giro del vehículo Φ , que depende de la orientación, la curvatura de carril y la velocidad de las ruedas, sirviendo de estímulo para obtener el desplazamiento lateral del vehículo según el sistema de estados filtrado mostrado en las ecuaciones (2.3) y (2.4).

$$\begin{aligned} x_{k+1\wedge k} &= Ax_{k\wedge k} + Bu_k \\ y_k &= Mx_k \end{aligned} \quad (2.3)$$

$$\begin{aligned} x &= [\phi, \dot{\phi} = \tan \theta, \ddot{\phi}, W]^T \\ A &= \begin{bmatrix} 1 & v\Delta t & \frac{(v\Delta t)^2}{2} & 0 \\ 0 & 1 & v\Delta t & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ Bu_k &= [0, \Phi\Delta t, 0, 0] \\ M &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (2.4)$$

Las variables de estado son la posición lateral ϕ y sus derivadas junto con el ancho de carril W . La velocidad lateral $\dot{\phi}$ se asocia con el giro del vehículo para diferenciales de tiempo Δt muy pequeños, el cual coincide con la orientación de las marcas viales. La velocidad v a lo largo de Z_s para el vehículo se asume constante.

La información para actualizar el vector de estado se extrae de la aplicación de la transformada Hough para el caso de marcas viales tipo línea, para el caso de reflectores circulares, se obtiene de su análisis estadístico.

Finalmente, la plataforma VioLET es probada en diferentes escenarios a lo largo de 65km de autopista, donde las marcas viales son líneas, líneas-reflectores o solo reflectores, al igual que bajo diferentes condiciones de iluminación como amanecer,

medio día, atardecer, y noche. Las métricas utilizadas fueron valor medio y desviación típica del error de posición lateral, del orden de unidades de cm, y error medio de velocidad en la salida de carril del orden de décimas de cm.

2.4.2.2. Urban Challenge-MIT

A continuación se hace una síntesis del trabajo presentado en [42] por el equipo Talos, representante de MIT (Massachusetts Institute of Technology) en el Urban Challenge, cuarto finalista de los 6 que terminaron la competición, y cuya estrategia de percepción geométrica de la carretera se basó en visión por computador.

A grandes rasgos, el esquema utilizado para la estimación de carriles múltiples, adoptado por Talos, se muestra en la fig. 2.3. Allí, varios algoritmos para la detección del modelo de la carretera se ejecutan paralelamente, 2 basados en imágenes y uno basado en información láser. Cada algoritmo genera poly-líneas (Polylines), que son fusionadas en un sistema de referencia común, antes de realizar el seguimiento de parámetros del conjunto de carriles.

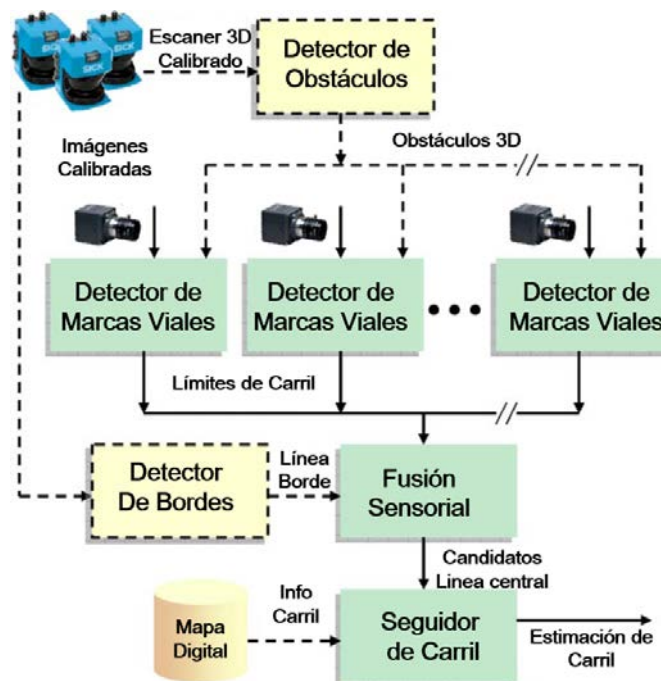


Figura 2.3: Diagrama general de la arquitectura Talos. Fuente [42].

Para disminuir el número de falsas detecciones de carril, se emplea un detector de obstáculos basado en sensores de distancia, velodyne y láser, con el fin de enmascarar en la imagen, las zonas donde se ubican los obstáculos, dado que se conoce la

calibración espacial de los sensores.

Los métodos para detección del modelo de la carretera basados en imágenes son:

- Detección de marcas viales por filtro unidimensional.

El primer método para la detección de marcas viales consiste en la aplicación de dos filtros unidimensionales (*matched filter*), de tamaño variable. El filtro es aplicado por convolución sobre filas y columnas independientemente, con el núcleo mostrado en la fig. 2.4, su objetivo es realzar las marcas viales.

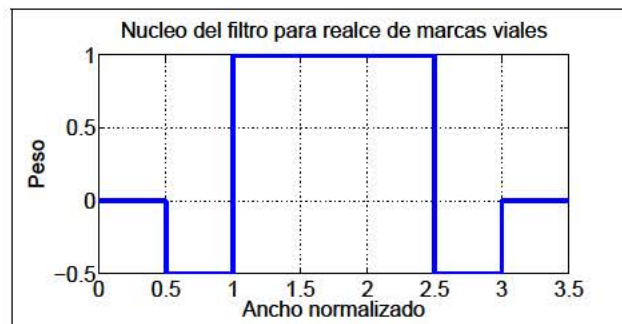


Figura 2.4: Núcleo del filtro direccional propuesto por Huang et al, [42].

A continuación se extraen los máximos locales por fila sobre la imagen filtrada horizontalmente y los máximos locales por columna sobre la imagen filtrada verticalmente, para crear una lista de posiciones, candidatos a ser parte de las marcas viales. Para cada característica se calcula la orientación de la línea subyacente encontrando la dirección de curvatura principal. Esta dirección se obtiene de la perpendicular del vector propio asociado al valor propio de mayor magnitud, extraído de la matriz hessiana valorada en el punto de la característica.

El paso siguiente, es obtener la transformación de distancia en relación al conjunto de características de línea halladas. El objetivo final de la propuesta es obtener la definición por parámetros de un conjunto de splines hermitianas cúbicas, que describan los bordes de carril presentes en la imagen. las curvas tipo spline se definen a través de un conjunto de puntos de control y las direcciones de la curva en dichos puntos. El resultado de cada una de las distintas fases del algoritmo se pueden apreciar en la fig. 2.5.

- Detección de marcas viales por contornos simétricos.

La hipótesis de partida es que las características que se encuentran pintadas en la carretera, son líneas bien definidas que contrastan con el fondo donde

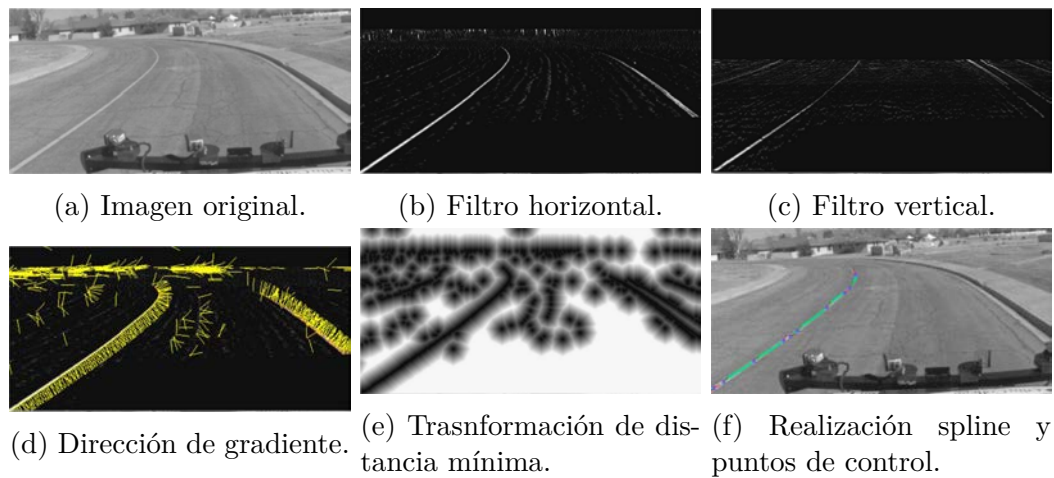


Figura 2.5: Algoritmo para la detección de marcas viales Huang et al [42].

se encuentran. La estrategia detecta en primer lugar bordes potenciales de la zona pintada en la carretera usando operaciones en el espacio del gradiente, luego se estima los centros de línea deseados, buscando por límites que encierren una región brillante, es decir, pares de borde paralelos y muy cercanos en el espacio de la imagen real y cuyos gradientes locales apunten entre sí. Cada centro, entra a formar parte de un contorno de curva. Finalmente, en una etapa de post-procesamiento de los contornos hallados, primero se unen fragmentos con curvatura coherente, luego se ajustan parábolas a dichos fragmentos, dividiéndolos en puntos de alta desviación y por último, se eliminan los fragmentos que no superen una longitud mínima.

2.4.2.3. Redes Neuronales Convolucionales

El uso de redes neuronales para la segmentación de imágenes y específicamente de carreteras, ha sido en las últimas décadas y hasta hace apenas un lustro, una tarea no sistemática. A principio de los años 90, en ALVIN [78] se puede observar cómo modificaciones de la topología estándar de una red neuronal eran requeridas para tratar el problema de cambios de iluminación. Trabajos posteriores en Aprendizaje de Máquina o Machine Learning, presentan la necesidad de diseñar descriptores invariantes hechos a medida según el tipo de carretera. Además, el hecho de entrenar las imágenes a partir de ventanas deslizantes con todas sus posibles variaciones, crea conexiones similares o redundancia a lo largo de la capa de entrada [57]. Las dificultades anteriores son la consecuencia de tratar cada ejemplo de entrenamiento de manera local, sub-regiones de interés que no tiene relación con la imagen de forma

global.

Las Redes Neuronales Convolucionales (Convolutional Neural Networks), CNN, solucionan los problemas antes mencionados, e incluyen la posibilidad de analizar información contextual de un pixel, es decir, su relación con el entorno e incluso con la imagen entera, cualidad que en trabajos como [35] se logra con descriptores de uso específico. Por ello, las CNN han sido utilizadas con éxito en múltiples aplicaciones comerciales como la lectura de cheques desarrollada por el grupo de investigación en redes neuronales en AT&T [58] o los sistemas de reconocimiento de texto desarrollados por Microsoft [86].

El interés en esta clase de algoritmos, que se encuentran enmarcados dentro del grupo denominado “Deep Learning”, se debe en gran medida a haber sido la técnica usada por [53], ganador del desafío en reconocimiento de objetos, ImageNet en 2010. En el campo del reconocimiento de carreteras, las CNN aparecen como fundamento teórico del algoritmo con más alto rendimiento del benchmark KITTI Vision [29] perteneciente al instituto de tecnología Karlsruhe y al instituto tecnológico Toyota en Chicago, el Marv3DNet (Multipurpose Deep Decoder Deconvolution Network), Red Deconvolucional basada en Decodificación Profunda Multi-proposito.

Un ejemplo de la aplicación de las Redes Neuronales Convolucionales en la segmentación de carreteras es el trabajo presentado por [2], donde se utilizan para aprender características de etiquetas ruidosas con el objetivo de recuperar la estructura 3D de la carretera. Un esquema de la arquitectura CNN utilizada se muestra en la fig. 2.6.

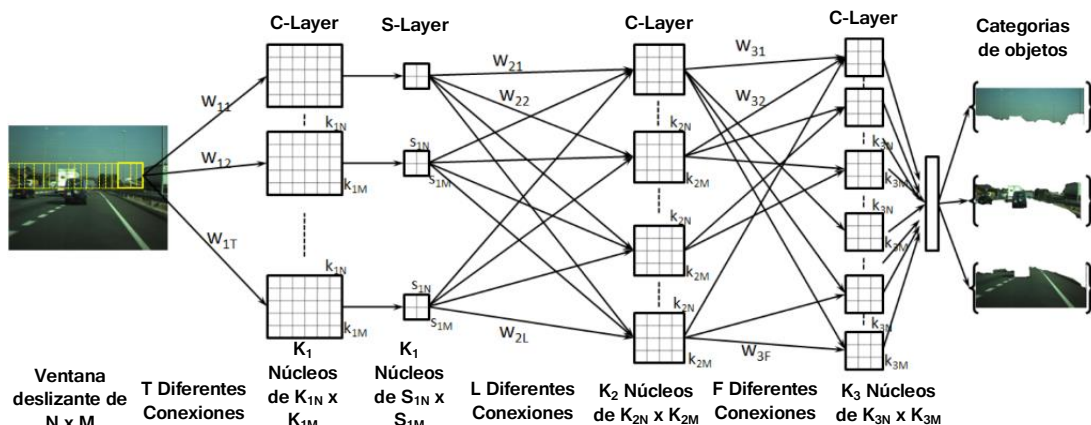


Figura 2.6: Esquema general de una CNN. Fuente [2].

La estructura consta de tres etapas, un banco de filtros convolucionales, una capa de transformación no lineal y una capa de muestreo o características espaciales. El

proceso de entrenamiento como es usual consiste en ajustar los núcleos y los pesos sinápticos dado un conjunto de entrenamiento, pero debido a que el etiquetado manual para un conjunto considerable puede ser intratable en tiempo, Alvarez et. al, propone usar un clasificador como generador automático de etiquetas y con ellos entrenar de forma supervisada la CNN.

Un esquema similar se propone en [70], la entrada a la red es un área de la imagen de 66×66 pixeles en tres canales de color, su salida es asociada a un área de 4×4 pixeles, donde se indica si el pixel pertenece o no a la carretera. La red contiene una capa de 32 filtros de convolución de 3×3 pixeles seguida por otra de 16 filtros de convolución sobre 1×1 pixel con el propósito de acelerar el proceso de aprendizaje, y una capa de promedio de 2×2 pixeles. Por último se encuentra una capa totalmente conectada de 1000 neuronas para terminar con una capa de dos neuronas, una por cada categoría de clasificación.

2.5. Conclusiones

La conducción semi-autónoma y completamente autónoma aún no son posibles a bajo costo y con la confiabilidad suficiente. Demostraciones como the Urban Challenge en 2006 [90] y the VisLab Intercontinental Autonomous Challenge en 2010 [6], en donde el estado de la técnica fue probada, evidenciaron problemas como la dificultad para obtener una calibración de instrumentos óptima, interferencias en las comunicaciones, dificultad para tratar con gran diversidad de escenarios y ausencia de mapas digitales en algunas zonas, entre otros aspectos, problemas que en el presente siguen solucionándose. Por tal razón, el desarrollo progresivo de los dispositivos avanzados de asistencia a la conducción, ADAS, será el camino a seguir hacia la conducción autónoma en las próximas décadas.

La detección de carriles es el núcleo de la mayoría de dispositivos ADAS comerciales. Allí, el uso de visión por computador en conjunto con posicionamiento GPS y reconstrucción tridimensional del entorno es una estrategia de percepción fiable. Así por ejemplo, cuando se cuenta con mapas digitales, la localización precisa del vehículo es posible a través de la lectura del GPS y las mediciones tomadas con una IMU (Inertial Measurement Unit). En los casos donde no se conoce información digital del entorno, el uso de un láser permite la detección del plano de la carretera y zona transitable de manera rápida y confiable. Así mismo, de la información visual derivada de la visión estéreo, se puede inferir el movimiento del vehículo (ego-

motion), la ubicación del plano de la carretera y zona transitable, al igual que de otros objetos importantes para el entendimiento de la escena, tales como las marcas viales, vehículos, peatones, etc. Por lo tanto, la suma de fuentes de información del entorno permitirá el entendimiento de la carretera del mismo modo que los seres humanos lo hacen, facilitando la tarea de planificación de trayectorias.

Los algoritmos para detección de carriles incluyen cuatro etapas estándar, pre-procesamiento, extracción de características de marcas viales, ajuste del modelo de los carriles e integración temporal o seguimiento de carriles, de allí el paso más importante es la extracción de características de forma robusta para detectar marcas viales en medio de sombras, cambios de iluminación y en condiciones de degradación y desvanecimiento de las líneas. En esta dirección son pocos y recientes los esfuerzos por mejorar el rendimiento de los detectores de carriles aún en este tipo de condiciones, por ejemplo, el trabajo de [59] propone el uso de filtros direccionales anisotrópicos para mejorar el rendimiento en marcas de carril de bajo contraste. En el presente trabajo se aborda el problema de detección de líneas degradadas a través de aprendizaje de máquina haciendo uso de descriptores hechos a medida y seleccionados a través de pruebas de rendimiento sobre secuencias de carretera tomadas sobre la plataforma IVVI 2.0 [66], así como también tomadas del repositorio KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago) [32].

Así mismo, recientemente las redes neuronales convolucionales o CNN (Convolutional Neural Networks), algoritmos representativos del campo de investigación denominado Deep Learning, han hecho su aparición para solucionar el problema de segmentación de objetos de forma contextual [75]. Estas arquitecturas para clasificación, no solo permiten segmentar cada zona de la imagen según un patrón de referencia, sino que, tanto las características con que describe las sub-regiones de la imagen, como las relaciones entre cada sub-región con el entorno a nivel global de la imagen, se auto configuran de forma óptima a través de un proceso de entrenamiento. Esta robustez hace posible segmentar carreteras con líneas degradadas sin tener que diseñar descriptores únicos para cada tipo de carretera o condición de la imagen. Sin embargo, el desafío que queda por superar, es lograr que este tipo de algoritmo de clasificación sea lo suficientemente rápido sin pérdida de robustez como para que pueda ser usado en tiempo de aplicación ADAS, alrededor de 50ms o menos, tiempo suficiente como para procesar toda la información visual de una cámara estándar, 20 fps (frames per second) sin la necesidad de utilizar GPUs (graphics processing units)

de última generación con los consecuentes altos costos derivados que imposibiliten llevar esta tecnología a nivel comercial en vehículos de gama media.

Capítulo 3

Reconstrucción Tridimensional de la Carretera

3.1. Visión Estéreo

La percepción de un entorno tridimensional se logra a partir de la captura sincronizada de dos imágenes provenientes de un par de cámaras separadas por una distancia conocida b , o línea base, como se observa en fig. 3.2a. Un ejemplo de este tipo de configuración es el utilizado en el desarrollo de la presente investigación, el dispositivo trinocular Bumblebee XB3, fabricado por Point Grey, cuyas especificaciones se muestran en la Tabla 3.1. Ésta cámara puede ser configurada para capturar dos imágenes en modo cercano $b = 12cm$, ó lejano $b = 24cm$ ó tres imágenes a la vez.

El proceso continúa con la corrección de distorsiones radiales y tangenciales debidas defectos en la fabricación de los lentes y su correspondiente ensamble. Este tipo de corrección no lineal es de la forma mostrada en la ec. (3.1).

$$\begin{aligned} \begin{bmatrix} x_d \\ y_d \end{bmatrix} &= (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \begin{bmatrix} x_r \\ y_r \end{bmatrix} \\ \begin{bmatrix} x_c \\ y_c \end{bmatrix} &= \begin{bmatrix} x_d \\ y_d \end{bmatrix} + \begin{bmatrix} 2p_1 x_d y_d + p_2 (r^2 + 2x_d^2) \\ p_1 (r^2 + 2y_d^2) + 2p_2 x_d y_d \end{bmatrix} \end{aligned} \quad (3.1)$$

Donde r es la componente radial del punto $[x_r, y_r]^T$, localización de un pixel con distorsión radial, y k_1, k_2, k_3, p_1, p_2 , son coeficientes de corrección.

Una vez se ha ajustado el error de proyección sobre el plano de cada imagen debido a las lentes y a su ensamble mecánico, se procede a la alineación de los

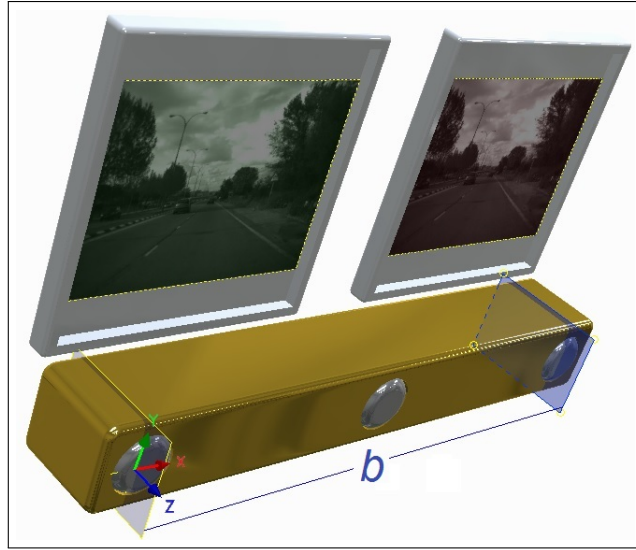


Figura 3.1: Cámara Estéreo XB3.

planos de las imágenes derecha e izquierda para lograr correspondencias en el nivel de intensidad sobre la misma coordenada fila en ambas imágenes. Este cambio de posición es el resultado de aplicar la transformación L para rectificación del par estéreo.

$$\begin{bmatrix} x_{pr} \\ y_{pr} \end{bmatrix}^{i,d} = L \begin{bmatrix} x_p \\ y_p \end{bmatrix} \quad (3.2)$$

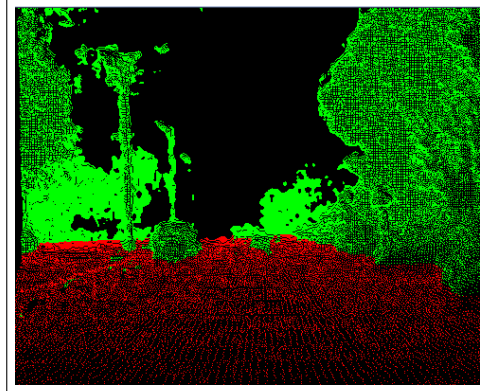
A continuación, con las imágenes de la cámara rectificadas izquierda y derecha I_{pr}^i y I_{pr}^d , es posible crear su respectivo mapa de disparidad. Para ello, se calcula en cada posición de la imagen de referencia $p_d = [u, v]_d$, la posición del pixel correspondiente al mismo punto en la escena $P_{3D} = [X, Y, Z]$, proyectado sobre la segunda imagen del par estéreo $p_i = [u^*, v^*]_i$. La disparidad d se define entonces como la distancia entre posiciones proyectadas desde un punto en la escena 3D, sobre ambas imágenes rectificadas.

$$d = dist(p_d, p_i) \quad (3.3)$$

El proceso para el cálculo de la disparidad, consiste en encontrar las posiciones correspondientes donde haya coincidencia en el nivel de intensidad entre las imágenes del par estéreo. Existen varios métodos para tal efecto: locales, semi-globales y globales; estos se diferencian en cuanto a la complejidad de la búsqueda que implementan, por ejemplo, los más simples como el SDA, suma de diferencias absolutas,



(a) Imagen derecha e izquierda sobrepuestas. (b) Mapa de disparidad, método SMG.



(c) Nube de puntos de la escena.

Figura 3.2: Reconstrucción tridimensional de la escena.

se basa en el cálculo de la correlación de intensidad en las cercanías a cada pixel p . Para el caso de estudio, se utiliza el algoritmo de correspondencia semi-global, SMG [39] por sus siglas en inglés, Semiglobal Matching Method, una idea basada en la minimización de una cierta función de energía $S(\mathbf{p}, d) = \sum_r L(\mathbf{p}, d)$, para cada posición p , que a su vez es la suma de las funciones de energía a lo largo de todos los caminos r que finalizan en el pixel p . Con ello se consigue una solución continua en 2D.

Finalmente, conocidas las correspondencias asociadas a la posición de cada pixel en la imagen de referencia, usualmente, la imagen derecha, es posible recuperar las coordenadas $P_{3D} = [X, Y, Z]$, de cada punto en la escena 3D, por triangulación, asumiendo el modelo de pin-hole, para la proyección de una escena sobre un sensor óptico.

$$Z = \frac{fb}{d}; X = \frac{Z(v_d - cv_d)}{f}; Y = \frac{Z(u_d - cu_d)}{f} \quad (3.4)$$

Donde f es la distancia focal de la cámara, cu_d y cv_d , son los centros ópticos, en cada eje, de la cámara de referencia.

3.2. Auto-calibración de parámetros extrínsecos

El proceso de auto-calibración para una cámara estéreo a bordo de un coche, comienza por la representación del entorno, la carretera, como un conjunto de puntos en el espacio 3D asociados a la posición de cada pixel en la imagen de referencia, esta reconstrucción, descrita en el sistema de referencia de la cámara $\{c\}$, se denomina nube de puntos o *PC*, por sus siglas en inglés, Point Cloud, como se puede apreciar en la fig. 3.2c.

La hipótesis de partida, es la asunción de que en el PC existe un plano que contiene una cantidad mayoritaria de puntos, y que corresponde al plano de la carretera. Por tanto, al aplicar un algoritmo para estimación paramétrica de modelos, como RANSAC [?], ajustado a la detección de planos en el espacio, es posible, obtener una definición en la forma de la ec.3.5, con parámetros $[a, b, c, d]$, ó en la forma hesiana de un plano, ec.3.6, donde \vec{n} , es un vector unitario perpendicular al plano detectado, y h , determina la distancia desde plano al sistema de referencia del *PC*.

$$\pi(\vec{p}) := ax + by + cz + d = 0 \quad (3.5)$$

Especificaciones XB3	
Tipo sensor de imagen	Escaneo progresivo CCD, <i>Sony</i> [®] 1/3" ICX445 (1280x960 max pixeles) 3,75 μ m pixeles cuadrados
Línea base	12 cm y 24 cm
Distancia Focal	3,8mm with 66° HFOV or 6mm with 43° HFOV
Apertura	f/2,0 (2,5mm and 3,8mm focal length), f/2,5 (6,0mm focal length)
Convertidor A/D	12 – bit
Balance de blanco	Manual (Modelo Color)
Tasa de cuadros	16 FPS
Requerimientos de voltage	8-30V
Consumo de potencia	4W@12V
Ganancia	<i>Automatica/Manual</i>
Obturador	<i>Automatica/Manual</i> , 0,01ms to 66,63ms @ 15FPS
Disparo	DCAM v1.31 Trigger Modes 0, 1, 3, y 14
Relación señal a ruido	54dB
Dimensiones	277 x 37 x 41.8mm
masa	505 g
Lentes	microlentes 3 x M12
Emisión EM	Ajustado a CE y Parte 15 Clase A de FCC
Temperatura operación	de 0° a 45°C

Cuadro 3.1: Especificaciones cámara estéreo

$$\pi(\vec{p}) := \vec{n} \cdot \vec{p} = h \quad (3.6)$$

3.2.1. Estimación del plano de la carretera

3.2.1.1. Reducción de la reconstrucción 3D

La reconstrucción tridimensional de una escena a partir de un par estéreo, genera un conjunto de puntos, PC , de tamaño igual a la cantidad de píxeles de cada imagen. Para una resolución típica de 480 líneas horizontales por 640 líneas verticales, el número de puntos en la escena es de 307200 puntos. Dado que los algoritmos ADAS deben ser ejecutados en tiempo real de aplicación, alrededor de 10 a 15 ciclos por segundo, la reducción del PC , sin afectar su uniformidad espacial, permite acelerar procesos posteriores, como la búsqueda del plano de la carretera.

Para reducir el entorno 3D se proponen dos alternativas, la primera consiste



Figura 3.3: Escena de referencia para reducción 3D.

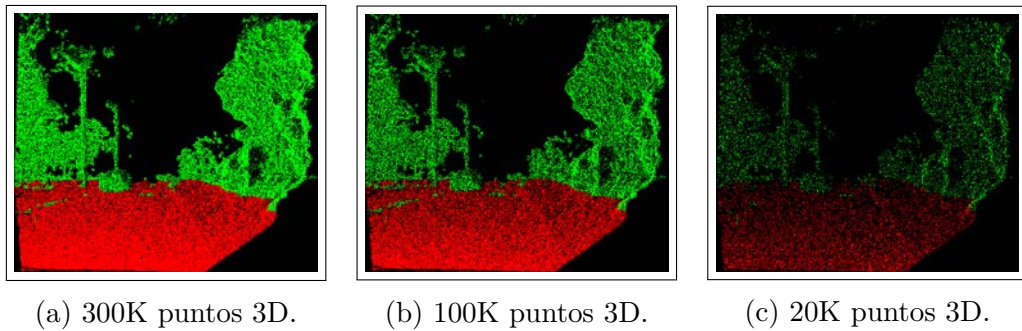


Figura 3.4: Reducción por muestreo uniforme.

en realizar un muestreo con una distribución de probabilidad uniforme del PC . La segunda alternativa consiste en dividir el espacio del PC en una rejilla tridimensional de cubos con un tamaño fijo. Dentro de cada cubo, se calcula el centroide de los puntos que contiene, y dicho punto es la simplificación del PC en el cubo al que corresponde. La aplicación de una u otra alternativa, dependerá del tiempo de ejecución para crear el entorno reducido, dado que como se concluye en [33], no se puede afirmar que los métodos determinísticos de muestreo, regularización, frente a los métodos pseudo-aleatorios, como el muestreo uniforme, mejoren el rendimiento de algoritmos cuya finalidad sea la exploración gradual de un espacio finito.

En la fig. 3.4. se presenta la reconstrucción de la escena correspondiente a la imagen de la Fig, 3.3. para diferentes resoluciones de muestreo, usando una distribución de probabilidad uniforme. Así mismo, en la fig. 3.5. se presenta la reconstrucción de la escena para diferentes tamaños de cubo o voxel en mm.

3.2.1.2. Métodos de estimación paramétrica

El objetivo de un algoritmo de estimación paramétrica es obtener el conjunto de parámetros, θ , que describe un modelo, $\mathcal{M}(\theta)$, a partir de un conjunto de observa-

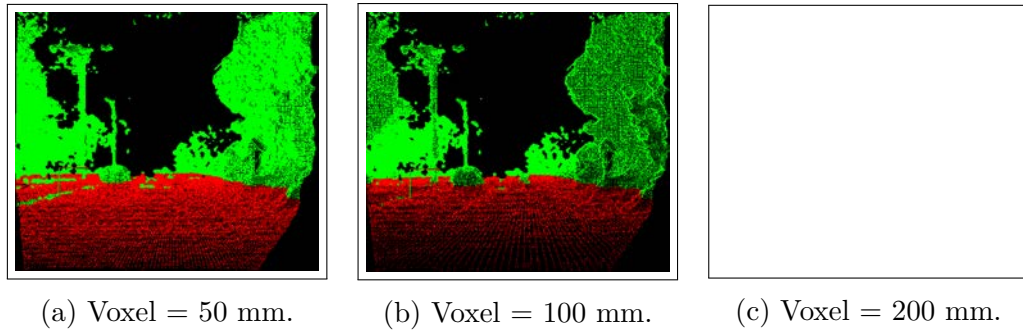


Figura 3.5: Reducción por regularización.

ciones, $m_i = [x_{1i}, x_{2i}, \dots, x_{pi}]^T$ con $i = 1, 2, \dots, n$, contaminadas por ruido gaussiano, que pertenecen al modelo llamadas inliers, y por observaciones que no pertenecen al modelo, llamadas outliers. Un outlier es un dato que no satisface el modelo descrito por un conjunto de parámetros propuesto dentro de algún umbral de error, que define la máxima desviación atribuible al efecto del ruido.

Existen diversos métodos para la estimación de modelos, llamados estimadores, dentro de los cuales, los más utilizados son:

LMedS Uno de los métodos que se utilizará para la estimación del plano de la carretera es el denominado LMedS, Least-Median-of-Squares. Como se describe en [82], se trata de la minimización de los residuos, $r_i = \theta m_i$, al cuadrado, que resultan al probar puntos del espacio de búsqueda con un vector de parámetros dado. Su definición se muestra en la ec. 3.7

$$\min_{\theta} \text{med}_i r_i^2 \quad (3.7)$$

Los pasos intermedios del algoritmo son como se explican a continuación:

- 1 Una técnica del tipo Monte Carlo es utilizada para seleccionar m subconjuntos de p elementos contenidos en m_i , dado que p es la dimensionalidad del modelo.
- 2 Para cada subconjunto j , se resuelve un sistema de p ecuaciones lineales y se obtiene un modelo paramétrico estimado θ_j .
- 3 Para cada θ_j , se calcula la mediana de los residuos cuadráticos.
- 4 Se almacena iterativamente el valor mínimo de la mediana a través de los m θ_j calculados.

Finalmente, se determina el valor de m , ec.3.9, es decir, cuantas veces se obtendrá un nuevo modelo en busca de un mínimo aceptable. Este cálculo se realiza asumiendo un porcentaje máximo de outliers, ϵ , en el conjunto de datos. Su complemento, $1 - \epsilon$, será la probabilidad de escoger un inlier, de esta forma, $(1 - \epsilon)^p$, es la probabilidad de que en un conjunto de p puntos, todos sean inliers, su complemento será la probabilidad de que el conjunto de p muestras seleccionado, solo contenga outliers, siguiendo esta lógica, $[1 - (1 - \epsilon)^p]^m$, es la probabilidad de que en m selecciones de p puntos, todas estén compuestas por outliers, y finalmente, su complemento indica la probabilidad de que al menos una selección de m ensayos, este compuesta por p inliers, con lo cual, es posible estimar un modelo correcto con una probabilidad P , ec.3.8.

$$P = 1 - [1 - (1 - \epsilon)^p]^m \quad (3.8)$$

$$m = \frac{\log(1 - P)}{\log[1 - (\epsilon)^p]} \quad (3.9)$$

Adicionalmente, con el propósito de hacer el algoritmo robusto a ruido gaussiano presente en el conjunto de datos, se utiliza un criterio de selección de puntos para el cálculo de la mediana, se trata de estimar iterativamente una desviación estándar de los datos en función de la mínima mediana obtenida en cada estimación de modelo, de tal forma, que si un residuo se encuentra fuera de dicha desviación típica, entonces no es considerado en el cálculo de la mediana. Este proceso convierte el LMedS originalmente propuesto, en un estimador robusto a ruido gaussiano, [69].

RANSAC Otra familia de métodos para estimación paramétrica, es el algoritmo RANSAC por sus siglas RANdom Sample And Consensus, junto a sus diferente variaciones. Fue propuesto por Fischler y Bolles en 1981, [11]. Consta de dos pasos que se repiten continuamente hasta alcanzar un resultado óptimo.

- 1 Generación de hipótesis: consiste en seleccionar un conjunto de muestras mínima llamado MSS (Minimal Sample Set), de tamaño k , del conjunto de datos total para calcular un modelo paramétrico, $\mathcal{M}(\theta)$.
- 2 Prueba: en este paso se revisa cuales elementos del conjunto total de datos, D , son consistentes con el modelo estimado a partir del MSS. Para ello, se calcula la distancia ortonormal de una observación m_i al modelo, $e_M(m_i, \theta)$ y se compara con un umbral δ , que limita con una probabilidad P_{inlier} , el

error generado por un inlier contaminado por ruido gaussiano $\eta \sim N(0, \sigma_\eta I)$. El conjunto obtenido se denomina consenso (CS), ec. 3.10.

$$CS(\theta) = \{m_i \in D : e_M(m_i, \theta) \leq \delta\} \quad (3.10)$$

El umbral δ se puede estimar como la distancia euclidiana acumulada por el ruido gaussiano de cada componente en el espacio del modelo, a demás, si el ruido gaussiano se normaliza respecto a su desviación típica, $\frac{\eta_i}{\sigma_\eta} \sim N(0, 1)$, ec. (3.11), la función de probabilidad resultante será una función de distribución acumulada de una variable aleatoria $X = \sum_{i=1}^p (\frac{\eta_i}{\sigma_\eta})^2$ con distribución de probabilidad χ_p^2 .

$$P_{inlier} = P[e_M(m_i, \theta) \leq \delta] = P\left[\sum_{i=1}^p \eta_i^2 \leq \delta^2\right] = P\left[\sum_{i=1}^p \left(\frac{\eta_i}{\sigma_\eta}\right)^2 \leq \left(\frac{\delta}{\sigma_\eta}\right)^2\right] \quad (3.11)$$

Por lo tanto el umbral δ puede ser obtenido según la ec. 3.12.

$$\delta = \sigma_\eta \sqrt{F_{\chi_p^2}^{-1}(P_{inlier})} \quad (3.12)$$

De manera similar a otros estimadores, es necesario calcular el número de veces que se debe escoger un MSS. Para ello se sigue el siguiente razonamiento, q es la probabilidad de seleccionar un MSS libre de outliers, entonces $1 - q$ es la probabilidad de escoger un MSS con al menos un outlier, si se realizan m experimentos, la probabilidad de que en todos se tome al menos un outlier es $(1 - q)^m$, su complemento será la probabilidad, ϵ , ec. 3.13 de que al menos un MSS de los m ensayos, genere un modelo correcto, esta probabilidad fija una cota inferior que puede ser seleccionada como una entrada al algoritmo. De esta definición se obtiene el número estimado de ensayos para obtener un modelo correcto con una certeza de al menos $1 - \epsilon$.

$$1 - (1 - q)^m \leq 1 - \epsilon \quad (3.13)$$

$$\hat{m} = \left\lceil \frac{\log(\epsilon)}{\log(1 - q)} \right\rceil \quad (3.14)$$

Para el cálculo de q , se asume que los inliers están libres de ruido y cualquier MSS genera un modelo fiable, a demás, si todas las observaciones tienen la misma probabilidad de ser seleccionadas, entonces, la ec. 3.15, muestra la probabilidad aproximada a posteriori para q , si $N_i \gg k$ y $N \gg k$, N el número total de datos y

N_I , el tamaño del consenso CS.

$$q = \frac{\binom{N_I}{k}}{\binom{N}{k}} = \frac{N_I!(N-k)!}{N!(N_I-k)!} \approx \left(\frac{N_I}{N}\right)^k \quad (3.15)$$

Como inicialmente no se conoce el número de inliers N_i o tamaño de CS, se toma un número grande y en la medida que se halle un CS, es posible calcular q real.

El algoritmo RANSAC termina cuando se han ejecutado un número mayor de iteraciones que el calculado según la ec. 3.14.

Finalmente, el criterio para escoger un conjunto de parámetros que definan un modelo mejor, se determina por un índice de calidad para el consenso. Esta definición determina las variaciones del algoritmo, RANSAC utiliza la cardinalidad del conjunto consenso, es decir, su tamaño.

Adicionalmente, el algoritmo RANSAC puede ser visto como un problema de optimización que minimiza la función de coste mostrada en la ec. 3.16.

$$C_M(D, \theta) = \sum_{i=1}^N \rho(m_i, \mathcal{M}(\theta))$$

$$\text{Donde : } \rho(m_i, \mathcal{M}(\theta)) = \begin{cases} 0 & |e_M(m_i, \theta)| \leq \delta \\ 1 & \text{En otro caso.} \end{cases} \quad (3.16)$$

En [?] se propone una función de calidad para el consenso mas sensible que tiene en cuenta la adaptación al modelo, esta modificación se encuentra en la ec. 3.17.

$$\rho(m_i, \mathcal{M}(\theta)) = \begin{cases} e_M(m_i, \theta) & |e_M(m_i, \theta)| \leq \delta \\ \delta & \text{En otro caso.} \end{cases} \quad (3.17)$$

Así mismo, otra variación del algoritmo RANSAC es PROSAC, propuesto por [17]. En esta modificación, el MSS se toma de un subconjunto de datos seleccionados por un índice de calidad. Las muestras que tienen mayor probabilidad de no estar contaminadas de ruido son examinadas primero en la etapa de generación de hipótesis. PROSAC toma el mismo conjunto de datos con el tiempo, pero en diferente orden.

3.2.1.3. Estimación de un plano

La particularización del algoritmo planteado para la estimación de un plano solo requiere la definición de la función que genere el modelo a partir de un MSS y la función que mide la distancia ortonormal, de un punto del conjunto de datos de

búsqueda al modelo.

Para el caso de un plano, el MSS se forma con 3 puntos del espacio, con los cuales se puede obtener un modelo como se muestra a continuación.

Primero se obtienen 3 puntos, que se asume, están sobre el plano, $\pi(X)$: P_1 , P_2 y P_3 , con los cuales se forma una base $[\vec{U}_p, \vec{V}_p] = [(P_2 - P_1), (P_3 - P_1)]$. A continuación, se define un vector genérico $\vec{P}_p = P_x - P_1$. Si los tres vectores con origen en P_1 : \vec{u}_p, \vec{v}_p , y \vec{P}_p , pertenecen al plano, entonces son coplanares y por tanto el determinante de la matriz formada por ellos como columnas o filas debe ser igual a cero, es decir:

$$\begin{vmatrix} \vec{P}_p & \vec{U}_p & \vec{V}_p \end{vmatrix} = 0 \quad (3.18)$$

Con ello se llega a la siguiente expresión:

$$P_{px}(U_{py}V_{pz} - U_{pz}V_{py}) + P_{py}(U_{pz}V_{px} - U_{px}V_{pz}) + P_{pz}(U_{px}V_{py} - U_{py}V_{px}) = 0 \quad (3.19)$$

Si $a = (U_{py}V_{pz} - U_{pz}V_{py})$, $b = (U_{pz}V_{px} - U_{px}V_{pz})$, y, $c = (U_{px}V_{py} - U_{py}V_{px})$, entonces:

$$P_{xx}a + P_{xy}b + P_{xz}c - (P_{1x}a + P_{1y}b + P_{1z}c) = 0 \quad (3.20)$$

La ec.3.20 es la definición de la ec.3.5 conocido el MSS.

Finalmente, para la implementación del algoritmo RANSAC aplicado a un plano, es necesario conocer la distancia de un punto del entorno 3D al modelo, en este caso, el plano definido por el MSS seleccionado. La función de distancia se obtiene como la mínima distancia entre un punto en el espacio P_x y un plano $\pi(X)$, el método utilizado para su deducción, son los multiplicadores de Lagrange, la conclusión se muestra en la ec.3.21.

$$d(P_x, \pi(X)) = \frac{|aP_{xx} + bP_{xy} + cP_{xz} + d|}{\sqrt{a^2 + b^2 + c^2}} \quad (3.21)$$

3.2.2. Extracción de extrínsecos

Los parámetros hallados en la definición del plano estimado, ec.3.6, se pueden interpretar como: \vec{n} , el vector unitario perpendicular a la carretera, que define los ángulos de orientación, cabeceo (α) y alabeo (σ) (pitch y roll) de la cámara por posición relativa, y así mismo, h , determina la distancia desde la carretera al sistema de referencia de la cámara, la altura del sensor en la coordenada z correspondiente a un sistema de referencia solidario con el plano tierra $\{m\}$, como se observa en la

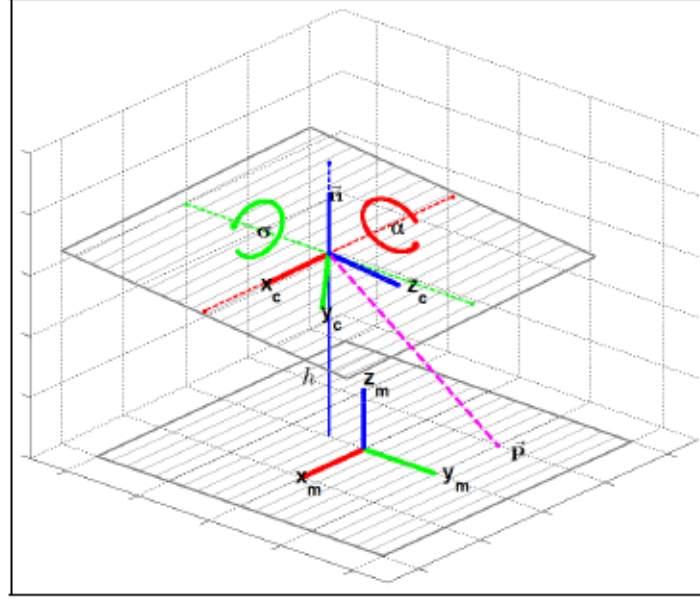


Figura 3.6: Representación gráfica de los parámetros extraídos de la forma hesiana del plano de la carretera.

fig. 3.6.

Ahora bien, si se conoce la posición del vector normal al plano de la carretera definido en el sistema de referencia de la cámara, ${}^c\vec{n}$, es posible definir dicho vector, como rotaciones sucesivas del eje Z del sistema de referencia de la carretera, ${}^m\vec{Z}$, ec.3.22 .La primera rotación ocurre en la dirección absoluta X de la carretera, un ángulo α , correspondiente al cabeceo de la cámara, y la segunda rotación ocurre con respecto al eje Y absoluto de la carretera, un ángulo σ , correspondiente al alabeo de la cámara.

$$\begin{aligned} {}^c\vec{n} &= {}^c R_m {}^m\vec{Z} \\ {}^c\vec{n} &= R_x(\alpha)R_y(\sigma) {}^m\vec{Z} \end{aligned} \quad (3.22)$$

$$\begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} = \begin{bmatrix} c\sigma & 0 & s\sigma \\ s\alpha s\sigma & c\alpha & -s\alpha c\sigma \\ -c\alpha s\sigma & s\alpha & c\alpha c\sigma \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.23)$$

$$\begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} = \begin{bmatrix} s\sigma \\ -s\alpha c\sigma \\ c\alpha c\sigma \end{bmatrix} \quad (3.24)$$

De la Ec. 3.24, se puede calcular el ángulo de alabeo σ y cabeceo α , así:

$$\begin{aligned}\sigma &= \arcsin(n_1) \\ \alpha &= \operatorname{atan}(-n_2, n_3)\end{aligned}\tag{3.25}$$

la determinación del ángulo de rotación con respecto al eje normal al plano de la carretera, así como el desplazamiento en X e Y, sobre dicho plano, se pueden extraer mediante un procesamiento posterior, donde sean detectados objetos de referencia en capturas consecutivas, dicho procesamiento se conoce como odometría visual.

3.2.3. Análisis de estabilidad de extrínsecos

La estabilidad de los parámetros extrínsecos es fundamental para poder preservar la coherencia visual que existe en imágenes capturadas consecutivamente dentro de un entorno, especialmente cuando se realiza un cambio de perspectiva tal y como se describe en el apéndice A.2.

Por tanto, es importante analizar y valorar la fiabilidad en la estimación de los parámetros que definen la posición de la cámara. La siguiente metodología muestra los pasos que se abordan con el ánimo de buscar los algoritmos más convenientes para una aplicación ADAS, así como, los ajustes necesarios para mitigar el impacto del ruido inherente a la captura de las imágenes y su reconstrucción tridimensional.

3.2.3.1. Estabilidad en una secuencia estática

En una secuencia estática de 100 fotogramas, similares a la imagen que se aprecia en la fig. 3.7, se compara el comportamiento de los 4 métodos de estimación paramétrica descritos en la Sec.3.2.1.2, para los dos tipos de reducción: muestreo uniforme y regularización, y con diferentes resoluciones. El objetivo del experimento es encontrar el algoritmo que cumpla con la mejor precisión en un tiempo de cómputo que se ajuste con el requerimiento de velocidad de la aplicación. La precisión en la medición, se observa calculando la desviación típica sobre el conjunto de capturas, como se muestra en la fig. 3.8 y en la fig. 3.9. Así mismo, a través de la desviación típica, es posible modelar el ruido inherente a la medición según el estimador utilizado y según el tipo de reducción del entorno.

El resultado de este experimento muestra cómo la desviación típica y por tanto, la precisión de todas las mediciones, así como con el tiempo de ejecución, es inferior para el caso de realizar la reducción del entorno tridimensional por muestreo uniforme, por tanto, el análisis posterior se centrará en las gráficas de la fig. 3.8. Allí,



Figura 3.7: Secuencia estática capturada desde IVVI 2.0.

el siguiente hallazgo encontrado en la medición de extrínsecos, es que el estimador LMedS exhibe la precisión más baja para todas las resoluciones de muestreo uniforme, aunque con un coste computacional significativamente más alto de un orden de magnitud, comparado con los otros estimadores del estudio. Así mismo, la precisión alcanzada por todos los métodos de estimación paramétrica, no muestra una tendencia decreciente en función de la resolución de muestreo y por tanto, se escoge la mínima resolución posible como mejor opción.

El estimador RANSAC al ser analizado bajo las opciones seleccionadas anteriormente, reducción por muestreo uniforme y mínima resolución de 2000 puntos, exhibe un precisión similar a LMedS, pero con tiempos de ejecución menores, es decir, a mínima resolución, el estimador RANSAC exhibe un tiempo de ejecución de 1.7ms frente a 44ms para LMedS. Por lo tanto, el método que resulta con la mejor relación tiempo - precisión, es el estimador RANSAC. En la tabla 3.2 se consolida los valores de precisión alcanzada con todos los métodos para la menor tasa de resolución por muestreo uniforme.

Precisión @ Muestreo Uniforme 2K Puntos				
Métodos \ Parámetro	Altura (mm)	Pitch (°)	Roll (°)	Tiempo (ms)
LMEDS	5.62	0.068	0.184	43.88
MSAC	8.94	0.113	0.366	1.97
PROSAC	11.76	0.139	0.512	1.39
RANSAC	9.33	0.109	0.368	1.71

Cuadro 3.2: Rendimiento de los estimadores usados en el experimento de análisis de estabilidad en una secuencia estática con reducción por muestreo uniforme de 2000 puntos sobre la secuencia de la fig.3.7

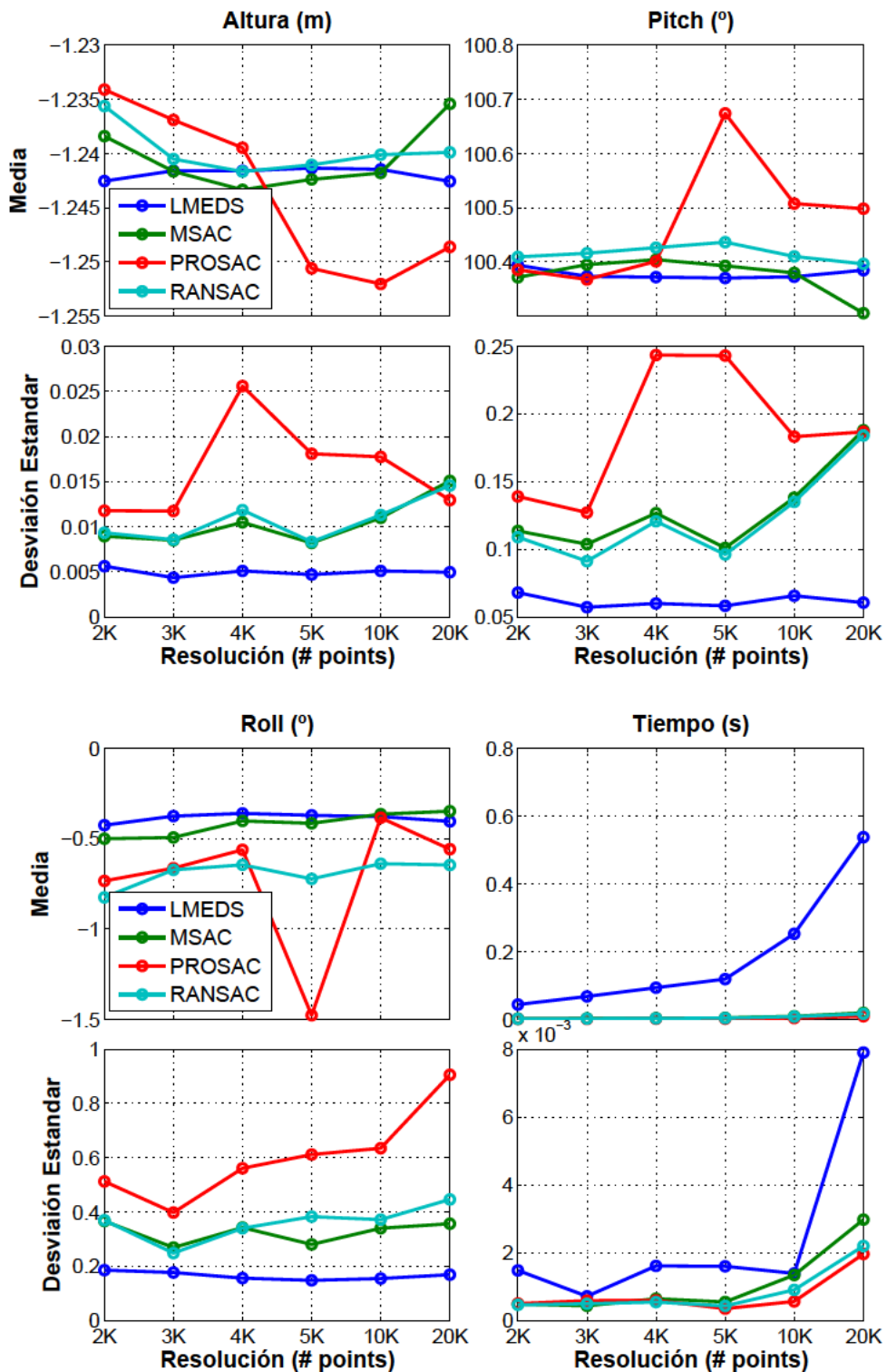


Figura 3.8: Comparación de métodos para estimación del plano de la carretera, en función de la resolución de muestreo.

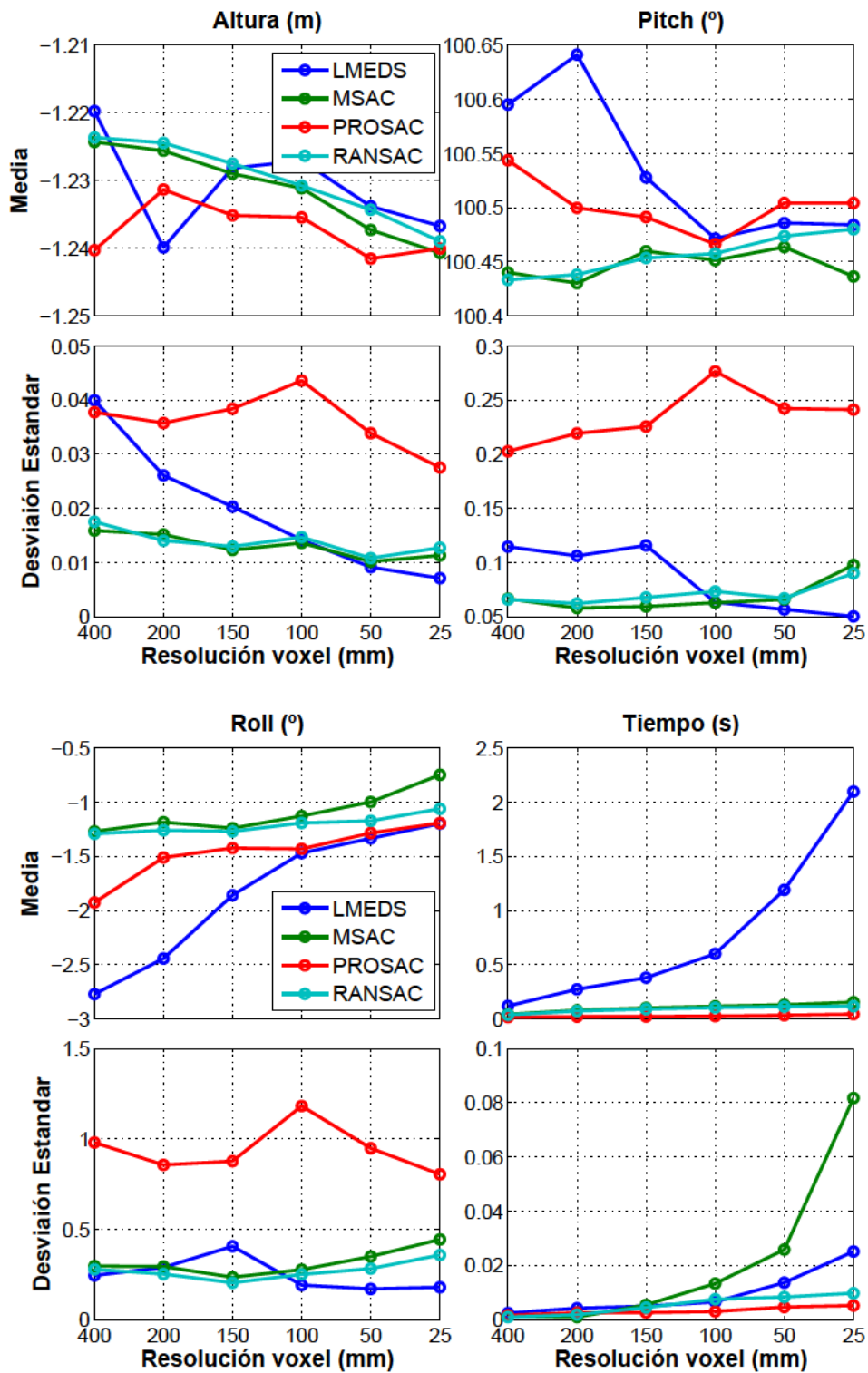


Figura 3.9: Comparación de métodos para estimación del plano de la carretera, en función del tamaño de voxel utilizado en la regularización del entorno.

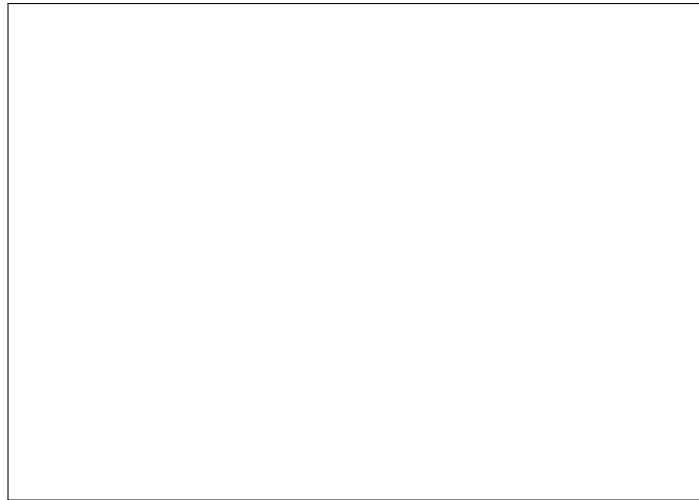
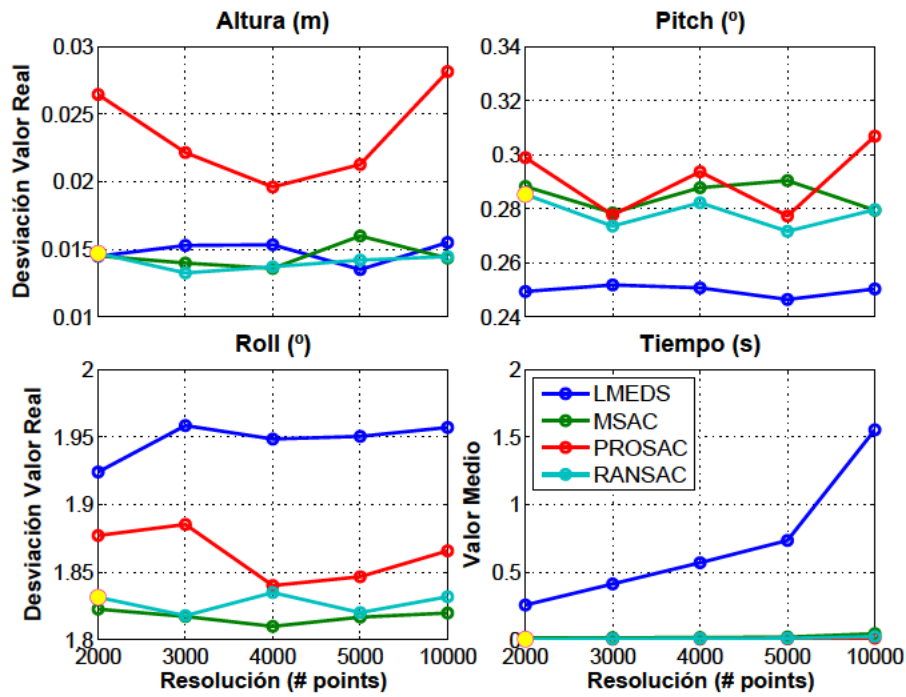


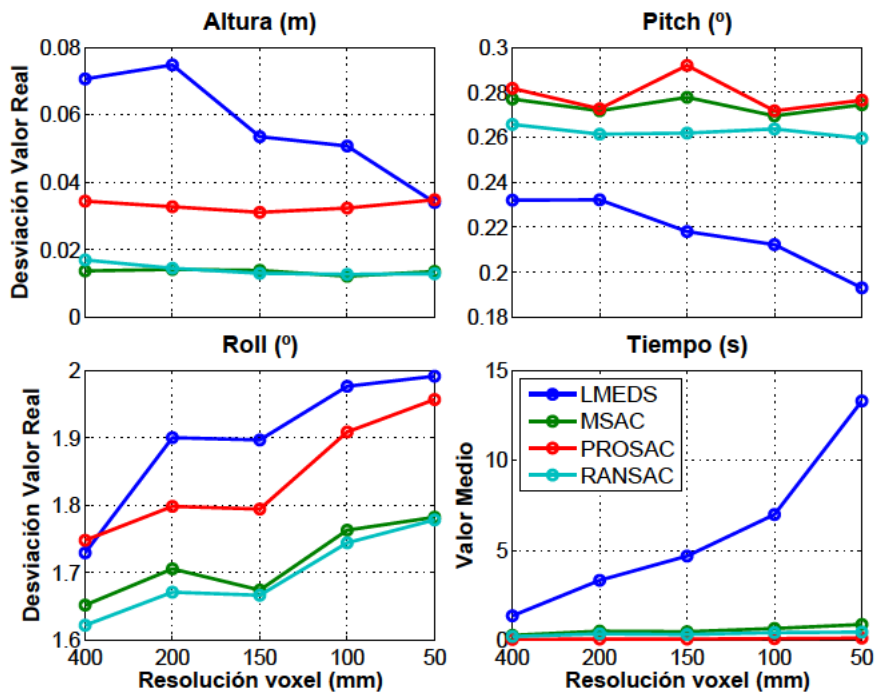
Figura 3.10: Secuencia de referencia Raw:Road:20112609_0015, tomada del KITTI.

3.2.3.2. Estabilidad en una secuencia dinámica

Se repite el procedimiento anterior sobre una secuencia dinámica de referencia, fig. 3.10, tomada del repositorio de secuencias de prueba para percepción en el KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago). Este paso permite establecer la exactitud de las mediciones, por cuanto se cuenta con medidas de referencia tomadas con una unidad IMU/GPS. La exactitud se calcula a través de la desviación típica referida al valor de referencia. Las mediciones se pueden observar en la fig. 3.11, donde nuevamente el estimador RANSAC, exhibe los valores mas bajos de desviación típica respecto al valor real, para los dos tipos de reducción, muestreo uniforme y regularización, con tiempos de ejecución del orden de décimas de ms y centésimas de ms, respectivamente. Por lo tanto, el método que resulta con la mejor relación exactitud-tiempo, es el estimador RANSAC con reducción por muestreo uniforme de 2000 puntos. La exactitud alcanzada con el método se muestra en la tabla 3.3.



(a) Muestreo Uniforme para diferentes resoluciones de muestreo.



(b) Regularización con diferentes tamaños de voxel.

Figura 3.11: Comparación de métodos para estimación del plano de la carretera con respecto a valores de referencia KITTI.

Exactitud @ Muestreo Uniforme 2K Puntos				
Métodos \ Parámetro	Altura (mm)	Pitch (°)	Roll (°)	Tiempo (ms)
LMEDS	14.5	0.25	1.92	258
MSAC	14.5	0.29	1.82	14.2
PROSAC	26.5	0.3	1.88	3
RANSAC	14.7	0.29	1.83	5

Cuadro 3.3: Rendimiento del estimador RANSAC con reducción por muestreo uniforme de 2000 puntos sobre la secuencia de la fig.3.10

3.2.3.3. Análisis temporal y filtrado de extrínsecos

Una vez definido el método que cumple con una óptima relación en cuanto a precisión, exactitud y tiempo de cálculo en la medición de los parámetros extrínsecos de la cámara, se procede a observar el comportamiento temporal de dichas mediciones, ver fig. 3.12, el objetivo es comprobar la correlación instantánea de las medidas con respecto a los valores de referencia tomados como verdaderos y que son suministrados dentro de las secuencias de prueba KITTI. El análisis de los datos medidos con respecto a los valores de referencia, revelan un error sistemático o fijo en cada medición, es decir, la exactitud obtenida en la tabla tabla 3.3, es la diferencia que debe ser ajustada en cada medición para obtener una correcta estimación. Así mismo, la precisión puede ser mejorada a través de procesos de filtrado, ya que como se aprecia en la figura fig. 3.12, existe un ruido que afecta las mediciones.

Con el ánimo de disminuir la incidencia del ruido de medida debido a múltiples fuentes como variaciones de luz a causa de la no perfecta sincronización en la captura de imágenes, los errores inherentes a la transducción foto-eléctrica y el ruido propio del ambiente automotriz, se aplica dos estrategia para la reducción de ruido sobre los parámetros extrínsecos medidos. La primera consiste en filtrar las mediciones usando un filtro digital pasa-bajos, con el objetivo de preservar las componentes frecuenciales bajas, propias del movimiento asociado a la amortiguación del vehículo. La segunda estrategia consiste en el uso de un filtro Kalman que estima de forma óptima el parámetro extrínseco donde es aplicado, conocida la dinámica de sistema que lo genera y el modelamiento del ruido presente tanto en el sistema como en la medición. En ambas propuestas se analiza las ventajas y desventajas de la utilización de estos dos tipos de filtro.

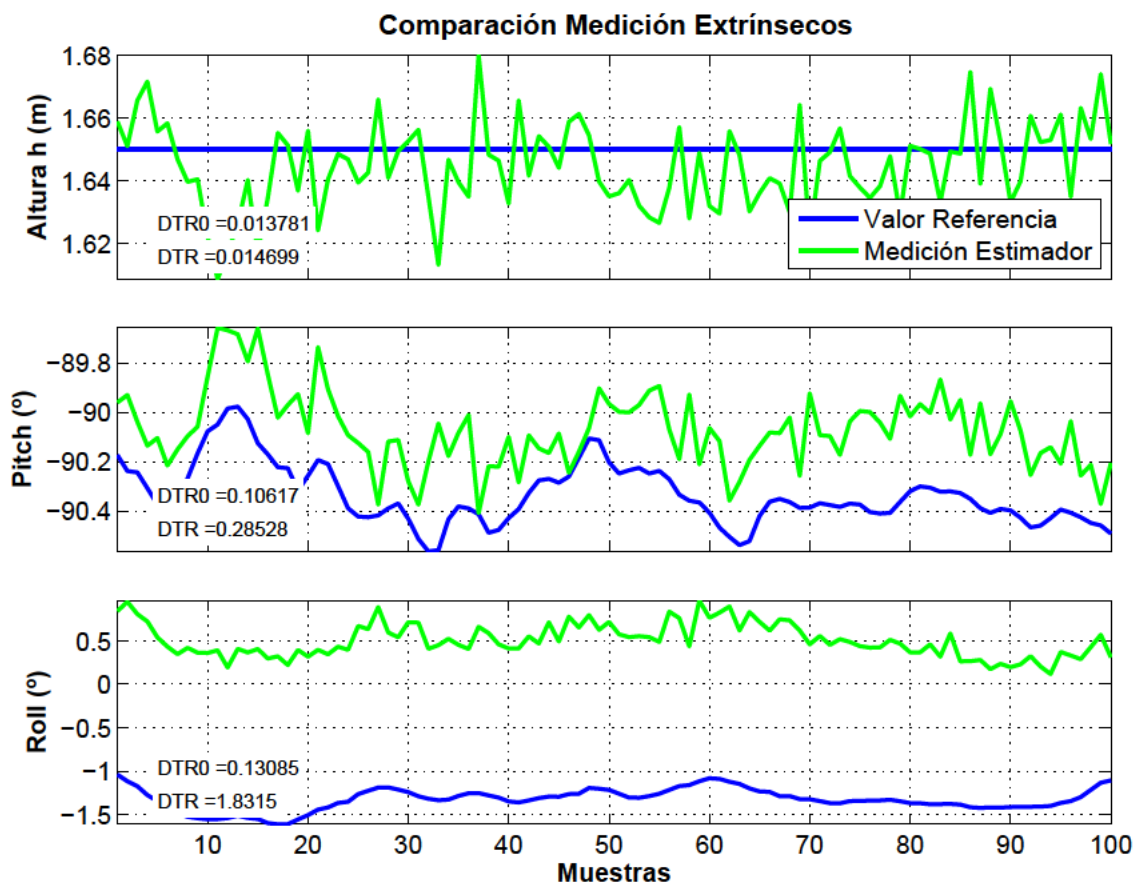


Figura 3.12: Comparación entre extrínsecos medidos por el método RANSAC @muestreo uniforme de 2000 puntos y valores de referencia KITTI, para la secuencia Road 2011_09_26_drive_0015. DTR (Desviación Típica respecto Referencia.)

Filtrado digital Existen dos tipos de filtros digitales [76], IIR, Infinite Impulse Response, y FIR, Finite Impulse Response. La diferencia entre ellos consiste en que un filtro FIR de orden N , solo se requiere N muestras del pasado de la señal de entrada $x(k)$, ec.3.26, mientras que el filtro IIR, necesita adicional mente contar con el conocimiento del pasado de la señal de salida $y(k)$, ec.3.27, hecho que genera un desfase no lineal a lo largo del espectro en frecuencia de $x(k)$. Sin embargo, una ventaja del filtro digital IIR en relación a su contra parte FIR, es que para obtener un rendimiento deseado, es requerido un orden mas bajo, es decir, un conocimiento menor del pasado de la señales. Ahora bien, el problema de requerir mas muestras pasadas de la señal de entrada, antes de poder generar una respuesta para el instante presente, k , es el retardo en la respuesta del filtro, efecto que impide su aplicación en tiempo real. En síntesis, implementar un filtro IIR, permite un una respuesta con desfase de magnitud baja, pero no lineal; implementar un filtro FIR, permite tener una respuesta con desfase lineal, pero con magnitud mayor.

$$y[k] = \sum_{i=0}^N b_i x[k-i] \quad (3.26)$$

$$y[k] = \frac{1}{a_0} \left(\sum_{i=0}^N b_i x[k-i] - \sum_{j=1}^P a_j y[k-j] \right) \quad (3.27)$$

El tipo de filtro seleccionado es el esquema FIR, ya que su respuesta es constante en cuanto al desfase que sufren todas las componentes de frecuencia de la señal de entrada, por tanto, el retardo de grupo, D , es estático. El efecto final del retardo o desfase de la señal, depende del orden del filtro, por lo cual, para obtener una rápida y drástica atenuación de las frecuencias altas, es necesario implementar un filtro de orden elevado, con el consecuente retardo asociado.

La respuesta de un filtro FIR de orden $N = 70$, con un retardo de grupo constante, $D = 35$ muestras, aplicado sobre la secuencia dinámica de la sección 3.2.3.2, se puede observar en la fig. 3.13. Allí, se almacenan y procesan las mediciones de la secuencia de entrada con los coeficientes resultantes del proceso de diseño del filtro, según se muestra en la tabla 3.4. La respuesta del filtro $y_i(k)$, para el respectivo extrínseco i , requiere D muestras del pasado de la señal $x_i(k)$, es decir, en la práctica, para estimar los parámetros filtrados en la captura $D+1$, se necesita procesar las D imágenes anteriores, con lo cual el filtro tiene un desfase en el tiempo discreto de D muestras, haciendo de esta estrategia una alternativa fuera de línea, ya que 35 muestras con una tasa de captura de 10 muestras por segundo, representa un

retardo de 3.5 segundos.

i	$\omega(i)$	i	$\omega(i)$	i	$\omega(i)$	i	$\omega(i)$	i	$\omega(i)$	i	$\omega(i)$	i	$\omega(i)$
1	0.001	11	-0.001	21	-0.005	31	0.056	41	0.056	51	-0.005	61	-0.001
2	0.001	12	0.000	22	-0.011	32	0.077	42	0.035	52	-0.000	62	-0.002
3	0.000	13	0.002	23	-0.016	33	0.096	43	0.016	53	0.004	63	-0.002
4	-0.000	14	0.004	24	-0.020	34	0.112	44	0.000	54	0.006	64	-0.002
5	-0.000	15	0.006	25	-0.021	35	0.122	45	-0.012	55	0.007	65	-0.001
6	-0.001	16	0.007	26	-0.019	36	0.125	46	-0.019	56	0.007	66	-0.001
7	-0.001	17	0.007	27	-0.012	37	0.122	47	-0.021	57	0.006	67	-0.000
8	-0.002	18	0.006	28	0.000	38	0.112	48	-0.020	58	0.004	68	-0.000
9	-0.002	19	0.004	29	0.016	39	0.096	49	-0.016	59	0.002	69	0.000
10	-0.002	20	-0.000	30	0.035	40	0.077	50	-0.011	60	0.000	70	0.001

Cuadro 3.4: Coeficientes del filtro FIR diseñado para mitigación de ruido en parámetros extrínsecos.

En la fig.3.13 se observa un error sistemático en la comparación realizada con la secuencia tomada del KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago) de 0.29 grados para el ángulo de cabeceo, 1.83 grados para el ángulo de alabeo y 14.7mm para la altura de la cámara, sin embargo, el comportamiento de las señales medidas y de referencia esta fuertemente correlado. Éste offset detectado se debe normalmente a la calibración inicial del instrumento de medida con el cual se toman los valores de referencia. La mayoría de unidades IMU/GPS tienen protocolos de calibración in situ, a través del modelado de ruido por software, que permite eliminar errores de medida sistemáticos.

Filtrado Kalman La reducción de ruido usando el filtro de kalman, es abordada como una estrategia en línea, dada la capacidad que tiene esta arquitectura para minimizar la varianza de las señales tratadas, utilizando únicamente, información del instante previo, $k - 1$.

La formulación del filtro junto con la solución al problema de optimización que plantea, se presenta en [46]. Allí, se describe un algoritmo recursivo de dos fases, predicción, estimación a priori y corrección, estimación a posteriori. Se parte del conocimiento dinámico del sistema sobre el cual se filtrará su señal de salida. La representación matemática de dicho sistema, es su definición en el espacio de estados. Para el caso de un sistema discreto LTI, (Linear Time Invariant), la ecuación de

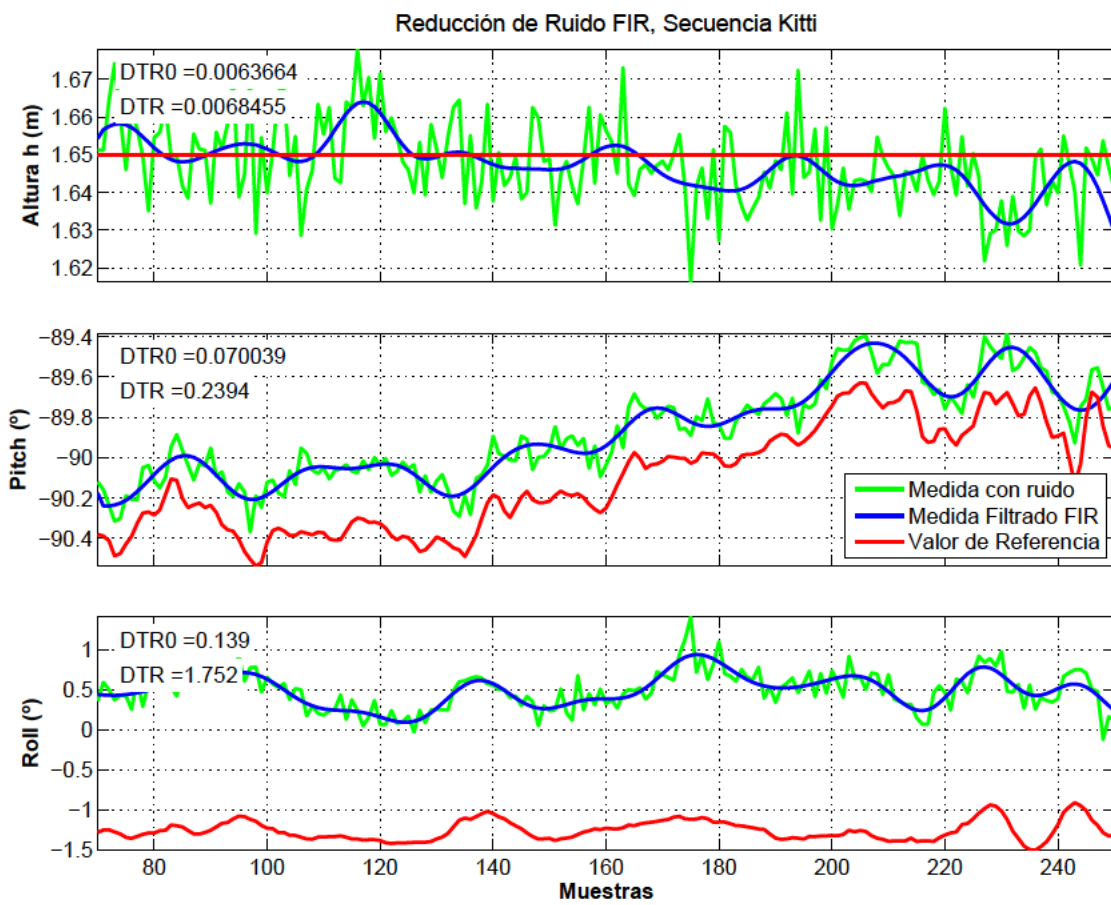


Figura 3.13: Reducción del ruido presente en los extrínsecos, en una escena dinámica.

estado y de salida, son como se muestran en la ec. 3.28.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + v(k) \\ y(k) &= Cx(k) + w(k) \end{aligned} \quad (3.28)$$

$x(k), x(k+1), v(k) \in \mathcal{R}^n; \quad u(k) \in \mathcal{R}^m; \quad y(k), w(k) \in \mathcal{R}^p$

Ahora bien, para la aplicación del filtro es necesario cumplir una serie de condiciones iniciales que incluyen: asumir $v(k)$ y $w(k)$, ruido de sistema y ruido de medida, respectivamente, normales y de media cero; estado inicial $x(0)$, normal y de media cero; conocidas las matrices de covarianza $Q = E(v(k)v(k)^t) \in \mathcal{R}^{nn}$ y $R = E(w(k)w(k)^t) \in \mathcal{R}^{pp}$. El problema siguiente, es entonces, estimar $x_e(k-1/k-1)$, estado a priori y $x_e(k/k)$, estado a posteriori, conocidos $x_e(k/k-1)$, $u(k)$ e $y(k)$. La solución, es de la forma mostrada en la ec. 3.29.

$$\begin{aligned} x_e(k/k-1) &= Ax_e(k-1/k-1) + Bu(k) \\ x_e(k/k) &= x_e(k/k-1) + H(k)(y(k) + w(k) - Cx_e(k/k-1)) \end{aligned} \quad (3.29)$$

$H(k)$ es una matriz libre que puede ser escogida de forma óptima según el criterio de minimizar la covarianza del error de predicción a posteriori, $\varepsilon(k/k)$, definido en la ecuación 3.30.

$$\begin{aligned} \varepsilon(k/k-1) &= x(k) - x_e(k/k-1) \\ \varepsilon(k/k) &= x(k) - x_e(k/k) \end{aligned} \quad (3.30)$$

La matriz de covarianza del error de predicción a priori, $P(k/k-1)$ se obtiene a través de la ec. 3.31, suponiendo que $\overline{\varepsilon(k/k-1)} = 0$.

$$\begin{aligned} P(k/k-1) &= E\left((\varepsilon(k/k-1) - \overline{\varepsilon(k/k-1)})(\varepsilon(k/k-1) - \overline{\varepsilon(k/k-1)})^t\right) \\ P(k/k-1) &= AP(k-1/k-1)A^t + Q \end{aligned} \quad (3.31)$$

La covarianza del error de predicción a posteriori, definida en la ec. 3.32, es la mínima posible obtenida al resolver el problema de optimización que se muestra en la ec. 3.33 para una matriz $H(k)$ óptima, ec. 3.34. Nuevamente, se supone que $\overline{\varepsilon(k/k)} = 0$.

$$P(k/k) = E\left((\varepsilon(k/k) - \overline{\varepsilon(k/k)})(\varepsilon(k/k) - \overline{\varepsilon(k/k)})^t\right) \quad (3.32)$$

$$\min_{H(k)} \alpha^t P(k/k) \alpha \quad \forall \alpha \in \mathcal{R}^n \quad (3.33)$$

$$\begin{aligned} H(k) &= P(k/k-1)C^t (CP(k/k-1)C^t + R)^{-1} \\ P(k/k) &= (I - H(k)C) P(k/k-1) \end{aligned} \tag{3.34}$$

En resumen, las fases del filtro de kalman se pueden relacionar como se muestra en la tabla 3.5.

Fase: Predicción	
Est. a priori	$x_e(k/k-1) = Ax_e(k-1/k-1) + Bu(k)$
Cov. error a priori	$P(k/k-1) = AP(k-1/k-1)A^t + Q$
Fase: Corrección	
Medida filtrada	$y_e(k) = Cx_e(k/k-1)$
Ganancia Kalman	$H(k) = P(k/k-1)C^t (CP(k/k-1)C^t + R)^{-1}$
Est. a post.	$x_e(k/k) = x_e(k/k-1) + H(k) (y(k) + w(k) - Cx_e(k/k-1))$
Cov. error a post.	$P(k/k) = (I - H(k)C) P(k/k-1)$

Cuadro 3.5: Algoritmo recursivo Filtro de Kalman.

Finalmente, para implementar el filtro de Kalman, se requiere ajustar apropiadamente los valores de covarianza del sistema, Q, y de medida, R. La matriz Q esta relacionada directamente con la incertidumbre o error que existe entre el modelo del sistema y el modelo propuesto, que para el caso, se asume como un una variable de velocidad cero. La matriz R por su parte, indica la fiabilidad de las mediciones hechas sobre los parámetros extrínsecos.

Así entonces, se realiza un experimento, utilizando la secuencia dinámica mostrada en la fig. 3.10, donde se prueba diferentes valores de varianza de ruido en el proceso, $np(i)$, y varianza de ruido en la medida, $nm(j)$, que definen las matrices $Q(i)$ y $R(j)$, como se muestra en la ec. 3.35. El objetivo de encontrar la combinación de valores que causen la menor medida de la desviación típica, del error existente entre la señal medida y la señal filtrada, como se define en la ec. 3.36. Los valores de $np(i)$ y $nm(j)$, se muestran en la tabla 3.6.

i, j	1	2	3	4	5	6	7	8
$np(i), nm(j)$	5e-1	1e-1	5e-2	1e-2	5e-3	1e-3	5e-4	1e-4

Cuadro 3.6: valores de varianza para ruido de proceso y ruido de medida, usados en el experimento de la fig. 3.14.

$$Q(i) = \begin{bmatrix} np(i) & 0 \\ 0 & np(i) \end{bmatrix} \quad R(j) = nm(j) \tag{3.35}$$

$$\begin{aligned} \varepsilon &= (y_e - \bar{y}_e) - (y_{ref} - \bar{y}_{ref}) \\ DTR0_e &= \sqrt{E((\varepsilon)(\varepsilon)^t)} \end{aligned} \tag{3.36}$$

El experimento de variación para np y nm , se observa en la fig. 3.14. Para cada cada parámetro extrínseco se determina un valor de Q y R que genere el valor de desviación típica con respecto al valor de referencia para la secuencia respectiva, menor, y con un comportamiento en el tiempo lo mas exacto posible a los datos de referencia, como se muestra en la fig. 3.15. Allí, para cada parámetro se ha escogido un punto adecuado de operación con máximo rendimiento, según se relaciona en la tabla 3.7.

	Ángulo de cabeceo	Ángulo de alabeo	Altura
np	5e-4	1e-4	1e-4
nm	5e-4	5e-4	5e-3

Cuadro 3.7: valores óptimos de varianza para ruido de proceso y ruido de medida, seleccionados por parámetro extrínseco

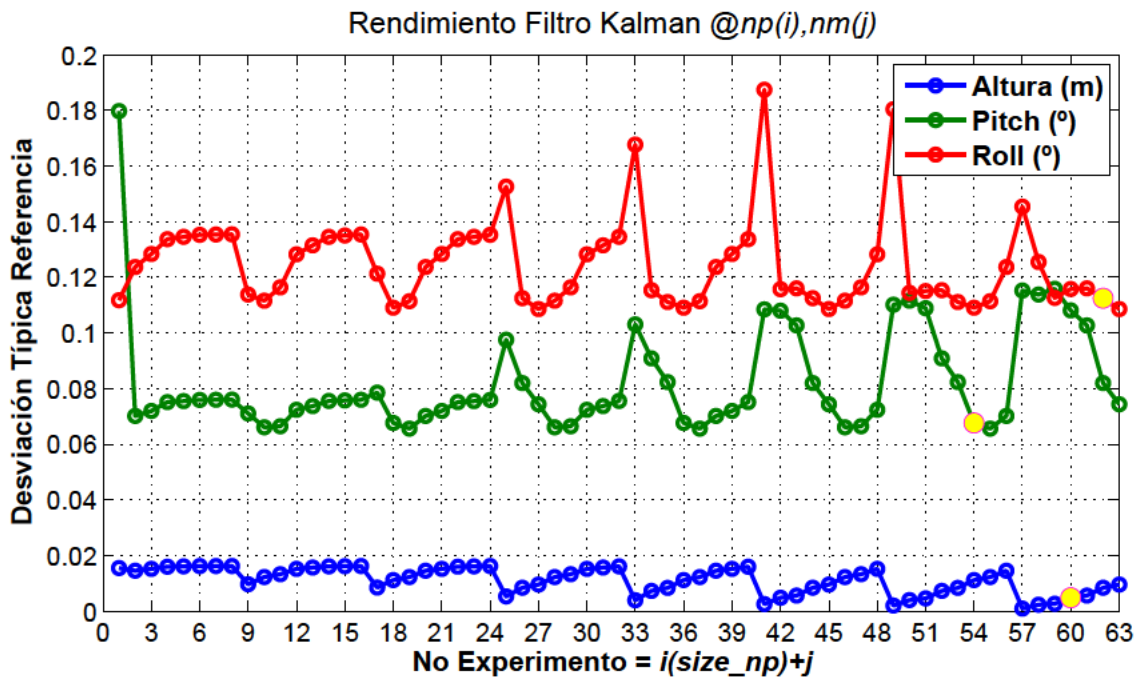


Figura 3.14: Sintonización del Filtro de Kalman para diferentes valores de desviación estándar en el ruido de proceso, np y para diferentes valores de desviación estándar en el ruido de medida, nm .

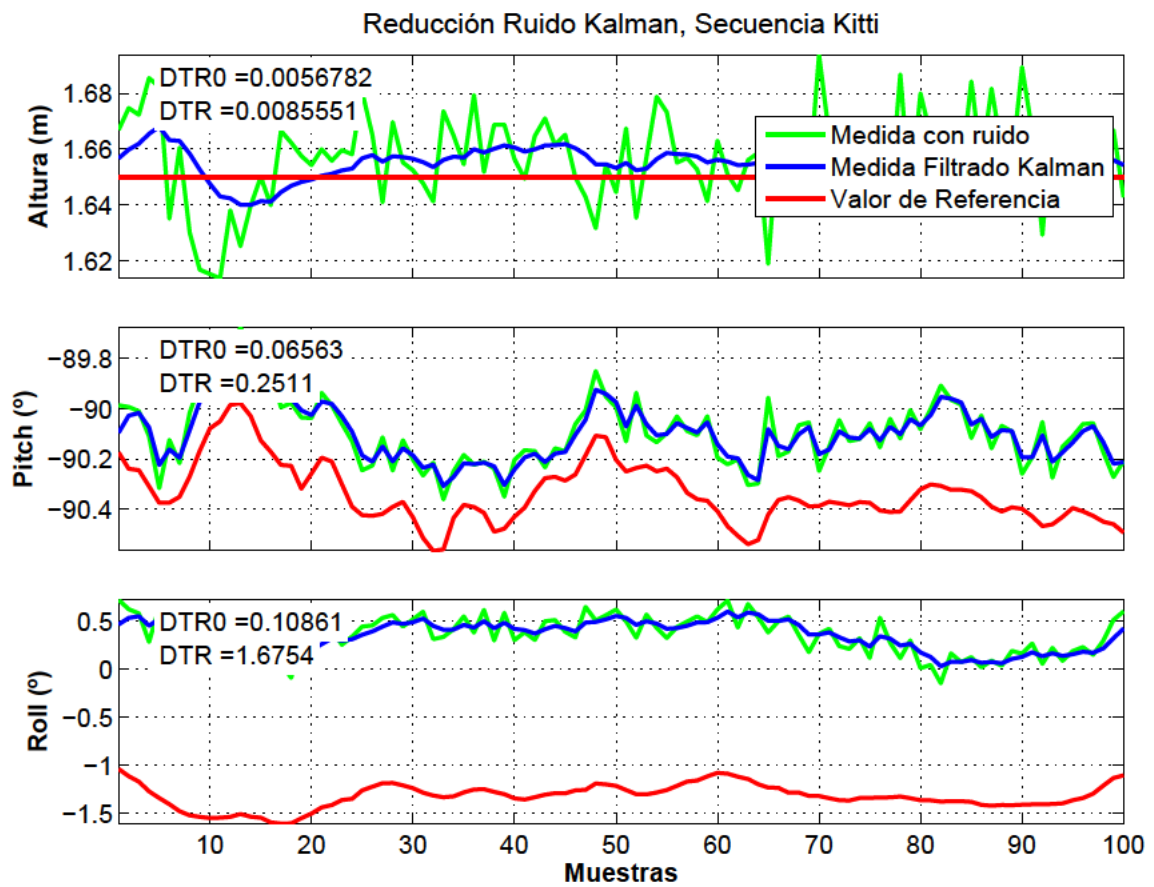


Figura 3.15: Reducción del ruido presente en los extrínsecos, en una escena dinámica usando Filtro de Kalman.

Capítulo 4

Modelado y Segmentación de la Carretera

En la fig. 4.1, se presenta el esquema general del sistema para segmentación de carreteras que se propone en el presente capítulo.

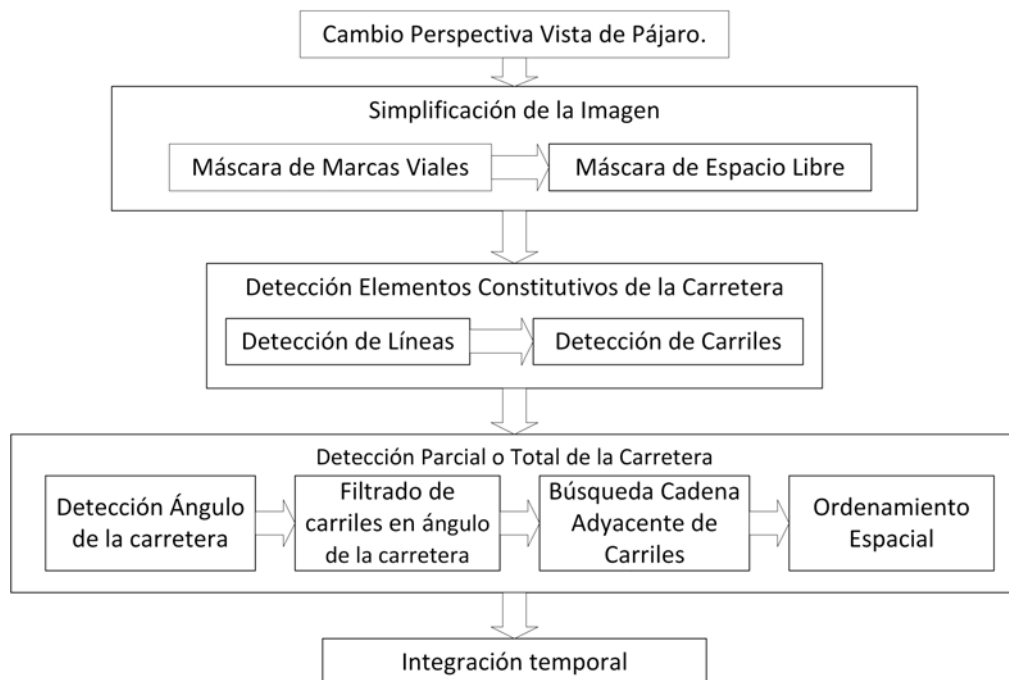


Figura 4.1: Algoritmo de Segmentación de carreteras según el estándar IC 8.2.

El flujo de datos desde que se captura la imagen hasta que se logra la identificación y numeración de carriles, utilizando información temporal, se efectúa en cuatro bloques principales. Primero se simplifica la imagen filtrando los objetos que

cumplen las características propias de las marcas viales, restringiendo la búsqueda a la zona de la imagen coincidente con el plano tierra. En el segundo bloque, el procesamiento continua con la detección de los elementos constitutivos de la carretera, las líneas de carril y los correspondientes carriles que se forman con ellas. Posteriormente, se efectúa la detección parcial o total de la carretera, que corresponde con la búsqueda de carriles adyacentes en el espacio. Finalmente, en el cuarto bloque, con las detecciones parciales o totales de carretera, formadas por una cadena de carriles, se aplica un proceso de casamiento carril a carril, para actualizar en el tiempo una cadena final completa, que tendrá la cualidad de rechazar oclusiones y errores por sombras y cambios de iluminación.

Ahora bien, el algoritmo para segmentación de carreteras presentado se inspira en dos ideas fundamentales:

- El modelado de la carretera, se efectúa en un sistema de referencia 2D, solidario al plano tierra, el cual se puede apreciar, cambiando la perspectiva en la imagen original, a la perspectiva superior y en dirección al plano tierra, llamada *vista de pájaro*, ver apéndice A.2. Esta transformación es útil en aplicaciones donde el movimiento de un cuerpo esta restringido a un plano en el espacio. La ventaja resulta del paralelismo existente entre el plano donde ocurre el movimiento del cuerpo, y el plano de la imagen. Esta condición permite el uso de técnicas de visión por computador para relacionar de manera directa, la segmentación de objetos en la imagen de la carretera con el movimiento descrito por ellos en el entorno 3D.
- Por otro lado, la carretera es uno de los pocos ambientes creados por el hombre para guiar las tareas que naturalmente ocurren allí, como la conducción o las interacciones vehículo-peatón, a través de marcadores visuales que cumplen patrones de simetría y paralelismo, tal y como se describe en la norma de carreteras IC 8.2 y se resume en el apéndice B. El modelado de la carretera entonces se puede construir aprovechando el paralelismo de las marcas visuales que aparecen en ella.

4.1. Máscara de Marcas Viales

Las líneas y los carriles de la carretera, son detectados sobre una máscara de marcas viales, \mathcal{RM} , según el estándar IC 8.2 [24]. Esta simplificación es útil, por

cuanto el espacio de aplicación para algoritmos posteriores se reduce, y además, porque filtrar la imagen teniendo en cuenta las características geométricas y de intensidad, que las marcas viales poseen, como el ancho de una línea separadora de carril y el contraste que existe con el fondo donde se encuentra pintada, permite eliminar artefactos espurios y patrones similares presentes en otros elementos de la escena.

4.1.1. Detección de marcas viales en el espacio del gradiente

El enfoque escogido es una modificación en tres fases, del método presentado en [18] y en [42].

En la primera fase se detecta los bordes en la imagen, utilizando los 3 primeros pasos del algoritmo Canny [14], con un estimador de gradiente distinto que el algoritmo original y sin seguimiento de contornos, esto debido a que los bordes fuertes se consideran inicialmente suficientes para el contraste que exhiben las marcas viales con el fondo de carretera y dado que no se requieren contornos cerrados. El procedimiento consiste en aplicar inicialmente un filtro por convolución tipo campana de Gauss para difuminar la imagen y prepararla para que la extracción del gradiente sea una nueva imagen continua y suave, la magnitud y dirección del gradiente son obtenidas utilizando las máscaras de Sobel. A continuación, se eliminan los bordes débiles, correspondientes a píxeles con magnitud de gradiente inferior a un umbral escogido convenientemente, de tal forma, que la mayoría de bordes sean conservados. Posteriormente, se realiza la supresión de no máximos sobre la imagen filtrada, este procedimiento se realiza para garantizar que los bordes sean delineados con un pixel de ancho, tal y como se puede apreciar en la fig. 4.2.

En la segunda fase, una vez se han obtenido los bordes de la imagen usando el detector de Canny parcial o Canny Original, se realiza la búsqueda de pares de puntos tipo borde, que forman los extremos de cada segmento transversal, $\{v_j\}$, de una marca vial. Dichos segmentos, a su vez, están formados por una colección de puntos sucesivos: $\{p_{j,k} : k \in \mathbb{N}; 1 \leq k \leq n(v_j)\}$, $n(v_j)$ es la cardinalidad del vector $\{v_j\}$, cuyos gradientes en la posición inicial y final, cumplen las restricciones dadas en la ec. (4.1).

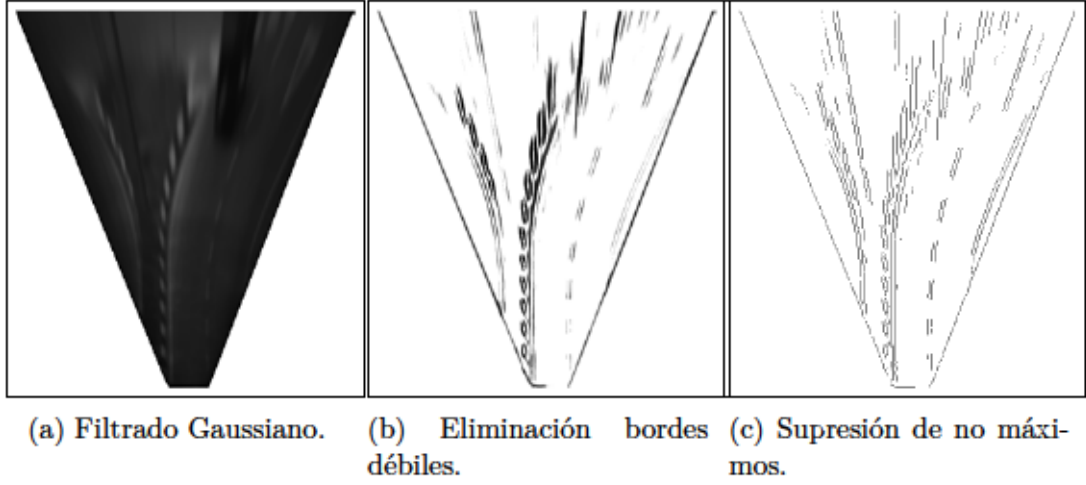
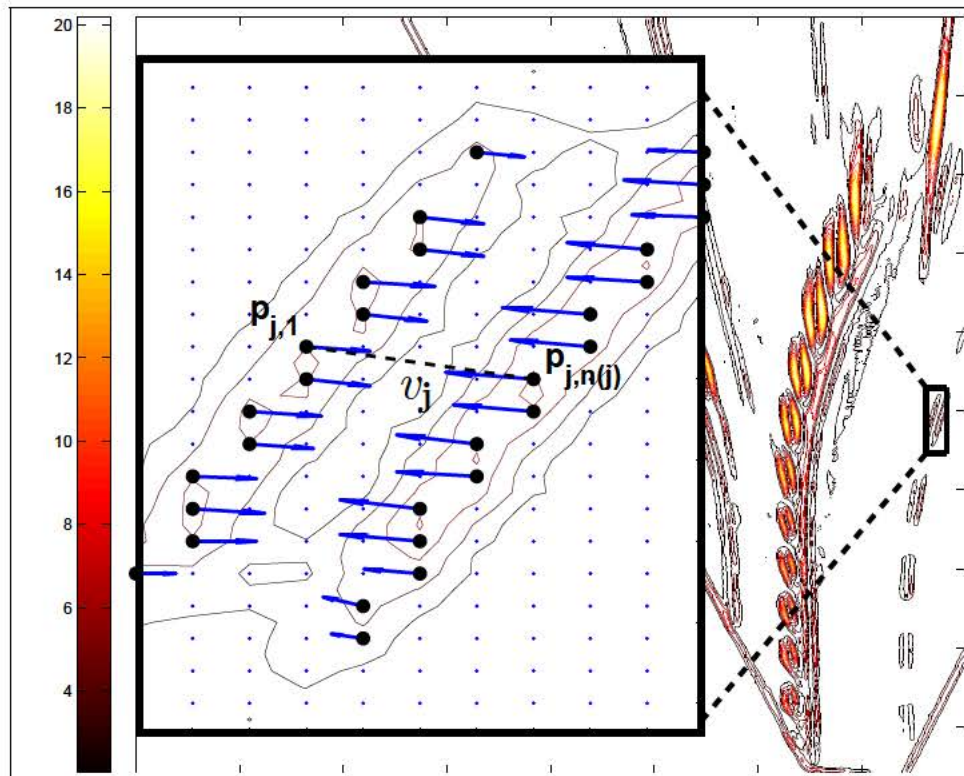


Figura 4.2: Fases del algoritmo para detección de marcas viales. Detección de bordes.

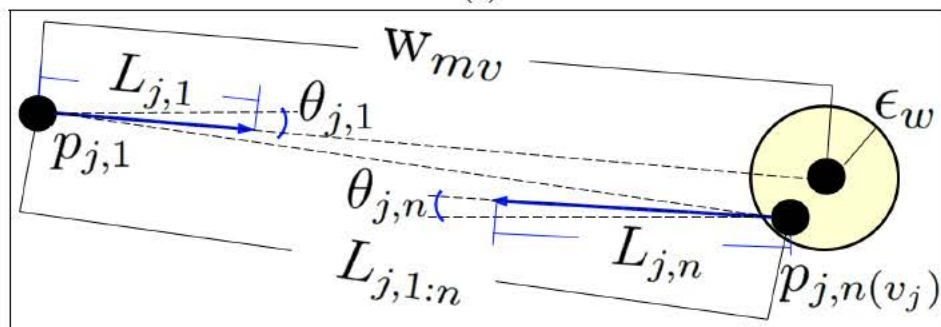
$$\begin{aligned}
 1. & \left| \|\nabla p_{j,1}\| - \|\nabla p_{j,n(v_j)}\| \right| < \epsilon_m, \\
 2. & \left| \angle \nabla p_{j,1} - \angle \nabla p_{j,n(v_j)} \right| - \pi < \epsilon_a, \\
 3. & \left| \|p_{j,1} - p_{j,n(v_j)}\| - w_{mv} \right| < \epsilon_w
 \end{aligned} \tag{4.1}$$

La restricción 1 de la ec. (4.1), establece que la magnitud de los gradientes en la posición inicial y final de cada segmento $\{v_j\}$, deben ser aproximadamente iguales. Por su parte, en la restricción 2, se verifica que la dirección del gradiente en los extremos, inicial y final de cada segmento, sean opuestos. Por último, la restricción 3, sirve para comprobar que la longitud del segmento sea cercana al ancho de una marca vial. El parámetro que contiene el ancho de marca vial, w_{mv} , es fijado correspondientemente con la altura de la cámara virtual en perspectiva *vista de pájaro*, z_{vp} , seleccionada en el cambio de perspectiva. ϵ_m , ϵ_a y ϵ_w son umbrales, escogidos lo suficientemente pequeños en relación al ruido del proceso.

Una representación gráfica de las condiciones que debe cumplir un segmento $\{v_j\}$ para ser considerado parte de una marca vial, se aprecia en la fig. 4.3. En la parte (a) de la figura, se observa dibujado en línea discontinua uno de los segmentos $\{v_j\}$ extendido desde el punto tipo borde de inicio $p_{j,1}$ hasta el punto de borde final de dicho segmento $p_{j,n(v_j)}$, los puntos tipo borde utilizados para el análisis, se obtienen como la salida de la fase 1 del algoritmo de detección de marcas viales. En la parte (b) de la figura, se detalla los elementos en el espacio de gradiente, correspondientes a cada segmento $\{v_j\}$, con los cuales se definen las restricciones de una marca vial. $L_{j,i}$ y $\theta_{j,i}$, representan la magnitud y la dirección del gradiente



(a)



(b)

Figura 4.3: Algoritmo para detección de marcas viales. Restricciones que definen una marca vial. (a) Detalle gradiente de la imagen sobre los bordes detectados en una marca vial. Los puntos negros representan los pixeles detectados como bordes en la primera fase del algoritmo. (b) Descripción gráfica de los elementos que conforman las restricciones de una marca vial. $L_{j,i}$ y $\theta_{j,i}$, representan la magnitud y la dirección del gradiente en el punto $p_{j,i}$.

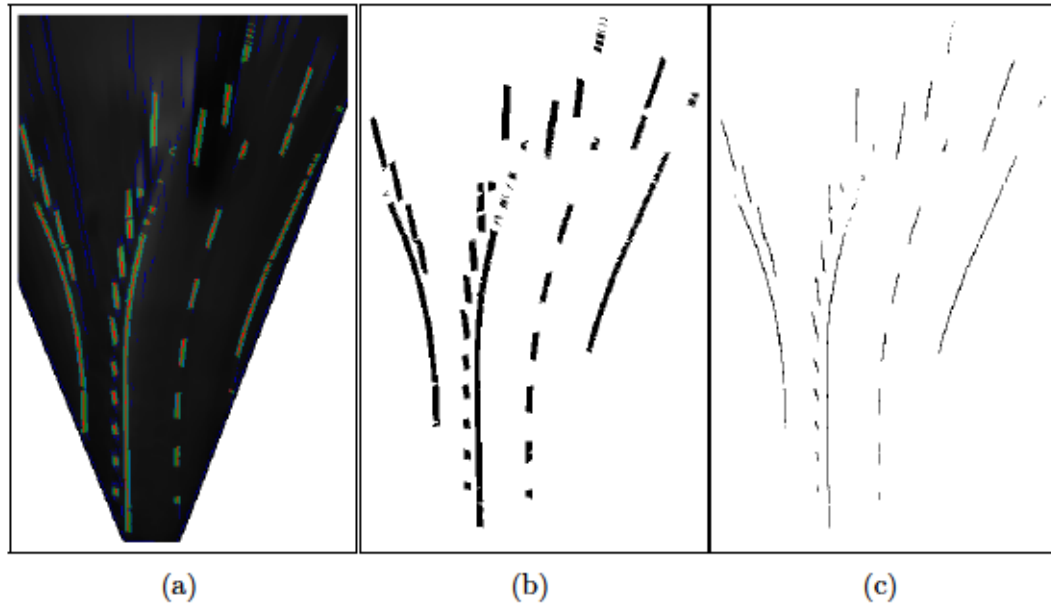


Figura 4.4: Algoritmo de detección de marcas viales. (a) Superposición de la máscara de marcas viales sobre la imagen de partida. (b) Máscara de marcas viales, \mathcal{RM} . (c) Centro de marcas viales.

en cada punto $p_{j,i}$ dentro del segmento $\{v_j\}$. $L_{j,i:n}$ representa la distancia entre los puntos inicial, i y final n del segmento en análisis j . Así mismo, la zona circular en amarillo representa el vecindario posible donde es aceptada la presencia de un punto tipo borde considerado como el extremo final del segmento $\{v_j\}$, es decir, el punto $p_{j,n(v_j)}$. Esta zona se encuentra centrada en el punto ideal donde debería aparecer el punto de marca vial complementario a $p_{j,1}$ en un radio de búsqueda ϵ_w .

Finalmente, en la fase 3, se construye la máscara de marcas viales, \mathcal{RM} , como la superposición de todos los segmentos $\{v_j\}$, dado que pertenecen a las líneas de la carretera. Adicionalmente, con el objetivo de extraer el centro de las marcas viales, como líneas de ancho un pixel, se realiza el siguiente análisis sobre cada segmento detectado $\{v_j\}$. La hipótesis de partida se basa en que para cada segmento, la magnitud del gradiente $\|\nabla v_j\|$, inicia y termina con un valor máximo, con lo cual, en el punto medio del segmento, debe existir un sub-segmento de magnitud mínima, $\|\nabla v_{j_{min}}\| < \epsilon_g$, correspondiente a la zona blanca de la marca vial. La posición del centro de la respectiva marca, se obtiene como el centroide del sub-segmento de magnitud de gradiente mínima. En la fig. 4.4, se puede apreciar, la máscara de marcas viales y la extracción de las líneas centrales de ancho un pixel.

4.1.2. Medición cuantitativa de la segmentación para marcas viales

Con el ánimo de analizar el impacto de algunas variaciones en el algoritmo propuesto en la sección 4.1.1, se han propuesto tres variaciones del esquema original que se detallan a continuación.

1. Segmentación de marcas viales basada en Kernel (Seg. Kernel)[42]. El proceso consta de tres pasos: filtrado gaussiano, filtrado por kernel y umbralización adaptativa.
2. Segmentación de marcas viales sin detección de bordes (Seg. grad.). En esta versión del algoritmo de segmentación presentado en la sec. 4.1.1, la fase 1, correspondiente a detección de bordes es omitida, por lo cual, la comprobación de restricciones se efectúa sobre toda la imagen, es decir, la búsqueda de segmentos $\{v_j\}$ se realiza analizando todos los puntos $p_{i,j}$ correspondientes a los pixeles de la imagen de partida, no solo analizando aquellos pixeles identificados como puntos borde.
3. Segmentación de marcas viales propuesta en la sección 4.1.1 (Seg. snm-grad.).
4. Segmentación de marcas viales con detección de bordes Canny (Seg. Canny-grad.). Variación del algoritmo de segmentación presentado en la sec. 4.1.1, donde la fase 1, se reemplaza por el detector completo Canny, con umbrales fijados como, $umbral\ max = 3 * umbral\ min$.

La evaluación comparativa de los métodos anteriores se realiza sobre la *secuencia de referencia 1* y sobre la *secuencia de referencia 2* para marcas viales, fig. 4.5. Se trata de 2 secuencias de 75 y 100 imágenes respectivamente. La *secuencia de referencia 1* es tomada sobre una carretera de 2 carriles, que inicialmente es atravesada por un puente, durante las primeras 20 capturas, seguidas por 40 imágenes de un trayecto recto con flujo bajo de vehículos y termina con 15 imágenes donde se observan 2 calzadas en diferentes sentidos, cada una de 2 carriles, con un ingreso a rotonda. La *secuencia de referencia 2*, por su parte, se toma de una carretera de 3 carriles con barandilla a lado y lado de la vía, con bajo tráfico de vehículos, en donde las líneas se observan bien definidas.

Para obtener valores cuantitativos que permitan evaluar el acierto de un detector de marcas viales es necesario estimar cuales pixeles sobre las secuencias de referencia

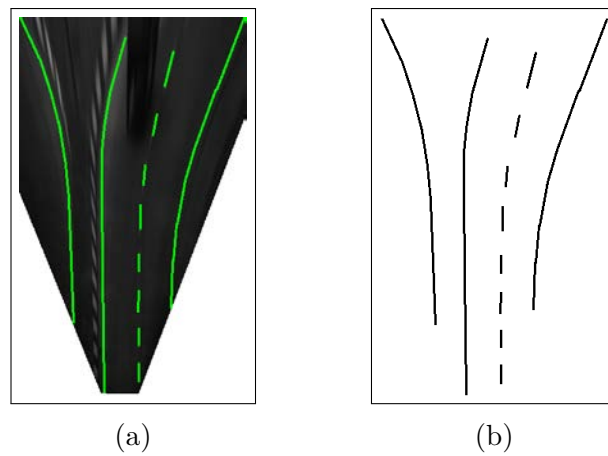


Figura 4.6: Secuencia de referencia 1, “Ground Truth”. (a) Coloreado manual de las marcas viales en secuencia de referencia. (b) Máscara de marcas viales en secuencia de referencia.

en realidad hacen parte de una marca vial. Para tal propósito se colorea manualmente cada una de las imágenes de prueba resaltando solo los píxeles correspondientes a marcas viales, fig. 4.6, construyendo así, el “Ground Truth” de las secuencias de referencia 1 y 2.

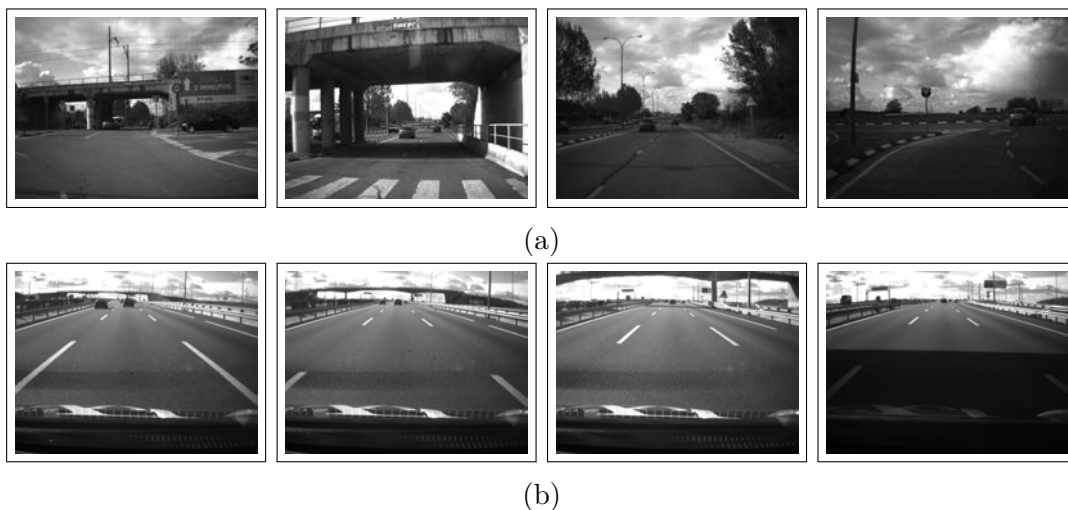


Figura 4.5: Algunas imágenes de la secuencia de referencia (a) 1 y (b) 2, para la detección de marcas viales.

Las medidas de rendimiento que permiten comparar la calidad de la segmentación para cada una de la variantes del algoritmo son:

- Sensibilidad. Fracción de píxeles clasificados correctamente en relación a todos

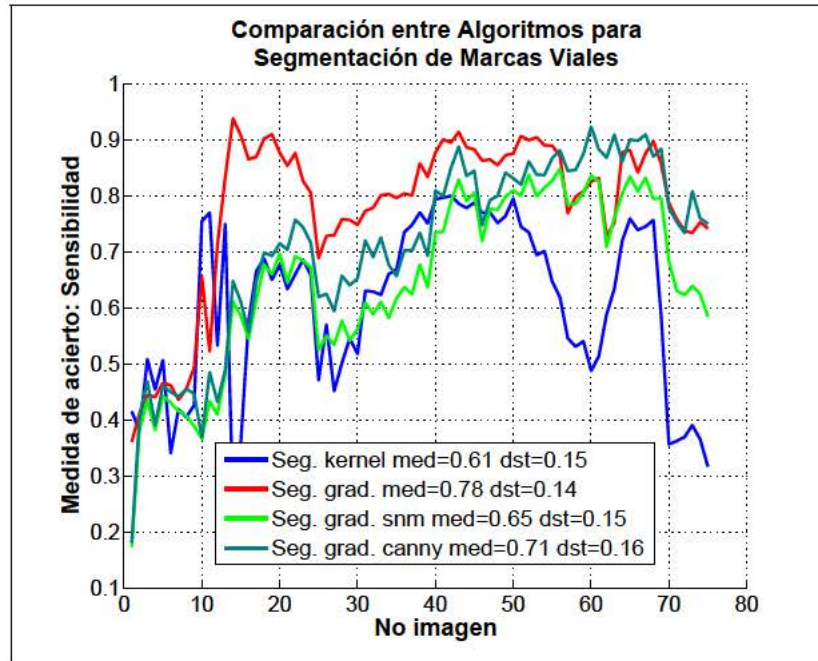


Figura 4.7: Comparación de índice de Sensibilidad entre variaciones del algoritmo para detección de marcas viales, sobre la secuencia 1 de referencia fig. 4.5a.

los pixeles que en verdad hacen parte de una marca vial, ec. (4.2).

$$Sensibilidad = \frac{\sum \mathcal{RM}_i \cap \mathcal{RM}_{ground\ truth}}{\sum \mathcal{RM}_{ground\ truth}} \quad (4.2)$$

- Precisión. Fracción de pixeles clasificados correctamente en relación a todos los pixeles clasificados como una marca vial, ec. (4.3).

$$Precision = \frac{\sum \mathcal{RM}_i \cap \mathcal{RM}_{ground\ truth}}{\sum \mathcal{RM}_i} \quad (4.3)$$

- Tiempo de ejecución.

A continuación, se realiza la comparación entre el resultado de los algoritmos para segmentación de marcas viales y el “Ground Truth” de la secuencia de referencia 1 y 2. Posteriormente, se calcula cada uno de los tres índices de calidad presentados anteriormente. Los datos son mostrados gráficamente en las fig. 4.7, 4.9, y 4.11. para la secuencia de referencia 1 y en las fig. 4.8, 4.10, y 4.12. para la secuencia de referencia 2.

De los resultados anteriores para las secuencias de referencia 1 y 2, se pueden enunciar los siguientes hallazgos:

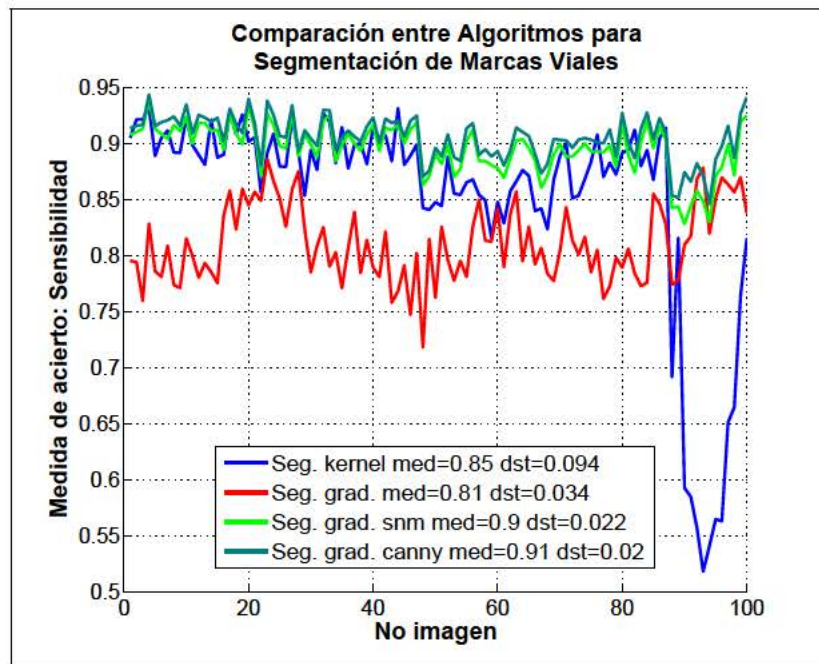


Figura 4.8: Comparación de índice de Sensibilidad entre variaciones del algoritmo para detección de marcas viales, sobre la secuencia 2 de referencia fig. 4.5b.

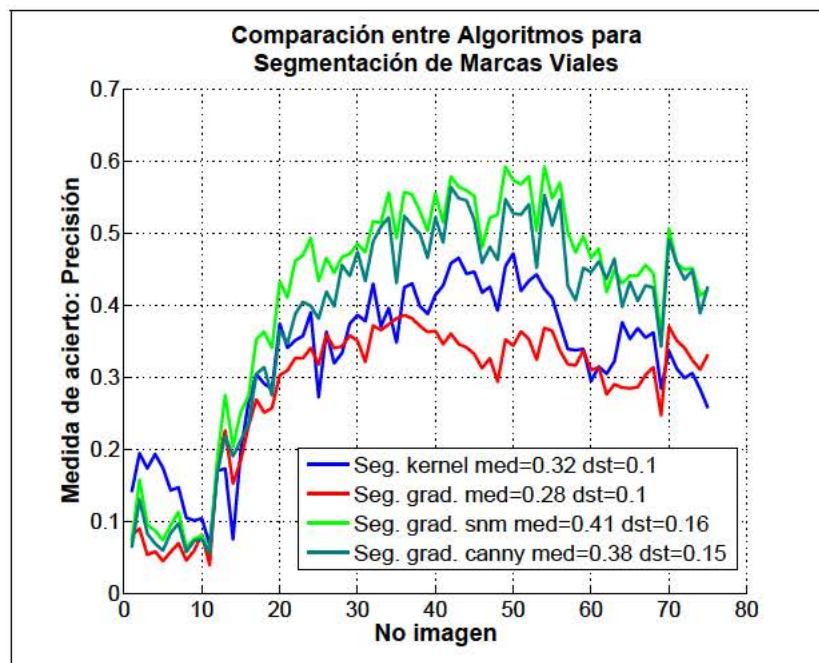


Figura 4.9: Comparación de índice de Precisión entre variaciones del algoritmo para detección de marcas viales, sobre la secuencia 1 de referencia fig. 4.5a.

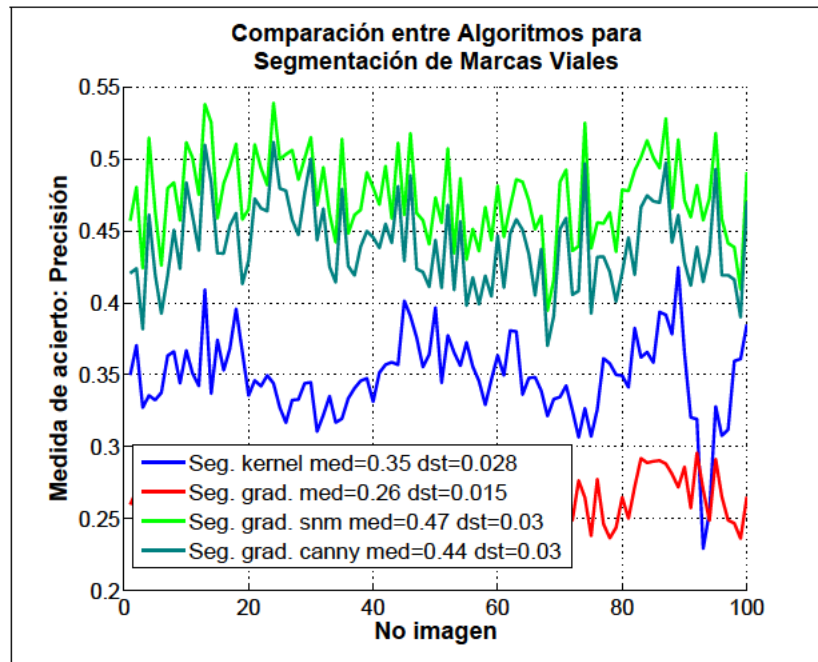


Figura 4.10: Comparación de índice de Precisión entre variaciones del algoritmo para detección de marcas viales, sobre la secuencia 2 de referencia fig. 4.5b.

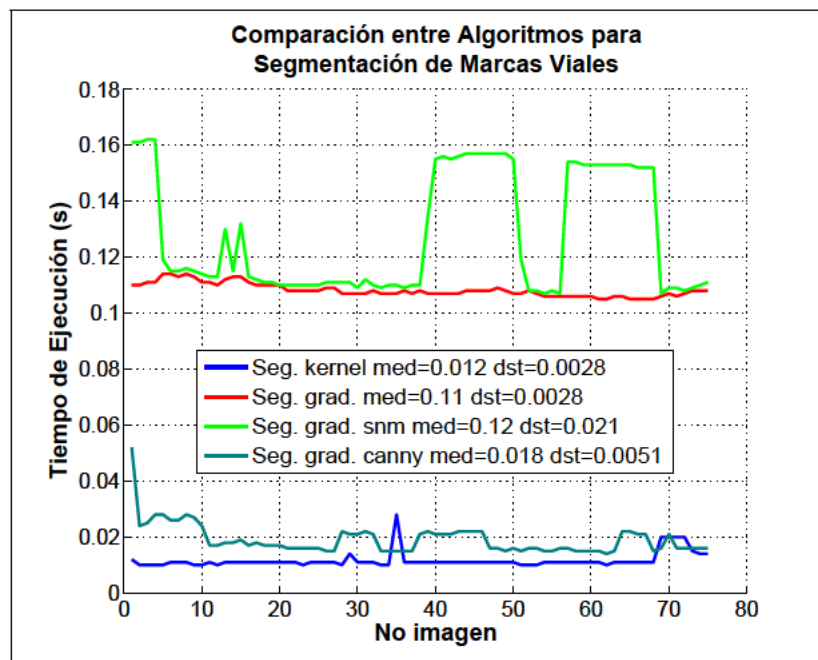


Figura 4.11: Comparación de índice de coste computacional entre variaciones del algoritmo para detección de marcas viales, sobre la secuencia 1 de referencia fig. 4.5a.

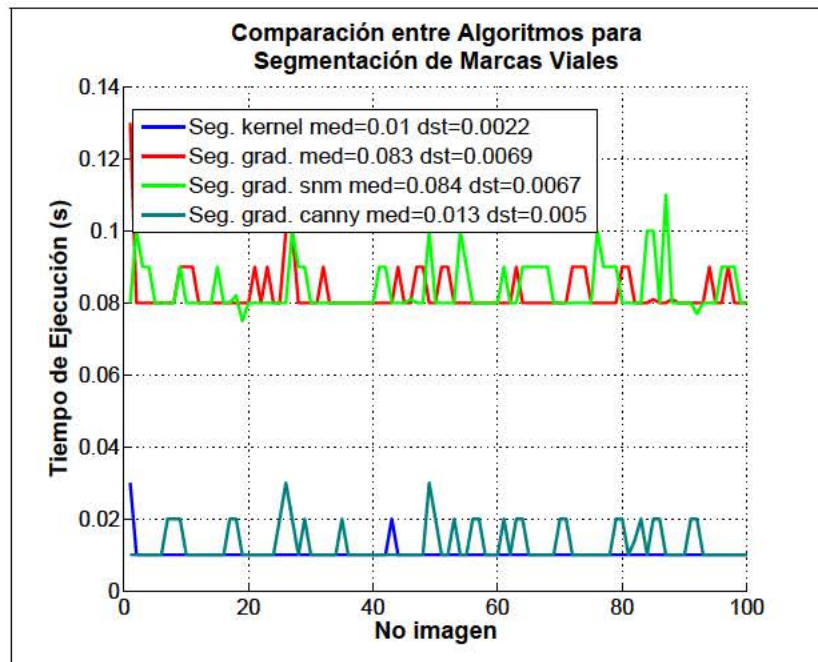


Figura 4.12: Comparación de índice de coste computacional entre variaciones del algoritmo para detección de marcas viales, sobre la secuencia 2 de referencia fig. 4.5b.

- El método Seg. kernel. grad. es el más rápido con un tiempo de ejecución de alrededor de 11ms, pero con la menor relación Sensibilidad+Precisión promedio de 53 %.
- El método Seg. grad. versión sin detección de bordes, mostró también la menor relación Sensibilidad+Precisión promedio de 53 %, con un tiempo de ejecución promedio de 96ms.
- El índice de calidad Sensibilidad+Precisión promedio, para el método Seg. grad. snm. (Canny parcial) es de 60 %, mientras que para el método Seg. grad. Canny es de 61 %, valores similares que hacen del factor tiempo, un diferenciador a la hora de seleccionar un mejor método, sin embargo, a pesar que la alternativa con fase 1 Canny es más compleja que su versión simplificada sin seguimiento de bordes, el tiempo de ejecución que mostró fue menor. Este resultado atípico se debe a que la implementación utilizada del detector Canny está optimizada para procesamiento paralelo y en consecuencia, se ejecuta en un tiempo menor, 15ms comparado con 102ms de la versión Seg. grad. snm (Canny parcial). Por tanto, aunque la calidad de la segmentación es similar

en ambos métodos, y el tiempo de ejecución debería ser menor en la versión simplificada del método, la versión del algoritmo de segmentación de marcas viales con fase 1 Canny, resulta más rápida y por tanto, mejor en relación a los 3 índices de rendimiento: sensibilidad, precisión y tiempo.

En síntesis, el algoritmo para segmentación de marcas viales con detección de bordes usando el método de Canny, con un *umbral min* = 8, exhibe la relación de rendimiento Sensibilidad+Precisión más alta para el tiempo de ejecución más bajo: con una Sensibilidad de 81 %, una Precisión de 41 %, y un coste computacional en tiempo de 15 *ms*.

Es importante aclarar que los valores de precisión medidos son bajos, para todas las variaciones del algoritmo, debido a la detección de patrones del tipo marca vial que ocurren en zonas de distorsión, causadas por factores como: proyección de objetos que no están en el plano de la carretera, desde la perspectiva original en donde fueron capturadas las imágenes, a la perspectiva en vista de pájaro, también por efectos de sombra que generan líneas claras entre zonas oscuras, caso que ocurre a lo largo de las barandillas en los costados de algunas carreteras. Un índice de precisión bajo indica que se han detectado una cantidad de falsos positivos comparable a la cantidad de marcas viales detectadas correctamente, evento que no genera inconvenientes, dado que el algoritmo de segmentación de carreteras contempla etapas de filtrado posteriores, donde se impondrán restricciones de forma a los carriles para eliminar aquellos que no sean parte de la carretera real. De esta forma, el índice de calidad decisivo en esta etapa del algoritmo de segmentación de carreteras es la sensibilidad, dado que esta nos señala que porcentaje de marcas viales de las que se deberían detectar, se han identificado correctamente, las marcas que no sean detectadas suponen un problema serio por cuanto en etapas posteriores, esta información permanecerá ausente del procesamiento de imágenes.

Ahora bien, con el ánimo de mejorar el índice de precisión del algoritmo de detección de marcas viales se implementa una nueva etapa de filtrado que restringe la zona donde los elementos detectados como marca vial, deberían existir, es decir, se definirá una máscara de espacio libre.

4.2. Máscara de Espacio libre

El problema de no poder evitar la detección de marcas viales en zonas de la imagen, vista desde la perspectiva superior, que no cumplen con la restricción de

mundo plano, se aborda a través de la segmentación de la carretera, definida como máscara de espacio libre o \mathcal{FSM} (Free Space Mask).

La segmentación de la carretera se puede realizar a través de la búsqueda de distintos descriptores tales como: la posición 3D de pixel, el color característico de la calzada,[3], ó por la textura del asfalto, [64]. En este sentido se proponen 3 métodos para obtener la \mathcal{FSM} :

1. Método Geométrico. La máscara de espacio libre se define a partir de todos los puntos ${}^m p_i$ en el PC o nube de puntos de la escena, que se encuentran a una distancia al plano de la carretera π , inferior a un cierto umbral δ , conjunto denominado consenso del plano carretera o ${}^m CS(\pi)$, ec. (4.4). Una vez segmentado el plano en las coordenadas del mundo $\{m\}$, se proyecta dicha información a la cámara $\{c\}$, a través de la transformación espacial ${}^c T_m$, donde se requiere definir la máscara. Las posiciones ${}^c CS(\pi)$ obtenidas en la imagen, permiten definir la \mathcal{FSM} , ec. (4.5).

$$\begin{aligned} {}^m CS(\pi) &= \{ {}^m p_i \in PC : d({}^m p_i, \pi) \leq \delta \} \\ {}^s {}^c CS(\pi) &= K {}^c T_m {}^m CS(\pi) \end{aligned} \quad (4.4)$$

$$\mathcal{FSM}(u, v) = \begin{cases} 1 & (u, v) \in {}^c CS(\pi) \\ 0 & \text{En otro caso.} \end{cases} \quad (4.5)$$

2. Método por multi-crecimiento de regiones. La segmentación se implementa a través de la búsqueda progresiva de pixeles, partiendo de un conjunto de puntos semilla escogidos uniformemente de ${}^c CS(\pi)$, cuyo valor de intensidad es tomado como referencia. Los vecinos con conectividad 4 u 8, son verificados para ser integrados a la región en crecimiento, un vecino se adiciona, si su valor de intensidad esta en el rango del valor semilla \pm un umbral.

Es importante aclarar que si la densidad del conjunto escogido es lo suficientemente grande, el proceso iniciará simultáneamente desde todas las posibles sub-regiones sobre la carretera, correspondientes a sombras, distintos tonos de asfalto debidos a la incidencia de la luz, y líneas pintadas sobre ella.

Finalmente, de las posiciones definidas como espacio libre, producto del crecimiento de regiones, se excluyen de la máscara final \mathcal{FSM} , aquellas que pertenecen al conjunto de no consenso ${}^m NCS(\pi)$ del plano de la carretera, ec. (4.6).

$${}^mNCS(\pi) = \{{}^mp_i \in PC : d({}^mp_i, \pi) \geq \delta\} \quad (4.6)$$

3. Método por transformada Watershed. Este método posee varias versiones, en particular se ha utilizado la propuesta por [71].

El algoritmo consiste en crecimiento de regiones simultaneas a partir de zonas fijadas como entrada al proceso. La idea que emula el algoritmo, es el movimiento de diferentes fluidos que no se mezclan, cuando son vertidos sobre un terreno desde diferentes puntos característicos, en la búsqueda de un punto de equilibrio.

Dado que la simple idea tiende a generar sobre segmentación debida al ruido presente en algunas texturas, se requiere, pre-procesar la imagen para obtener un conjunto adecuado de puntos que representen adecuadamente cada zona de la imagen que se requiera limitar.

- Inicialmente se escoge un conjunto de marcadores desde donde el crecimiento comienza. A cada marcador se le asocia una etiqueta que representa la región donde se encuentra ubicado el punto.
- Se inserta los vecinos de cada área marcada en una cola ordenada según la intensidad de los pixeles.
- Se prueba el pixel con menor intensidad. Si todos los vecinos ya etiquetados, del pixel bajo prueba, poseen la misma clasificación, entonces, dicho pixel recibe la etiqueta común. Los vecinos aun no marcados son integrados a la cola.
- Se repite el paso anterior hasta que la cola este vacía.

Los pixeles no marcados constituyen las líneas divisorias, llamadas watersheds.

Para limitar el número de regiones a 2, se escogen los marcadores del paso 1 como inliers y outlier, extraídos de la definición de cercanía o no, al plano de la carretera, ec. (4.4) y (4.6).

Una vez se han marcado todos los puntos sobre la imagen que cumplen el criterio de búsqueda, se realiza la transformación de perspectiva inversa a vista de pájaro de la \mathcal{FSM} segmentada sobre la imagen en perspectiva real. Como consecuencia de que para algunas escenas la segmentación deja fuera de la \mathcal{FSM} las líneas de la carretera, es necesario aplicar operaciones morfológicas de cerramiento hasta conseguir un área uniforme que cubra la zona de interés, de ello depende la aceleración de

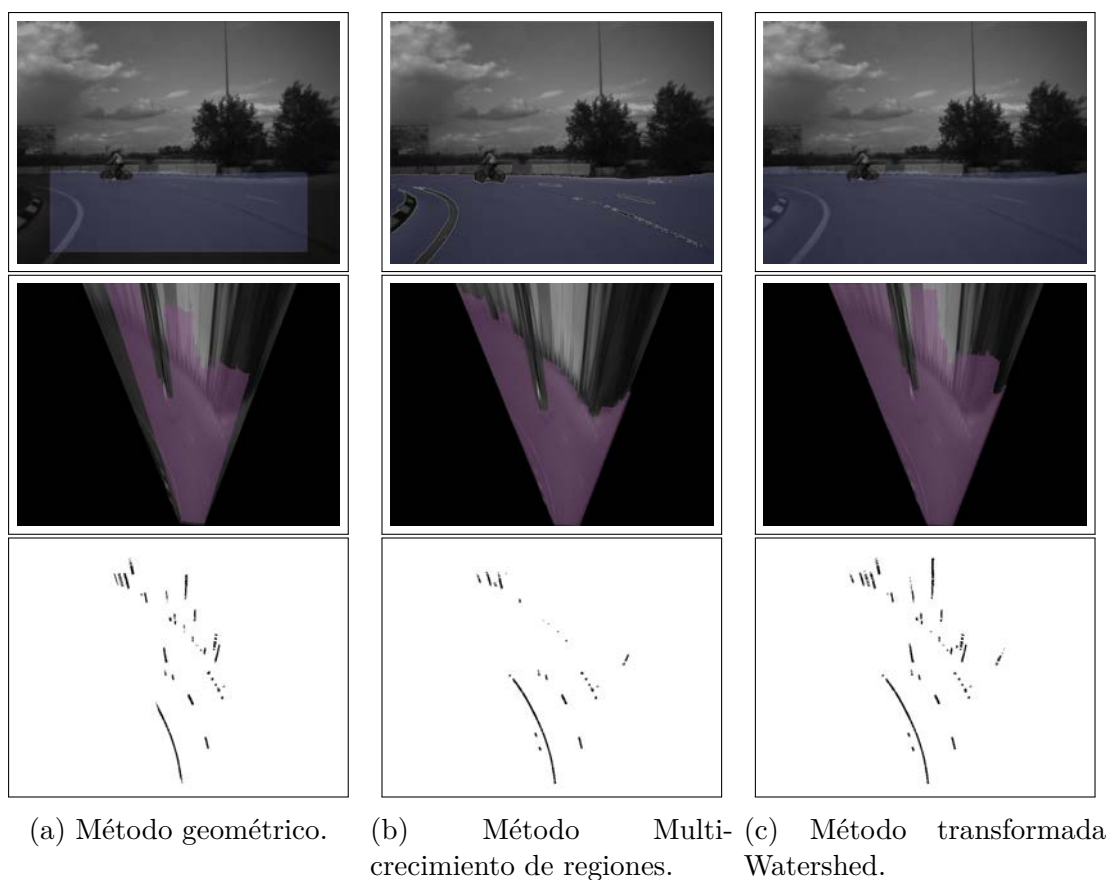


Figura 4.13: Comparación: \mathcal{FSM} resultantes.

procesos posteriores como la búsqueda de líneas. En la fig. 4.13, se puede observar las máscaras de espacio libre resultantes de cada método.

Por último, con el objeto de medir el efecto que produce la aplicación de la máscara de espacio libre, obtenida por los tres métodos enunciados anteriormente, en la detección de marcas viales, se repite el experimento de medición cuantitativa de marcas viales para el método Seg. Canny-grad. presentado en la sec. 4.1.1. Los resultados, representados en los índices de sensibilidad y precisión para la secuencia de referencia de marcas viales 2 fig. 4.5b, se muestran en las fig. 4.14 y 4.15.

Los resultados anteriores, indican que el método de detección de marcas viales propuesto con fase 1 por detector Canny, junto con la aplicación de la máscara de espacio libre denominada Método geométrico, aplicada sobre la secuencia de referencia para marcas viales 2 fig. 4.5b, produjo la mejor relación Sensibilidad+Precisión promedio, 67%. Se observa una leve disminución en el índice de sensibilidad de 8%, fig. 4.14, en relación al obtenido sin máscara de espacio libre fig. 4.8, al tiempo que ocurre un incremento en el índice de Precisión de 6%, fig. 4.15, en relación también

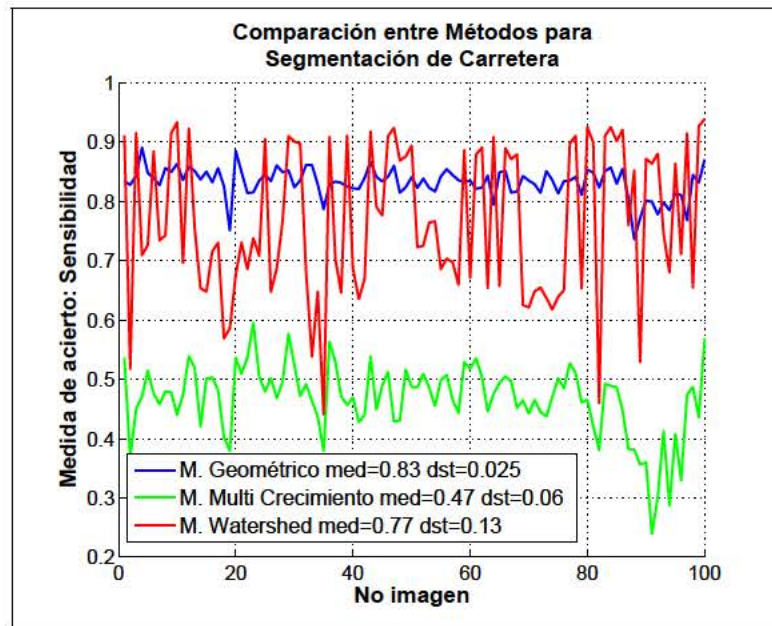


Figura 4.14: Comparación del índice de Sensibilidad aplicando la máscara de espacio libre obtenida por 3 diferentes métodos, sobre la secuencia 2 de referencia fig. 4.5b.

al obtenido sin máscara de espacio libre 4.10.

Por otra parte, la máscara de espacio libre denominada método de multi-crecimiento de regiones, aunque exhibe la mayor precisión, con un 74 %, exhibe también el menor índice de sensibilidad, 47 %, condición no deseada, como se explicó anteriormente porqué la cantidad de marcas viales no reconocidas es mayor a la mitad de todas las posibles. Este comportamiento se debe a que la máscara es muy restrictiva y no solo elimina regiones de distorsión, sino además zonas donde las marcas viales existen.

Con respecto a la máscara de espacio libre denominada método Watershed, ésta muestra la mayor desviación típica para el índice de sensibilidad y precisión, lo cual indica que en algunas imágenes, las condiciones de iluminación, llevan a la convergencia de zonas de carretera completamente erróneas.

4.3. Detección de Líneas y Carriles Viales

La detección de líneas, lin_i se realiza luego de haber construido la máscara de marcas viales, y posterior a ello, haber ajustado la zona de interés al espacio libre, a través de la mascarada creada para tal fin, \mathcal{FSM} . La búsqueda de líneas se lleva a cabo, a través del algoritmo de Hough en su versión probabilística [67], como se observa en la fig. 4.16. Allí, la transformada es aplicada sobre una imagen de carretera tipo

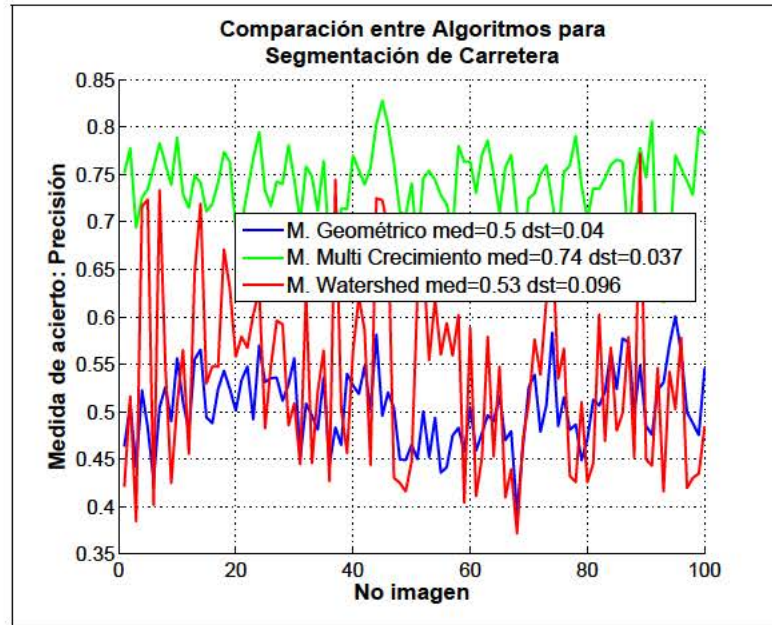


Figura 4.15: Comparación del índice de Precisión aplicando la máscara de espacio libre obtenida por 3 diferentes métodos, sobre la secuencia 2 de referencia fig. 4.5b.

Parámetro	Definición
umb_{hs}	Umbral para la selección de línea en el espacio Hough.
l_{min}	Mínima longitud de línea detectada.
s_{max}	Máxima separación entre segmentos de línea, que hacen parte de una línea detectada.

Cuadro 4.1: Parámetros de configuración para la transformada Hough.

rotonda, donde a demás, es posible comparar el efecto de la máscara \mathcal{FSM} , sobre la detección de líneas. Cuando el enmascaramiento es implementado, los resultados muestran una reducción significativa de líneas detectadas, no coherentes con marcas viales, mientras que las líneas reales de la carretera se mantienen constantes.

Los parámetros usados en el algoritmo Hough restringen las características de las líneas detectadas. Por tanto, dado que el siguiente objetivo en la modelación de la carretera, es detectar líneas viales bajo el estándar IC 8.2 [24], los valores de los parámetros Hough, se expresan en función de las magnitudes que tienen las marcas viales en el sistema de referencia del mundo, tal como se resume en el cuadro 4.1 y cuyas relaciones analíticas se muestran en la ec. (4.7).

$$l_{min} = ss_{disc} \frac{f_x}{Z_{cbv}} \quad s_{max} = sb_{inc} \frac{f_x}{Z_{cbv}} \quad (4.7)$$

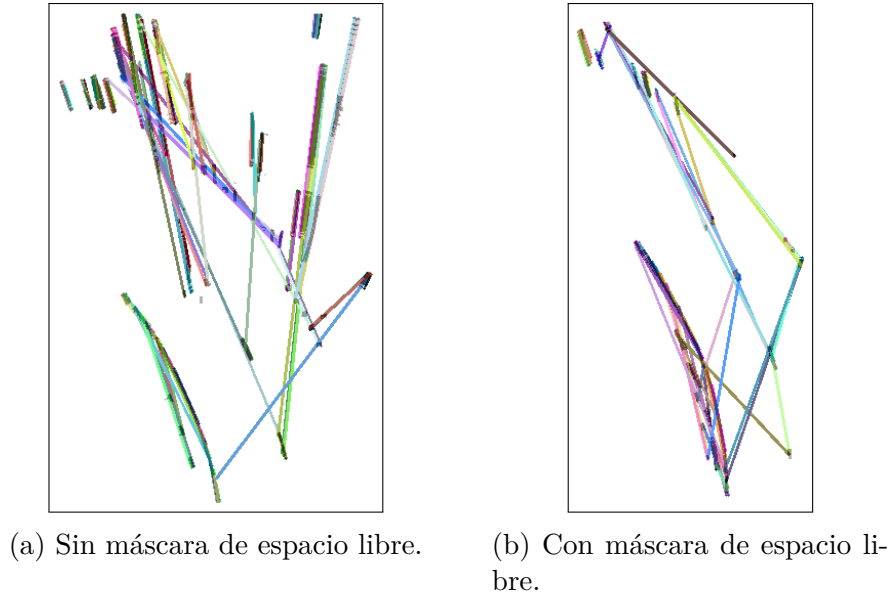


Figura 4.16: Detección de líneas de la carretera, basada en la transformada de Hough.

Donde, ss_{disc} , es la longitud del espacio entre segmentos en una línea discontinua. sb_{inc} , es la longitud del segmento blanco de una línea de incorporación. Z_{cbv} , es la altura de la cámara virtual desde donde es observada la vista de pájaro, f_x es la distancia focal de la cámara. Todos los parámetros son tomados en metros.

Las líneas resultantes de la aplicación de la transformada de Hough, se definirán en adelante por sus puntos extremos, como una dupla $(lin_i.p_1, lin_i.p_2)$, ó por sus parámetros característicos: pendiente m , e intercepto b , también como una dupla $(lin_i.m, lin_i.b)$. Así, el conjunto total de líneas se define como $Lin = \{lin_i = (lin_i.p_1, lin_i.p_2) : i \in \mathbb{N} ; 1 \leq i \leq n\}$

Ahora bien, el conjunto de líneas detectadas, por sí solo, no proporciona información suficiente para interpretar adecuadamente la carretera, por lo tanto, es necesario ascender en la identificación de elementos con mayor jerarquía conceptual, tal es el caso de los carriles, y finalmente, conjuntos consecutivos de ellos, denominados carreteras.

La búsqueda de objetos tipo ‘carril’, lan_j , se realiza filtrando pares de líneas, $(lin_k, lin_l) : lin_k \in Lin, y lin_l \in Lin$, paralelas entre si, y separadas una distancia igual a la longitud transversal de carril estándar rlw , según la norma IC 8.2 [24]. Las restricciones que deben cumplir los elementos constitutivos de los nuevos carriles, se muestran en la ec. (4.8).

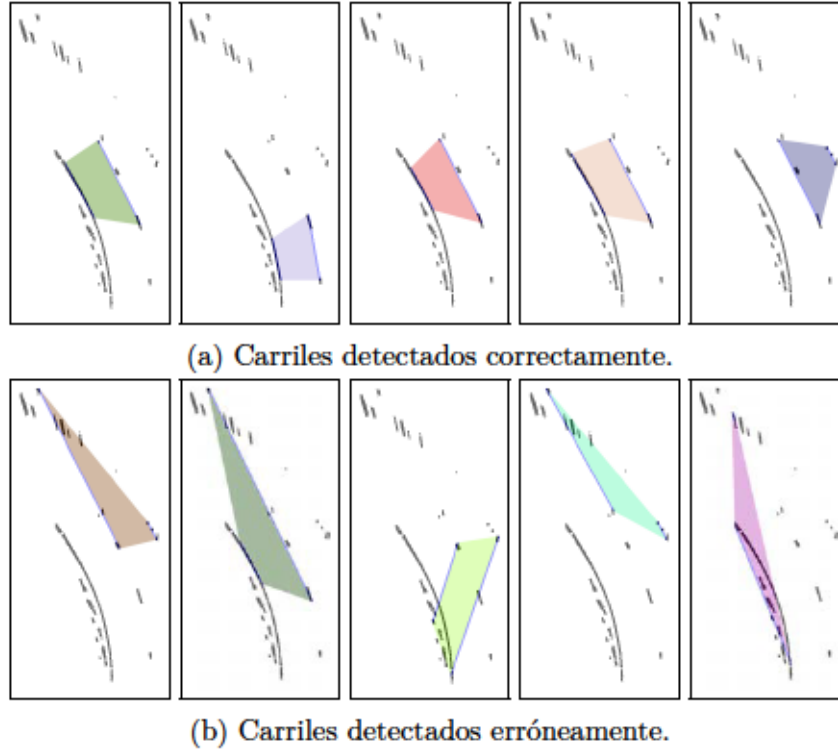


Figura 4.17: Detección de carriles viales de la carretera, basada en la transformada de Hough.

1. $\text{máx}(d_{\min}(\text{lin}_k \cdot p_m, \text{lin}_l), d_{\min}(\text{lin}_l \cdot p_m, \text{lin}_k)) < rlw,$
 2. $|\angle \text{lin}_k - \angle \text{lin}_l| < \epsilon_{prl}$
- (4.8)

Donde, d_{\min} , es la distancia mínima entre un punto del espacio, p y una recta, lin , según se presenta en la ec. (4.9). $\text{lin}_k \cdot p_m$, es el punto medio de la línea lin_k .

$$d_{\min}(p, \text{lin}) = \frac{(\text{lin}.m * p.x) - (p.y - \text{lin}.b)}{\sqrt{\text{lin}.m^2 + 1}} \quad (4.9)$$

El conjunto total de carriles detectados, denominado $\mathcal{R} = \{\text{lan}_j = (\text{lin}_1^j, \text{lin}_2^j) : j \in \mathbb{Z}; 1 \leq j \leq m\}$, se puede apreciar en la fig. 4.17, diferenciados como: correctamente detectados en la primera fila, y carriles erróneos en la segunda fila. Las detecciones se realizaron sobre la máscara de marcas viales reducida a espacio libre de la fig. 4.13.

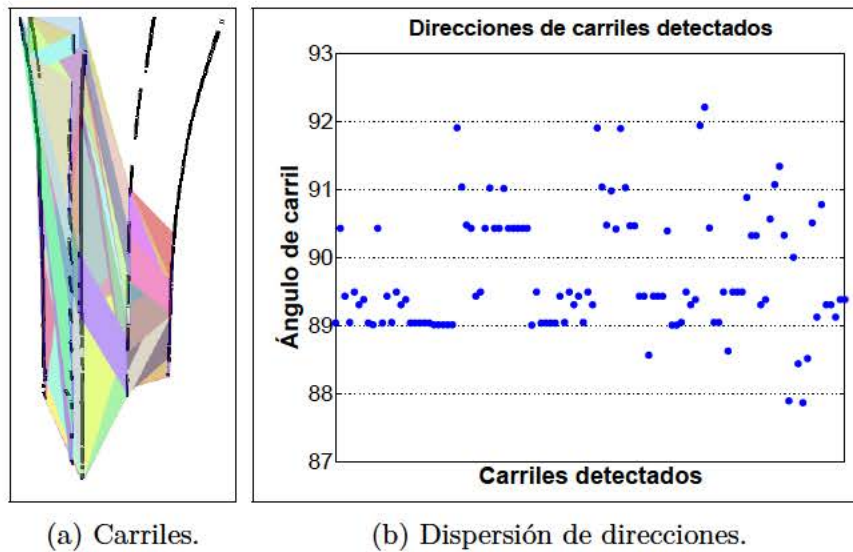


Figura 4.18: Direcciones en carriles detectados.

4.3.1. Detección del ángulo de la carretera.

El conocimiento del ángulo de la carretera ϕ_C , permite filtrar los carriles que realmente pertenecen a la vía, de aquellos que, cumpliendo las restricciones de la ec. (4.8), se presentan coincidentalmente en lugares aleatorios y con direcciones dispersas.

El problema de encontrar la dirección que represente a la mayoría de los carriles detectados en el procedimiento anterior, parte de la hipótesis siguiente: “la identificación de carriles sobre los contornos de líneas viales, deben exhibir una mayor probabilidad de ocurrencia, dada la imposición de las restricciones: (4.1) y (4.8)”.

Para comprobar la hipótesis de partida, primero se analiza una imagen del tipo entorno carretera, en la cual, puedan ser identificados carriles en diferentes direcciones. La dispersión de direcciones $D = \{d_j = (\angle \text{lin}_1^j + \angle \text{lin}_2^j)/2; j \in \mathbb{Z}; 1 \leq j \leq m\}$, asociada al conjunto de carriles detectados \mathcal{R} , se puede observar en la fig. 4.18. Allí, se muestra la dificultad de escoger una dirección representativa, dados los múltiples puntos de acumulación. Para dar solución a este problema, a continuación se presentan dos enfoques que abordan la detección del ángulo ϕ_C , en ambos métodos, el ángulo corresponde al centro del grupo de direcciones, alrededor del cual se encuentra la mayor acumulación de puntos en la dispersión de carriles detectados.

4.3.1.1. Histograma de conjuntos soporte.

El método comienza con la definición de los conjuntos soporte $SP = \{sp_j : j \in \mathbb{Z}; 1 \leq j \leq m\}$, asociados a la dirección d_j de cada carril detectado lan_j , fig. 4.19, imagen superior. El soporte de la dirección del carril lan_j , se conforma por todas las direcciones cercanas a d_j en D , dentro de un radio de longitud r . Posteriormente, se construye el histograma con los tamaños ó cardinalidad de sp_j , fig. 4.19, imagen inferior. Finalmente, se extrae el conjunto de direcciones cuya cardinalidad es máxima, y entonces, el ángulo de la carretera se obtiene como la media del conjunto, tal y como se indica en la ec. (4.10).

$$\phi_C = \overline{d_j : |sp_j| = \max(|SP|)} \quad (4.10)$$

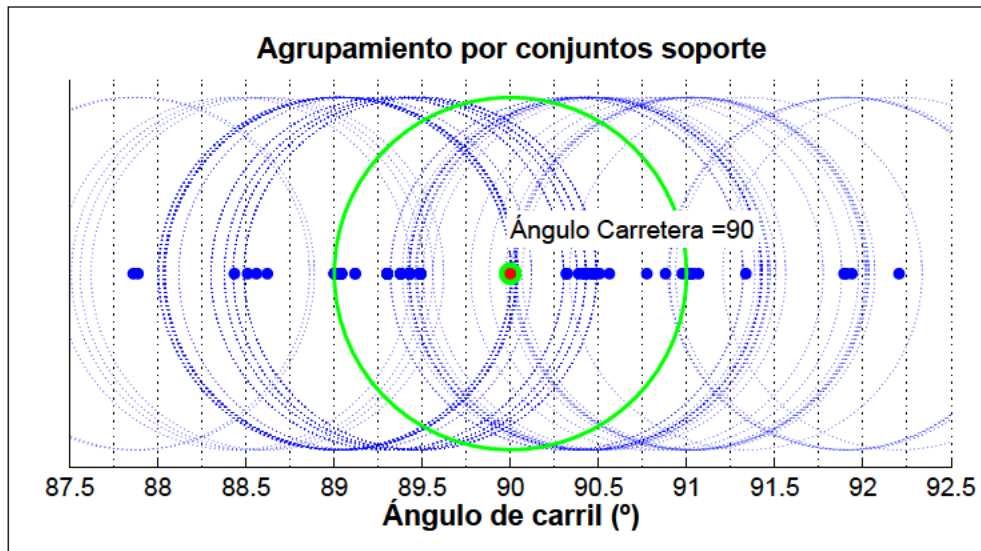
4.3.1.2. Agrupamiento Mean Shift.

El método consiste en aplicar la técnica de agrupamiento denominada Mean Shift, la cual se adapta perfectamente al problema, ya que a priori no se conoce el número de grupos, pero sí se tiene un estimado del tamaño de los conjuntos. El radio de comprobación de vecinos se limita a la precisión con que se espera conocer el ángulo de la carretera y se asume como la mitad del umbral ϵ_{prt} , utilizado como restricción de ancho en la detección de carriles. El resultado de aplicar la técnica a la dispersión D , de la fig. 4.18, se aprecia en la fig. 4.20.

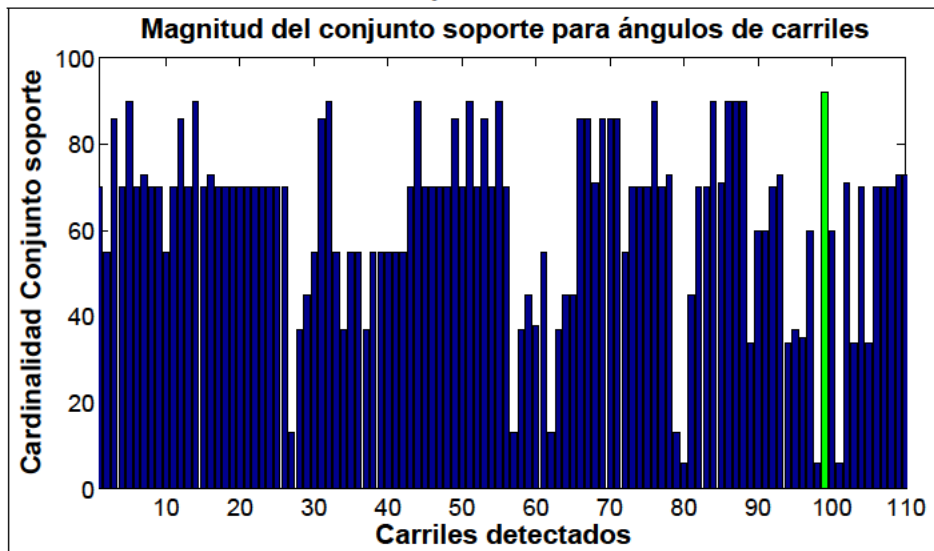
En este caso, como consecuencia de tener grupos de puntos separados una distancia inferior al radio de vecindario, el método de extracción del ángulo de la carretera por conjunto de soporte, tiende a encontrar el centro de toda la dispersión D , mientras que el método basado en Deslizamiento de Media ó Mean Shift, encuentra dos grupos, dada su robustez para tratar ruido inter-clase, y así, lograr identificar el centro del grupo con mayor número de vecinos y por lo tanto, un valor mas preciso que en el primer método.

4.3.1.3. Análisis comparativo de los métodos para detección del ángulo de carretera.

Para observar el comportamiento de los 2 algoritmos propuestos para detección del ángulo de la carretera se utilizarán las 2 secuencias de referencia que aparecen en la fig. 4.21. La secuencia que se muestra en fig. 4.21a corresponde a un trayecto de carretera levemente curvo de 2 carriles, conformado por 100 imágenes en donde



(a) Conjuntos soporte. Los límites de cada conjunto soporte se muestran como círculos azules punteados. Los conjuntos de soporte máximo se muestran en verde. La dirección de la carretera se observa como un punto rojo.



(b) Histograma que muestra la magnitud de los conjuntos soporte asociados a cada dirección de carril.

Figura 4.19: Detección del ángulo de la carretera a través de conjuntos soporte.

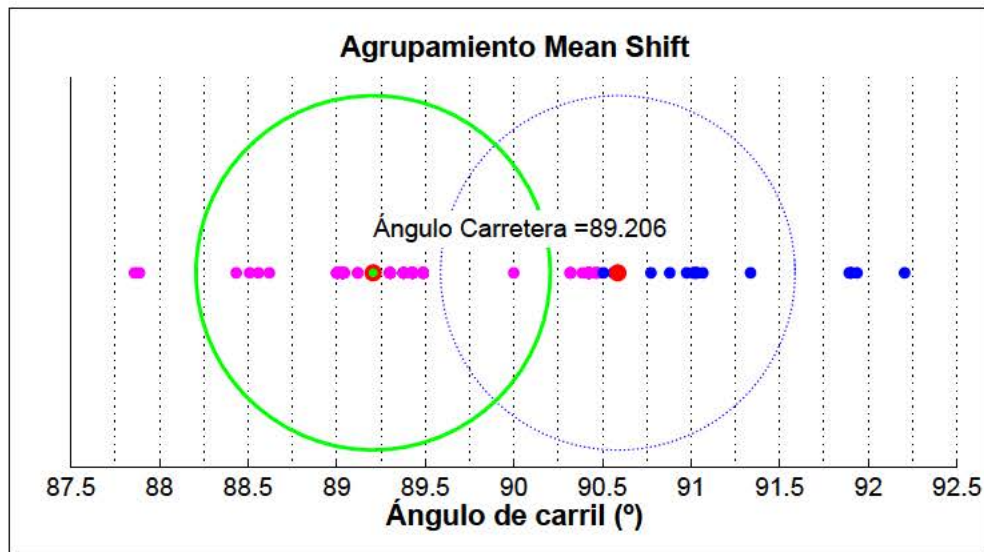


Figura 4.20: Detección del ángulo de la carretera usando agrupamiento Mean Shift.

aparece barandilla en ambos costados de la vía. La secuencia que se muestra en fig. 4.21b corresponde a una sección de rotonda de 5 carriles conformada por 60 imágenes.



Figura 4.21: Algunas imágenes de la secuencia de referencia (a) 1 y (b) 2, para la detección de ángulo de carretera.

A continuación en las fig. 4.22 y 4.23, se presenta el resultado de utilizar el método basado en Histograma de Soporte de ángulo y el método denominado Mean Shift, para detección del ángulo referencia e los carriles que pertenecen a la carretera. En las gráficas central e inferior de cada figura, se muestra el número de carriles detectados que cumplen con la definición de la ec. 4.8 en gris, así cómo, el número de carriles que son soporte del ángulo detectado por cada método.

Tanto para el caso del tramo recto de carretera de la fig. 4.21a, como para la sección de rotonda de la fig. 4.21b, los dos métodos se comportaron aproximadamente igual. En tramo recto la diferencia detectada en media entre los métodos fue de 0.04 grados, mientras que para el caso de trayecto curvo fue de 0.84 grados. La

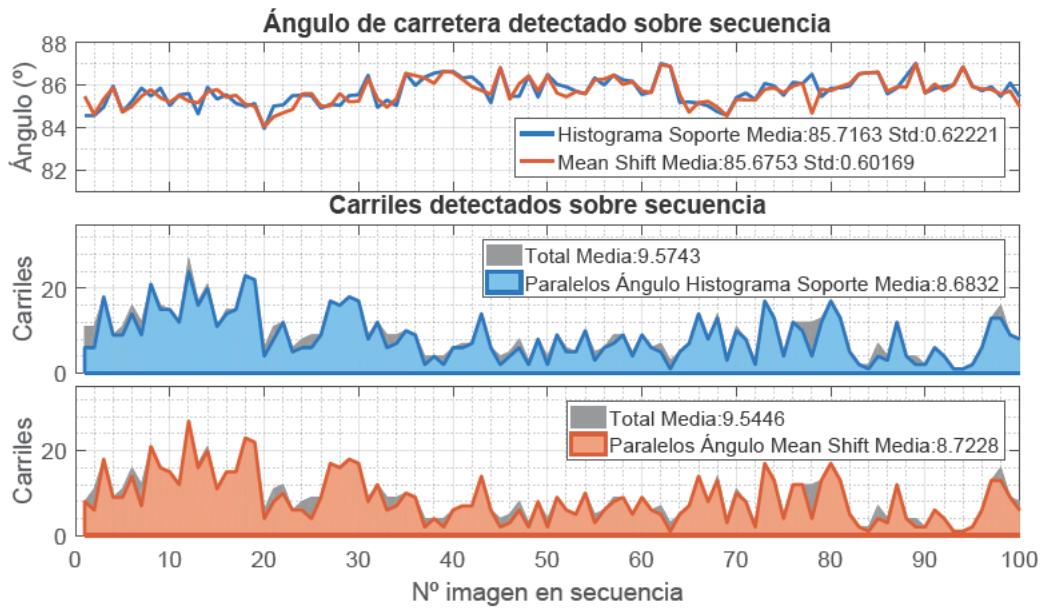


Figura 4.22: Comparación de los métodos para detección de ángulo de carretera sobre la secuencia de la fig. 4.21a.

comparación entre valores de media no se ve influenciado por las variaciones de ángulo de carretera detectados por efecto del movimiento del vehículo, intentando corregir su dirección para seguir el contorno de la rotonda. La desviación típica en un tramo recto, aunque se ve afectada también por el movimiento del vehículo, refleja una medida indirecta de precisión en la estimación del ángulo de la carretera, de alrededor de 0.6 grados medidos por ambos métodos. La desviación típica de un trayecto curvo refleja el movimiento correctivo del vehículo, de 6.5 grados para el caso de la medición por histograma soporte y 5.4 grados para el caso del método de clustering Mean Shift.

Finalmente, se observan errores en 4 imágenes sobre la secuencia tipo rotonda, en las cuales, la medición del ángulo de carretera falla para uno o ambos métodos, eventos que se traducen en una tasa de detección de alrededor del 94 %.

4.3.2. Identificación de Conjuntos Adyacentes de Carriles

Se denominará en adelante *estructura de carretera*, $\mathcal{R}_{perc} = \{P_i = lan_i; i = 1, \dots, n_P\}$, al subconjunto de carriles, extraído del conjunto, \mathcal{R}_{ϕ_C} , que son consistentes con la dirección ϕ_C . \mathcal{R}_{perc} se forma buscando una cadena de carriles físicamente adyacentes, y no solapados, como se muestra en la fig. 4.24. Es importante aclarar que la detección de un carril se puede dar de manera parcial, solo es necesario que dos

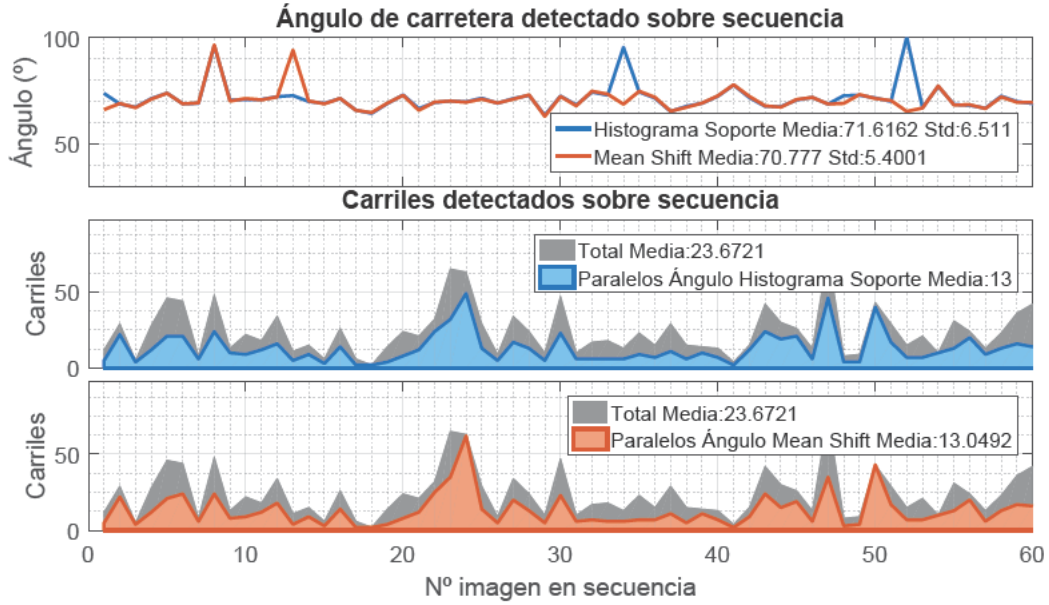


Figura 4.23: Comparación de los métodos para detección de ángulo de carretera sobre la secuencia de la fig. 4.21b.

sub-segmentos de línea sean detectados a lado y lado de un carril para considerarlo como carril detectado, y por lo tanto, el resultado de la búsqueda de carretera es un lista de cuadriláteros adyacentes en el sentido de que un par de sus líneas separadoras extrapoladas sean adyacentes.

La búsqueda de \mathcal{R}_{perc} se implementa de la siguiente manera. Sea $\mathcal{R}_{\phi_C} = \{lan_i : lan_i \in \mathcal{R}; 1 \leq i \leq m \wedge \angle lan_i \parallel \phi_C\}$, se escoge aleatoriamente un carril inicial, $lan_i = lan_{i^*} \in \mathcal{R}_{\phi_C}$, a partir de allí, se comprueba con cada carril restante en \mathcal{R}_{ϕ_C} , las restricciones de la ec. (4.11).

- 1) $|\angle lan_i - \angle lan_j| \leq \theta_{atin}$
donde $\angle lan = \frac{\angle lan.lin_1 + \angle lan.lin_2}{2}$ (4.11)
- 2) si $dist_{n,m} = d(lan_i.lin_n, lan_j.lin_m) | n = 1, 2 \wedge m = 1, 2$
entonces $\exists! n^*, m^* | dist_{n^*, m^*} \leq \epsilon_{slp}$

Cada vez que un carril es insertado en \mathcal{R}_{perc} , desaparece del conjunto de carriles en cola de búsqueda de \mathcal{R}_{ϕ_C} . Dado que la restricción de carril adyacente, adiciona un nuevo elemento lan_j a \mathcal{R}_{perc} , la segunda inserción, define una dirección de búsqueda, y por tanto, una vez encontrada la primera subcadena, \mathcal{R}_{perc}^{d1} , es necesario retornar al

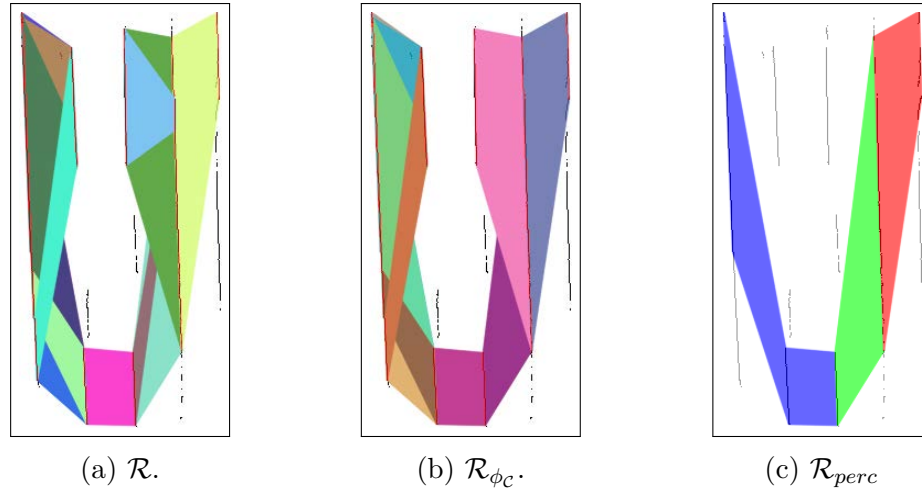


Figura 4.24: (a) Carriles detectados según la ec. (4.8). (b) Carriles detectados en la dirección próxima a ϕ_C . (c) Resultado de la búsqueda de carriles adyacentes y no solapados en dirección ϕ_C . En verde se representa el carril inicial de la búsqueda lan_{i^*} , en azul y rojo, aparecen los carriles obtenidos de la búsqueda en direcciones opuestas partiendo del carril inicial, \mathcal{R}_{perc}^{d1} y \mathcal{R}_{perc}^{d2} .

carril de partida de la primera búsqueda, para encontrar la subcadena en la dirección complementaria, \mathcal{R}_{perc}^{d2} , si esta llegase a existir. El resultado final de la búsqueda se encuentra a través de la unión de ambas subcadenas así: $\mathcal{R}_{perc} = \mathcal{R}_{perc}^{d1} \cup \mathcal{R}_{perc}^{d2}$.

4.4. Integración Temporal

Finalmente, se lleva a cabo un filtrado temporal de los carriles, basado en la actualización de dos conjuntos de datos persistentes entre capturas sucesivas: $\mathcal{R}_{mem} = \{L_i; i = 1, \dots, n_L\}$, que contiene los carriles L_i que componen el modelo de la carretera, y $\mathcal{R}_{cand} = \{C_i; i = 1, \dots, n_C\}$, compuesta por los carriles candidatos a formar parte de dicho modelo de acuerdo a las detecciones extraídas de las últimas capturas. La información obtenida a partir del procesamiento de la imagen actual está contenida en \mathcal{R}_{perc} , y es la que se emplea para actualizar convenientemente las dos estructuras anteriores, de acuerdo al Alg. 4.4.1.

La descripción del algoritmo 4.4.1 se realiza a continuación. En la primera iteración del algoritmo todas las banderas son puestas a valor falso, además \mathcal{R}_{mem} comienza con el contenido de \mathcal{R}_{perc} , los carriles detectados de la primera captura. En capturas sucesivas se realizan los siguientes pasos:

1. Se compara cada carril L_i de \mathcal{R}_{mem} , con todos los carriles P_j de \mathcal{R}_{perc} , si

hay coincidencia se actualiza P_j en L_i a través de una copia de contenido, y además, se registra en las banderas: $L_i.act$ que L_i se actualizó, y en $P_j.mem$, que P_j modificó algún L_i de \mathcal{R}_{mem} . Ver Definición D2 y D3.

2. Para cada carril L_i de \mathcal{R}_{mem} , después de verificar el paso 1, en dependencia de si L_i ha sido actualizado o no, se procede a incrementar o a disminuir su contador de vida en una unidad. Los valores posibles que puede tomar el contador de vida de un carril en cualquier estructura de carretera estarán comprendidos entre 0 y $count_{max}$.
3. A continuación se verifica que los contadores de vida de los carriles: inicial y final de \mathcal{R}_{mem} (vistos espacialmente como derecho e izquierdo), no estén en cero, de ser así, se eliminan de la estructura.
4. Ahora se compara cada carril P_i de \mathcal{R}_{perc} , con todos los carriles C_j de \mathcal{R}_{cand} , si hay coincidencia y P_i no había actualizado antes a ningún carril de \mathcal{R}_{mem} , entonces se actualiza P_i en C_j a través de una copia de contenido, y además, se registra en las banderas: $C_j.act$ que C_j se actualizó, y en $P_i.cand$, que P_i modificó algún C_j de \mathcal{R}_{cand} . Ver Definición D2 y D3.
5. Para cada carril P_i de \mathcal{R}_{perc} , después de verificar el paso 4, si P_i no ha actualizado ningún carril en \mathcal{R}_{mem} ni tampoco en \mathcal{R}_{cand} , entonces es insertado en \mathcal{R}_{cand} , fijando su correspondiente bandera $P_{nuevo.act}$ a true, con lo cual se garantiza que su contador de vida se incremente en el siguiente paso y entonces, tenga una oportunidad de sobrevivir.
6. Posteriormente, para cada carril C_i de \mathcal{R}_{cand} , según si C_i haya sido actualizado o no, se procede a incrementar o a disminuir su contador de vida en una unidad.
7. si el contador de vida de C_i ha llegado al valor máximo y es contiguo a \mathcal{R}_{mem} por izquierda o derecha, ver definición D12, entonces se inserta C_i en \mathcal{R}_{mem} con contador de vida igual al máximo valor, para garantizar que sea persistente por algunas capturas, si éste no llegase a ser detectado en capturas posteriores. Si no se cumple la condición, no se produce acción.
8. finalmente, si se verifica que el contador de vida del carril C_i no sea cero, de ser así, se elimina de la estructura \mathcal{R}_{cand} .

La síntesis del algoritmo anterior se puede entender a través de la evolución del estado de la carretera memorizada \mathcal{R}_{mem} , y su interacción con las demás estructuras, \mathcal{R}_{cand} y \mathcal{R}_{perc} , según como se observa en la fig. 4.25.

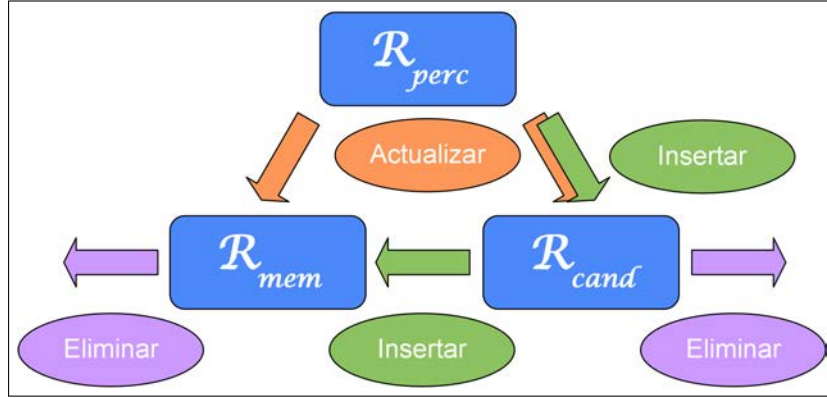


Figura 4.25: Relación entre estructuras de carriles para generar integración temporal.

Una vez comienza la integración temporal, en el instante inicial, la estructura \mathcal{R}_{mem} recibe los carriles detectados en la carretera, almacenados en \mathcal{R}_{perc} .

A partir del instante $k = 1$, los carriles detectados en \mathcal{R}_{perc} , P_i , pueden actualizar carriles en \mathcal{R}_{mem} ó \mathcal{R}_{cand} si existe correspondencia espacial con alguno de ellos, ver definición D3. En caso de no encontrarse correspondencia espacial, los carriles P_i pueden ser insertados en \mathcal{R}_{cand} , quedando en periodo de prueba. Cada vez que un carril es actualizado por un carril P_j , la función de energía del carril actualizado, $L_i.vida$ ó $C_i.vida$, se incrementa, ver definición D5. Si ningún carril P_j , actualiza los carriles L_i ó C_i , sus funciones de energía disminuyen.

Los carriles en observación, presentes en \mathcal{R}_{cand} pueden ser insertados en \mathcal{R}_{mem} ó eliminados de la estructura de carretera. La inserción ocurre si el valor de energía $C_i.vida$ ha alcanzado su valor máximo y se comprueba que existe contigüidad con el primero o último carril de la estructura \mathcal{R}_{mem} . Adicionalmente, un carril C_i es eliminado de \mathcal{R}_{cand} si su valor de energía $C_i.vida$ ha disminuido hasta cero o si ha alcanzado su valor máximo y no pudo ser insertado en \mathcal{R}_{mem} , ver definición D10.

Finalmente, aquellos carriles que pertenecen a \mathcal{R}_{mem} , que alcanzan un valor de energía, $C_i.vida$ igual a cero, son eliminados de la estructura si están en la primera o última posición.

Este procedimiento permite incrementar la fiabilidad del algoritmo frente a oclusiones o ruido espúreo generado por el movimiento de los otros vehículos en el campo de visión y frente a la deformación inducida por objetos que no pertenecen al plano

de la calzada.

Algoritmo 4.4.1: CONSISTENCIATEMPORAL(\mathcal{R}_{perc} , \mathcal{R}_{cand} , \mathcal{R}_{mem})

```

si  $\mathcal{R}_{mem} = \{ \}$ 
    entonces  $\mathcal{R}_{mem} \leftarrow \mathcal{R}_{perc}$ 
    {
        LIMPIARBANDERAS( $\mathcal{R}_{perc}$ ,  $\mathcal{R}_{cand}$ ,  $\mathcal{R}_{mem}$ )
        para cada  $L_i \in \mathcal{R}_{mem}$ 
            haz {
                para cada  $P_j \in \mathcal{R}_{perc}$ 
                    haz {
                        si  $(L_i \approx P_j)$ 
                            entonces {
                                 $L_i \leftarrow P_j$ ;  $L_i.act \leftarrow \text{verd.}$ 
                                 $P_j.mem \leftarrow \text{verd.}$ 
                            }
                        si  $L_i.act = \text{verd.}$ 
                            entonces INCREM( $L_i.vida$ )
                            si no DECREM( $L_i.vida$ )
                    }
            }
         $L_0 \leftarrow \text{EXTREMIZQ}(\mathcal{R}_{mem})$ ;
        si  $L_0.vida = 0$ 
            entonces EXTRAE( $\mathcal{R}_{mem}$ ,  $L_0$ )
         $L_f \leftarrow \text{EXTREMDER}(\mathcal{R}_{mem})$ ;
        si  $L_f.vida = 0$ 
            entonces EXTRAE( $\mathcal{R}_{mem}$ ,  $L_f$ )
    }
si no {
        para cada  $P_i \in \mathcal{R}_{perc}$ 
            haz {
                para cada  $C_j \in \mathcal{R}_{cand}$ 
                    haz {
                        si  $(P_i \approx C_j) \wedge (P_i.mem = \text{falso})$ 
                            entonces {
                                 $C_j \leftarrow P_i$ ;  $C_j.act \leftarrow \text{verd.}$ 
                                 $P_i.cand \leftarrow \text{verd.}$ 
                            }
                        si  $(P_i.mem = \text{falso}) \wedge (P_i.cand = \text{falso})$ 
                            entonces {
                                INSERTA( $\mathcal{R}_{cand}$ ,  $P_i$ );  $C_{nuevo}.act \leftarrow \text{verd.}$ 
                            }
                    }
                para cada  $C_i \in \mathcal{R}_{cand}$ 
                    haz {
                        si  $C_i.act = \text{verd.}$ 
                            entonces INCREM( $C_i.vida$ )
                            si no DECREM( $C_i.vida$ )
                        si {
                                 $(\text{ESMAX}(C_i.vida)) \wedge (\text{ESCONTIGUO}(C_i, \mathcal{R}_{mem}))$ 
                            }
                            entonces {
                                LLENA( $C_i.vida$ ); INSERTA( $\mathcal{R}_{mem}$ ,  $C_i$ )
                            }
                        si no, si  $C_i.vida = 0 \vee (\text{ESMAX}(C_i.vida)) \wedge$ 
                            ( $\neg \text{ESCONTIGUO}(C_i, \mathcal{R}_{mem})$ )
                            entonces EXTRAE( $\mathcal{R}_{cand}$ ,  $C_i$ )
                    }
            }
    }
    
```

Algunas definiciones que aparecen en el alg. 4.4.1 son:

- D1: $X_i.act$: Bandera que indica que el carril $X_i \in \mathcal{R}_x$ ha sido actualizado.
- D2: $X_i.y$: Bandera que indica que el carril $X_i \in \mathcal{R}_x$ ha actualizado al menos un carril $Y_j \in \mathcal{R}_y$.
- D3: $X_i \approx Y_j$: función de detección que define la equivalencia espacial entre 2 carriles, cumpliendo con las restricciones de paralelismo y solapamiento. El solapamiento se define como la contigüidad simultanea de dos carriles en sus dos líneas constitutivas.
- D4: $X_i \leftarrow Y_j$: Función que copia el contenido del carril Y_j en X_i .
- D5: $Increm(X_i.vida)$: Función que indica que el carril X_i incrementa su valor de energía en una unidad.
- D6: $Decrem(X_i.vida)$: Función que indica que el carril X_i reduce su valor de energía en una unidad.
- D7: $ExtremIzq(\mathcal{R}_x)$: Función que extrae el último carril por la izquierda de la estructura de carretera \mathcal{R}_x .
- D8: $ExtremDer(\mathcal{R}_x)$: Función que extrae el último carril por la derecha de la estructura de carretera \mathcal{R}_x .
- D9: $Extrae(\mathcal{R}_y, X_i)$: Función que extrae el carril X_i de la estructura de carretera \mathcal{R}_y .
- D10: $Inserta(\mathcal{R}_y, X_i)$: Función que inserta el carril X_i en la estructura de carretera \mathcal{R}_y .
- D11: $EsMax(X_i.vida)$: Función que verifica si el carril X_i ha alcanzado el valor máximo de energía.
- D12: $EsContiguo(X_i, \mathcal{R}_y)$: Función que establece la contigüidad entre el carril X_i y la estructura de carreteras \mathcal{R}_y .
- D13: $Llena(X_i.vida)$: Función que fija en el valor máximo la energía del carril X_i .

En la fig. 4.26 se muestra la evolución temporal de la estructura de carretera \mathcal{R}_{mem} en función de los carriles detectados en la estructura de carretera \mathcal{R}_{perc} . Allí, cada sub-figura corresponde a un tiempo discreto k donde es analizada la carretera en una secuencia de 27 fotogramas. En cada cuadro se observan dos imágenes, en la parte superior aparece el estado de la estructura \mathcal{R}_{mem} , mientras que en la imagen inferior, aparece el estado de la estructura \mathcal{R}_{perc} . Los carriles en color verde que se observan en las imágenes superiores, representan aquellos carriles que han sido memorizados en el pasado y que en el tiempo presente, no han sido detectados. Los carriles que se observan en color rojo, son aquellos que se encontraban memorizados y que además, han sido actualizados por coincidir espacialmente con un carril de la percepción actual de la carretera \mathcal{R}_{perc} . Los carriles que se muestran en color azul sobre las imágenes inferiores, permiten conocer el estado de la estructura de carretera que ha sido detectada en el instante actual. Como se puede observar a lo largo de la secuencia, en los primeros cuadros de (1) a (11), la cantidad de carriles del modelo de carretera memorizados fueron 2, de ahí en adelante, se integraron 2 carriles adicionales, uno a la vez en los cuadros (12) y (13), completándose el número total de carriles reales de la carretera de 4 carriles. En cuadros posteriores, el número de carriles permanece invariante, aún cuando en la estructura de carretera \mathcal{R}_{perc} , se detecten menos carriles (15) o más carriles (17), por efecto de oclusiones o distorsiones de la imagen.

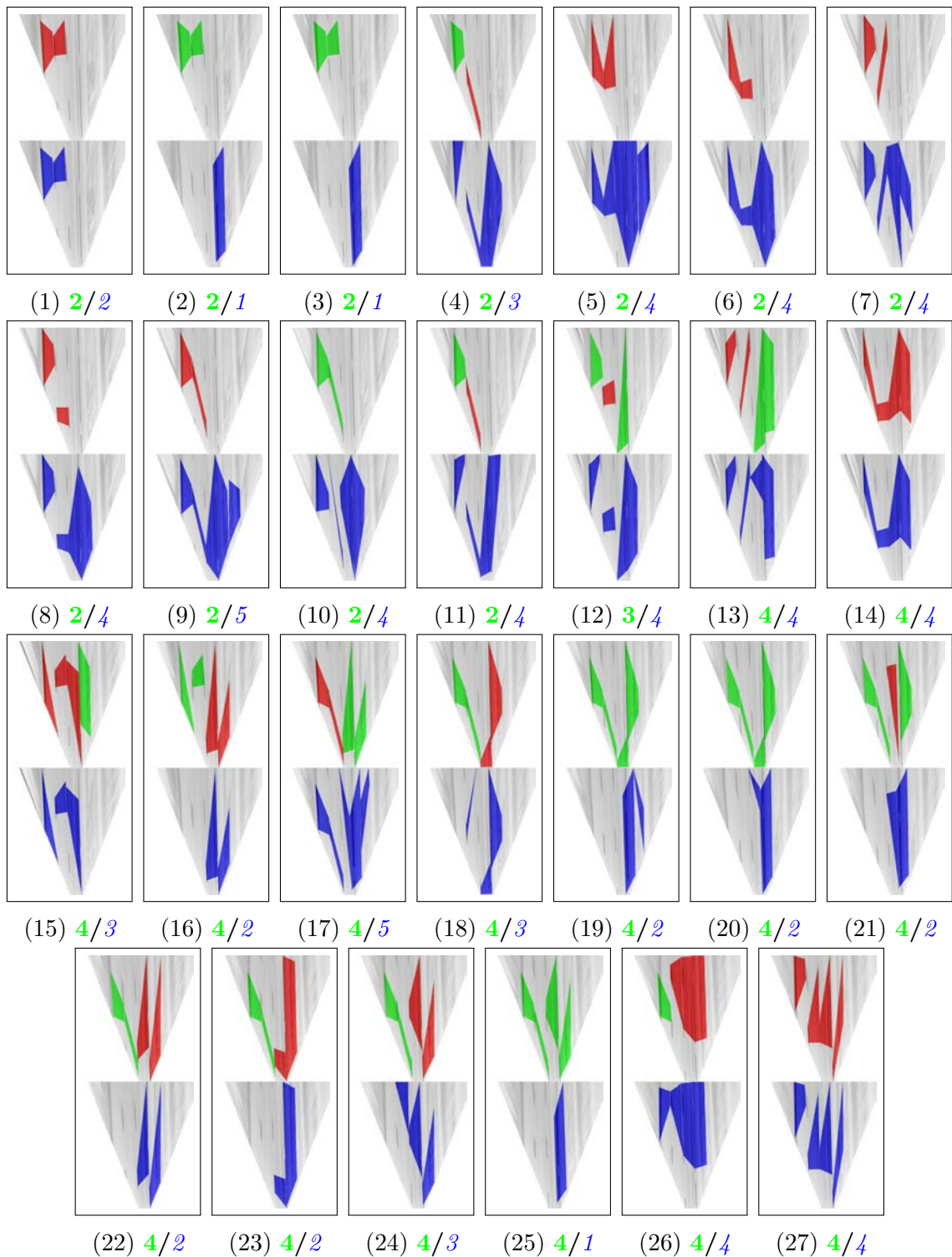


Figura 4.26: Integración temporal a lo largo de 27 detecciones consecutivas de carriles en una carretera interurbana de 4 carriles. La imagen superior de cada cuadro muestra el contenido de \mathcal{R}_{mem} en color verde, mientras que la imagen inferior, muestra el contenido de \mathcal{R}_{perc} en color azul. En color rojo se marca las actualizaciones de carriles desde \mathcal{R}_{perc} a \mathcal{R}_{mem}

4.5. Rendimiento del Algoritmo para Detección de Carreteras

La comprobación funcional del algoritmo propuesto para la detección de carreteras con y sin integración temporal, se realiza sobre dos secuencias de imágenes tomadas desde la plataforma de pruebas IVVI 2.0 [66], denominadas secuencia 1 para detección de carriles, fig. 4.27 y secuencia 2 para detección de carriles, fig. 4.28. Además se utilizan dos secuencias de referencia del repositorio para pruebas de percepción KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago) [32], secuencia 3 para detección de carriles fig. 4.29 y secuencia 4 para detección de carriles fig. 4.30. Adicionalmente, se hace uso de la secuencia 1 para detección de marcas viales fig. 4.5a, usada en la sección 4.1.2.

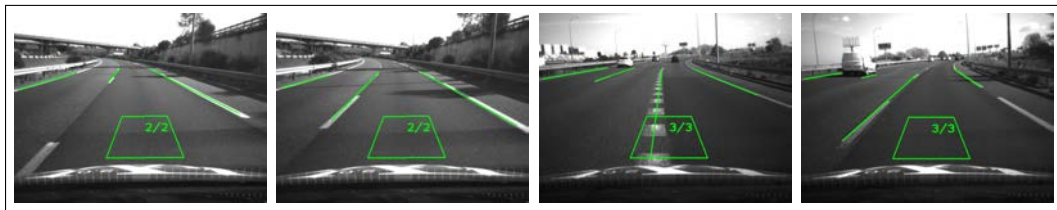


Figura 4.27: Secuencia 1 para la detección de carreteras, tamaño 1000 imágenes.

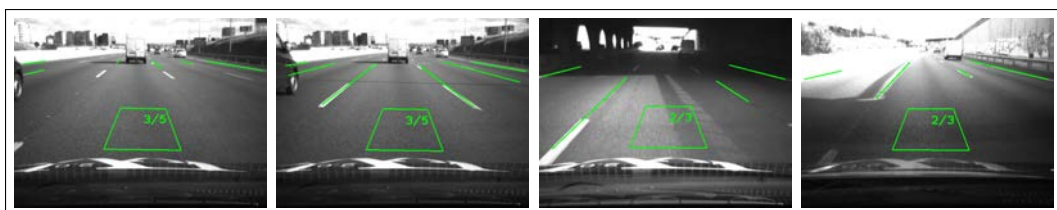


Figura 4.28: Secuencia 2 para la detección de carreteras, tamaño 500 imágenes.

La secuencia 1 es un tramo de carretera de 2 carriles que se convierte en una carretera de 3 carriles, con poco tráfico vehicular, líneas bien demarcadas e iluminación promedio, condiciones ideales para el algoritmo base de detección de carreteras. La secuencia 2 por su parte, es una carretera de 4 carriles que se convierte en una de 3 carriles, con flujo vehicular medio y una alta iluminación solar, condiciones exigentes para el algoritmo de detección de carreteras propuesto.



Figura 4.29: Secuencia 3 para la detección de carreteras, Raw:Road:Drive.0015, tomada del KITTI [32], tamaño 297 imágenes.



Figura 4.30: Secuencia 4 para la detección de carreteras, Raw:Road:Drive.0027, tomada del KITTI [32], tamaño 188 imágenes.

La secuencia 3 es un carretera de 2 carriles, donde cada carril tiene un ancho diferente, esto como consecuencia de la cercanía del carril izquierdo con una línea de tren, además, la línea izquierda, en la mayoría de imágenes, no se observa por estar cubierta de hierva. La secuencia 4 por otro lado, es una carretera de 2 carriles interurbana, con vegetación en ambos costados, con poco transito de vehículos y muchas sombras proyectadas en la acera.

A continuación se enuncian los pasos necesarios para medir el rendimiento del algoritmo de detección de carreteras propuesto, sobre cada secuencia:

- Primero se extrae las líneas de referencia de cada imagen utilizando un software diseñado para tal fin, fig. 4.31, en donde se puede etiquetar cada línea con su posición y clase correcta, comprobando visualmente el patrón de líneas generado tanto en vista normal de la carretera como en vista de pájaro.
- En segundo término, se divide cada secuencia en 10 grupos de igual tamaño N . Por cada sub-grupo se realiza un experimento donde se analiza cada imagen usando el algoritmo propuesto en la sec. 4.3. Cada línea del conjunto de líneas detectadas $lin_i^k; k = 1, \dots, N \wedge i = 1, \dots, n_P$, se compara con el grupo de líneas marcadas manualmente del paso anterior, que son tomadas como referencia ó ‘Ground True’, $linRef_j^k; k = 1, \dots, N \wedge j = 1, \dots, n_{GT}$, la línea que cumple con los criterios de proximidad y alineación descritos en la ec. (4.12), es asumida como línea detectada correctamente y por tanto, se calcula

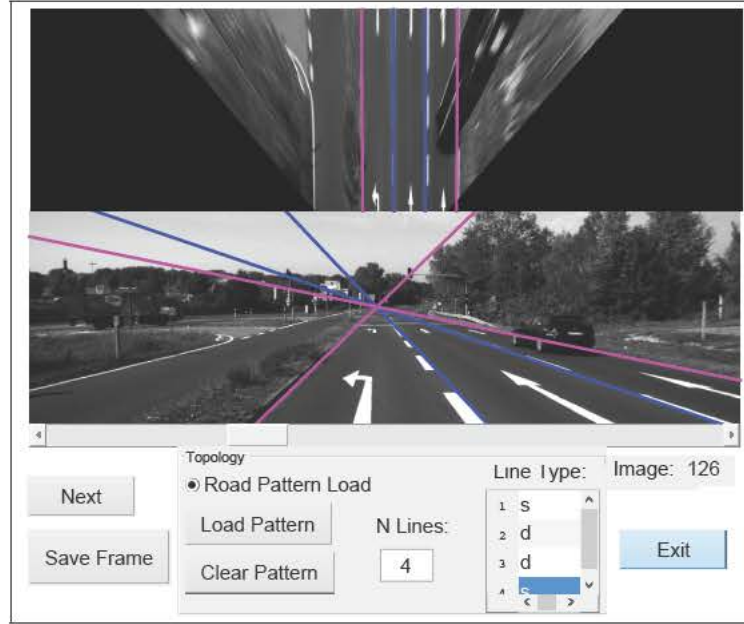


Figura 4.31: Aplicación para etiquetado de líneas de carretera.

la precisión de casamiento a la línea de referencia correspondiente, la definición de la medida de precisión es presentada en la ec. (4.13). El resultado de los experimentos realizados, donde se muestra el número de líneas detectadas correctamente e incorrectamente, y la precisión media de las líneas detectadas correctamente dentro de cada experimento, se muestra en las fig. 4.32, 4.33, 4.34, 4.35 y 4.36 correspondientes a cada secuencia de análisis.

$$\begin{aligned}
 dist &= \max \left(d_{\min}(lin_i^k.m, linRef_j^k), d_{\min}(linRef_j^k.p_m, lin_i^k) \right) < \epsilon_{dist} \\
 delta_{ang} &= \left| \angle lin_i^k - \angle linRef_j^k \right| < \epsilon_{\theta_{alin}}
 \end{aligned} \tag{4.12}$$

$$\begin{aligned}
 precs_{dist} &= \frac{\epsilon_{dist} - dist}{\epsilon_{dist}} \\
 precs_{ang} &= \frac{\epsilon_{\theta_{alin}} - delta_{ang}}{\epsilon_{\theta_{alin}}} \\
 precs &= \frac{precs_{dist}}{2} + \frac{precs_{ang}}{2}
 \end{aligned} \tag{4.13}$$

La medida de precisión referida a la calidad de asociación entre una línea detectada, lin_i^k , y una línea dentro del patrón de referencia, $linRef_j^k$, establece dos criterios que de forma ponderada, forman un único índice de precisión

en el casamiento de líneas clasificadas como reconocidas correctamente. Los criterios tomados en consideración son nuevamente, la alineación y distancia entre líneas. La precisión de cada criterio es el error relativo con respecto a la máxima desviación permitida, diferencia angular máxima entre líneas, $\epsilon_{\theta_{atm}}$ y máxima distancia entre líneas, ϵ_{dist} .

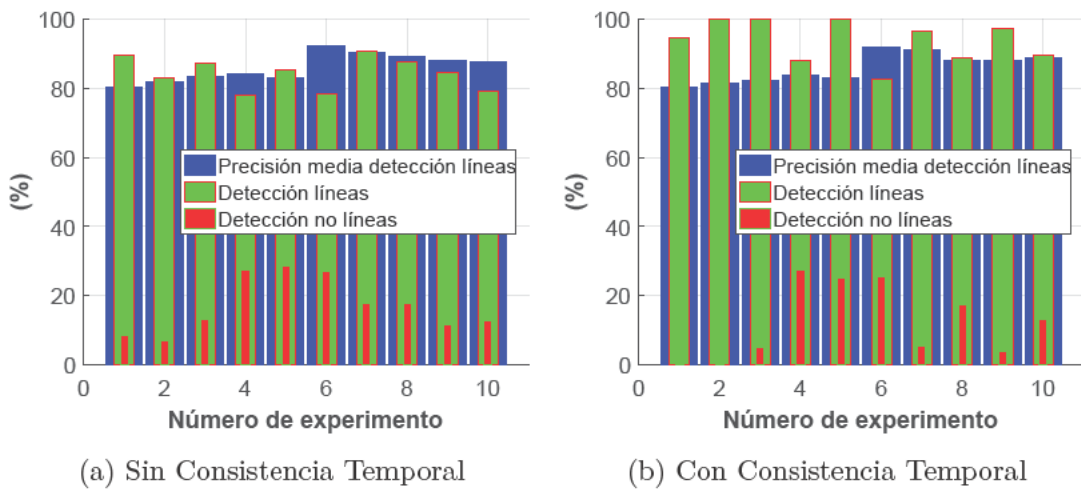


Figura 4.32: Rendimiento del algoritmo Detección de Líneas Viales Secuencia 1.

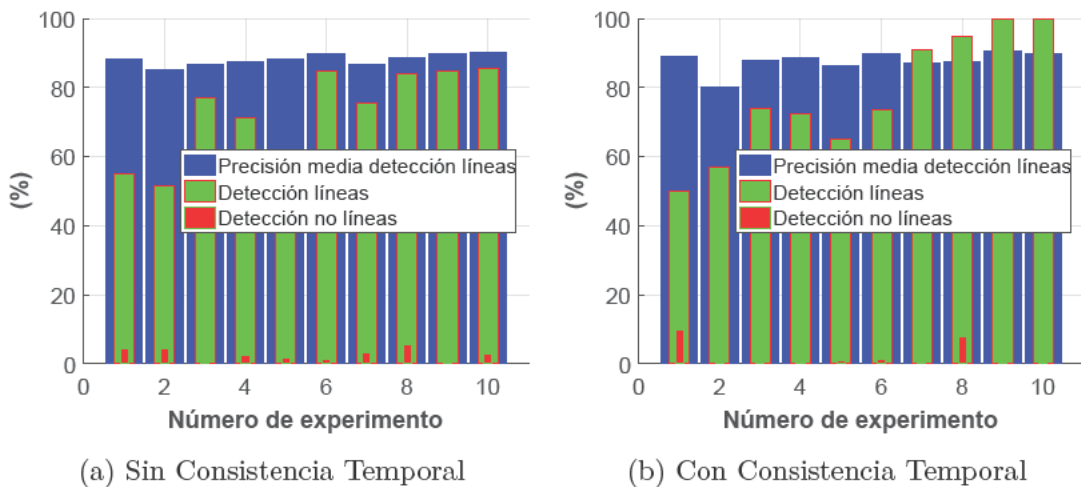


Figura 4.33: Rendimiento del algoritmo Detección de Líneas Viales Secuencia 2.

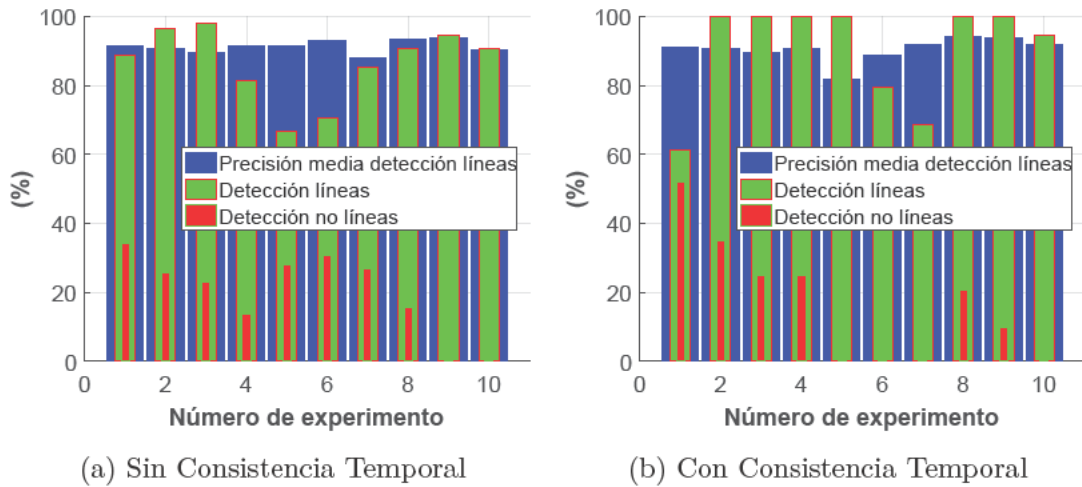


Figura 4.34: Rendimiento del algoritmo Detección de Líneas Viales Secuencia 3 del KITTI Drive_0015.

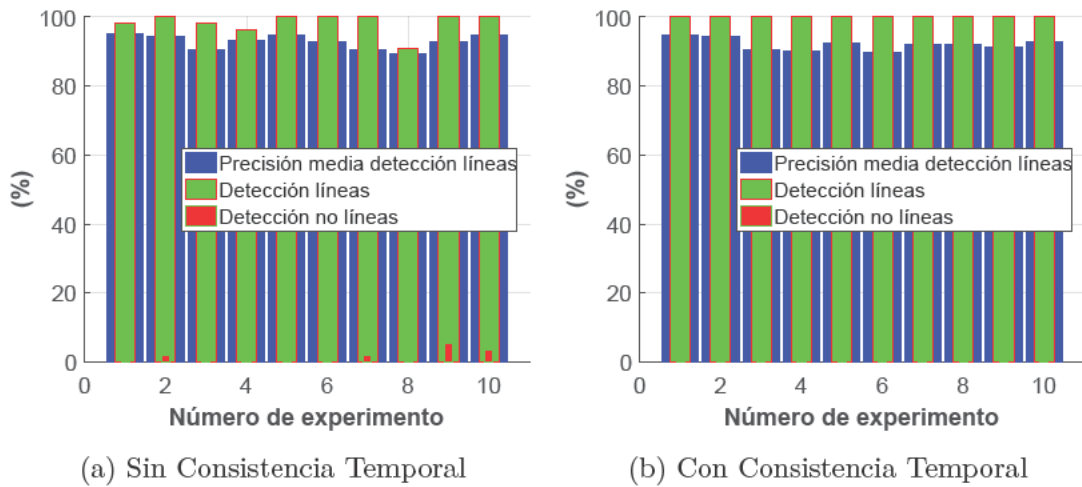


Figura 4.35: Rendimiento del algoritmo Detección de Líneas Viales Secuencia 4 del KITTI Drive_0027.

Los resultados mostrados para las secuencias tomadas en la plataforma IV-VI 2.0 fig. 4.32 y 4.33 comparadas con las secuencias tomadas del repositorio KITTI, fig. 4.35 y 4.34, exhiben grados de dificultad baja y media respectivamente en cada caso, para la detección de carriles, condiciones consistentes con las descripciones respectivas de cada secuencia.

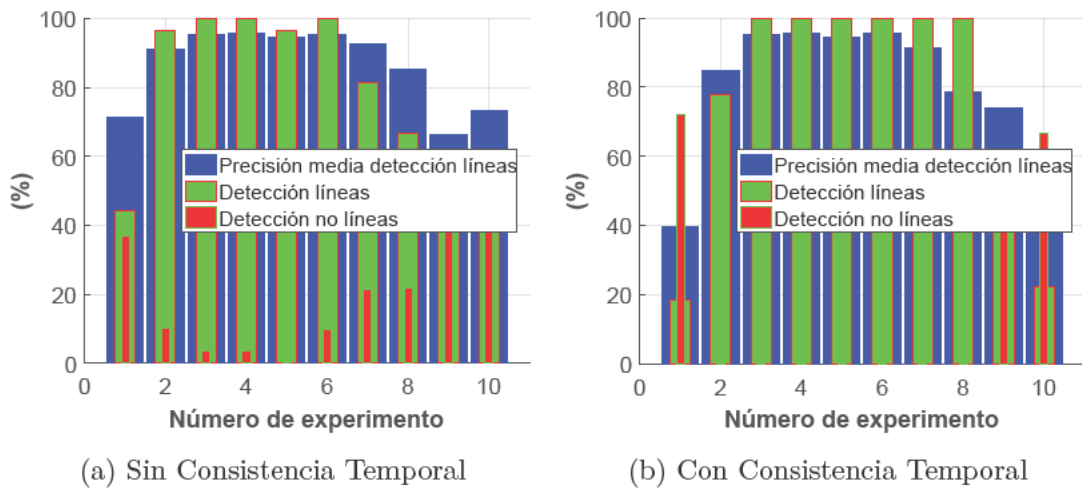


Figura 4.36: Rendimiento del algoritmo Detección de Líneas Viales Secuencia 5. fig. 4.5 (a)

Los resultados del algoritmo de detección de carriles con y sin integración temporal, según se observa en la fig. 4.36 muestran que las características de la secuencia 5 para detección de carriles son especialmente difíciles dado que inicia con la entrada a un túnel donde el mapa de disparidad no se define correctamente, y por tanto el cambio de perspectiva de las imágenes se distorsiona, obteniéndose altos índices de fallo en las detecciones de carril, así mismo, en la parte final de la secuencia se hace el ingreso a una rotonda en cuya transición igualmente se presenta un número apreciable de falsos positivos, situación que es corregida con la aplicación de la componente de integración temporal en el algoritmo.

- Finalmente, se obtiene tres índices globales que describen el rendimiento en la aplicación del método para detección de carreteras sobre cada secuencia: la proporción media de líneas detectadas correctamente, verdaderos positivos; la proporción media de falsas detecciones o detecciones imprecisas, falsos positivos; y por último, la precisión media de las detecciones correctas. Los resultados consolidados se pueden apreciar en el cuadro 4.2.

Sec.	Algoritmo	Media de Parámetros de Rendimiento (%)		
		Precisión	Líneas Mal Detectadas	Líneas Detectadas
1	Sin I. T.	86.08	17.08	84.37
	I. T.	85.93	12.20	93.78
	Variación	-0.15	-4.88	9.41
2	Sin I. T.	88.20	2.64	72.89
	I. T.	87.73	2.01	77.80
	Variación	-0.47	-0.63	4.91
3	Sin I. T.	91.27	19.82	86.3
	I. T.	90.46	16.77	90.37
	Variación	-0.81	-3.1	4.07
4	Sin I. T.	94.00	1.25	98.33
	I. T.	93.31	0.00	100.00
	Variación	-0.69	-1.25	1.67
5	Sin I. T.	86.11	19.13	76.66
	I. T.	80.54	18.67	76.3
	Variación	-5.57	-0.46	-0.36

Cuadro 4.2: Resultados generales del método propuesto para detección de carreteras Sec. 4.3 sin I. T. (Integración temporal) y con I.T Sec. 4.4.

Los valores porcentuales globales obtenidos como índices de calidad para el algoritmo de detección de carreteras se relacionan con las condiciones particulares de cada secuencia. Así por ejemplo, los índices más altos se obtienen sobre la secuencia 1 y 4, secuencias que exhiben líneas bien demarcadas, poco tráfico y carriles simétricos. Los índices más bajos se obtienen para la secuencia 2, 3 y 5. En la secuencia 2 se presentan cambios de carril, flujo vehicular medio, y una alta iluminación. En la secuencia 3, se presenta carriles asimétricos con líneas degradadas o desvanecidas, mientras que en la secuencia 5, como ya se mencionó, existe un ingreso a túnel y a rotonda al inicio y al final de la secuencia. Por otro lado, En la secuencia 1 se aprecia la mayor variación en el índice de detección de carriles, la menor variación en la precisión para detección de líneas y en líneas mal detectadas, es decir, es la secuencia donde el módulo de integración temporal mostró los mejores resultados, consecuencia de la eliminación de falsos positivos provenientes de la existencia de una barandilla en el costado derecho del tramo de carretera y las condiciones ideales

de la secuencia, bajo tráfico, poco movimiento entre carriles, la gran longitud de la secuencia y las condiciones de iluminación estables.

Otras métricas para la medición del rendimiento de algoritmos para detección de carreteras se presentan en [30] y son usadas por el KITTI para la comparación estandarizada de algoritmos sobre secuencias de prueba. Algunas de las métricas utilizadas son precisión, recall, f-measure y accuracy. Estas mediciones no son aplicables para el algoritmo propuesto por cuanto el ground truth proporcionado en el repositorio de secuencias KITTI se orienta a la detección de la zona transitable y no se proporciona información sobre los carriles demarcados por líneas, objetivo distinto al reconocimiento de carriles y carreteras basado en la norma para líneas viales de cada país.

Capítulo 5

Aprendizaje de Máquina para Clasificación de Carriles

El aprendizaje de máquina ó ML, Machine Learning por sus siglas en inglés, es una poderosa herramienta para el reconocimiento ó clasificación de patrones. En palabras de Bishop [9], “El campo de reconocimiento de patrones se refiere al descubrimiento automático de regularidades en la información a través del uso de programas informáticos, y con ellas, se toma acciones tales como clasificar la información en diferentes categorías.” Esta herramienta es especialmente útil en aplicaciones donde los elementos de las categorías tratadas tienen una descripción compleja ó donde el tamaño de las categorías es intratable con recursos computacionales limitados. La visión por computador es un claro ejemplo de este tipo de problemas, la información procesada en este contexto es compleja y su representación, matrices de píxeles multidimensionales, pueden ser de tamaños del orden de millones de datos independientes. Además, las posibles instancias que una categoría puede abarcar, por ejemplo: peatones, automóviles, tipos de calzadas, son incalculables en tamaño, por cuanto el número de elementos en los conjuntos no es cuantificable a priori y aumenta con el tiempo.

Existen muchas aplicaciones en la literatura que resuelven problemáticas relacionadas con ADAS usando aprendizaje de máquina. Por ejemplo, la detección de intención de cambio de carril, en [55] se aborda el problema usando información de velocidad y dirección del vehículo, aplicando un clasificador tipo máquina de soporte vectorial, SVM, multiclase como etapa intermedia hacia un clasificador tipo Bayes, que garantiza mínima detección de falsos positivos. Otras aplicaciones como detección y clasificación de señales de tráfico [99] y detección de concentraciones de agua

sobre la calzada [49], hacen parte de la gama de aplicaciones.

En el presente capítulo se hará uso de las principales técnicas de clasificación por aprendizaje de máquina, incluyendo las estrategias clásicas y modernas, para dar solución a dos etapas que hacen parte de la arquitectura del Sistema Avanzado de Asistencia a la Conducción, ADAS por sus siglas en inglés, presentado en el capítulo 4. Como se observa en la fig. 5.1. Las tareas escogidas son el reconocimiento de píxeles correspondientes a marcas viales y la clasificación de líneas separadoras de carril.

El capítulo se estructura de la siguiente manera: primero se desarrolla cada una de las etapas donde se hace uso del aprendizaje de máquina en el contexto ADAS, para luego realizar un análisis cuantitativo que permite concluir la efectividad de los algoritmos para clasificación de patrones presentados en el apéndice C.

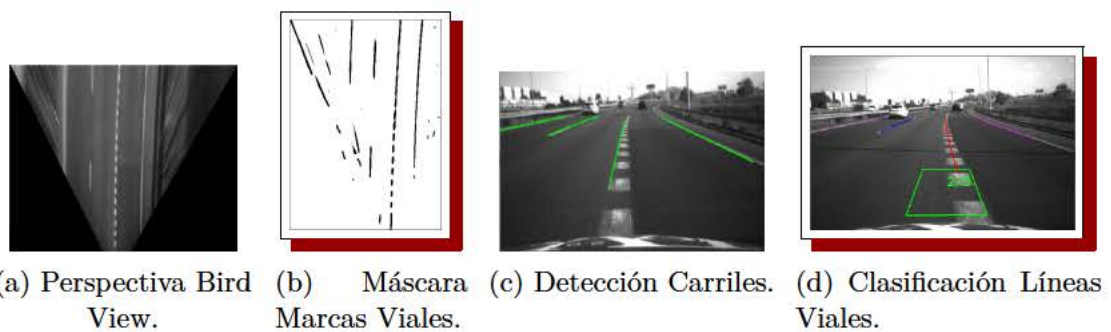


Figura 5.1: Diagrama de bloques general para el sistema avanzado de asistencia a la conducción presentado en el capítulo 4.

5.1. Fundamentos del aprendizaje de máquina

Existen dos aspectos centrales a abordar en la comprensión de las herramientas para clasificar patrones, denominadas algoritmos de aprendizaje de máquina: el primero, es la metodología para interpretar una tarea o necesidad del mundo real como un problema de clasificación, y el segundo, son los algoritmos concretos para implementar el reconocimiento de patrones. A continuación se desarrolla la metodología general para aplicar el aprendizaje de máquina, los tipos de clasificadores más utilizados se describen brevemente en el Anexo C.

La metodología para dar solución a un problema de clasificación haciendo uso del aprendizaje de máquina, puede sintetizarse en las etapas que se muestran en la figura 5.2. Como caso especial, cuando se cuenta con grandes cantidades de ejemplos de

entrenamiento, se puede abordar el problema de clasificación a través de un enfoque tipo Deep Learning [34], donde la primera etapa, descripción de características, es una tarea que se aprende automáticamente del conjunto de entrenamiento.

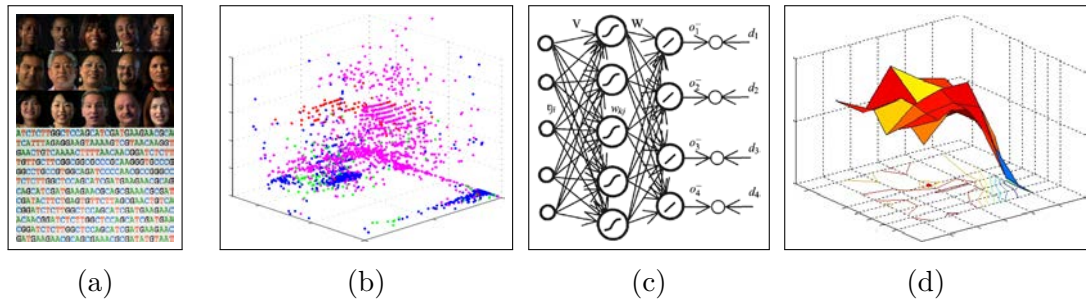


Figura 5.2: Metodología para la implementación de un clasificador por aprendizaje de máquina: (a) Descripción característica. (b) Mapeado en espacio de características. (c) Selección del clasificador. (d) Ajuste parámetros del clasificador.

La primera etapa de la metodología consiste en describir los objetos que se clasificarán en distintas categorías, esta descripción asocia, un vector de características denominado descriptor, a cada posible elemento a clasificar. El significado que se da a cada elemento del vector de características, depende del problema en particular, así por ejemplo, una descripción para la representación de los seres humanos, podría ser su código genético, ya que es único para cada individuo y se puede representar numéricamente.

La segunda fase se refiere a la representación de los objetos en el espacio de características. Un aspecto importante en la definición de los códigos que definen a los objetos en el contexto del aprendizaje de máquina, es la necesidad de encontrar el espacio donde la representación de los objetos exhiba máxima dispersión a lo largo de los ejes directores del espacio. Para ello, existen transformaciones como el análisis de componente principal, PCA: Principal Component Analysis, por sus siglas en inglés, ó el discriminante de Fisher, ambas herramientas matemáticas convierten el vector de características original en uno de menor tamaño y con máxima dispersión en la representación de los objetos.

Una definición formal de PCA, se encuentra en [45], “La idea central del análisis de componente principal (PCA) es reducir la dimensionalidad del conjunto de datos conformado por un considerable número de variables interrelacionadas, conservando tanto como sea posible la variación presente en el conjunto de datos. Esta representación se logra transformando los datos a un nuevo conjunto de variables, las componentes principales, las cuales no están correlacionadas, y las cuales se orde-

nan de tal forma para que las primeras conserven la mayor variación presente en todas las variables originales.”

Posteriormente, en la etapa tres de la metodología, es necesario seleccionar una estrategia de clasificación adecuada y ajustada a los datos disponibles, esto es: si se cuenta con conjuntos de entrenamiento entrada-salida, es posible implementar un clasificador supervisado, de lo contrario, uno no supervisado; si se requiere clasificar en más de tres categorías, existe la posibilidad de implementar una estrategia multiclase tal como, redes neuronales ó máquinas de soporte vectorial, SVM: Support Vector Machine, por sus siglas en inglés. Adicional mente dentro de la variedad de posibles algoritmos clasificadores se pueden encontrar dos grupos generales diferenciados en cuanto a su aparición histórica en el estado de la técnica, se trata de los clasificadores clásicos versus los algoritmos de clasificación modernos. Los principales representantes del primer grupo son los clasificadores Bayesianos y el clasificador KNN ó k-Nearest Neighbors algorithm, por sus siglas en inglés, mientras que ejemplos representativos de clasificadores modernos son las estrategias multiclase antes mencionadas.

Una vez se ha elegido un clasificador adecuado, es necesario realizar su entrenamiento para poder agrupar objetos dentro de un conjunto de categorías preestablecidas. El uso de un clasificador, por tanto, consta de dos fases: entrenamiento y predicción. Para la primera fase, si se trata de un clasificador supervisado, se requiere de un conjunto de entrenamiento de pares objeto-categoría, la estructura interna del clasificador es ajustada progresivamente con cada elemento del conjunto de entrenamiento, se trata de minimizar el error existente entre la categoría generada por el clasificador para el objeto en entrenamiento, y la categoría real, asociada a dicho objeto. El funcionamiento por etapas, del clasificador seleccionado, se observa en la fig. 5.3. La fase de predicción, permite utilizar el conocimiento acumulado en el clasificador, para generalizar una respuesta, es decir, una categoría coherente con la naturaleza de los objetos que se introduzcan como entrada y que no hayan sido entrenados previamente. Para el caso de clasificar seres humanos según su nacionalidad, se trata de presentar un nuevo individuo, el cual será clasificado según la representación interna que el clasificador, en el proceso de entrenamiento, haya construido.

Finalmente, en la última fase de la metodología, de acuerdo al tipo de clasificador escogido, se realiza el ajuste de los parámetros que configuran su funcionamiento, así, si se trata de una red neuronal, se debe seleccionar la mínima cantidad de capas

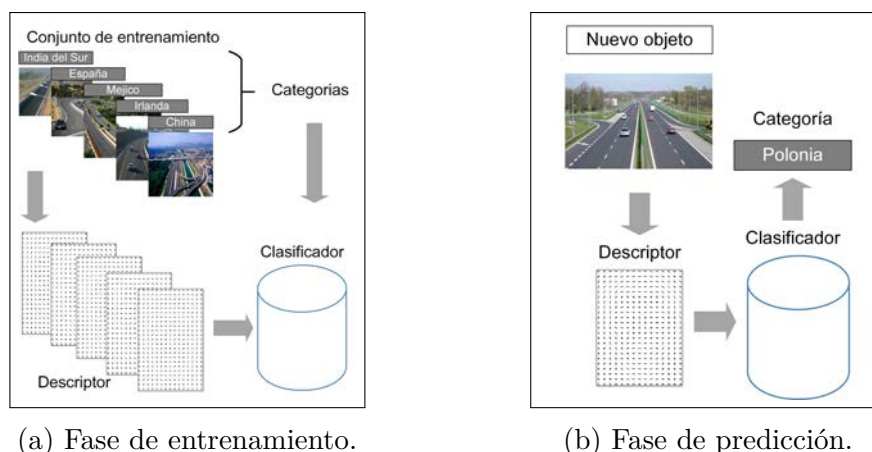


Figura 5.3: Funcionamiento del clasificador.

y de neuronas por capa, que resuelvan el problema con un índice de rendimiento superior a un umbral de calidad escogido. Para el caso de un SVM, el parámetro de regularización que previene el sobre-ajuste al conjunto de entrenamiento y que limita la capacidad de generalización del clasificador, debe ser ajustado minuciosamente, así como la definición de un kernel ó núcleo adecuado.

5.2. Generación máscara de marcas viales

Una de las etapas centrales sobre la cual se construye la mayoría de los sistemas avanzados de asistencia a la conducción, es la generación de la máscara de marcas viales. El reconocimiento de las marcas viales aporta toda la información esencial sobre el contexto de la carretera, por ejemplo: las líneas separadoras de carril, los avisos de parada, las indicaciones de giro, los límites de velocidad, entre otros. Es por ello que la aplicación de técnicas sistemáticas y robustas como el aprendizaje de máquina en la segmentación de marcas viales, permite abordar escenarios diversos en textura de asfalto, configuraciones de sombra, iluminación y visibilidad, por cuanto los conjuntos de entrenamiento se pueden extender incorporando nuevas condiciones de forma ilimitada.

Un aspecto importante que no ha sido abordado en muchos trabajos de reconocimiento de carreteras, usando visión por computador, es la detección de marcas viales cuando las líneas pintadas en la calzada no se encuentran bien definidas, y por tanto, se observan degradadas e imperceptibles a inspección manual, confundándose con la textura del asfalto. Los experimentos que se presentan en la presente sección,

abordan tanto carreteras con líneas bien definidas, secuencia de referencia 1, fig. 4.5(a), así como, carreteras con líneas degradadas, secuencia de referencia 2, fig. 5.12.

5.2.1. Descriptor de marcas viales

Para la descripción de marcas viales, las imágenes capturadas se analizan por bloques de tamaño $N \times M$. Cada bloque en su forma base, escala de grises, es tratado como un vector de $N \times M$ elementos ordenado de forma lexicográfica, todas las filas ordenadas, una a continuación de otra, en un único vector fila.

Con el ánimo de comparar y buscar la máxima eficiencia computacional, se utilizará dos descriptores distintos: ROI, (Region Of Interest) por sus siglas en inglés, en escala de grises, e histograma orientado de gradiente asociado, es decir, la respectiva representación HOG[21] del bloque. Ver fig. 5.4.

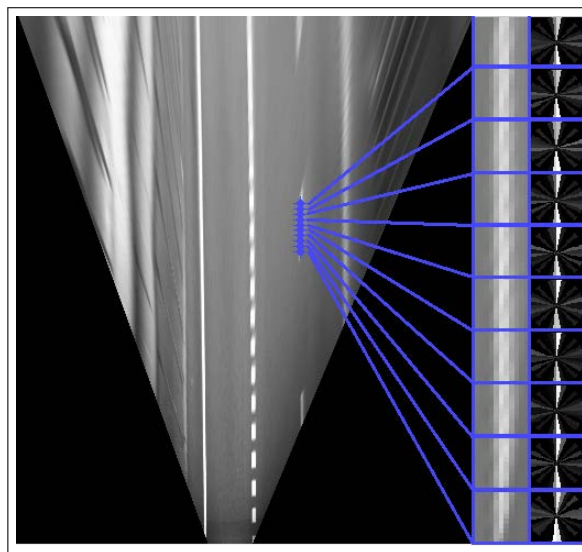


Figura 5.4: Descriptor de marcas viales. ROIs en escala de grises y su correspondiente representación HOG.

5.2.2. Mapeado en espacio de características

Para el análisis de separabilidad entre categorías, en este caso, marca y no marca vial, se utiliza la secuencia de referencia 1 para marcas viales mostrada en la fig. 4.5.

La representación de los descriptores correspondientes al conjunto de entrenamiento extraído de la secuencia de referencia 1, se muestra visualmente utilizando

el espacio PCA, en particular las dos primeras componentes del espacio, si bien la información proyectada disminuye de dimensionalidad, estas dos componentes reflejan la máxima dispersión de los descriptores y por tanto permiten observar su nivel de intersección. El objetivo es comparar el efecto en la separación de los datos a clasificar, para distintos filtros aplicados a la imagen: filtro gaussiano junto con filtro de mediana, filtro de difusión anisotrópica e imagen en forma base, y para los dos descriptores propuestos, ROI en escala de grises y representación HOG del ROI. El conjunto de entrenamiento por clase, representado en las dos primeras componentes PCA para todas las variaciones mencionadas se presentan en la fig. 5.5.

El índice de solapamiento entre las clases correspondientes a marca vial y a no marca vial, se mide asignado a cada descriptor proyectado en el plano PCA(1)-PCA(2), una etiqueta de solapamiento si cumple con la condición de encontrarse en un vecindario circular de radio ϵ alrededor suyo, con al menos un punto descriptor de clase diferente, es decir, si el descriptor bajo prueba es de clase marca vial, se comprueba la presencia en el vecindario de descriptores tipo no marca vial, en caso de tratarse de un descriptor tipo no marca vial se hace la comprobación opuesta. Una vez se han marcado los puntos solapados, se establece el índice de solapamiento como el número de puntos solapados sobre el total de puntos.

De los resultados del experimento mostrado en la fig. 5.5, se puede encontrar una relación directa entre el grado de solapamiento de los datos y la efectividad esperada para el clasificador con la configuración del descriptor relacionada en el análisis comparativo. Así, para la gráfica de la fila 1 columna 1 correspondiente al descriptor en forma base, escala de grises, vectores de 11 por 7 píxeles organizados lexicográficamente, y para la gráfica de la fila 2 columna 1 correspondiente al descriptor con filtrado anisotrópico, escala de grises, vectores de 11 por 7 píxeles organizados lexicográficamente, el solapamiento entre clases se hace mínimo con valores de 39% y 38% respectivamente, por lo tanto, dado que ambos descriptores muestran un nivel de solapamiento similar, se escoge el descriptor con la configuración de menor coste computacional, es decir, un ROI de 11 por 7 píxeles organizados lexicográficamente, tomado de la imagen en escala de grises sin filtros aplicados, para el entrenamiento del clasificador que se abordará a continuación.

5.2.3. Detección de marcas viales basada en clasificación

Para la segmentación de marcas viales en una escena tipo carretera inter-urbana, primero, se compara el uso de distintos clasificadores clásicos junto con la estrategia

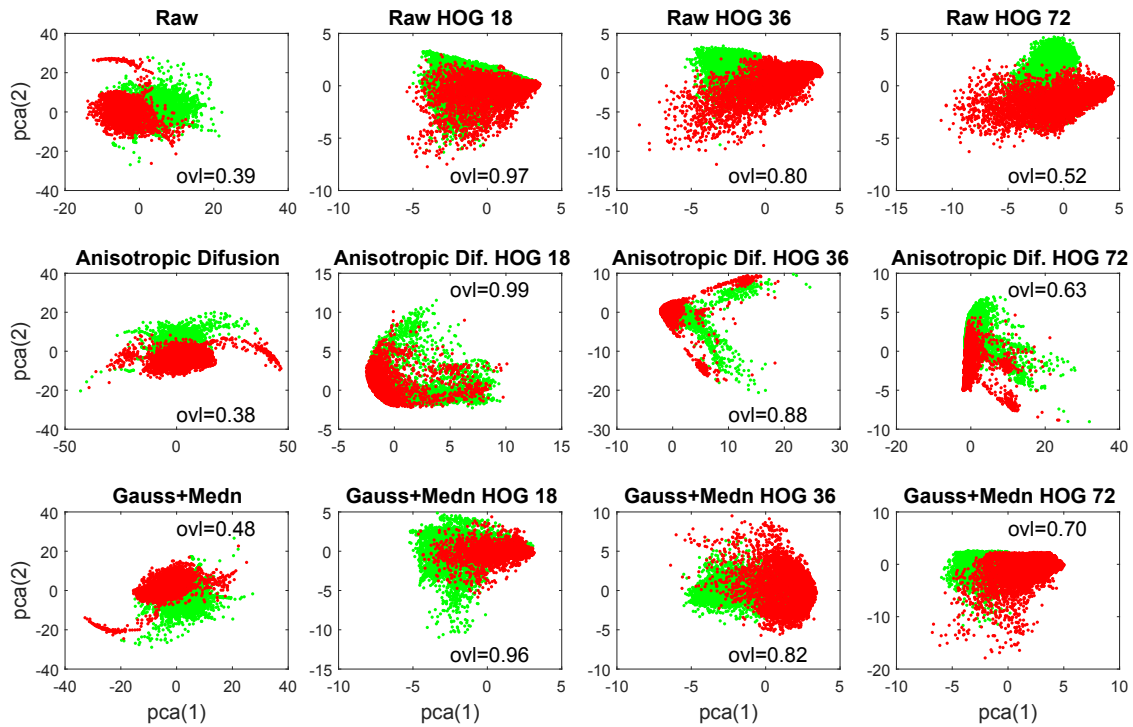


Figura 5.5: Experimento de comparación entre las representaciones del conjunto de entrenamiento para distintos filtros aplicados a la imagen de entrada y para los descriptores propuestos: ROI escala de grises y HOG con diferentes resoluciones, 18, 36 y 72 ángulos.

SVM en la secuencia de referencia 1. fig. 4.5. Para ello, se realiza el entrenamiento de las tres estructuras correspondientes a los clasificadores: KNN, Bayes y SVM, con el conjunto de entrenamiento extraído de la secuencia 1 para marcas viales. Luego, se obtienen las máscaras de marcas viales aplicando cada clasificador sobre cada una de las imágenes de la secuencia, posteriormente, se comparan con su correspondiente “Ground Truth” a través de los índices: sensibilidad, ec. (4.2), y precisión, ec. (4.3). Finalmente, los resultados se muestran en las fig. 5.6 y 5.7.

Los anteriores resultados muestran como la segmentación de marcas viales basada en máquinas de soporte vectorial, SVM, exhibe el mayor índice de sensibilidad con un 83 %, mientras que para el índice de precisión, su rendimiento es intermedio con respecto a los demás clasificadores clásicos, con un valor medio de 36 %.

5.2.3.1. Sintonización de parámetros para clasificación de marcas viales

El algoritmo escogido es la estrategia SVM, dada su estructura y rendimiento óptimo frente a otros tipos de clasificadores. Así mismo, el descriptor seleccionado

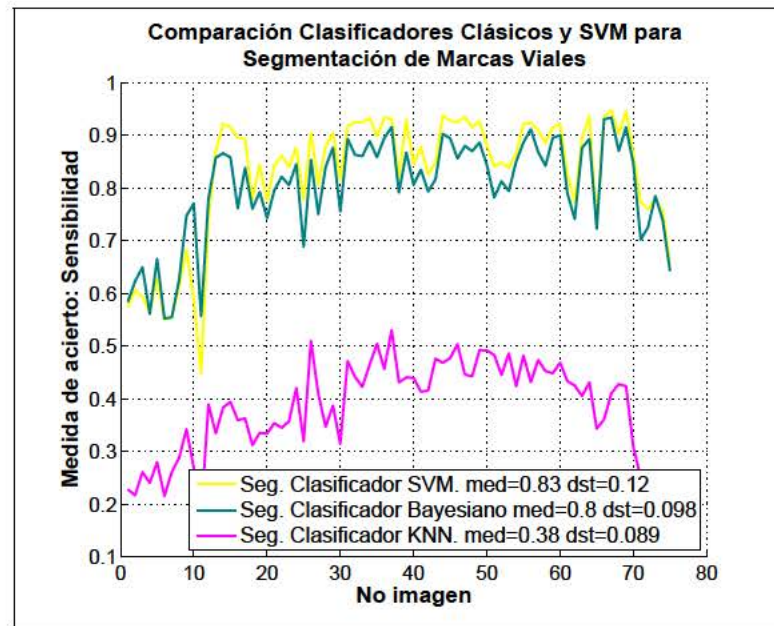


Figura 5.6: Índice de Sensibilidad medido sobre secuencia de referencia 1 para clasificadores clásicos y SVM.

según el solapamiento inter-clase mostrado en la sección anterior, sugirió como mejor alternativa para representar marcas viales, la utilización de regiones de interés de 11 píxeles columna por 7 píxeles fila en escala de grises. Una vez seleccionado el tipo de clasificador, a continuación es necesario seleccionar el valor adecuado de los parámetros que configuran dicho clasificador: el radio del kernel tipo campana gaussiana, y el valor del parámetro de regularización que controla el ajuste de los datos de entrenamiento C . La selección de valores óptimos se realiza a través de un experimento de variación paramétrica donde se prueban los valores para γ desde $1e0$ hasta $1e-5$ y los valores para C desde $1e0$ hasta $1e6$. En este experimento se hará uso de un conjunto de entrenamiento de 6000 muestras tipo “marca vial” y 6000 muestras tipo “no marca vial”, extraídas de la secuencia de referencia 1. Con cada posible combinación de parámetros γ y C se entrenará un clasificador SVM. En cada aprendizaje, se hará uso del 80 % de los datos como ejemplos de entrenamiento, mientras que el 20 % restante será utilizado para validación cruzada.

Es importante mencionar que para poder obtener buenos resultados se han normalizado los datos de entrenamiento. Cada descriptor es ajustado a magnitud unitaria, con el ánimo de lograr invarianza a los cambios de intensidad en grises ocasionados por variaciones en la iluminación de la escena. También se aplica un segundo tipo de normalización para ajustar uniformemente el rango de cada característica

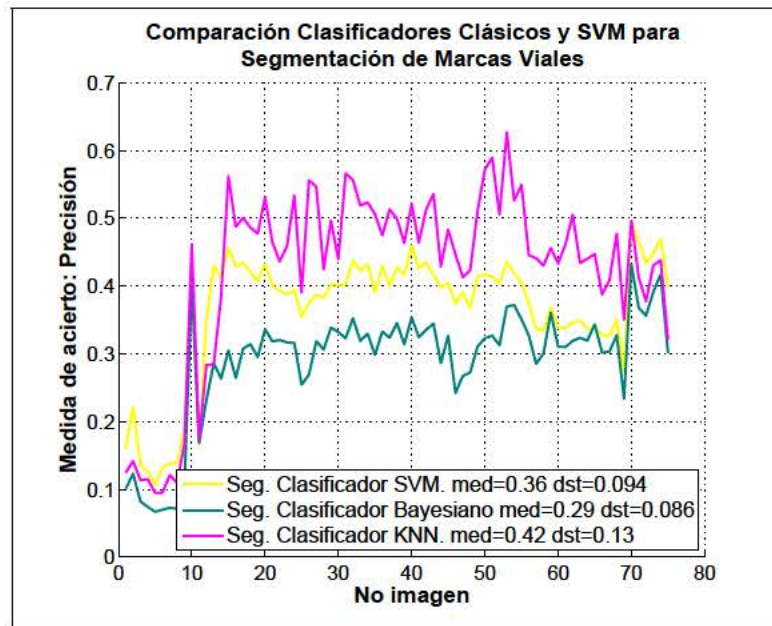


Figura 5.7: Índice de Precisión medido sobre secuencia de referencia 1 para clasificadores clásicos y SVM.

del descriptor a media cero y desviación típica uno.

El resultado del experimento se muestra en la fig. 5.8. Allí se puede ubicar el valor óptimo de $\text{gama} = 1e-2$ y $C = 1e2$, para un índice de rendimiento medido en la exactitud global correspondiente a 97.8%. La definición de exactitud se encuentra en la ec. (5.1)

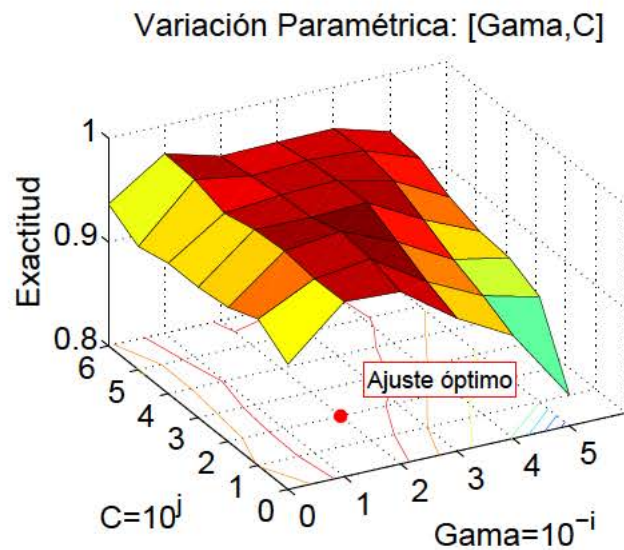


Figura 5.8: Experimento de variación de los parámetros: Gama (radio de kernel) y C (parámetro de regularización).

$$Exactitud_{SVM} = \frac{\text{Traza}(CM)}{\sum (CM)} \quad (5.1)$$

5.2.3.2. SVM para Clasificación de marcas viales bien definidas

En esta etapa se comprueba el rendimiento del clasificador SVM en relación con las propuestas basadas en el espacio del gradiente de la sección 4.1.2, con el mismo procedimiento que se desarrollo para la comparación con clasificadores clásicos. Los resultados se muestran en la fig. 5.9 y fig. 5.10.

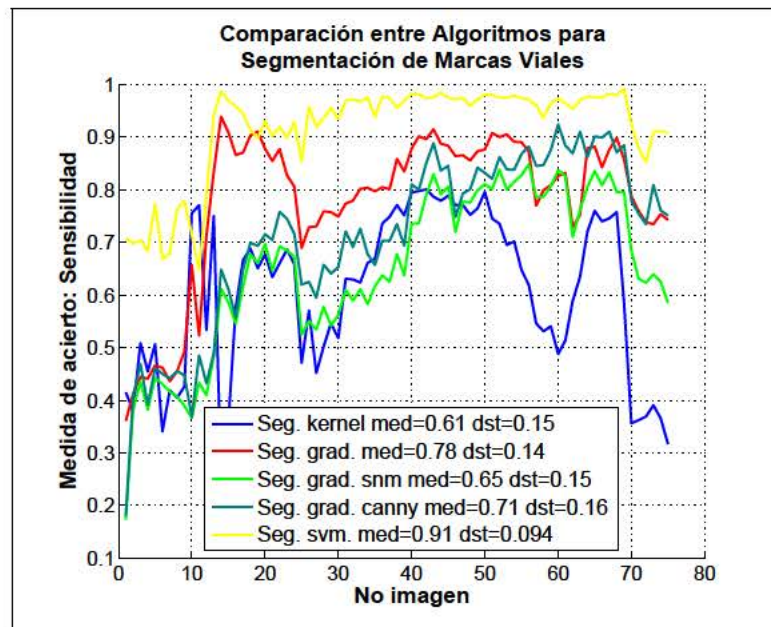


Figura 5.9: Comparación de la sensibilidad entre algoritmos para segmentación de marcas viales vrs SVM.

Las gráficas comparativas, muestran como el método SVM mejora considerablemente la detección de verdaderos positivos con un 91 % de efectividad, un 13 % de mejora en el índice de sensibilidad sobre los demás métodos, sin una disminución significativa en la precisión del clasificador, 31 %, 10 puntos porcentuales por debajo del método mas preciso y 3 puntos porcentuales por encima del método menos preciso.

Por otra parte, es posible analizar los resultados obtenidos desde el punto de vista cualitativo, comparando visualmente las máscaras asociadas con cada algoritmo aplicado sobre la imagen de la fig.4.6. Ver fig.5.11.

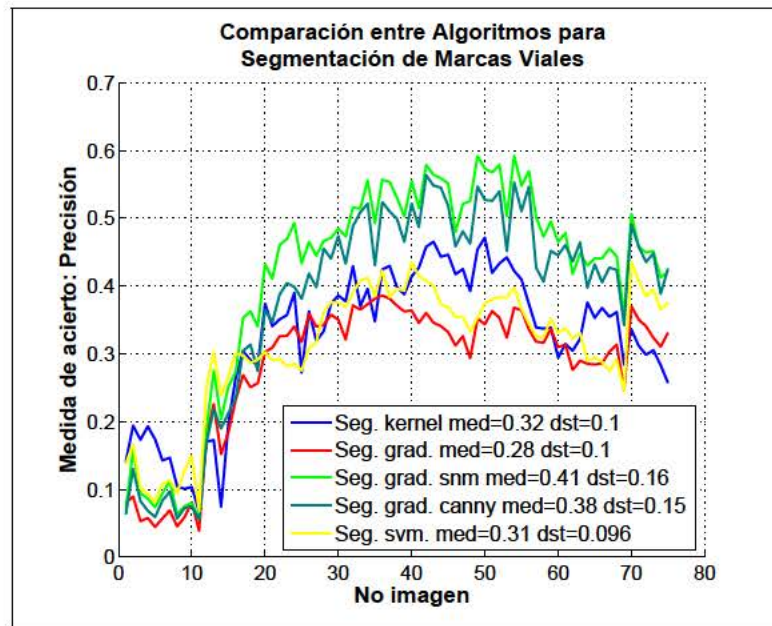


Figura 5.10: Comparación de la precisión entre algoritmos para segmentación de marcas viales vrs SVM.

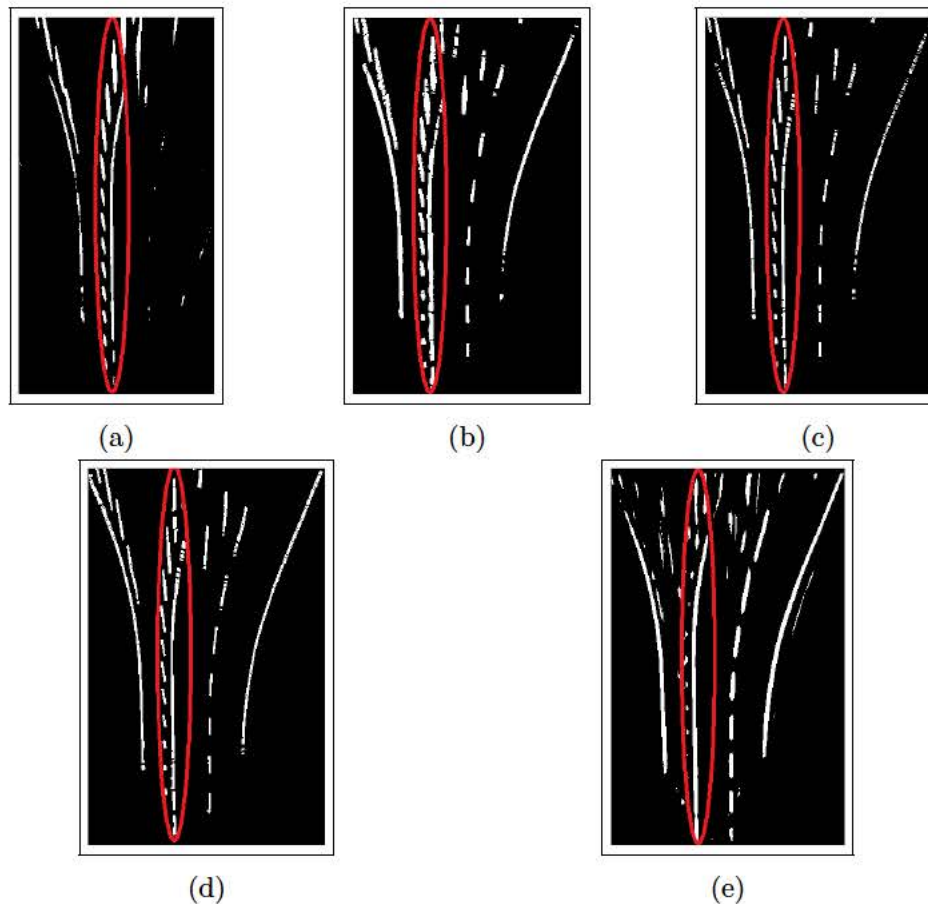


Figura 5.11: Comparación cualitativa entre algoritmos para segmentación de marcas viales presentados en la Sec. 4.1.2 (a) Seg. Kernel, (b) Seg. Grad, (c) Seg. Grad. SNM y (d) Seg. Grad. Canny. vrs (e) SVM.

Solo en la máscara mostrada en la fig. 5.11 cuadro (e), correspondiente al método basado en aprendizaje de máquina, se puede apreciar dos aspectos destacables del conjunto: el primero, relacionado con el alto índice de sensibilidad, es la buena definición de las marcas viales coincidentes con el ground truth, el segundo aspecto es el desvanecimiento de las líneas diagonales cercanas al límite izquierdo de la calzada, enmarcadas en las elipses de color rojo de cada sub-imagen, las cuales no fueron identificadas como marca vial válida en los ejemplos de entrenamiento. Esta diferencia muestra cómo un clasificador tipo SVM permite filtrar de manera selectiva, patrones que cumplen no solo con la descripción de marca vial a nivel de gradiente y dimensión, sino también en relación con su contexto.

5.2.3.3. SVM para Clasificación de marcas viales degradadas

Por otro lado, una de las ventajas que se observó en el uso del algoritmo SVM para detección de marcas viales, fue su flexibilidad para reconocer patrones de línea en imágenes donde la demarcación de la calzada es difusa o está degradada por el paso del tiempo. Esta conclusión es el resultado de un experimento realizado sobre una secuencia de referencia 2 con marcas viales degradadas formada por 75 imágenes, fig 5.12, donde difícilmente un observador humano consigue reconocer líneas sobre la calzada.



Figura 5.12: Secuencia de referencia 2 con líneas degradadas ó poco visibles para la detección de marcas viales.

De la secuencia 2 de referencia se extrajo un conjunto de entrenamiento de 1000 muestras, 5000 tipo “marca vial degradada” y 5000 tipo “no marca vial”, para repetir

el procedimiento desarrollado con imágenes de líneas claramente demarcadas. Se efectuó un nuevo entrenamiento en este entorno y se aplicaron todos los algoritmos desarrollados para detección de marcas viales comparándolos con la versión SVM. Los resultados cuantitativos se observan en las fig. 5.13 y 5.14.

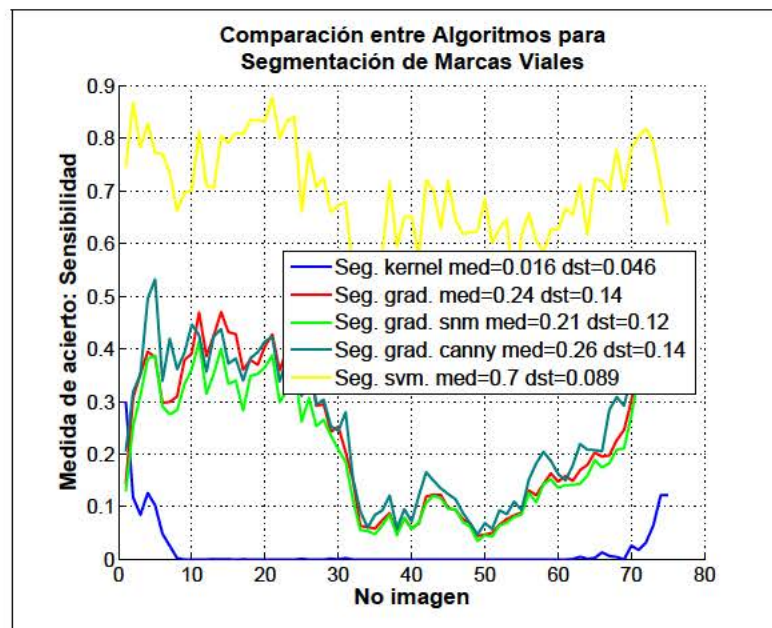


Figura 5.13: Comparación de la sensibilidad entre algoritmos para segmentación de marcas viales degradadas vrs SVM.

Los resultados relacionados con los índices de sensibilidad y precisión tomados de la secuencia de referencia 2, para líneas degradadas, muestra nuevamente un valor medio de sensibilidad más alto en un 44 %, por encima del siguiente método más sensible. En referencia a la precisión, el algoritmo SVM exhibe un valor medio similar al de los demás detectores de marcas viales, 3 % por debajo del algoritmo más preciso.

En la fig. 5.15, de igual manera, se consolidan los resultados cualitativos de la comparación, es decir, las máscaras de marcas viales para todos los métodos utilizados.

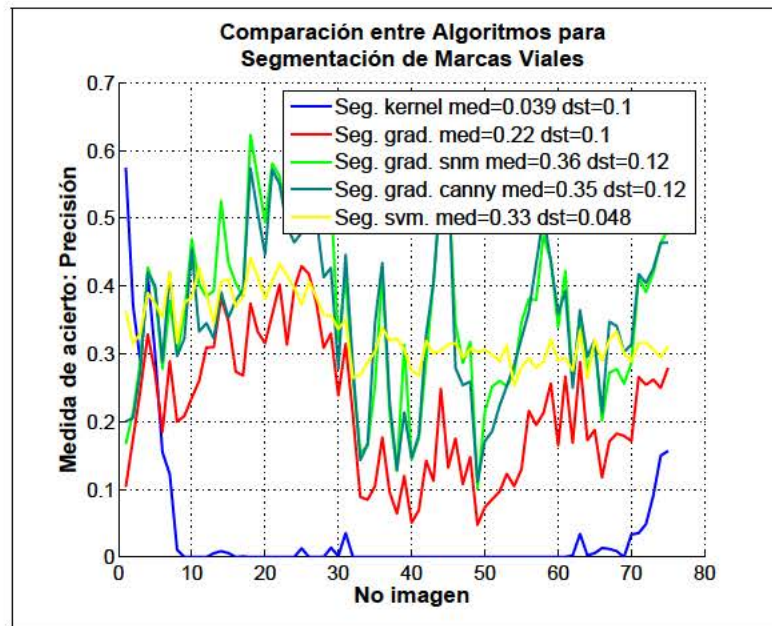


Figura 5.14: Comparación de la precisión entre algoritmos para segmentación de marcas viales degradadas vrs SVM.

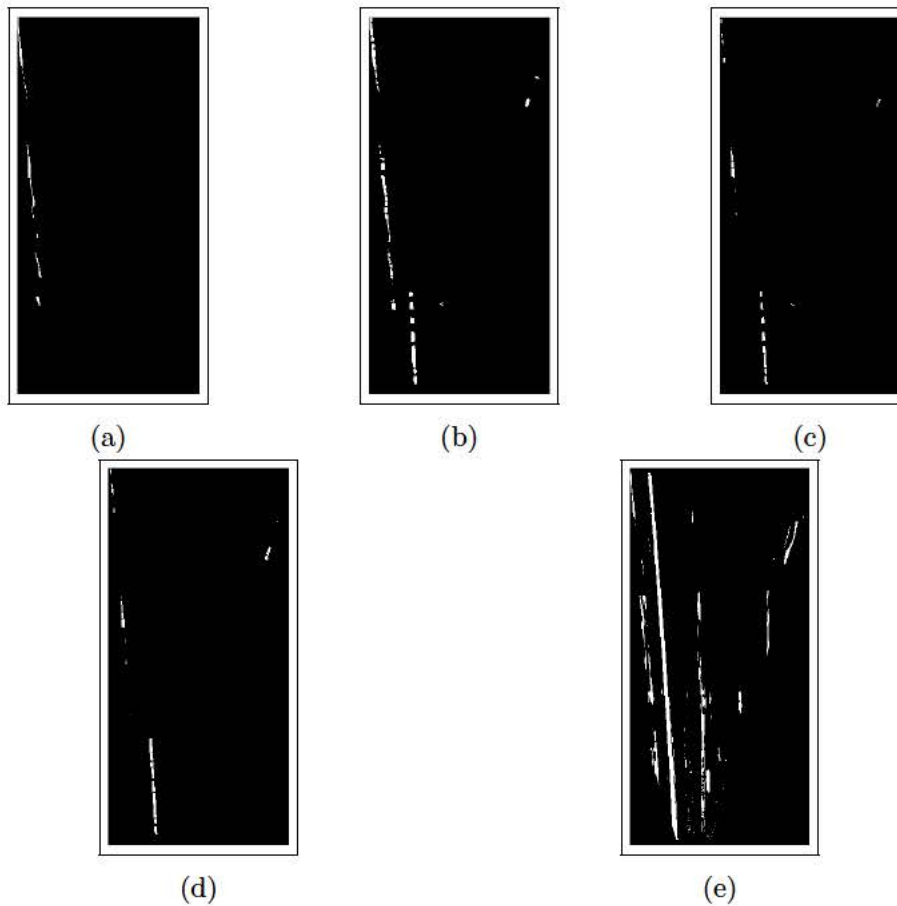


Figura 5.15: Comparación cualitativa entre algoritmos para segmentación de marcas viales presentados en la Sec. 4.1.2 (a) Seg. Kernel, (b) Seg. Grad, (c) Seg. Grad. SNM y (d) Seg. Grad. Canny. vrs (e) SVM, aplicados sobre marcas viales degradadas.

Finalmente, el detalle de la imagen de referencia con líneas degradadas utilizada para el análisis cualitativo se puede observar en la fig. 5.16, junto con su respectiva representación en perspectiva vista de pájaro, la segmentación manual ground truth de las líneas viales, así como la segmentación SVM.

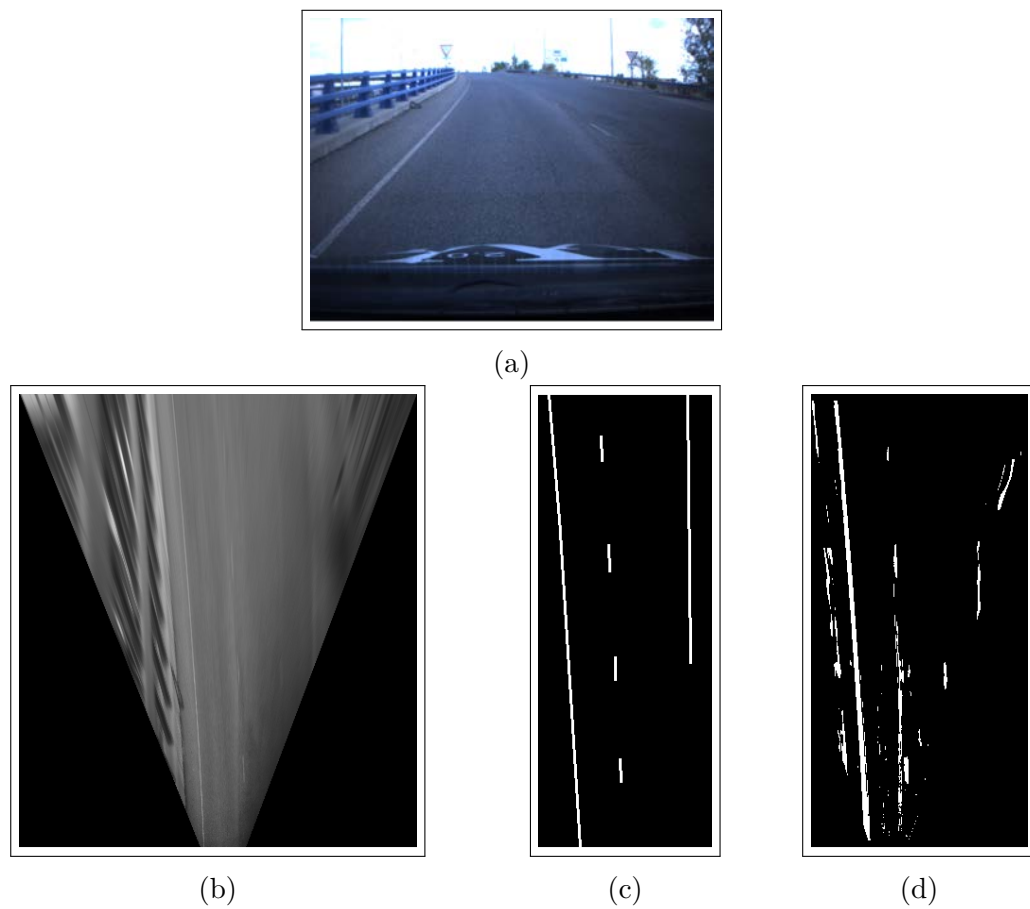


Figura 5.16: Segmentación SVM sobre imagen de referencia para marcas viales degradadas. (a) Vista original. (b) Vista de pájaro. (c) Máscara marcas viales ground truth. (d) Máscara de marcas viales SVM.

De las fig. 5.15 y 5.16, es claro cómo el clasificador SVM es el único capaz de detectar las casi imperceptibles marcas viales que se confunden con la textura del asfalto.

5.2.4. Detección de Carreteras usando Máquinas de soporte vectorial

En esta sección se evaluará el efecto de reemplazar el proceso de detección de marcas viales basado en gradiente por el método propuesto en la sección 5.2.3 basado en clasificación tipo máquinas de soporte vectorial o SVM, dentro del algoritmo general para detección de carreteras presentado en el capítulo 4.

Para la evaluación se aplicará el método descrito en la sección 4.5 sobre la secuencia de referencia 5 fig. 4.5 (a). Los resultados se pueden apreciar en la fig. 5.17.

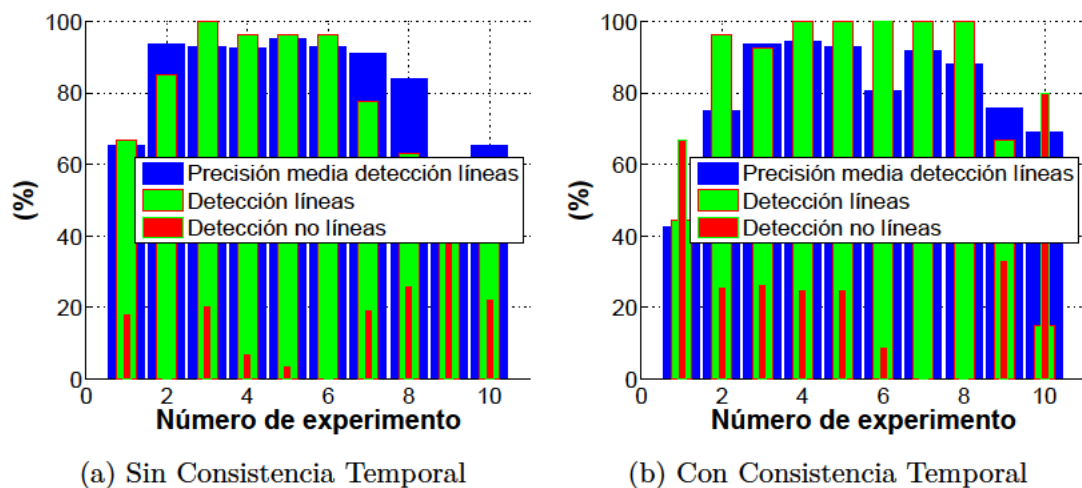


Figura 5.17: Rendimiento del algoritmo Detección de Líneas Viales usando SVM. fig. 4.5 (a).

Finalmente, en la tabla 5.1 se sintetizan los resultados del efecto en el reemplazo de la detección de marcas viales SVM en el algoritmo general para detección de carreteras a través de la proporción media de líneas detectadas correcta e incorrectamente y la precisión media de las detecciones correctas.

Sec.	Algoritmo	Media de Parámetros de Rendimiento (%)		
		Precisión	Líneas Mal Detectadas	Líneas Detectadas
5 GRAD	Sin I. T.	86.11	19.13	76.66
	I. T.	80.54	18.67	76.3
	Variación	-5.57	-0.46	-0.36
5 SVM	Sin I. T.	83.36	15.93	77.4
	I. T.	80.4	29.1	82.96
	Variación	-2.96	13.17	5.56

Cuadro 5.1: Resultados generales del método propuesto para detección de carreteras Sec. 4.3 sin I. T. (Integración temporal) y con I.T Sec. 4.4. Comparando el efecto de usar para la detección de marcas viales el algoritmo basado en gradiente versus el algoritmo basado en clasificación SVM

Como se observa en el cuadro 5.1, la precisión de las líneas detectadas disminuye con integración temporal en ambos algoritmos, con detección de marcas viales en el espacio del gradiente en un 5,6% y usando el método SVM en un 3%. En cuanto al porcentaje de líneas detectadas correctamente, éste disminuye en un 0,4% con el método Gradiente y aumenta en un 5,6% con SVM, variaciones que parten de alrededor de 77%. Finalmente, la cantidad de líneas mal detectadas no mejora con el uso del algoritmo de marcas viales SVM. Usando I.T. en el método Gradiente se tiene un porcentaje de 18,77% mientras que con SVM es de 29,1%, comportamiento que se explica dada la corta longitud de la secuencia, 100 imágenes, 10 por cada experimento, contra la ventana de vida de los carriles en las estructuras que memorizan la topología de la carretera, 12 capturas de imagen para el caso del experimento en mención. Lo que significa que un error capturado en la estructura de carretera tardará en corregirse más allá de cada experimento.

En general, el efecto de utilizar detección de marcas viales a través de clasificación SVM mejora el rendimiento del algoritmo general para detección de carreteras presentado en el capítulo 4, ya que los resultados obtenidos sobre la secuencia de la fig. 4.5 (a), muestran que la precisión de las líneas se mantuvo aproximadamente constante, la detección de líneas aumento, aunque la detección de líneas erróneas aumento también, efecto que puede ser atenuado en secuencias más grandes, dando tiempo al algoritmo de integración temporal para eliminar errores, falsos carriles debidos a distorsiones en la imagen.

5.3. Clasificación de líneas

La clasificación de líneas es una de las etapas más importantes para el entendimiento de la carretera. Esta información permite a un *ADAS* detectar maniobras peligrosas y dar la respectiva alarma, siempre que se conozca adicionalmente el movimiento que describe la cámara. Procesos posteriores, tales como seguimiento de carriles en la carretera u odometría visual, pueden ser mejorados con este conocimiento.

La información de entrada al clasificador es el perfil binario \mathcal{P}_i , extraído de la máscara de marcas viales \mathcal{RM} mostrado en la fig. 5.18.

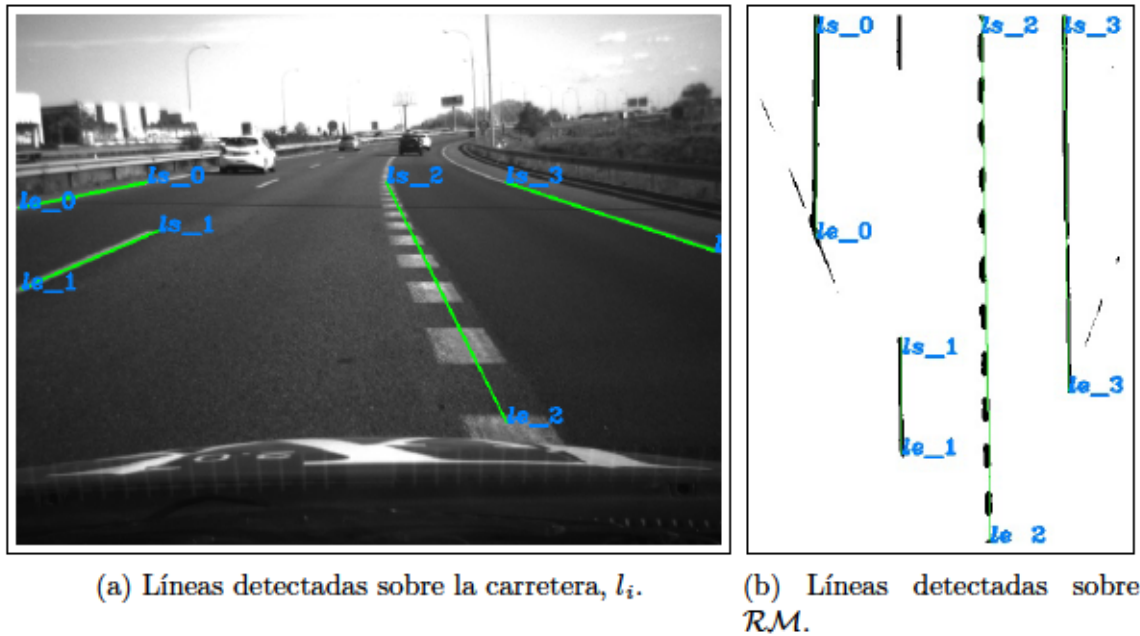


Figura 5.18: Extracción de perfiles de línea vial.

Cada perfil binario es una colección de puntos, resultado de la interpolación lineal entre los extremos de cada línea l_i , que pertenece a la estructura de carretera, \mathcal{R} , reconocida previamente, ec (5.2). Las líneas que definen los perfiles, se extraen de los externos de la cadena de carriles, \mathcal{R} , y de cada límite común a dos carriles de la cadena.

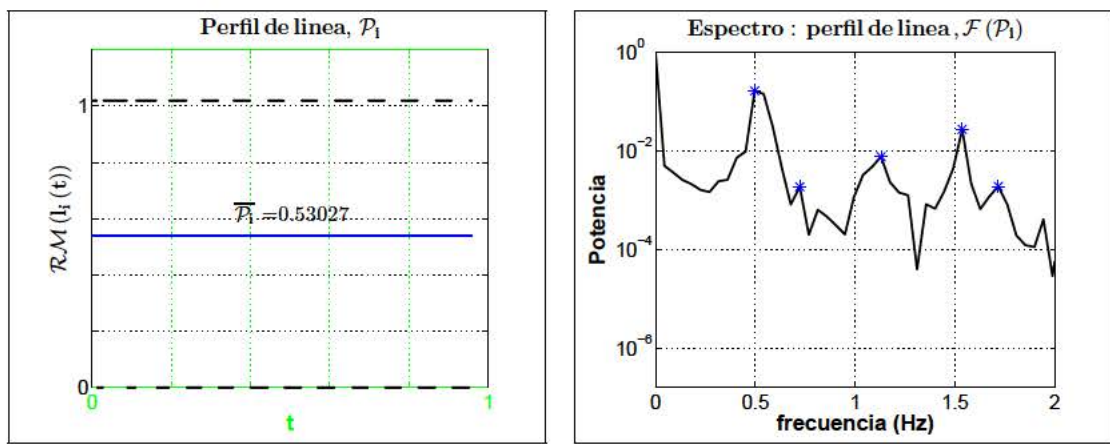
$$\begin{aligned} \mathcal{P}_i(t) &= \mathcal{RM}(l_i(t)) \\ l_i(t) &= (\text{longitud}(l_i) - t) P_{s_i} + t P_{e_i} \end{aligned} \quad (5.2)$$

Donde $i = 1, 2, \dots, \# \text{ líneas} : i \in \mathcal{R}$, $t = 0, \dots, 1 = \text{longitud}(l_i) : t \in \mathcal{R}$, y P_{s_i}

y Pe_i son el punto inicial y final de la línea i -ésima detectada en la estructura de carretera \mathcal{R} .

5.3.1. Descriptor de líneas viales

El descriptor escogido en este caso es una combinación de características provenientes del dominio espacial y de frecuencia, asociados a cada perfil de línea \mathcal{P}_i , ver fig. 5.19.



(a) Perfil de línea.

(b) Espectro de potencia de un perfil de línea.

Figura 5.19: Descriptor de líneas viales.

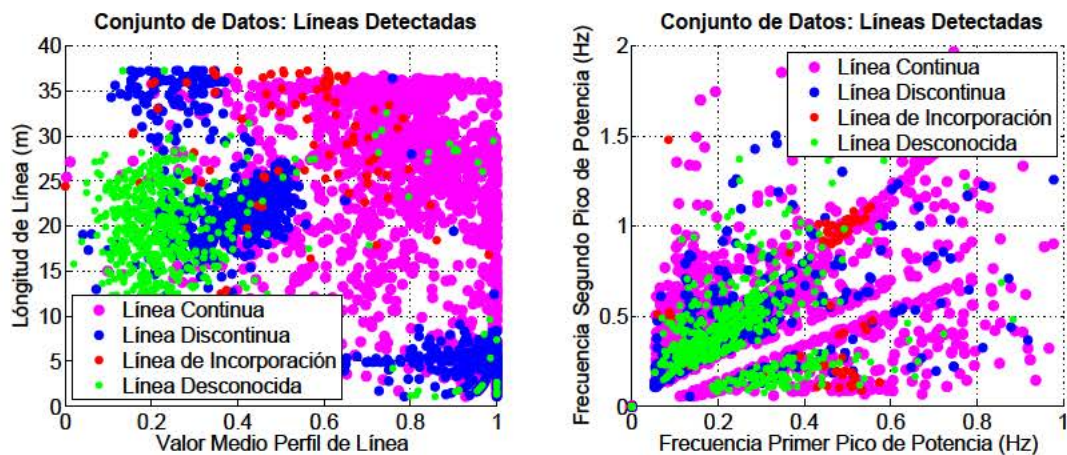
Las dos primeras características son el valor medio, \overline{P}_i , del i -ésimo perfil de línea y su longitud medida en metros, $longitud(\mathcal{P}_i) = longitud(l_i) = dist(Ps_i, Pe_i)$. Las otras características corresponden a las frecuencias donde se encuentran los n primeros picos del espectro de potencia asociado a su correspondiente perfil de línea, en orden descendente, $Pico_j(\mathcal{F}(\mathcal{P}_i))$; $j = 1, 2, 3, \dots, n$, ec. (5.3).

$$Descriptor_{linea} = [\overline{P}_i, longitud(\mathcal{P}_i), Pico_1(\mathcal{F}(\mathcal{P}_i)), Pico_2(\mathcal{F}(\mathcal{P}_i)), \dots, Pico_n(\mathcal{F}(\mathcal{P}_i))] \quad (5.3)$$

5.3.2. Mapeado en espacio de características

Del análisis de la secuencia mostrada en la fig. 4.27, se extrajo una colección de seis mil líneas detectadas y representadas a través de su correspondiente perfil,

las cuales fueron etiquetadas manualmente. La representación del conjunto de líneas detectadas en el espacio de características: valor medio de líneas-Longitud de línea, y frecuencia primer pico de potencia-frecuencia segundo pico de potencia, se puede observar en la fig. 5.20. El color de cada punto indica la clase que fue asociada a cada línea en el proceso de etiquetado manual. Las clases utilizadas para las líneas son: continua, discontinua, de incorporación y no vial.



(a) Características en el dominio del espacio. (b) Características en el dominio de la frecuencia.

Figura 5.20: Representación en espacio de características de un conjunto de líneas, etiquetadas manualmente.

Existe una relación directa entre la ubicación donde los vectores característicos de un cierto tipo de línea detectada se acumula, y la norma IC 8.2 para demarcación de carreteras. Así por ejemplo, el cluster formado por las líneas discontinuas, color azul, en la fig. 5.20a, correspondientes a las características en espacio, se pueden asociar al grupo de marcas longitudinales discontinuas función D1 para velocidad media mayor o igual a 100Km/h, ver tabla B.2. Como se puede observar en la figura existen dos puntos de acumulación en el espacio para líneas discontinuas: el primero en 22m de longitud de línea y un valor medio de 0.5, corresponde a un perfil compuesto por dos tramos continuos de 5m y un espacio entre segmentos de 12m, con lo cual se obtiene una señal de 22m con valor medio de 0.42. El segundo punto de acumulación en 5m y un valor medio de 1, corresponde a un perfil compuesto sólo por un segmento continuo de 5m.

Los puntos correspondientes a líneas continuas se ubican en valores medios altos, mayores a 0.6, como consecuencia del ruido presente en las detecciones, y longitudes grandes superiores a 20m.

En la fig. 5.20b, correspondiente a la descripción sobre el dominio de frecuencia en donde se observan los picos de potencia mas altos asociados a los perfiles de línea, es posible encontrar visualmente un patrón distintivo para el caso de marcas de incorporación. Allí se observa que el cluster en color rojo se ubica alrededor de 0.5 Hz como frecuencia del primer pico más alto de potencia, lo que significa que en 1m, el perfil de línea se repite 0.5 veces, es decir, que su periodo es de 2m. Este tipo de perfil corresponde al grupo de marcas viales longitudinales, función D3, D4 y D5, independiente de la velocidad media, ver tabla B.2, compuesto por segmentos continuos de 1m intercalados con espacios entre segmento de también 1m de longitud.

Una representación gráfica de la proyección del conjunto de datos de entrada sobre las tres primeras componentes del espacio PCA, se encuentra en la fig. 5.21. Allí se puede apreciar como las líneas se representan en conjuntos de puntos que aunque con solapamiento inter-clase, pueden ser separados, ya que se concentran en zonas diferentes del espacio PCA. Sin embargo, en la representación de líneas según su clase asociada, usando características intuitivas y heurísticas, se logra una mejor diferenciación visual de los datos.

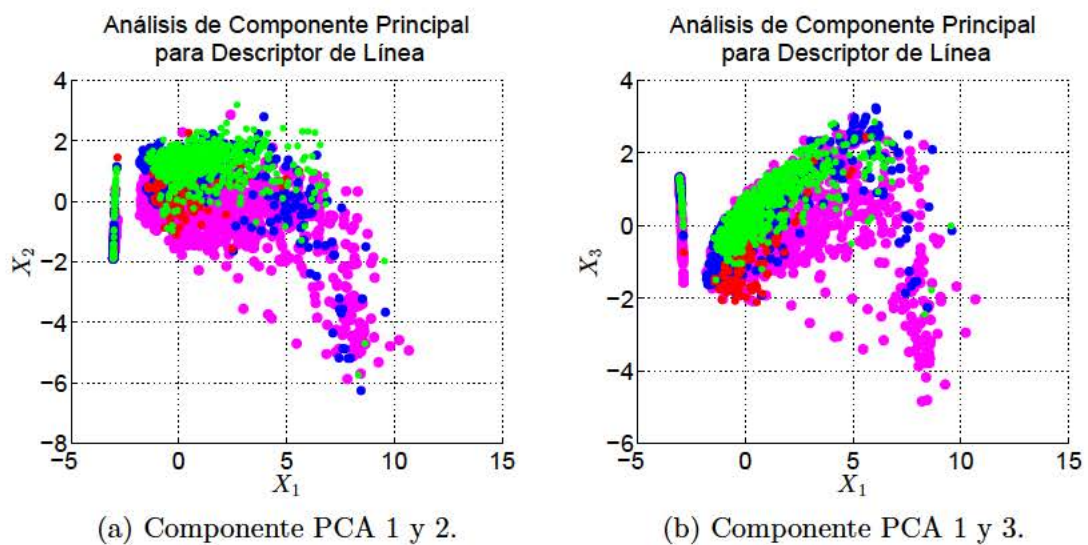


Figura 5.21: Representación PCA del conjunto de líneas extraído de la secuencia mostrada en la fig. 4.27.

5.3.3. Clasificador de líneas viales y sintonización de parámetros

En la fig. 5.22 se observa el comportamiento del clasificador basado en SVM, para un descriptor de tan solo dos características en el dominio espacial, el valor medio del perfil de línea y la longitud de la línea en metros. El resultado muestra una agrupación de las observaciones según su respectiva clase, con lo cual se puede confirmar que el uso de información a priori para la selección de características, permite alcanzar altos rendimientos en la clasificación de los datos.

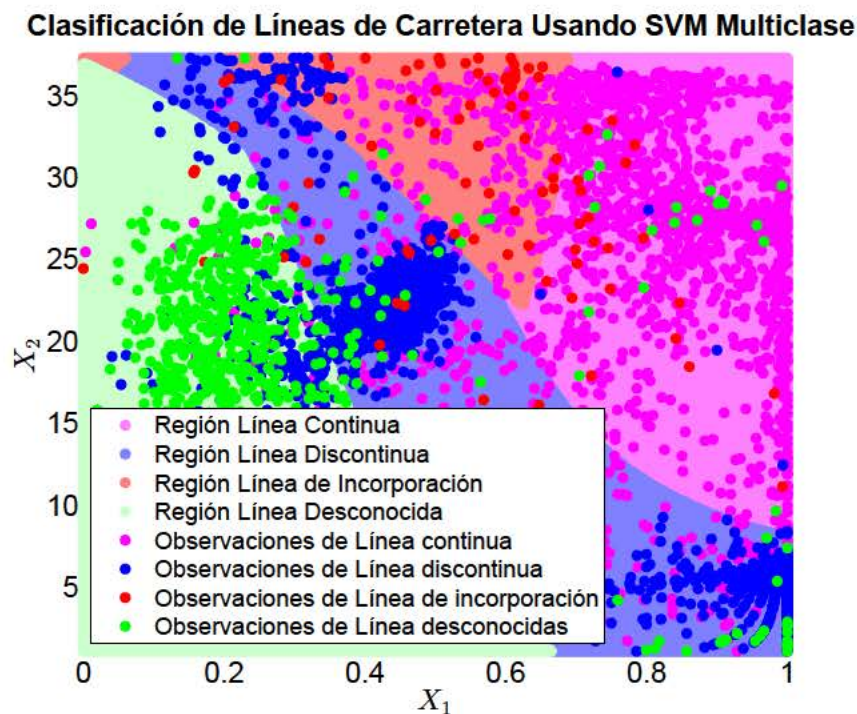


Figura 5.22: Comportamiento del Clasificador SVM para la detección de líneas viales para un descriptor de 2 características: Valor medio de perfil de línea y longitud de línea en metros.

Una vez seleccionado el tipo de clasificador, que para el caso es la topología Máquinas de Soporte Vectorial, SVM, es necesario seleccionar el valor adecuado de los parámetros que configuran dicho clasificador: el radio de acción de la campana gaussiana del kernel y el valor del parámetro de regularización que controla el sobre ajuste de los datos de entrenamiento C . La selección de valores óptimos se realiza a través de un experimento de variación paramétrica donde se prueban los valores para gama de $[1e - 1, 1e - 2, 1e - 3, 1e - 4, 1e - 5]$ y los valores para C

de $[1e2, 1e3, 1e4, 1e5, 1e6, 1e9]$. En este experimento se ha entrenado un SVM con todas las posibles combinaciones de γ y C , utilizando como vector descriptor el valor medio del perfil de línea, la longitud de línea en metros y las cuatro primeras frecuencias donde ocurren los picos mas altos de potencia presentes en el espectro de un perfil de línea. Así mismo, en cada aprendizaje, se ha utilizado el 80% de los datos como ejemplos de entrenamiento, mientras que el 20% restante se ha usado para validación cruzada. Adicionalmente, se han normalizado los datos de entrenamiento por clase, de tal forma que los conjuntos normalizados posean media cero y desviación típica uno.

El resultado del experimento se muestra en la fig. 5.23. Allí se puede ubicar el valor óptimo de $\gamma = 1e-1$ y $C = 1e4$, para un índice de rendimiento medido en la exactitud global correspondiente a 85.6%. La definición de exactitud se encuentra en la ec. (5.1)

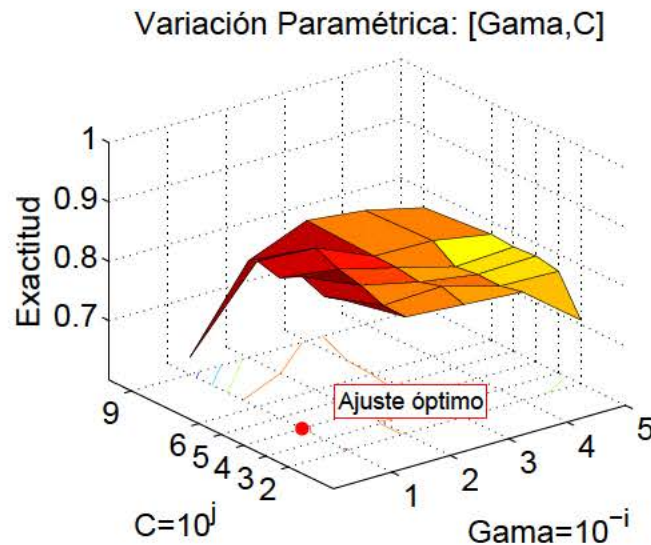


Figura 5.23: Experimento de variación de los parámetros Gama, radio de kernel, y C , parámetro de regularización.

Donde CM es la matriz de confusión asociada a los resultados del entrenamiento SVM para una pareja de valores (γ , C) dados. En particular, la matriz de confusión para los parámetros óptimos arrojados en el experimento de la fig. 5.23, se encuentran en la tabla 5.2.

	L. Contn.	L. Disctn.	L. Incorp.	L. Descd.
L. Contn.	353	47	10	18
L. Disctn.	8	346	2	21
L. Incorp.	6	3	20	0
L. Descd.	11	13	2	120

Cuadro 5.2: Matriz de confusión del entrenamiento SVM para clasificación de líneas de carretera para valores óptimos de ajuste.

Capítulo 6

Conclusiones y Trabajo Futuro

6.1. Reconstrucción Tridimensional de la Carretera

El método de auto-calibración propuesto para la extracción de parámetros extrínsecos presentado en la sección 3.2, mostró resultados satisfactorios, con precisión de décimas de grado para los ángulos y milésimas de metro, para la altura.

El algoritmo que presenta una mejor relación precisión versus tiempo es el estimador RANSAC con muestreo uniforme, tal y como se concluye de los experimentos ilustrados por las fig. 3.8, 3.9 y 3.11.

Se observó un error sistemático en la comparación realizada con la secuencia tomada del KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago) de 0.29 grados para el ángulo de cabeceo, 1.83 grados para el ángulo de alabeo y 14.7mm para la altura de la cámara, sin embargo, el comportamiento de las señales medidas y de referencia esta fuertemente correlado, tal y como se observa en la fig.3.12, en la fig.3.13 y en la fig.3.15.

Por último, el efecto de filtrar los parámetros extrínsecos por los dos métodos mostrados en la sec. 3.2.3.3, se puede apreciar a través del índice DTR0, definido en la ec. 3.36, cuyos valores se reúnen en la tabla 6.1. Los datos muestran cómo la alternativa del filtro de Kalman, en presencia de ruido de media cero, arroja los mejores resultados, logrando aumentar la precisión para la altura a 5.6 mm, para el ángulo de cabeceo a 0.06 grados y para el ángulo de alabeo a 0.1 grados. Adicionalmente, su implementación es adecuada para una aplicación tipo ADAS, dado que es un algoritmo recursivo.

Método filtrado	DTR0		
	Ang. Cabeceo	Ang. Alabeo	Altura
Sin filtro	0.10617	0.13085	0.013781
FIR	0.070039	0.139	0.0063664
Kalman	0.06563	0.10861	0.0056782

Cuadro 6.1: Desviación típica respecto a referencia, eliminando el error sistemático de medida, de los parámetros extrínsecos medidos en la secuencia de referencia, Raw:Road:20112609_0015, tomada del KITTI.

6.2. Modelado y Detección de la Carretera

El capítulo 4 desarrolla un método novedoso para detección de carreteras, aportando ideas originales como la detección del ángulo de la carretera por agrupamiento y la integración en el tiempo de carriles detectados a través de la interacción de estructuras de carretera, ver fig 4.25.

Uno de los pasos fundamentales en el algoritmo para detección de carreteras es el cambio de perspectiva a vista superior o vista de pájaro. El apéndice A.2 presenta la descripción del cambio de perspectiva como una homografía aplicada a una imagen de partida, construida por el producto de transformaciones homogéneas que describen movimientos de cámara en el espacio, y matrices de proyección que reflejan la naturaleza de la conversión espacio a sensor de imagen, utilizando los parámetros extrínsecos de la cámara, calculados en tiempo real en el capítulo 3, con un nivel de precisión aceptable, suficiente para mantener aproximadamente estable el paralelismo de las líneas en vista de pájaro.

El procedimiento para detección de carreteras fue probado progresivamente para todos los pasos del flujo de proceso correspondientes a los bloques de la fig. 4.1. La medida de rendimiento de cada módulo es el resultado de recabar una base de información de referencia y a través de la comparación con los resultados de varios métodos, se clasifican las alternativas solución bajo criterios de calidad como: ec. (4.2), ec. (4.3) para el caso del cálculo de la máscara de marcas viales y la máscara de espacio libre o ec. (4.13) para el caso de la detección de líneas.

Se muestra a lo largo del capítulo 4, la aplicación de restricciones basadas en el estándar IC 8.2, en cada una de las etapas para detección de carreteras a partir de sus componentes fundamentales: marcas, ec. (4.1), ancho de marca vial; líneas, ec. (4.8), mínima longitud de línea y máxima longitud de espacio separador entre

líneas; y carriles, ec. (4.8), ancho de carril.

Como muestra el comportamiento de las gráficas que aparecen en la fig. 4.32 a 4.36, y que se sintetiza en el cuadro 4.2, se observa un incremento en el número de detecciones, en promedio de 9,41 %, 4,91 %, 4,07 %, 1,67 % y $-0,36$ % respectivamente, al utilizar integración temporal, así como una reducción en promedio de 4,88 %, 0,63 %, 3,1 %, 1,25 % y 0,46 % en el número de falsas detecciones a lo largo de todos los experimentos, mientras que la precisión de las mediciones se mantuvo invariante con excepción de los resultados de la secuencia 5, donde la integración temporal decremento levemente la precisión de las detecciones de línea. Con lo cual se puede concluir que la integración temporal proporciona robustez al algoritmo. Así mismo, se observa una relación directa entre la complejidad del entorno, iluminación, tráfico y movimiento entre carriles, con los índices de calidad que muestran el comportamiento del algoritmo de detección de carreteras.

6.3. Aprendizaje de máquina en Clasificación de Carriles

La utilización de la metodología propuesta para la aplicación del aprendizaje de máquina en el contexto de los Sistemas Avanzados de Asistencia a la Conducción, ADAS, por sus siglas en inglés, para detectar marcas viales y clasificar tipos de línea separadora de carril arrojó resultados exitosos.

La segmentación de marcas viales a través de un clasificador SVM en diferentes escenarios: secuencia de referencia 1 y 2, con líneas bien definidas y líneas degradadas respectivamente, permitieron comprobar la robustez del método, por cuanto los dos conjuntos de entrenamiento pueden ser integrados en un solo conjunto unificado, logrando ampliar el grado de generalización del clasificador.

Un factor a destacar, necesario para obtener resultados cualitativos como los de la fig. 5.11, cuadro f, es la normalización unitaria de los vectores descriptores de marca vial. Este ajuste permite que la variación local de iluminación en la imagen, no afecte la representación de las marcas al interior del clasificador y por tanto se pueda realizar una acertada generalización. Para el caso de líneas degradadas, este procedimiento es la diferencia entre los resultados mostrados por otros métodos y los buenos resultados generados con SVM, tal y como se observa en la fig. 5.15.

Por otro lado, en la segmentación de marcas viales, si bien el índice de sensibilidad o la razón de verdaderos positivos sobre el total de detecciones positivas, fue

alta, para los experimentos realizados a lo largo de las secuencias de referencia 1 y 2, con valores de 91 %, en el caso de líneas bien definidas (13 % por encima del siguiente mejor método) y 70 % en el caso de líneas degradadas (44 % por encima del siguiente mejor método), la precisión generó valores moderados, de 31 % y 33 % respectivamente. La razón es consecuencia de la definición de las máscaras “Ground Truth” de cada secuencia, allí, solo se contemplan las líneas separadoras de carril, excluyendo cualquier otra línea, con lo cual, reconocimientos relacionados con la no correspondencia del mundo plano en la perspectiva vista de pájaro de las imágenes, ocasionan falsos positivos que se equiparán en cantidad con el número de píxeles de marcas viales reales. Sin embargo, los resultados en el índice de precisión en relación con los métodos de segmentación basados en gradiente de la sec. 4.1.2, estuvieron en valores promedio a 10 % y 3 % por debajo del método mas preciso, respectivamente para los escenarios con líneas bien definidas y líneas degradadas.

La estrategia por aprendizaje de máquina SVM para detección de marcas viales de la sec. 5.2.4 fue utilizada en el algoritmo general para detección de carreteras presentado en el capítulo 4 con resultados satisfactorios. Usando el protocolo de prueba de la sec. 4.5 sobre la secuencia de referencia 5 fig. 4.5 (a), se mostró como la alternativa SVM mejora la detección de líneas a un 83 %, en relación al 77 % obtenido con métodos para detección de marcas viales basados en gradiente.

La clasificación de líneas separadoras de carril es una tarea que solo se trató con la estrategia SVM en la sec. 5.3, la medida de éxito de esta tarea se mide a través del índice de exactitud utilizado en el experimento de sintonización paramétrica de la fig. 5.23. En este caso, se logró un índice de reconocimiento del 85.6 %. La discriminación del comportamiento del mejor clasificador alcanzado para valores óptimos de $\text{gama}=0.1$ y $C=1e4$, se muestra en la tabla 5.2.

6.4. Trabajo Futuro

Uno de los aspectos cruciales en la segmentación de carreteras demarcadas por líneas viales es la correcta transformación de perspectiva a vista superior, para ello es esencial que la calibración de parámetros extrínsecos sea lo suficientemente precisa. La lectura directa de extrínsecos a través de una Unidad Inercial IMU (Inertial Measurement Unit) incrementa el coste de la aplicación, además que se debe integrar las dos fuentes de información, extrínsecos calculados por visión con las lecturas del sensor a través de una estrategia de fusión [26] para mejorar la relación señal a

ruido y la precisión final de la medición. Por lo tanto, queda abierto el camino para buscar estrategias que minimicen el impacto del ruido en las mediciones inferidas a través de reconstrucción 3D de la carretera. En el desarrollo del presente trabajo una de las posibilidades que fue vislumbrada es el uso de técnicas de control para ajustar la calibración en cada imagen capturada, es decir, los parámetros extrínsecos son señales de actuación que pueden modificar la perspectiva de pájaro del mundo plano observado desde una cámara virtual, en donde las imágenes deben exhibir características conocidas, de las cuales se puede inferir tamaño de los objetos y paralelismos, mediciones que pueden ser observadas para controlar la variables de ajuste o actuación hasta lograr llevar las señales observadas a los valores deseados, en otras palabras, fijar una referencia o set point en un bucle realimentado. Este tipo de sistemas de control se enmarca dentro del contexto del control avanzado, por cuanto se debe modelar y tratar el ruido de los parámetros extrínsecos dentro de las estrategias de control.

Otro aspecto por desarrollar es el incremento en la robustez de la segmentación de carriles basados en segmentación de líneas viales. Para ello es necesario la implementación de un nivel más alto de comprensión de la carretera que haga uso de la detección y segmentación de objetos sobre el plano tierra de la escena, especialmente los elementos relacionados con la carretera, tales como barandillas, barreras, aceras y arcenes. En esta capa se deberá estimar la forma de la carretera, cambio en el número de carriles, carriles de incorporación, túneles y rotondas. El conocimiento de si se está realizando la incorporación a una rotonda es necesario para ajustar los parámetros de búsqueda de carriles, dado que el ancho de carril cambia pasando de 3.5m en promedio a 4.5m, dada la baja velocidad que supone el tránsito en una rotonda. Una vez, la robustez en la detección de carriles es alcanzada en cualquier escenario de carretera, el siguiente paso es la planificación de trayectorias en una jerarquía de análisis de información de aún más alto nivel, utilizando distintos módulos ADAS como: detección y seguimiento de objetos sobre la carretera, detección y seguimiento de carriles, identificación de señales de tráfico, identificación del punto de atención en la carretera por parte del conductor, entre otros.

Por otro lado, un enfoque que ha cobrado cada vez más relevancia en la visión por computador es el aprendizaje de máquina utilizando redes neuronales convolucionales, con ellas se podrá lograr que la segmentación de marcas viales sea posible en cualquier tipo de escenario: líneas bien definidas, líneas degradadas, día, noche, etc, gracias a su capacidad para inferir el conjunto de características adecuado sin

intervención de un experto, de la base de imágenes anotadas de la que se disponga. Además de adquirir la estructura paramétrica adecuada para realizar el reconocimiento con la precisión suficiente. En la actualidad se ha aplicado el Aprendizaje de máquina profundo o “Deep Learning” en otras áreas como reconocimiento de objetos, texto y voz, pero en el contexto de las marcas viales hay un largo camino por recorrer. Así mismo, otra ventaja de este enfoque de aprendizaje de máquina es la posibilidad de usar fuentes distintas de información, es decir, la posibilidad de hacer fusión sensorial, la descripción de las escenas que se requieren comprender a partir de un descriptor compuesto por cadenas de datos de fuentes diferentes, cámaras, láser, IMU y GPS entre otros.

Finalmente, una tarea que queda abierta a trabajos posteriores es la integración del algoritmo basado en detección de carreteras a través de arreglos de carriles acoplados con tiempos de vida finitos, y el seguimiento de carriles individualmente a través de estrategias computacionales bio-inspiradas como enjambres [8].

Apéndice A

Cambio de Perspectiva a Vista de Pájaro

A.1. Introducción

El cambio de perspectiva en una imagen se efectúan a través de una matriz de modelo proyectivo, que en dependencia de cuantos parámetros contenga, permite movimientos a los objetos proyectados con mas o menos grados de libertad. Así por ejemplo, el modelo proyectivo de similaridad, con 4 grados de libertad, permite escalado y traslación, paralelos al plano de la imagen y rotación en dirección normal al plano de la misma. En el caso del modelo proyectivo homográfico, se permite todos los movimientos en el espacio, 3 rotaciones y 3 desplazamientos, con 6 grados de libertad. Con ello, el plano tierra proyectado sobre la imagen resultante en la perspectiva en vista de pájaro, puede ser ajustado en cualquier caso para que sea paralelo al plano de la imagen.

Existen múltiples trabajos donde se hace uso de la transformación vista de pájaro. En [60] por ejemplo, se estima un modelo aproximado de perspectiva inversa para asistir al conductor en la tarea de aparcarse. De forma similar, con el mismo propósito, en [63], se obtiene el entorno circundante de un vehículo en vista de pájaro, usando 6 cámaras ojo de pez alrededor del coche, dispuestas en forma de mosaico; el método incluye la búsqueda de la trayectoria de unión óptima entre imágenes, definida sobre la zona de solapamiento, que genere la transición mas suave y coherente posible. Otros trabajos como [28], utilizan el cambio de perspectiva a vista de pájaro para estudiar la detección general de marcas viales: flechas, cruces de peatón, pasos de cebra, palabras y pictogramas, que son menos estudiadas que

las marcas de carril. Así mismo, en [94], se aborda el reconocimiento de este tipo de patrones, usando casamiento de imágenes panorámicas, para obtener un mapa de entorno local, donde es posible realizar detección contextual de marcas viales, a través de un proceso de segmentación basado en umbralización local y filtrado por máxima saturación, clasificación de contornos a través de máquinas de soporte vectorial, *SVM*, con caracterización por firma de contorno, y clasificación conceptual, usando campos aleatorios de Markov.

La obtención de la homografía, puede ser abordada por tres caminos diferentes: En primer lugar, esta la estimación del modelo de proyección basado en la detección y casamiento de características puntuales, sobre las imágenes fuente y destino de la transformación, descrito en [36]. Un segundo enfoque, utilizado por algunos autores como [74], utiliza el punto de fuga en una imagen, para estimar la matriz de cambio. Finalmente, el modelo proyectivo se puede obtener a través de la composición matricial que representa los fenómenos que experimenta la luz, reflejada sobre los objetos y capturada en las dos perspectivas que relaciona el cambio homográfico, este enfoque se abordará a continuación.

A.2. Homografía por Composición Matricial

Una vez obtenidos los parámetros extrínsecos de la cámara, las imágenes pueden ser proyectadas a una perspectiva más conveniente, llamada “*vista superior*” ó “*vista de pájaro*”, \mathcal{VP} , donde el paralelismo que existe entre las líneas de un carril estándar en autopista es conservado visualmente. Este cambio de perspectiva se efectúa a través de la transformación homográfica, mostrada en ec. A.1.

$$s^cP = {}^cP_s = {}^cH_{vp} {}^{vp}P \quad (\text{A.1})$$

Donde ${}^{vp}P$ es la proyección de un punto $P_{3D} = [X, Y, Z]$, desde la escena 3D, sobre la imagen en la perspectiva de *vista de pájaro* y cP es la proyección del mismo punto P_{3D} sobre la imagen en la perspectiva *original*. s es un factor de escala para el cual la transformación es válida.

Así, la matriz homográfica tiene tres componentes:

$${}^cH_{vp} = K^c T_m {}^mT_{vp} K^{-1} \quad (\text{A.2})$$

Donde K y K^{-1} son la matriz de intrínsecos y su inversa, cT_m es una matriz de

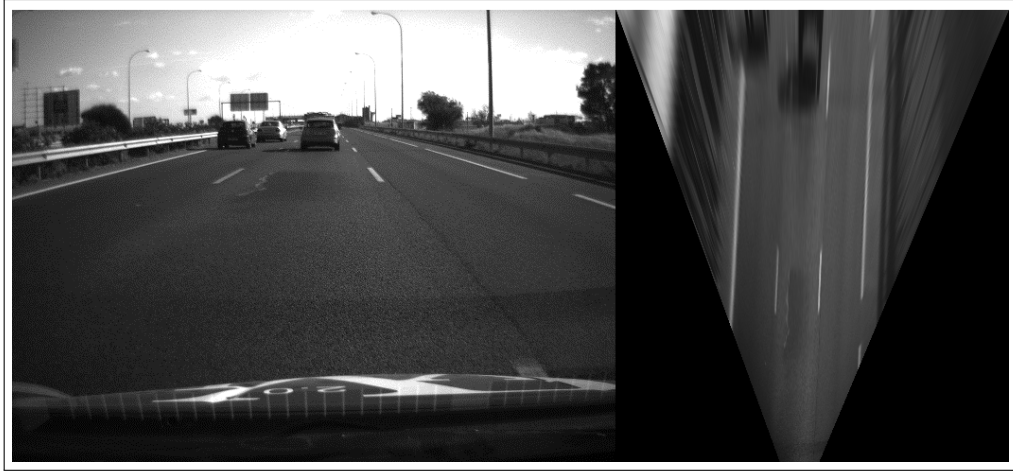


Figura A.1: Transformación homográfica desde la perspectiva real de la cámara a la perspectiva en *vista de pájaro*, $ROI_{vp}(x_{bv}, y_{bv}, z_{bv})$.

transformación homogénea que cambia la descripción espacial de un punto en el mundo 3D al sistema de referencia solidario con la cámara y ${}^mT_{vp}$ es la matriz de transformación homogénea desde el sistema de referencia solidario a la ubicación de la cámara, donde se observa la perspectiva *vista de pájaro*, al sistema de referencia en el mundo 3D:

$${}^cT_m = \begin{bmatrix} \mathbf{R}_x(-\alpha) \mathbf{R}_y(-\sigma) & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\ \mathbf{0} & \begin{bmatrix} \mathbf{I} & \begin{bmatrix} 0 \\ 0 \\ -h \\ 1 \end{bmatrix} \\ \mathbf{0} & \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} \quad (\text{A.3})$$

$${}^mT_{bv} = \begin{bmatrix} \mathbf{R}_x(\pi) & \begin{bmatrix} x_{bv} \\ y_{bv} \\ z_{bv} \\ 1 \end{bmatrix} \\ \mathbf{0} & \begin{bmatrix} x_{bv} \\ y_{bv} \\ z_{bv} \\ 1 \end{bmatrix} \end{bmatrix}$$

La posición de la cámara desde donde un observador tiene como punto de vista la perspectiva de pájaro $[x_{vp}, y_{vp}, z_{vp}]$, permite cambiar la región de interés mostrada en la imagen: mover la escena horizontalmente, verticalmente, o hacer acercamiento, respectivamente (fig. A.1).

Una representación gráfica de la descomposición homográfica mostrada en la Ec. A.2, se encuentra en la fig. A.2.

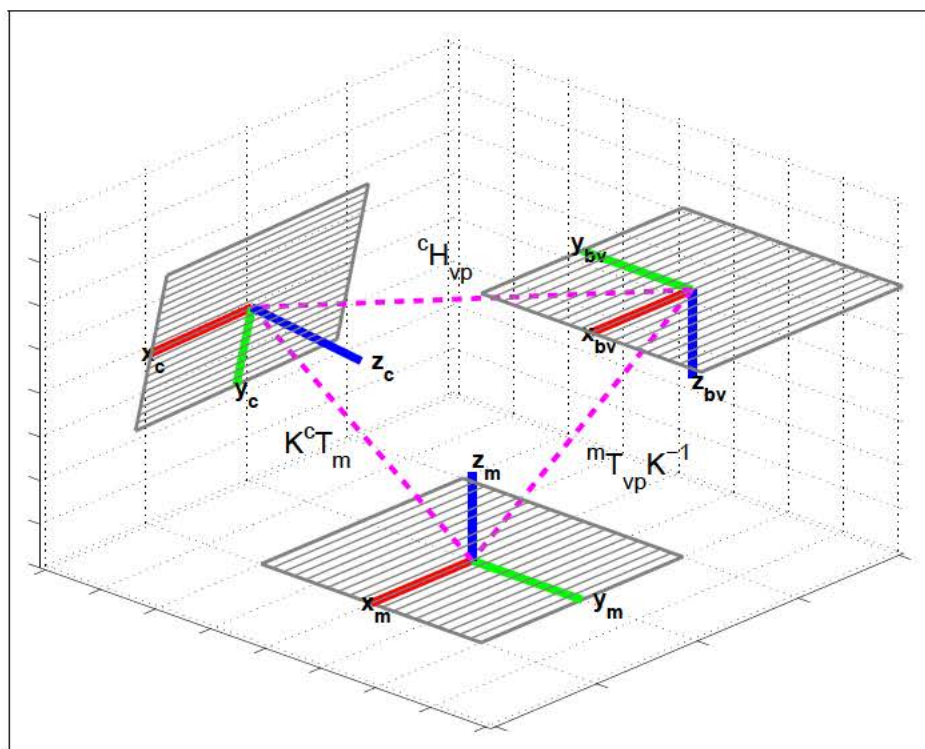


Figura A.2: Representación gráfica de la transformación homográfica desde la perspectiva *real* de la cámara a la perspectiva en *vista de pájaro*

Apéndice B

Elementos del Estándar IC 8.2

El problema de identificar de manera autónoma las líneas que puedan existir como límites de un carril en carreteras inter urbanas, se puede reducir a la detección de líneas longitudinales discontinuas ó continuas, ya que los demás tipos, son combinaciones de estas dos formas básicas.

por otro lado, los tipos de línea para la demarcación de la carretera, según el estándar IC 8.2, se dividen en grupos de la siguiente manera:

- M1. Longitudinales discontinuas.
- M2. Longitudinales continuas.
- M3. Longitudinales continuas adosadas a discontinuas.
- M4. Transversales.
- M5. Flechas.
- M6. Inscripciones.
- M7. Otras marcas.

Por lo tanto solo se abordara la revisión en detalle del grupo correspondiente a marcas longitudinales discontinuas y continuas, con todas sus posibles variaciones.

Ahora bien, las dimensiones que definen los diferentes tipos de línea, varían en función de la velocidad máxima permitida VM para una clase de carretera en particular, tal y como se muestra en la tabla B.2 y B.1.

De allí se puede apreciar que la mínima distancia posible entre segmentos de línea es 0.1m, la mínima longitud de segmento es 0.1m, la máxima distancia entre

segmentos corresponde a 12m para separación de carriles normales en carreteras con VM permitida de más de 100km/h, finalmente, se puede observar que la máxima dimensión de un segmento de línea es 20m, en marcas de borde de calzada sobre carreteras de más de 100km/h.

Función	Descripción
D1	Separación carriles normales.
D2	Separación carriles reversibles.
D3	Separación carriles incorporación entrada ó salida.
D4	Separación carriles especiales.
D5	Pre aviso de bifurcación.
D6	Pre aviso marca continua o de peligro.
D7	Borde de calzada cuando el ancho de calzada es inferior a 6,25m.
C1	Separación carriles mismo sentido, prohibición cambio de carril.
C2	Separación de sentidos en calzada de dos o tres carriles.
C3	Separación de sentidos en calzada de cuatro o más carriles.
C4	Separación carriles especiales.
C5	Separación de carriles de entrada o salida.
C6	Borde de calzada.

Cuadro B.1: Función de los tipos de marcas viales del estándar IC 8.2., referidos en la tabla B.2.

Tipos de marcas viales IC 8.2				
Grupo	Función	Velocidad máxima		
		$VM \leq 60$	$60 \leq VM \leq 100$	$VM \geq 100$
Marcas longitudinales discontinuas	D1			
	D2			
	D3			
	D4			
	D5			
	D6			
	D7			
Marcas continuas	C1			$\alpha = 0.1$
	C2			$\alpha = 0.2$
	C3			
	C4		$\alpha = 0.3$	$\alpha = 0.4$
	C5		$\alpha = 0.3$	$\alpha = 0.4$
	C6		$\alpha = 0.1$ (arcén $\leq 1,5$), 0.15 (arcén $\geq 1,5$)	$\alpha = 0.2$

Cuadro B.2: Tipos de marcas viales según el estándar IC 8.2 medidas en m. y VM en Km/h.

Apéndice C

Clasificadores

Los clasificadores pueden dividirse en dos grandes bloques, por un lado están los clasificadores clásicos, basados en métodos no paramétricos o de naturaleza estadística y en la segunda categoría se encuentran las estrategias paramétricas.

Los clasificadores clásicos se definen según el conocimiento estadístico con que se cuente, es decir, si se conoce la distribución analítica de los datos observados y su correcta categoría asociada θ , o si no se tiene información estadística a priori, excepto la inferida del conjunto de muestras observadas. En el primer caso, un análisis basado en la teoría de decisión de Bayes producirá un error de probabilidad en la clasificación mínimo. En el segundo caso, la decisión de clasificar una nueva observación x dentro de una categoría θ depende solo de una colección de n muestras correctamente clasificadas $(x_1 \theta_1), (x_2 \theta_2), \dots, (x_n \theta_n)$, y de un proceso de asociación no paramétrico, ejemplo: k-Nearest Neighbors, KNN.

Los clasificadores modernos por su parte son funciones parametrizables que permiten dividir un espacio de n dimensiones en dos sub-espacios a través de un hiperplano definido en $n-1$ dimensiones. El aprendizaje de los clasificadores consiste en asignar valores adecuados a los parámetros que definen el hiperplano separador, de tal forma que la información de entrada al clasificador, correspondiente a dos categorías distintas, quede dividida en partes diferentes del espacio, a lado y lado del hiperplano separador. Esta idea se puede observar tanto en las redes neuronales utilizadas como clasificador, cómo en las máquinas de soporte vectorial, SVM, así como en sus variantes, SVM difuso [61].

C.1. Clasificador Bayesiano

Sea $\mathbf{X} = X_1, \dots, X_n$ un vector de variables aleatorias observadas, donde cada característica toma valores de su dominio D_i . El conjunto de todos los valores (ejemplos o estados), se denomina $\Omega = D_1 \times \dots \times D_n$. Sea C una variable aleatoria observada que corresponde con la clase de un ejemplo, donde C puede tomar uno de m valores $c \in 0, \dots, m-1$. X_i representa variables, mientras que x_i representa valores; \mathbf{X} representa vectores.

Una función $g : \Omega \rightarrow 0, \dots, m-1$, donde $g(\mathbf{x}) = C$, representa un concepto a ser aprendido. la función $g(x)$ determinística corresponde a un concepto sin ruido, la cual asigna siempre la misma clase a un ejemplo dado. En general, sin embargo, un concepto puede ser ruidoso, produciendo una función aleatoria $g(x)$.

Un clasificador es definido por una función determinista $h : \Omega \rightarrow 0, \dots, m-1$ (una hipótesis) que asigna una clase a cualquier ejemplo dado. Un enfoque común es asociar cada clase i con una función discriminante $f_i(\mathbf{x})$, $i = 0, \dots, m-1$, y entonces, el clasificador selecciona la clase con máxima función discriminante para un ejemplo dado: $h(\mathbf{x}) = \operatorname{argmax}_{i \in 0, \dots, m-1} f_i(\mathbf{x})$.

El clasificador de Bayes $h^*(\mathbf{x})$ (también llamado clasificador óptimo de Bayes y nombrado como BO(\mathbf{x})) [81], usa como funciones discriminante, la probabilidad a posteriori dado un vector de características, es decir, $f_i^* = P(C = i | \mathbf{X} = \mathbf{x})$. Aplicando la regla de Bayes se obtiene $P(C = i | \mathbf{X} = \mathbf{x}) = \frac{P(\mathbf{X}=\mathbf{x}|C=i)P(C=i)}{P(\mathbf{X}=\mathbf{x})}$, donde $P(\mathbf{X} = \mathbf{x})$ es un factor común para todas las clases, y por lo tanto es ignorado. Esto genera las funciones discriminantes de Bayes.

$$f_i^* = P(\mathbf{X} = \mathbf{x} | C = i) P(C = i) \quad (\text{C.1})$$

donde $P(\mathbf{X} = \mathbf{x} | C = i)$ es llamada la distribución de probabilidad conjunta de clase, CPD por sus siglas en inglés. Así es clasificador de Bayes se define así.

$$h^*(\mathbf{x}) = \operatorname{argmax}_i P(\mathbf{X} = \mathbf{x} | C = i) P(C = i) \quad (\text{C.2})$$

El clasificador de Bayes calcula la máxima hipótesis de probabilidad a posteriori dado un ejemplo \mathbf{x} . Sin embargo la estimación directa de $P(\mathbf{X} = \mathbf{x} | C = i)$ desde un conjunto de ejemplos de entrenamiento es una tarea intratable cuando el espacio de características es de un orden dimensional elevado. Es por ello que comúnmente es utilizada para simplificar el problema, la suposición de que las características son independientes entre si dada una clase. Este raciocinio lleva a la definición de un

clasificador Bayesiano ingenuo o NB, naive Bayes Classifier, como es encontrado en la literatura especializada en lengua inglesa.

$$f_i^{NB} = \prod_{j=1}^n P(X_j = x_j | C = i)P(C = i) \quad (C.3)$$

C.2. Clasificador K-Vecinos más Próximos

También conocido como KNN (K-Nearest Neighbors) [20], basado en el conjunto de k vecinos más cercanos se puede definir formalmente como:

Sea un conjunto de pares observación-categoría: $(x_1 \theta_1), (x_2 \theta_2), \dots, (x_n \theta_n)$, donde la observación x_i toma valores en un espacio métrico X sobre el cual se define una métrica d , y la categoría θ_i , toma valores en el conjunto $[1, 2, \dots, M]$. Cada θ_i es el índice asociado para cada observación x_i .

Ahora bien, sea un nuevo par (x, θ) , para la medición observable x se estima la categoría θ correspondiente utilizando la información contenida en el conjunto de puntos clasificados correctamente. Se define entonces al vecino mas cercano a x como: $x'_n \in \{x_1, x_2, \dots, x_n\}$

$$\min d(x_i, x) = d(x'_n, x) \quad i = 1, 2, \dots, n. \quad (C.4)$$

La regla de decisión del vecino mas cercano asocia a x la categoría θ'_n de su vecino mas cercano x'_n . Para el caso de la regla de decisión de los k vecinos mas próximos, se realiza una ponderación de los k vecinos para asociar la categoría final.

C.3. Redes Neuronales

En la fig. C.1 se presenta la estructura base de un perceptrón multicapa. El ajuste paramétrico de la función de salida de la red neuronal o perceptrón multicapa, denominado proceso de aprendizaje de la red, se realiza a través del algoritmo de propagación hacia atrás ó back-propagation por si siglas en inglés. En la fig. C.2 se presenta esquemáticamente el diagrama del algoritmo de aprendizaje, el cual se compone de dos fases: propagación hacia adelante de la señal de entrada hasta generar una señal de salida, y una segunda fase correspondiente a la propagación del error hacia atrás, producido al comparar la salida generada con el valor esperado para la entrada. El esquema es un caso particular de dos capas, $L = 2$, tres neuronas de entrada, tres neuronas en la capa oculta y tres neuronas en la capa de salida, $m_0 =$

$m_1 = m_2 = 3$. La parte superior de la imagen muestra la fase de propagación hacia adelante, mientras que la parte inferior de la imagen presenta la fase de propagación de error hacia atrás.

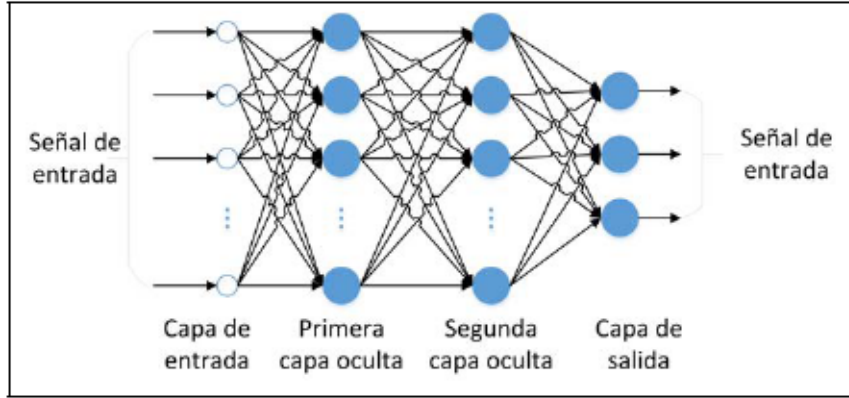


Figura C.1: Esquema básico perceptrón multicapa. Imagen obtenida de [37].

El algoritmo de entrenamiento que actualiza secuencialmente los pesos w_{ji} de la red, a través de los ejemplos de entrenamiento $\{(x(n), d(n))\}_{n=1}^N$, se presenta a continuación:

1. Inicialización: Los pesos de las neuronas, w_{ji} , se escogen de tal manera que los campos inducidos v_j , el estímulo acumulado de cada neurona, no sature las respectivas funciones de activación de cada neurona presente en la red.
2. Entrenamiento de ejemplos: Se presenta un conjunto completo de ejemplos de entrenamiento de tamaño N a la red, propagando los datos a través de las dos fases descritas en los apartados 3 y 4.
3. Propagación hacia adelante: Sea un conjunto de entrenamiento $(\vec{x}(n), \vec{d}(n))$ para la iteración n , $n = 1, 2, \dots, N$, con el vector de entrada $\vec{x}(n)$ aplicado a la capa de entrada y el vector de respuesta deseado $\vec{d}(n)$ aplicado a la capa de salida, se calcula los campos locales inducidos y las señales propagadas a las siguientes capas, capa por capa de la red. El campo local inducido $v_j^{(l)}$ por la neurona j en la capa l se muestra en la eq. (C.5).

$$v_j^{(l)} = \sum_{i=0}^{m_0} w_{ji}^{(l)}(n) y_i^{(l-1)}(n) \quad (C.5)$$

Donde $y_i^{(l-1)}(n)$ es la señal de salida de la función de activación de la neurona i en la capa $l - 1$ en la iteración n y $w_{ji}^{(l)}$ es el peso sináptico de la neurona j en

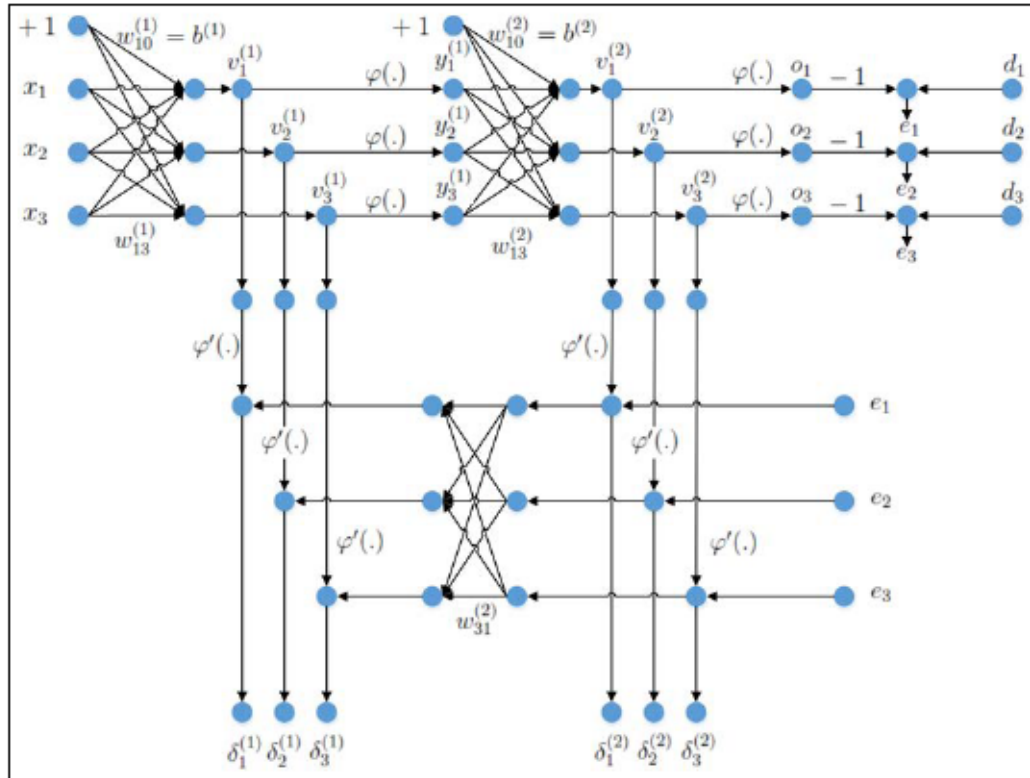


Figura C.2: Resumen gráfico del flujo de señales para el algoritmo de aprendizaje: “Back-Propagation”. Imagen obtenida de [37].

la capa l , cuya conexión viene de la neurona i en la capa $l - 1$. Para $i = 0$, se tiene que $y_0^{(l-1)}(n) = +1$ y $w_{j0}^{(l)}(n) = b_j^{(l)}(n)$ es el sesgo aplicado a la neurona j en la capa l . Tomando como función de activación de las neuronas, la función sigmoideal, la señal de salida de la neurona en la capa l es $y_j^{(l)} = \varphi_j(v_j(n))$. Si la neurona j está en la capa de entrada, $l = 0$: $y_j^{(0)} = x_j(n)$, donde $x_j(n)$ es el elemento j -ésimo del vector de entrada $\vec{x}(n)$. Si la neurona j está en la capa de salida, $l = L$, entonces $y_j^{(L)} = o_j(n)$. Ahora se calcula la señal de error de acuerdo a la ec. (C.6).

$$e_j(n) = d_j(n) - o_j(n) \tag{C.6}$$

Donde $d_j(n)$ es el elemento j -ésimo del vector deseado $\vec{d}(n)$.

4. Propagación hacia atrás: Se calcula los gradientes locales δs de la red neuronal según la ec. (C.7)

$$\delta_j^{(l)}(n) = \begin{cases} e_j^{(l)}(n)\varphi_j'(v_j^{(l)}(n)) & \text{para neurona } j \text{ capa } L \\ \varphi_j'(v_j^{(l)}(n)) \sum_k \delta_k^{(l+1)}(n)w_{kj}^{(l+1)}(n) & \text{para neurona } j \text{ capa } l \end{cases} \quad (\text{C.7})$$

Donde $\varphi'(\cdot)$ significa la diferenciación respecto al argumento. Ahora se ajusta los pesos sinápticos de acuerdo a la regla delta generalizada mostrada en la ec. (C.8).

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha [w_{ji}^{(l)}(n-1)] + \eta \delta_j^{(l)}(n) y_i^{(l-1)}(n) \quad (\text{C.8})$$

Aquí η representa la velocidad de aproximación al valor de pesos óptimo, mientras que α es un parámetro de regularización que ajusta el sobre entrenamiento.

4. Ciclo: Se realizan los cálculos de propagación hacia adelante y hacia atrás presentando conjuntos completos de entrenamiento hasta que el criterio de parada sea alcanzado.

Finalmente, el orden de los elementos del conjunto de entrenamiento, deben ser reorganizados aleatoriamente de ciclo en ciclo, y los valores de η y α deben cambiar, decrementalmente conforme los ciclos avanzan.

C.4. Máquinas de Soporte Vectorial

Las máquinas de soporte vectorial ó SVM Support Vector Machine, son clasificadores relativamente modernos, posteriores a las redes neuronales, propuestos inicialmente por [88]. La idea detrás de este tipo de clasificador es construir una hiper superficie no lineal en un espacio de características definidas por funciones llamadas Kernels, de tal forma que sea posible agrupar en dos categorías colecciones de datos no separables. La formulación matemática de una máquina de soporte vectorial se presenta en la ec. (C.9), allí se observa un problema de optimización sencillo sujeto a múltiples restricciones lineales, de tal forma que por tratarse de la minimización de una función convexa, se tendrá un mínimo global, característica que destaca frente a la minimización de error propia de las redes neuronales.

Sea $x \in \mathbb{R}^n$, la representación de un dato de entrada a clasificar, se calcula la representación auxiliar en el espacio de características definidas por la función de

núcleo $f = \phi(x) : f \in \mathbb{R}^{m+1}$, donde n es la dimensión de los datos de entrada y m es la dimensión del conjunto de entrenamiento.

$$\begin{aligned} \min_{\theta} C \sum_{i=1}^m y^i \log\left(\frac{1}{1 + e^{-\theta^T f^{(i)}}}\right) + (1 - y^{(i)}) \log\left(1 - \frac{1}{1 + e^{-\theta^T f^{(i)}}}\right) + \frac{1}{2} \sum_{j=1}^n \theta_j^2 \\ \text{subject to} \\ y = 1, \theta^T f \geq 0. \end{aligned} \tag{C.9}$$

La solución al problema se efectúa computacionalmente a través de una de las librerías especializadas disponibles como por ejemplo libsvm [15].

Bibliografía

- [1] Yaniv Alon, Andras Ferencz, y Amnon Shashua. Off-road path following using region classification and geometric projection constraints. En *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, tomo 1, págs. 689–696. IEEE, 2006.
- [2] Jose M Alvarez, Theo Gevers, Yann LeCun, y Antonio M Lopez. Road scene segmentation from a single image. En *European Conference on Computer Vision*, págs. 376–389. Springer, 2012.
- [3] José M Alvarez, A Lopez, y Ramon Baldrich. Illuminant-invariant model-based road segmentation. En *Intelligent Vehicles Symposium, 2008 IEEE*, págs. 1175–1180. IEEE, 2008.
- [4] José Manuel Álvarez, Antonio M López, y Ramon Baldrich. Shadow resistant road segmentation from a mobile monocular system. En *Iberian Conference on Pattern Recognition and Image Analysis*, págs. 9–16. Springer, 2007.
- [5] Anand Atreya, Bryan Cattle, y B Collins. Darpa urban challenge-princeton university-technical paper. *Princeton University*, 2007.
- [6] M Bertozzi, L Bombini, A Broggi, M Buzzoni, E Cardarelli, S Cattani, P Cerri, S Debattisti, R Fedriga, M Felisa, et al. The vislab intercontinental autonomous challenge: 13,000 km, 3 months, no driver. En *Proc. 17th World Congress on ITS, Busan, South Korea*. 2010.
- [7] Massimo Bertozzi y Alberto Broggi. Gold: A parallel real-time stereo vision system for generic obstacle and lane detection. *Image Processing, IEEE Transactions on*, 7(1):62–81, 1998.
- [8] A Bio-Inspired. Bio-inspired artificial intelligence.

-
- [9] Christopher M Bishop. Pattern recognition. *Springer*, 2006.
- [10] C Blaschke, F Breyer, B Färber, J Freyer, y R Limbacher. Driver distraction based lane-keeping assistance. *Transportation research part F: traffic psychology and behaviour*, 12(4):288–299, 2009.
- [11] Robert C Bolles y Martin A Fischler. A ransac-based approach to model fitting and its application to finding cylinders in range data. En *IJCAI*, tomo 1981, págs. 637–643. 1981.
- [12] Amol Borkar, Monson Hayes, y Mark T Smith. Robust lane detection and tracking with ransac and kalman filter. En *ICIP*, págs. 3261–3264. 2009.
- [13] Karel A Brookhuis, Dick De Waard, y Wiel H Janssen. Behavioural impacts of advanced driver assistance systems an overview. *European Journal of Transport and Infrastructure Research*, 1(3):245–253, 2001.
- [14] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.
- [15] Chih-Chung Chang y Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.
- [16] Hsu-Yung Cheng, Bor-Shenn Jeng, Pei-Ting Tseng, y Kuo-Chin Fan. Lane detection with moving vehicles in the traffic scenes. *IEEE Transactions on intelligent transportation systems*, 7(4):571–582, 2006.
- [17] Ondřej Chum y Jiří Matas. Matching with prosac-progressive sample consensus. En *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, tomo 1, págs. 220–226. IEEE, 2005.
- [18] Juan Manuel Collado Hernáiz. Detección y modelado de carriles de vías interurbanas mediante análisis de imágenes para un sistema de ayuda a la conducción. 2009.
- [19] European Commission. Eurostat regional yearbook 2014. *Statistical books*, 2014.
- [20] Thomas M Cover y Peter E Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

- [21] Navneet Dalal y Bill Triggs. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 1(1):886–893, 2005.
- [22] Radu Danescu y Sergiu Nedevschi. Probabilistic lane tracking in difficult road scenarios using stereovision. *IEEE Transactions on Intelligent Transportation Systems*, 10(2):272–282, 2009.
- [23] Joyce Dargay, Dermot Gately, y Martin Sommer. Vehicle ownership and income growth, worldwide: 1960-2030. *The Energy Journal*, págs. 143–170, 2007.
- [24] Instrucción de Carreteras. 8.2 ic “marcas viales.”. 1987.
- [25] Dirección General de Tráfico (DGT). Las principales cifras de la siniestralidad vial. España 2014. 2014.
- [26] Jiang Dong, Dafang Zhuang, Yaohuan Huang, y Jingying Fu. Advances in multi-sensor data fusion: Algorithms and applications. *Sensors*, 9(10):7771–7784, 2009.
- [27] N Minoiu Enache, Mariana Netto, Said Mammar, y Benoît Lusetti. Driver steering assistance for lane departure avoidance. *Control engineering practice*, 17(6):642–651, 2009.
- [28] Philippe Foucher, Yazid Sebsadji, Jean-Philippe Tarel, Pierre Charbonnier, y Philippe Nicolle. Detection and recognition of urban road markings using images. En *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, págs. 1747–1752. IEEE, 2011.
- [29] Jannik Fritsch, Tobias Kuehnl, y Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. En *International Conference on Intelligent Transportation Systems (ITSC)*. 2013.
- [30] Jannik Fritsch, Tobias Kuehnl, y Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. En *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, págs. 1693–1700. IEEE, 2013.
- [31] Behnam Ganji, Abbas Z Kouzani, Sui Yang Khoo, y Mojtaba Shams-Zahraei. Adaptive cruise control of a hev using sliding mode control. *Expert Systems with Applications*, 41(2):607–615, 2014.

-
- [32] Andreas Geiger, Philip Lenz, Christoph Stiller, y Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.
- [33] RJ Geraerts y Mark H Overmars. Sampling techniques for probabilistic road-map planners. 2004.
- [34] Ian Goodfellow, Yoshua Bengio, y Aaron Courville. Deep learning, 2016. URL <http://www.deeplearningbook.org>. Book in preparation for MIT Press.
- [35] Raghavan Gopalan, Tsai Hong, Michael Shneier, y Rama Chellappa. A learning approach towards detection and tracking of lane markings. *Intelligent Transportation Systems, IEEE Transactions on*, 13(3):1088–1098, 2012.
- [36] Richard Hartley et al. In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(6):580–593, 1997.
- [37] Simon HAYKIN. Neural networks, a comprehensive foundation. 1994.
- [38] Aharon Bar Hillel, Ronen Lerner, Dan Levi, y Guy Raz. Recent progress in road and lane detection: a survey. *Machine vision and applications*, 25(3):727–745, 2014.
- [39] Heiko Hirschmüller. Stereo processing by semiglobal matching and mutual information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):328–341, 2008.
- [40] Khashayar Hojjati-Emami, Balbir Dhillon, y Kouroush Jenab. Reliability prediction for the vehicles equipped with advanced driver assistance systems (adas) and passive safety systems (pss). *International Journal of Industrial Engineering Computations*, 3(5):731–742, 2012.
- [41] Yi Hou, Praveen Edara, y Carlos Sun. Situation assessment and decision making for lane change assistance using ensemble learning methods. *Expert Systems with Applications*, 42(8):3875–3882, 2015.
- [42] Albert S Huang, David Moore, Matthew Antone, Edwin Olson, y Seth Teller. Finding multiple lanes in urban road networks with vision and lidar. *Autonomous Robots*, 26(2-3):103–122, 2009.

-
- [43] Ruyi Jiang, Reinhard Klette, Tobi Vaudrey, y Shigang Wang. New lane model and distance transform for lane detection and tracking. En *International Conference on Computer Analysis of Images and Patterns*, págs. 1044–1052. Springer, 2009.
- [44] Yan Jiang, Feng Gao, y Guoyan Xu. Computer vision-based multiple-lane detection on straight road and in a curve. En *2010 International Conference on Image Analysis and Signal Processing*, págs. 114–117. IEEE, 2010.
- [45] Ian Jolliffe. Principal component analysis. *Wiley Online Library*, 2002.
- [46] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.
- [47] Dong-Joong Kang y Mun-Ho Jung. Road lane segmentation using dynamic programming for active safety vehicles. *Pattern Recognition Letters*, 24(16):3177–3185, 2003.
- [48] Ioannis Katramados, Steve Crumpler, y Toby P Breckon. Real-time traversable surface detection by colour space fusion and temporal analysis. En *International Conference on Computer Vision Systems*, págs. 265–274. Springer, 2009.
- [49] Jisu Kim, Jeonghyun Baek, Hyukdoo Choi, y Euntai Kim. Wet area and puddle detection for advanced driver assistance systems (adas) using a stereo camera. *International Journal of Control, Automation and Systems*, 14(1):263–271, 2016.
- [50] ZuWhan Kim. Robust lane detection and tracking in challenging scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):16–26, 2008.
- [51] Hui Kong, Jean-Yves Audibert, y Jean Ponce. Vanishing point detection for road detection. En *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, págs. 96–103. IEEE, 2009.
- [52] Chris Kreucher y Sridhar Lakshmanan. Lana: a lane extraction algorithm that uses frequency domain features. *IEEE Transactions on Robotics and automation*, 15(2):343–350, 1999.

- [53] Alex Krizhevsky, Ilya Sutskever, y Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. En *Advances in neural information processing systems*, págs. 1097–1105. 2012.
- [54] Tobias Kühnl, Franz Kummert, y Jannik Fritsch. Monocular road segmentation using slow feature analysis. En *Intelligent Vehicles Symposium (IV), 2011 IEEE*, págs. 800–806. IEEE, 2011.
- [55] Pranaw Kumar, Mathias Perrollaz, Stéphanie Lefevre, y Christian Laugier. Learning-based approach for online lane change intention prediction. En *Intelligent Vehicles Symposium (IV), 2013 IEEE*, págs. 797–802. IEEE, 2013.
- [56] Raphaël Labayrade, Jerome Douret, Jean Laneurit, y Roland Chapuis. A reliable and robust lane detection system based on the parallel use of three algorithms for driving safety assistance. *IEICE transactions on information and systems*, 89(7):2092–2100, 2006.
- [57] Yann LeCun y Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.
- [58] Yann LeCun, Léon Bottou, Yoshua Bengio, y Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [59] Chuanxiang Li, Bin Dai, Ruili Wang, Yuqiang Fang, Xingsheng Yuan, y Tao Wu. Multi-lane detection based on omnidirectional camera using anisotropic steerable filters. *IET Intelligent Transport Systems*, 2016.
- [60] Chien-Chuan Lin y Ming-Shi Wang. A vision based top-view transformation model for a vehicle parking assistant. *Sensors*, 12(4):4431–4446, 2012.
- [61] Chun-Fu Lin y Sheng-De Wang. Fuzzy support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):464–471, 2002.
- [62] Christian Lipski, Bjorn Scholz, Kai Berger, Christian Linz, Timo Stich, y Marcus Magnor. A fast and robust approach to lane marking detection and lane tracking. En *Image Analysis and Interpretation, 2008. SSIAI 2008. IEEE Southwest Symposium on*, págs. 57–60. IEEE, 2008.

- [63] Yu-Chih Liu, Kai-Ying Lin, y Yong-Sheng Chen. Bird's-eye view vision system for vehicle surrounding monitoring. En *Robot Vision*, págs. 207–218. Springer, 2008.
- [64] Keyu Lu, Jian Li, Xiangjing An, y Hangen He. A hierarchical approach for road detection. En *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, págs. 517–522. IEEE, 2014.
- [65] Riccardo Marino, Stefano Scalzi, y Mariana Netto. Integrated driver and active steering control for vision-based lane keeping. *European journal of control*, 18(5):473–484, 2012.
- [66] D Martín, F García, B Musleh, D Olmeda, G Peláez, P Marín, A Ponz, C Rodríguez, A Al-Kaff, A De La Escalera, et al. Ivv 2.0: An intelligent vehicle based on computational perception. *Expert Systems with Applications*, 41(17):7927–7944, 2014.
- [67] J Matas, C Galambos, y J Kittler. Progressive probabilistic hough transform. En *Proceedings of the British Machine Conference, pages*, págs. 26–31.
- [68] Joel C McCall y Mohan M Trivedi. Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *IEEE transactions on intelligent transportation systems*, 7(1):20–37, 2006.
- [69] Peter Meer, Doron Mintz, Azriel Rosenfeld, y Dong Yoon Kim. Robust regression methods for computer vision: A review. *International journal of computer vision*, 6(1):59–70, 1991.
- [70] Caio César Teodoro Mendes, Vincent Frémont, y Denis Fernando Wolf. Exploiting fully convolutional networks for fast road detection. En *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016.
- [71] F. Meyer. Color image segmentation. págs. 303–306, 1992.
- [72] Michael Montemerlo, Jan Becker, Suhrid Bhat, Hendrik Dahlkamp, Dmitri Dolgov, Scott Ettinger, Dirk Haehnel, Tim Hilden, Gabe Hoffmann, Burkhard Huhnke, et al. Junior: The stanford entry in the urban challenge. *Journal of field Robotics*, 25(9):569–597, 2008.

- [73] Marcos Nieto, Luis Salgado, Fernando Jaureguizar, y Jon Arróspide. Robust multiple lane road modeling based on perspective analysis. En *2008 15th IEEE International Conference on Image Processing*, págs. 2396–2399. IEEE, 2008.
- [74] Marcos Nieto, Luis Salgado, Fernando Jaureguizar, y Julian Cabrera. Stabilization of inverse perspective mapping images based on robust vanishing point estimation. En *Intelligent Vehicles Symposium, 2007 IEEE*, págs. 315–320. IEEE, 2007.
- [75] Gabriel L Oliveira, Wolfram Burgard, y Thomas Brox. Efficient deep models for monocular road segmentation. En *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, págs. 4885–4891. IEEE, 2016.
- [76] Alan V Oppenheim, Ronald W Schafer, John R Buck, et al. *Discrete-time signal processing*, tomo 2. Prentice-hall Englewood Cliffs, 1989.
- [77] World Health Organization et al. Global status report on road safety 2013: supporting a decade of action: summary. 2013.
- [78] Dean A Pomerleau. Alvin: An autonomous land vehicle in a neural network. Inf. téc., DTIC Document, 1989.
- [79] Vivek Pradeep, Gerard Medioni, y James Weiland. Piecewise planar modeling for step detection using stereo vision. En *Workshop on computer vision applications for the visually impaired*. 2008.
- [80] J Rebut, A Bensrhair, y G Toulminet. Image segmentation and pattern recognition for road marking analysis. En *Industrial Electronics, 2004 IEEE International Symposium on*, tomo 1, págs. 727–732. IEEE, 2004.
- [81] Irina Rish. An empirical study of the naive bayes classifier. En *IJCAI 2001 workshop on empirical methods in artificial intelligence*, tomo 3, págs. 41–46. IBM New York, 2001.
- [82] Peter J Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984.
- [83] F Samadzadegan, A Sarafraz, y M Tabibi. Automatic lane detection in image sequences for vision-based navigation purposes. *ISPRS Image Engineering and Vision Metrology*, 2006.

-
- [84] Bok-Suk Shin, Junli Tao, y Reinhard Klette. A superparticle filter for lane detection. *Pattern Recognition*, 48(11):3333–3345, 2015.
- [85] Bok-Suk Shin, Zezhong Xu, y Reinhard Klette. Visual lane analysis and higher-order tasks: a concise review. *Machine vision and applications*, 25(6):1519–1547, 2014.
- [86] Patrice Y Simard, David Steinkraus, y John C Platt. Best practices for convolutional neural networks applied to visual document analysis. En *ICDAR*, tomo 3, págs. 958–962. 2003.
- [87] Jongin Son, Hunjae Yoo, Sanghoon Kim, y Kwanghoon Sohn. Real-time illumination invariant lane detection for lane departure warning system. *Expert Systems with Applications*, 42(4):1816–1824, 2015.
- [88] Johan AK Suykens y Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.
- [89] Brian C Tefft. Asleep at the wheel: the prevalence and impact of drowsy driving. 2010.
- [90] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, MN Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.
- [91] Jyun-Guo Wang, Cheng-Jian Lin, y Shyi-Ming Chen. Applying fuzzy method to vision-based lane detection and departure warning system. *Expert Systems with Applications*, 37(1):113–126, 2010.
- [92] Yue Wang, Dinggang Shen, y Eam Khwang Teoh. Lane detection using spline model. *Pattern Recognition Letters*, 21(8):677–689, 2000.
- [93] Yue Wang, Eam Khwang Teoh, y Dinggang Shen. Lane detection and tracking using b-snake. *Image and Vision computing*, 22(4):269–280, 2004.
- [94] Thomas Woudsma, Lykele Hazelhoff, Peter HN de With, y Ivo Creusen. Automated generation of road marking maps from street-level panoramic images. En *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, págs. 925–930. IEEE, 2015.

-
- [95] Shing-Jen Wu, Hsin-Han Chiang, Jau-Woei Perng, Chao-Jung Chen, Bing-Fei Wu, y Tsu-Tian Lee. The heterogeneous systems integration design and implementation for lane keeping on a vehicle. *IEEE Transactions on Intelligent Transportation Systems*, 9(2):246–263, 2008.
- [96] Mosquet Xavier, Andersen M, y Arora A. A roadmap to safer driving through advanced driver assistance systems. 2015.
- [97] Koichiro Yamaguchi, Akihiro Watanabe, Takashi Naito, y Yoshiki Ninomiya. Road region estimation using a sequence of monocular images. En *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, págs. 1–4. IEEE, 2008.
- [98] Feng You, Ronghui Zhang, Guo Lie, Haiwei Wang, Huiyin Wen, y Jianmin Xu. Trajectory planning and tracking control for autonomous lane change maneuver based on the cooperative vehicle infrastructure system. *Expert Systems with Applications*, 42(14):5932–5946, 2015.
- [99] Fatin Zaklouta y Bogdan Stanciulescu. Real-time traffic sign recognition in three stages. *Robotics and autonomous systems*, 62(1):16–24, 2014.
- [100] Geng Zhang, Nanning Zheng, Chao Cui, Yuzhen Yan, y Zejian Yuan. An efficient road detection method in noisy urban environment. En *Intelligent Vehicles Symposium, 2009 IEEE*, págs. 556–561. IEEE, 2009.
- [101] Baltsaviase Zhange y A Gruen. Knowledge based image analysis for 3d road reconstruction. *Proceedings of the 21st Asian Conference on Remote Sensing, Zurich: Institute of Geodesy and Photogrammetry, ETH Hoenggerberg*, 100:105, 2000.
- [102] Dongbin Zhao, Zhaohui Hu, Zhongpu Xia, Cesare Alippi, Yuanheng Zhu, y Ding Wang. Full-range adaptive cruise control based on supervised adaptive dynamic programming. *Neurocomputing*, 125:57–67, 2014.