

Detección de líneas de carril en autopista

José Luis Gómez Zurita

Resumen — La detección de líneas de carril es un recurso muy utilizado para la asistencia en la conducción, ya que ayuda tanto a conductores como a sistemas de conducción autónoma a mantener el vehículo dentro del carril. Uno de los métodos y objeto de este artículo, es la detección de líneas en autopista basada en un sistema de visión con cámara monocular. Uno de los mejores métodos del estado del arte se basa en un descriptor de crestas [1]. Partiendo de un código no funcional que implementa [1], será reimplementado y mejorado mediante los métodos explicados a continuación. El descriptor de crestas aplica un filtro que resalta las líneas de carril en la imagen. Aplicando el algoritmo de Hough sobre la imagen de crestas se detectan las líneas más importantes. A continuación se utiliza un algoritmo para seleccionar las líneas del carril por donde circula el vehículo. Finalmente se utiliza un sistema de seguimiento basado en filtros de Kalman para mantener una coherencia temporal y así eliminar detecciones espurias, obteniendo un sistema más robusto. Para realizar los experimentos se han tomado secuencias de imágenes propias en autopista en diferentes momentos del día, donde se analizan los resultados obtenidos.

Palabras Clave—Conducción autónoma, filtro de Kalman, cresta, Transformada de Hough, tracking.

Abstract — Lane detection is a resource used a lot for driving assistance, it helps drivers and autonomous systems to maintain the vehicle inside the lane. One of the methods and object of this paper, is the highway lane detection based in a vision system with a monocular camera. One of the best methods of the art state is based on a ridge descriptor [1]. Starting with a non-functional code that implements [1], it will be reimplemented and improved by the following methods explained. The ridge descriptor applies a filter that remarks lanes lines on the image. Applying on this filtered image the Hough Transform algorithm, followed by a selection mechanism for obtain the line that are more adequate to the lane where the vehicle is driving, assisted by tracking methods that use Kalman filters with the detection on the Hough transform space and it helps to maintain a coherence with the tracking detection that its being performing, for make a final detection more robust. For realize the experiments it has toke own highway image sequences at different time of the day, where analyze the results obtained.

Index Terms — Autonomous driving, Kalman filters, ridge, Hough Transform, tracking.



1. INTRODUCCIÓN

El contexto de este trabajo se sitúa en el campo de la visión por computador, aplicado a la automoción para realizar sistemas de ayuda a la conducción autónoma.

Actualmente, multitud de empresas, desarrollan sistemas de asistencia a la conducción. Entre estos sistemas de asistencia están la conducción autónoma por autopista o ciudad, siendo el caso de empresas como Google y Tesla. Se trata de un mercado en auge y que se está explotando e informando constantemente de sus avances, ya que supondrá sin duda una revolución en la conducción y el día a día de la sociedad.

El grupo de investigación Advanced Driver Assistance Systems (ADAS) del Centro de Visión por Computador (CVC), se encarga de investigar y desarrollar tecnología aplicada a la conducción autónoma. Para ello dispone de un coche eléctrico dotado de cámaras y un sistema para poder controlar el coche de manera autónoma. El objetivo de ADAS es dotar este vehículo de cámaras que permitan al coche circular de manera autónoma.

Para que el coche pueda empezar a circular de manera autónoma existen varios módulos imprescindibles entre

los que se encuentra la detección de líneas de carril, que será el módulo a implementar y objetivo de este trabajo. El trabajo se centrará en la detección de líneas de carril en autopista pero sin dejar de lado el entorno urbano, realizando pruebas de funcionamiento en ambos entornos.

2. ESTADO DEL ARTE

Numerosos vehículos cuentan ya con sistema avanzados de navegación y asistencia que hacen uso de la detección de líneas de carril. Generalmente estos sistemas se usan como una medida de seguridad para evitar salidas de involuntarias del carril [2]. Entre estos sistemas se encuentran los basados en visión con cámaras frontales o laterales y los basados en otros sensores como infrarrojos.

Entre los sistemas de visión basados en cámaras frontales, existen los que solo requieren de una cámara estereoscópica (3D). Este trabajo, se enfocará en la detección de carriles mediante sistema de cámara monocular, por ser un método robusto y económico. Los sistemas de cámaras estereoscópicas suponen un coste más elevado de hardware y requieren cómputo del 3D.

Existen numerosos trabajos acerca de la detección de líneas de carril mediante cámaras 2D [3], [4], [5]. Estos métodos tienen en común el uso de detector de contornos y filtros para detectar el cambio de intensidad que genera la línea de carril pintada sobre el asfalto.

Uno de los mejores métodos del estado del arte [1]

- E-mail de contacto: joselui-master@hotmail.com
- Mención realizada: Computación.
- Trabajo tutorizado por: Antonio López Peña
- Curso 2015/16

desarrollado en el CVC por Antonio López para Volkswagen en el año 2003. En este trabajo se tomará como punto de partida por disponer del código fuente. Este código fuente recuperado, se encuentra no funcional debido a incompatibilidades de arquitectura y librerías obsoletas, e integra el descriptor de crestas, altamente detallado en [1], [6], [7]. Anteriormente, había sido utilizado como detector de líneas, junto a más módulos y otras técnicas, pero implementaba una lógica de alto nivel pobre, que necesitaba mejorar. Por lo tanto, como punto de partida es importante recuperar el código, modernizarlo y reutilizarlo junto a nuevas técnicas que doten de una lógica de alto nivel, a la hora de detectar líneas, más robusta.

Se debe tener en cuenta, que a la hora de realizar la detección se distingue entre líneas rectas y curvas, ya que el método de detección varía considerablemente. Siendo las líneas rectas lo más primordial ya que predominan en carreteras urbanas, autopistas y otros tipos de vías, donde existe más riesgo de distracción debido a la monotonía de la vía.

3. OBJETIVOS

El objetivo principal del trabajo es diseñar un detector de líneas de carril robusto y con una buena funcionalidad de alto nivel, basándose en [1], [6] y [7] y a partir del código no funcional existente, se integrarán nuevas metodologías que mejoren el resultado que se obtenía en estas referencias. Para ello, el detector estará centrado en detectar líneas rectas.

Será esencial, que en tramos rectos o con poca curva, se haga una detección como el mostrado en la figura 1, manteniéndose así incluso en condiciones lumínicas menos favorables, posibles obstáculos o ausencia de detección puntual.

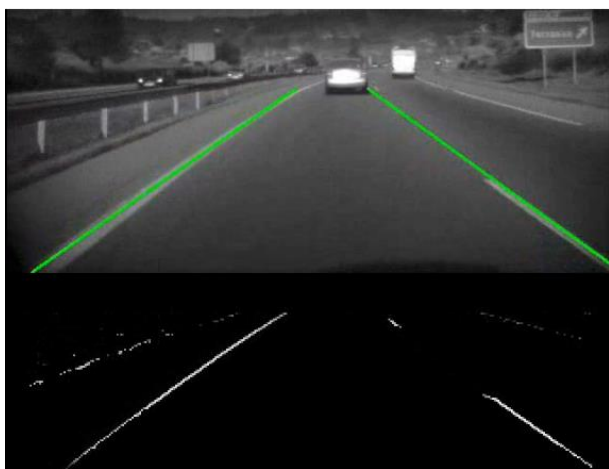


Fig. 1 - Ejemplo de un resultado satisfactorio de procesamiento de una imagen y muestra de resultados.

Como subobjetivos se plantea:

- Estudio y análisis del estado del arte para proponer mejoras a [1].
- Adaptación del código que implementa [1] a C++ utilizando la librería OpenCV, con la cual se trabajará para hacer procesamiento de imágenes de forma más

eficiente, estructurándolo en clases y acondicionándolo para que sea escalable e iterable. Hacer que funcione y obtener resultados del baseline.

- Mejorar la lógica actual del sistema para obtener resultados robustos ante situaciones como los cambios de carril.
- Realizar los experimentos necesarios para evaluar el comportamiento del sistema.

4. METODOLOGÍA Y ENTORNO

Es muy importante establecer una metodología adecuada y realizar un programa que implemente todo lo mencionado en los objetivos de manera clara, estructurada y amigable puesto que este proyecto es reutilizable y ampliable en el futuro.

4.1. Planificación

La planificación que se ha llevado a cabo ha seguido un formato Scrum, realizando un proyecto modular e incremental. Se definieron reuniones semanales con el tutor, en las que se mostraba el estado de los módulos implementados y se planteaban los siguientes pasos del desarrollo. La composición del proyecto se ha dividido en fases que van culminando los objetivos derivados planteados (figura 2).

Fase inicial: estudio del estado del arte necesario para poder planificar, preparar adecuadamente el trabajo.

Fase 1: obtener los primeros resultados sobre imágenes, por lo que se necesita tener el código base que se ha proporcionado funcional e ir mejorándolo y añadiendo funcionalidades de manera iterativa.

Fase 2: realizar pruebas en secuencias de imágenes y a partir de los resultados, estudiar dónde hay que realizar mejoras o cambios para mejorar el resultado obtenido.

Fase 3: tareas de tracking, aplicando una lógica de más alto nivel que responda ante el hecho de realizar un seguimiento de la progresión de la vía. Además de realizar pruebas en secuencias tomadas en autopista a diferentes horas del día.

Fase final: se ajustan parámetros para obtener variaciones favorables en los resultados y se aplican funcionalidades extra como son la detección de cambio de carril y de si la línea es continua o discontinua.

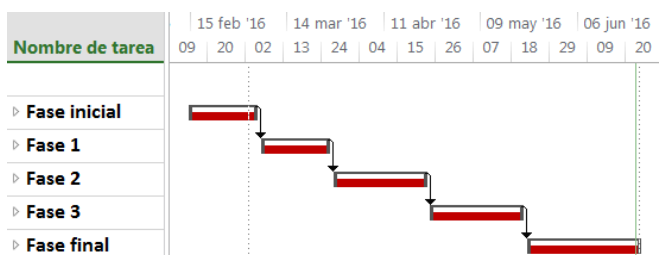


Fig. 2 - Diagrama de Gantt que muestra brevemente el calendario seguido para cada una de las fases y que corresponde a cada una de las iteraciones realizadas.

4.2. Entorno

El entorno de trabajo donde se ha realizado el programa, ha sido el sistema operativo Ubuntu 14.04 (Linux), programado en C++ [8], haciendo uso de la librería gráfica

OpenCV 2.4.12, utilizando el compilador gcc de linux y como debugger el Qt Creator 5.5.1. Este entorno elegido ha sido porque se utiliza ROS (Robot Operative System), que es un framework en el que se puede implementar software para robots, en este caso para el vehículo de conducción autónoma.

El vehículo de conducción autónoma es el mostrado en la figura 3. Se caracteriza por ser un vehículo pequeño, dotado de un ordenador completo en el maletero, con placa base, procesador y tarjeta gráfica de alto rendimiento, dónde realizar los cálculos a la hora de procesar las imágenes que se captan por las cámaras.



Fig. 3 - Imágenes del vehículo eléctrico que se utiliza para las pruebas de conducción autónoma en el CVC.

4.3. Diagrama de clases

La estructura del programa consta de un manager, en este caso una clase (LaneDetectorManager), que instancia una clase detector de líneas (LaneDetector), siendo esta una clase abstracta de la cual heredará cada uno de los tipos de detector que se podrán realizar y que por ahora son dos, el detector de líneas rectas (StraightLaneDetector) y el detector de líneas curvas (CurveLaneDetector), figura 4.

LaneDetector es una clase abstracta porque contiene métodos comunes protegidos que utilizan sus hijos de forma común, como son la conversión de formato de imagen Mat a vector o viceversa, printados por pantalla, dibujado de líneas, entre otros. Además de tener las variables privadas de la clase que serán comunes independientemente del tipo concreto (hijo) de detector.

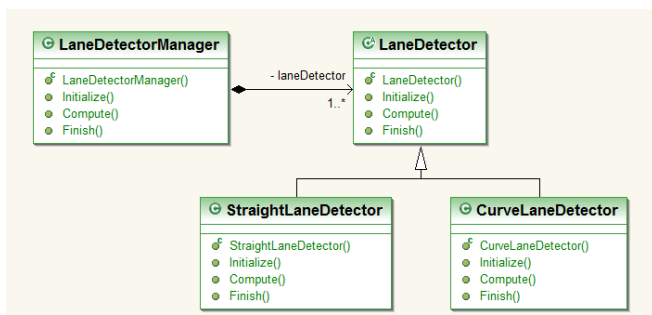


Fig. 4 - Diagrama de las principales clases con la estructura base del programa

4.4. Estilo

Se utiliza Doxygen [9] como herramienta para generar la documentación del código, siendo este adecuadamente estructurado, siguiendo un orden de aparición de los métodos que integran las diferentes clases, los archivos

incluidos, variables globales y opciones de compilación. Todo archivo ya sea de cabecera (.h) o código (.cpp) tiene una descripción inicial del módulo al que pertenece, autor, fecha y descripción utilizando Doxygen. Lo mismo se aplica a todos los métodos, estos tienen una cabecera antes que indica el nombre, descripción, parámetros de entrada, autor y fecha.

4.5. Fichero de parámetros

La necesidad de un fichero de parámetros que contenga toda la configuración del programa se ha hecho latente ante diferentes datasets realizados con diferentes cámaras y puntos de grabación diferentes. Ya que esto cambia por un lado el punto de fuga, afectando a la anchura de las carreteras y ángulos de detección en las líneas, además los cambios de resolución en las mediciones establecidas en pixeles suponen un problema con las variables prefijadas del programa.

Aparte los métodos de activación y desactivación de los filtros y trackings dependen de los frames procesados, por lo que afecta la velocidad de grabación en frames por segundo de la cámara.

Se crea una función de lectura de fichero de parámetros cuando se inicializa la clase del detector de líneas. De esta manera se puede facilitar el fichero que más convenga para el dataset a probar.

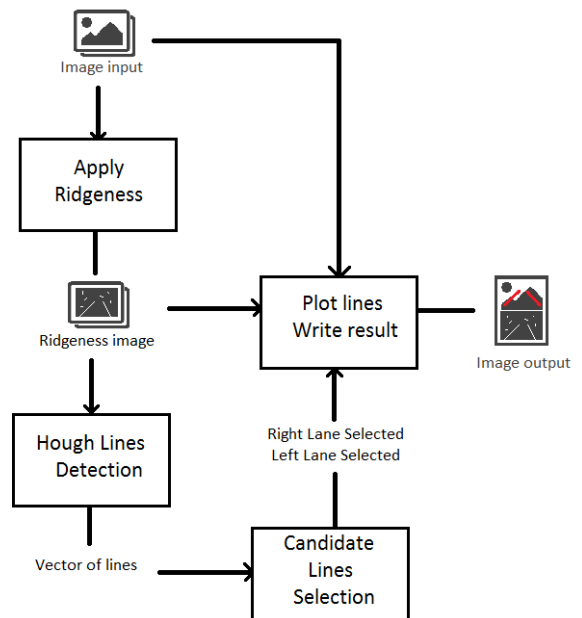


Fig. 5 - Esquema de bloques que describe los procesos que realiza el módulo de detección.

5. DETECCIÓN DE LÍNEAS DE CARRIL

El programa a implementar debe realizar la detección de líneas de carril similar a como lo realizaba el código no funcional base. Por lo que en primer lugar, la implementación se centrará en aplicar los módulos y funcionalidades observables en la figura 5. Siendo estos, la imagen filtrada con las crestas (Ridges), reutilizada del código base, aplicar la transformada de Hough de la librería

OpenCV para detección de rectas en vez de implementarla manualmente como estaba en el código de [1], obtener un resultado que muestre la imagen original con las líneas remarcadas sobre ella y debajo la salida de aplicar el filtro de crestas a la imagen.

5.1. Crestas

El concepto de Crestas (Ridges) [1], [6], [7], utilizado en el código fuente de partida, intenta ir más allá del gradiente y detección de contornos para ser más robusto y constante. Este concepto consiste en dividir por capas de intensidad la imagen en escala de grises, siendo las crestas los puntos de mayor intensidad, de esta manera se obtiene el grado de similitud de cada pixel en el hecho de estar próximo a una cresta, los pixeles próximos a crestas tienen unos valores más altos siendo estos más fáciles de filtrar (figura 6).

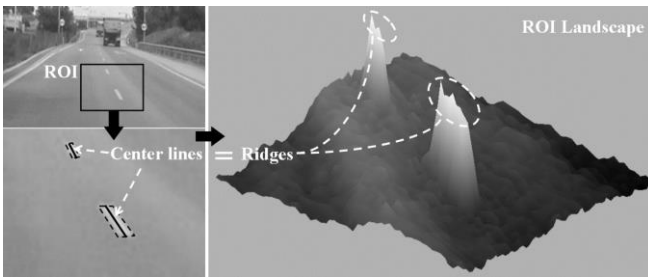


Fig. 6 - Representación de la región de interés (ROI) de la imagen de la carretera, proyectada según la intensidad de sus píxeles para ver la variación de valores y donde se situarían las crestas (imagen sacada del artículo "Ridgeness for Detecting Lane Markings" [1]).

Para conseguir los valores de proximidad a una cresta se realiza mediante el descenso de gradiente, obteniendo de aquí los valores de los vectores propios que determinan el grado de pendiente, para saber cuan próximos a una cresta están. Realizando el tensor estructural sobre la imagen y los vectores dominantes del gradiente se obtiene la dirección predominante y se determina qué parte es una cresta y por lo tanto línea de carril (figura 7).

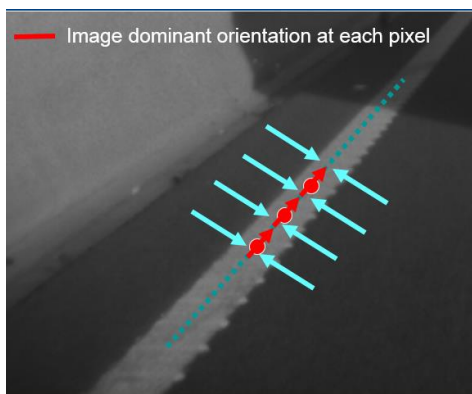


Fig. 7 - Representación de los vectores del gradiente y la orientación dominante.

5.2. Transformada de Hough

Una vez obtenida la imagen binarizada de crestas, el siguiente paso es utilizar la transformada de Hough para detectar las líneas rectas y seleccionar las más convenientes [10].

tes [10].

En primer lugar, para implementar el algoritmo que aplica la transformada de Hough, se utiliza una función de la librería OpenCV [11]. Con esta función, se obtienen líneas rectas que satisfacen un filtro, para así obtener solo líneas con suficiente largo como para determinar que son de carril. Estas líneas están en coordenadas polares y se trabajará en estas coordenadas a la hora de realizar una selección.

Cuando se utiliza esta función, se obtiene como resultado una lista de líneas detectadas en la imagen, ordenadas de mayor a menor calidad. Para seleccionar qué líneas corresponden al carril donde circula el vehículo, se divide la búsqueda en línea derecha del carril e izquierda, pues tendrán unas propiedades diferentes. Para discriminar entre lado derecho e izquierdo de la imagen, se calcula el punto por el que pasa cada línea al principio de la imagen (correspondiente al plano más cercano) en coordenadas cartesianas a partir de las polares, de este modo se sabe si la línea pertenece a un lado u otro.

Una vez detectado el lado donde empieza la línea se mira si esta cumple con un ángulo (en radianes) coherente al que debería tener una línea de carril. Por ejemplo para el lado izquierdo, encontrar una línea con ángulos comprendidos entre $(30^\circ, -90^\circ)$ y para el lado izquierdo ángulos entre $(90^\circ, 150^\circ)$. Para realizar esto, se aprovecha el ángulo θ , para obtener el ángulo que cumple las condiciones mencionadas en el ejemplo, véase figura 8.

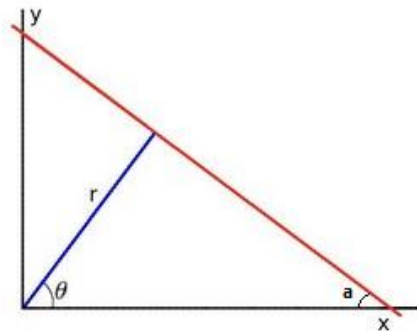


Fig. 8 - Representación del sistema de coordenadas polares (r, θ) y su equivalente en coordenadas cartesianas (línea roja). El ángulo a corresponde al ángulo a calcular para determinar si una línea tiene coherencia en la imagen.

Una vez determinada qué líneas cumplen los requisitos definidos, se guardan las 10 primeras líneas, de haberlas, para cada lado, pues serán las más idóneas como línea de carril. Acto seguido se hace un tratamiento de estas líneas candidatas más estricto para obtener la mejor para cada lado, en base a criterios de ancho de carril y la distancia al centro. Una vez realizada la selección de líneas se pueden obtener los resultados indicados en la figura 9.

Posibles detecciones

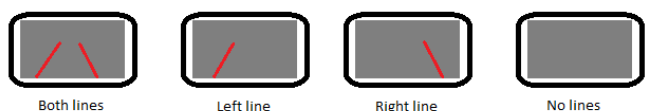


Fig. 9 - Posibles resultados obtenidos en el espacio de Hough

5.3. Resultados Iniciales

Los resultados obtenidos en primera instancia de aplicar las técnicas mencionadas hasta ahora, se pueden observar en la figura 11. Estos resultados muestran cómo se obtienen las líneas de manera clara en la imagen de crestas, además de que la selección de candidatos en el espacio de Hough consigue ajustarse adecuadamente a las líneas de carril.

Aun vistos estos resultados, a continuación se enumeran situaciones desfavorables en las que no se obtienen resultados o resultados no deseados:

- Líneas discontinuas: si las líneas discontinuas están muy distantes, la detección en este carril no aparecerá o se irá a la línea del siguiente carril, hasta volver a tener suficiente línea discontinua para detectar nuevamente.
- Cambios de carril o invasión de línea: cuando el vehículo empieza a pisar la línea, esta deja de detectarse por funcionalidad de selección e intenta buscar otra línea próxima, resultando en un resultado incorrecto.
- Líneas desgastadas: si el desgaste es muy pronunciado la imagen de crestas no detectará la línea.
- Vehículos u otros objetos: los vehículos a los lados u otros objetos pueden producir en el espacio de crestas líneas no deseadas que se detectan como posibles líneas (outliers).
- Zona urbana: pasos de peatones, zonas con muchas líneas marcadas, resaltos de todo el carril entre otros elementos, producen detecciones erróneas.

Para solucionar toda esta serie de problemas, en los que la detección falla en momentos más puntuales, es necesario preparar una lógica de alto nivel, que responda adecuadamente ante estas situaciones, es por eso que surge la necesidad de realizar seguimientos sobre la vía en la que se encuentra el vehículo (tracking).

6. SEGUIMIENTO DE LÍNEAS DE CARRIL Y LÓGICA DE ALTO NIVEL

La necesidad de realizar tracking surge de pensar en técnicas que ayuden a predecir el resultado o hacerlo más robusto ante las situaciones desfavorables mencionadas anteriormente. Es importante realizar un seguimiento de algún tipo a medida que se ha ido avanzando, se pueda predecir o asegurar un margen fiable donde hallar un resultado de detección razonable y suprimir estos outliers.

Posibles detecciones con filtros de Kalman

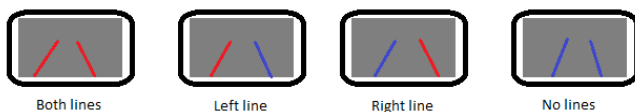


Fig. 10 – Posibles resultados aplicando los filtros de Kalman en color azul para asistir el resultado

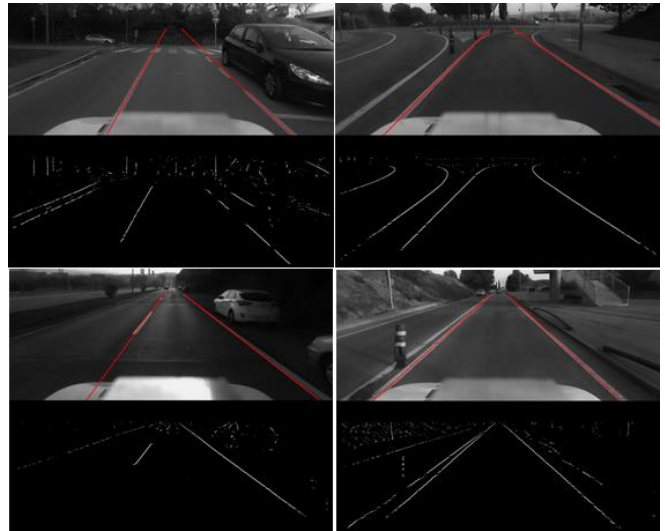


Fig. 11 – Primeros resultados obtenidos de aplicar el algoritmo de crestas para filtrar la imagen original (imagen inferior de cada frame) y resultado en rojo de realizar la selección de líneas en el espacio de Hough.

6.1. Filtros de Kalman

Es muy común hacer uso de filtros de Kalman [12], [13], [14] para realizar un tracking del elemento a seguir, para poder predecir un resultado aproximado en un estado siguiente (por ejemplo siguiente frame). Aplicado al detector, esto se traduce en dos filtros de Kalman, uno para la línea de carril derecha y otro para la izquierda. Las líneas detectadas están en coordenadas polares, por lo que el tracking mediante el filtro será en este espacio.

El tracking con filtros de Kalman se utiliza como sistema de soporte en el detector de líneas. En primer lugar se toman en cuenta los posibles resultados obtenibles al analizar la imagen y buscar las líneas en el espacio de Hough (Figura 9).

Al tener dos filtros de Kalman activos, uno para cada línea, primero se inicializan estos a unos valores de línea inicial conocidos en coordenadas polares, por ejemplo una primera imagen ya comprobada que se detecten líneas adecuadamente. Hecho esto, por cada detección producida del espacio de Hough, se reajusta el filtro de Kalman correspondiente, en caso de no tener detección se busca la predicción del filtro de Kalman y se utiliza como resultado (Figura 10).

De este modo se pueden evitar outliers, tal y como se verá en el apartado de Experimentos, ya que se puede ir mostrando como resultado la línea predicha por los filtros, siendo dentro de un margen una línea adecuada como resultado.

Ante el uso de estos filtros surge una gran problemática y es determinar qué detecciones son realmente adecuadas para realizar el reajuste del filtro, porque estas podrían ser detecciones erróneas y es preferible evitarlas de ser posible, para no aumentar el error en la predicción. Además habrá situaciones en las que no haya línea que detectar y el filtro de Kalman no deba de calcular ninguna predicción.

Para tratar el primer problema se utilizan más elementos de tracking, tal y como se verá más adelante. Para el

segundo problema, se utiliza un sistema de activación y desactivación de los filtros de Kalman, que se basa en contadores.

6.1.1. Activación y desactivación de filtros Kalman

Cuando se habla de un sistema de activación/desactivación, se hace referencia al hecho de hacer uso del resultado de la línea predicha por el filtro de Kalman para realizar cálculos posteriores que determinarán qué resultado elegir como salida final.

Este sistema resulta necesario, ya que no siempre se detectan líneas de carril, ya sea porque estén borradas o por el tipo de vía u otros factores, como glorietas, cruces con multilíneas o pasos de peatones. A fin de evitar en zonas donde claramente no hay líneas de carril que se muestre una salida incorrecta, se crea un mecanismo de activación y desactivación que dependerá a priori del número de frames consecutivos dándose una acción concreta.

La activación/desactivación dependerá de ciertas situaciones de las cuatro ya mencionadas que se den en la detección. Las situaciones afectarán independientemente al filtro de Kalman, correspondiente a la línea derecha e izquierda. Por lo que se tendrán cuatro situaciones posibles, ambos filtros desactivados, uno activo y el otro no y viceversa y ninguno activo.

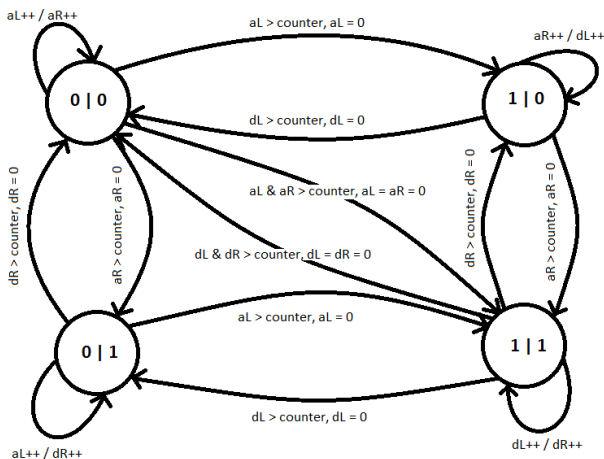


Fig. 12 - Diagrama de transiciones entre los estados de activación y desactivación del filtro de Kalman. El contador de activación aR para el Kalman derecho, aL para el izquierdo; el contador de desactivación dR para el Kalman derecho, dL para el Kalman izquierdo. Estas variables contador pueden o no ser incrementadas cuando se permanece en un estado y son reinicializadas a 0 cuando cambian de estado.

Estas situaciones se muestran como estados en la Figura 12, donde se puede observar como dependiendo del contador establecido, en este caso serían número de frames para activar o desactivar el filtro de Kalman (este puede coincidir o ser diferente y depender del número de frames por segundo de la cámara); cambia de un estado a otro. Dentro de cada estado, al recibir una nueva detección, cabe la posibilidad de incrementar el contador de activación, de desactivación o ninguno, según el caso. El incremento depende de si detecta una línea en el espacio de Hough adecuada, se incrementará la variable de acti-

vación aX (siendo $X \in \{R, L\}$ según derecha o izquierda) o sino detecta, lo cual incrementará la variable de desactivación dX .

De modo que se crea un flujo de activación y desactivación en función de la detección, de esta manera se activa el filtro de Kalman después de un número de frames fijado detectando líneas adecuadas en el espacio de Hough, para ajustarlo en la medida de lo posible a las mediciones tomadas hasta el momento y poder hacer uso de estos filtros para mejorar la robustez del resultado. De la misma manera, se desactiva para evitar el uso de la predicción de Kalman cuando se ha dejado de detectar líneas durante una cantidad de frames fijadas.

6.2. Tracking de ancho de carretera

A parte de los filtros de Kalman que predicen líneas en coordenadas polares, también resulta útil realizar un tracking de la anchura del carril de la vía actual por la que circula el vehículo, de esta forma se tiene otra referencia más para detectar outliers o elegir mejor entre una detección o la predicción de Kalman, para situaciones más ambiguas que se resuelvan cómodamente solo mirando la anchura del carril.

Esta anchura se mide en píxeles actualmente, además de contar con un sistema de activación y desactivación similar al filtro de Kalman, para solventar situaciones en las que se ha podido cambiar de vía y su ancho y así evitar un overfitting que pueda dar problemas en zonas urbanas, donde los anchos de vía varían fácilmente.

De la misma manera que el filtro de Kalman, su activación depende del caso en posibles detecciones, donde se detectan ambas líneas, ya que el ancho depende de conocer ambas líneas. En caso de tener coherencia esta detección, se añade a la media calculada hasta el momento y después de varias detecciones parcialmente consecutivas se activa para ayudar en la selección de líneas. Si deja de detectar ambas líneas durante un número de frames determinado, se desactiva y se reinicia el cálculo a espera de activarse nuevamente.

6.3. Tracking del punto de fuga

Otro cálculo de soporte muy interesante es hacer un tracking del punto de fuga del carril detectado, por lo que análogamente al ancho de carretera, contará con un sistema de activación y desactivación que dependerá de las mismas condiciones, además de tener una tolerancia al error alto debido a que el punto de fuga de la carretera resulta muy cambiante y sensible a curvas, subidas y bajadas.

El uso dado a este tracking es para evitar outliers que no guardan coherencia con el ángulo en su coordenada polar, ya que casualmente puede tener coherencia con el ancho de la carretera, pero su progresión tener un ángulo que quede muy alejado del punto de fuga promedio.

6.4. Implementación de técnicas de tracking

La implementación de las técnicas de tracking explicadas, se realiza en el código del detector de líneas de carril mediante una función, que trata de contemplar todas las posibles situaciones a la hora de determinar la selección final de la línea.

Icono	Caso	Descripción
<i>Detección en el espacio de Hough</i>		
	Ambas líneas	Detección de dos líneas con alta probabilidad de corresponder a las líneas de carril.
	Línea derecha	Línea derecha detectada con alta probabilidad de corresponder a la línea de carril.
	Línea izquierda	Línea izquierda detectada con alta probabilidad de corresponder a la línea de carril.
	Ninguna línea	No se ha detectado ninguna línea candidata.
<i>Posibilidades con los filtros de Kalman</i>		
	Ambas líneas	Ambas líneas detectadas en el espacio de Hough encajan con la predicción del filtro de Kalman.
	Línea derecha	Solo la línea derecha encaja con la predicción de Kalman.
	Línea izquierda	Solo la línea izquierda encaja con la predicción de Kalman.
	Ninguna línea	Ninguna línea encaja con la predicción de Kalman.
<i>Estado de los filtros de Kalman</i>		
	Ambos filtros Activos	Con ambos filtros activos se podrán dar cualquiera de las situaciones descritas en posibilidades con los filtros Kalman.
	Filtro derecho activo	Solo el filtro derecho está activo, limitando los casos de posibilidades al lado derecho y ninguno.
	Filtro izquierdo activo	Solo el filtro izquierdo está activo, limitando los casos de posibilidades al lado izquierdo y ninguno.
	Ningún filtro activo	No se puede dar ningún caso de posibilidades con filtros Kalman.

Tabla 1 - Variables de estado principales que determinan las situaciones posibles si se combinan entre ellas.

En la tabla 1, se muestran las principales variables de estado que determinan las diferentes situaciones que se puede encontrar la función. Cada caso concreto se genera de la combinación de estas variables de estado. Siempre se parte de la detección obtenida en el espacio de Hough, dándose cualquiera de los cuatro casos ya visto y a partir de este punto, según los filtros activos se procederá a las posibilidades con estos filtros, por lo que las variables de estados del filtro de Kalman guardan relación directa con las posibilidades al ver si encajan con las del espacio de Hough. Por ejemplo si están ambos filtros activos y el

espacio de Hough detecta ambas líneas, puede darse cualquier posibilidad con los filtros.

Es necesario contemplar cada caso, pues responde e influye de manera diferente a la hora de seleccionar el resultado final. Los posibles resultados finales pueden observarse en la Figura 13.

Posibles outputs

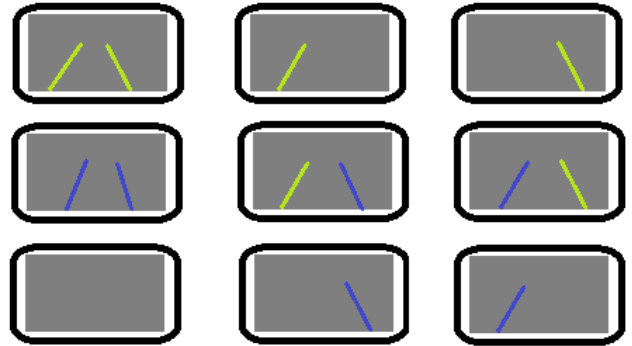


Fig. 13- Casos posibles de selección de línea como resultado final, siendo las líneas verdes las obtenidas en el espacio de Hough y las líneas azules la predicción de los filtros de Kalman.

Una vez contemplado un caso, combinación de las variables comentadas, solo falta determinar mediante una función de decisión, que con ayuda de los tracking de ancho de carretera y punto de fuga seleccionará el output final de resultado. Esta función de decisión, recibirá según el caso las líneas detectadas y las líneas predichas con Kalman, medirá distancia respecto a la media de ancho de carretera y medirá el punto de fuga y lo comparará, quedándose con el resultado que mejor se adapte al tracking realizado.

Para el caso en el que no están activos los filtro de Kalman, el resultado es el obtenido en el Hough tal y como se puede observar en la Figura 14.

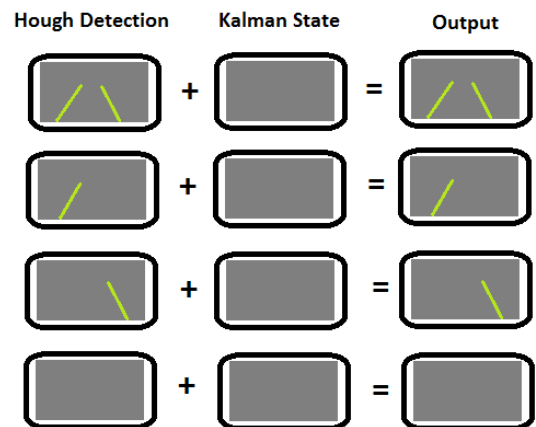


Fig. 14- outputs para los casos en los que los filtros de Kalman no están activos.

La Figura 15, muestra los casos en que se detectan ambas líneas en el espacio de Hough, dependiendo de los filtros activos y como encajan los resultados.

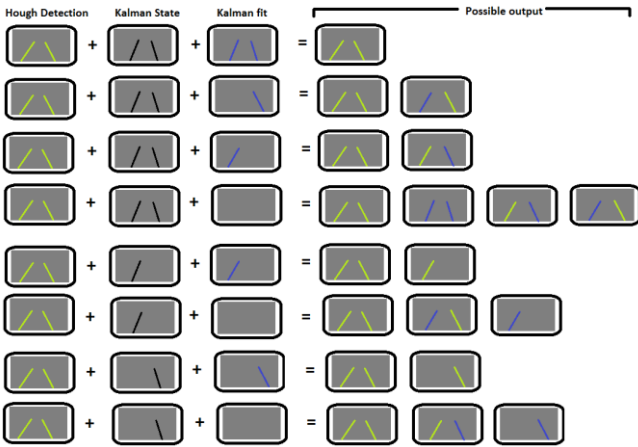


Fig. 15- Posibles salidas ante detección de ambas líneas en el espacio de Hough.

Para los casos que solo se detecta una de las dos líneas en el espacio de Hough, los posibles output se muestran en la Figura 16.

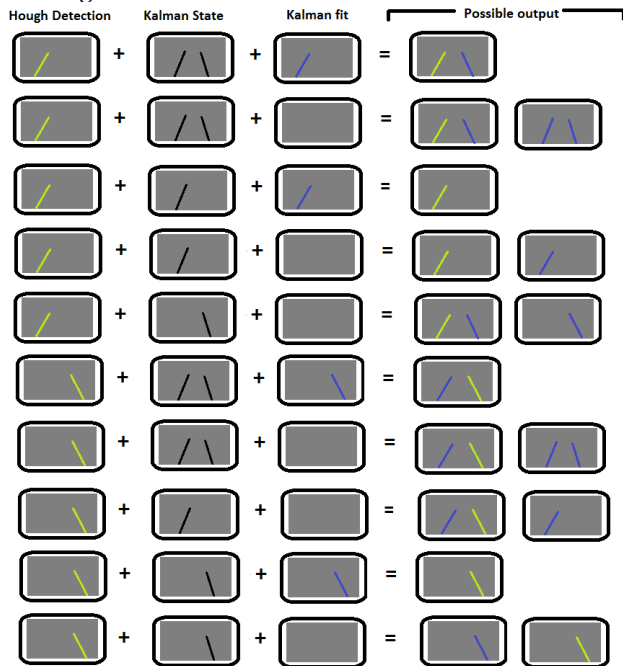


Fig. 16 - Posibles salidas ante detección en espacio de Hough de una sola línea.

Finalmente, para los casos en que no se detecta ninguna línea en el espacio de Hough, Figura 17.

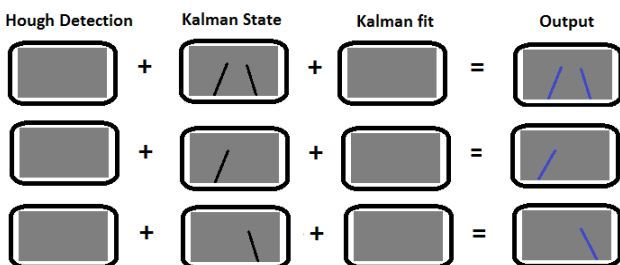


Fig. 17 - Salidas ante ninguna detección en el espacio de Hough.

Los casos de output múltiples observados en la Figuras 7 y 8, se deben a que este último paso lo decide la función

de decisión final ya comentada, que determina si quedarse con la predicción del Kalman, con las líneas detectadas en el espacio de Hough o ninguna.

Entrando en más detalle sobre los cálculos que realiza esta función, dependiendo de la entrada, siempre recibe dos líneas de un mismo lado como parámetro de entrada, la de Kalman y Hough y se empareja con la línea del lado contrario, ya sea solo Hough, Kalman o ambas. De este modo se emparejan las posibles combinaciones de resultados finales y se selecciona el que mejor se ajusta a las medias de los tracking realizados, siempre priorizando las líneas de la predicción de Kalman en caso de duda.

7. FUNCIONES ADICIONALES

7.1. Cambio de carril

Una de las funciones adicionales implementadas, ha sido la detección de cambio de carril. La idea de este módulo es detectar cuando el vehículo está haciendo un desplazamiento de carril (de ser posible), para ello se monitoriza la posición de la detección de líneas respecto al centro, cuando se detecta un movimiento en una dirección y una de las líneas se aproxima al centro y la otra se aleja, se considera un desplazamiento hacia la línea que se va aproximando.

Para que el módulo funcione, necesita detectar el carril entero o al menos la línea que se traspasará en el desplazamiento, puesto que se necesita una detección para determinar la posición respecto al centro, si varía o no. Por lo tanto, se realiza un seguimiento del movimiento del vehículo y si se detecta durante una cantidad definida de frames, un desplazamiento continuo hacia uno de los lados, se activará el cambio de carril hacia el lado correspondiente, junto a las implicaciones que esto conlleva.

Las implicaciones que conlleva el desplazamiento de carril es por una parte, indicar en la parte superior mediante una flecha hacia donde se está realizando el desplazamiento. Por otro lado, desactivar los filtros de Kalman, ya que daría mediciones erróneas durante el desplazamiento. Finalmente, permitir que la línea que se traspasa siga siendo detectada y mostrada en el resultado y ver cómo cambia del lado derecho al izquierdo durante la secuencia de cambio de carril.

7.2. Detección del tipo de línea

Otra función adicional implementada es determinar si la línea que se está detectando es continua o discontinua.

Para ello se utiliza la recta correspondiente a la línea detectada y se hace un seguimiento por ella en la imagen de crestas, buscando si hay o no continuidad en los píxeles en blanco correspondientes a la línea. El barrido de búsqueda en la línea se hace con un tamaño de 5 píxeles, siendo el centro el punto donde pasa la recta y el resto un margen de tolerancia por si la detección no es óptima pasando por encima de los puntos o si la línea no es completamente recta. De modo que la suma de los píxeles y las agrupaciones de píxeles consecutivos y espacio negro determinarán si la línea es continua o no.

Actualmente, el algoritmo determina en un frame que la línea es continua cuando más del 60% del barrido ha

detectado píxeles blancos correspondiente a la línea. Este número es debido a la imagen de crestas pierde eficacia conforme aumenta la distancia, además de que una línea continua no es perfecta en todos los píxeles, por lo que durante el barrido no se espera que el 100% detecte la línea continua. Para la línea discontinua, en caso de no cubrir el 60% definido antes, se mira si hay un segmento suficientemente grande de línea seguido de un segmento sin ella, de esta manera se presupone una discontinuidad en la línea.

Como el cálculo por cada frame está muy ligado a la imagen de crestas y es susceptible al ruido, las mediciones pueden ser erróneas en momentos puntuales. Debido a esto

8. EXPERIMENTOS Y RESULTADOS

Los experimentos realizados en este apartado tienen como objetivo demostrar la utilidad de las técnicas de alto nivel utilizadas. Mostrando como se hace uso de los filtros de Kalman y de los otros módulos de tracking adicionales del ancho de carretera y el punto de fuga, para obtener resultados robustos ante ciertas situaciones.

Para ello es necesario una secuencia de imágenes que muestren como se activan y desactivan estos sistemas y dejen claro su funcionamiento en la imagen final de resultado. Es por ello que las pruebas realizadas a continuación serán siempre una secuencia de imágenes, donde aparecerá en rojo la línea que da como resultado el programa, en azul la línea predicha por el filtro de Kalman y en verde la detección de Hough.

A continuación se muestran los experimentos realizados, en primer lugar con el dataset correspondiente al campus de la UAB.

8.1. Pruebas en Campus UAB

El dataset correspondiente al campus de la UAB cuenta con 12.996 imágenes, tomadas con la cámara ZED a una resolución de 1280x720 píxeles, recortadas a 1280x576. Se disponen de videos que muestran la secuencia como si se estuviera circulando y detectando las líneas de carril, pero para mostrar el experimento se realizan tres test con pequeñas secuencias que mostrarán el funcionamiento de lo implementado.

En este primer test, el objetivo es mostrar cómo se realiza la activación del filtro de Kalman, tanto para el carril izquierdo como el derecho a un valor inicial prefijado y como este a medida que se recalcula con nuevas mediciones se va ajustando a lo que visualmente corresponde el resultado. La figura 18 muestra esta progresión, en el primer frame se activa el filtro de Kalman para el carril izquierdo y conforme pasan los frames este se va ajustando al resultado obtenido del espacio de Hough. En el cuarto frame sucede lo mismo con el filtro de Kalman del carril derecho.

8.2. Pruebas en autopista de día

Para este otro experimento, se dispone de un dataset de imágenes de la autopista AP7, con un total de 14.000 imágenes tomadas por la mañana con la cámara IDS a una resolución de 960x480 píxeles y 25 frames por segundo.

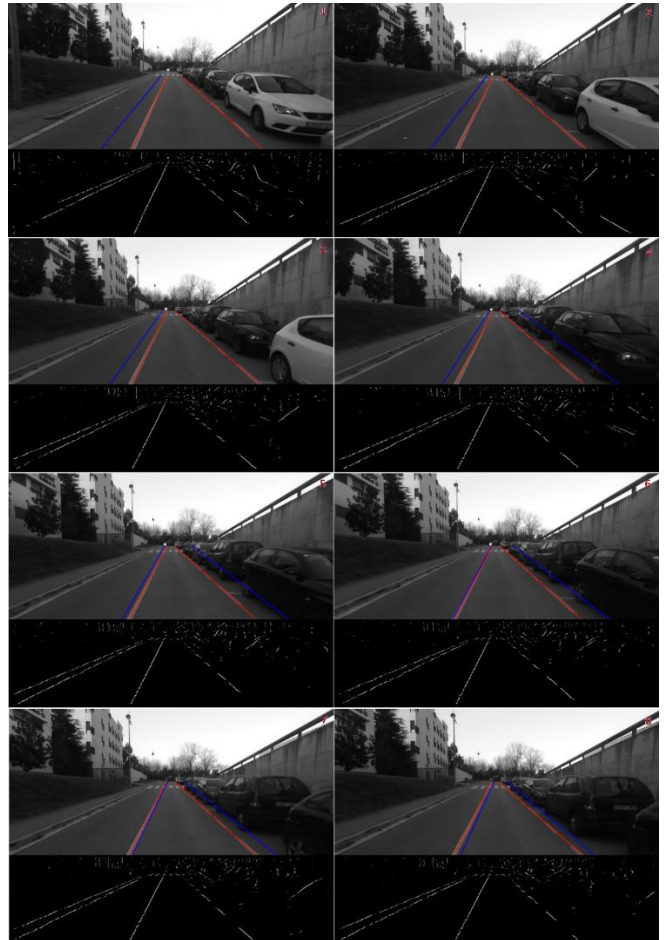


Fig. 18 – Secuencia de 8 imágenes que muestran la activación de las líneas correspondiente a la predicción del filtro de Kalman inicializadas a un valor previo no ajustado a la vía y como a medida que progresa se va ajustando a la detección.

Para este test en autopista, la idea es mostrar la robustez del detector cuando se está circulando por el mismo carril, aunque las líneas sean discontinuas y haya frames donde no se detectan en el espacio de Hough. Si se observa la Figura 19, se muestra la detección en todos los frames con ayuda de los filtros de Kalman.

8.3. Resultados

Los resultados obtenidos en las pruebas, son visualmente correctos. Por lo general el detector en autopista, con los filtros de Kalman activos y manteniéndose en el carril tiene más de un 95% de acierto (desde un punto de vista cualitativo). Siendo las situaciones de cambio de carril la que provoca algunas detecciones erróneas debido al uso del Kalman y su predicción hasta que se desactiva. Por otro lado hasta que se reactivan estos filtros, las líneas discontinuas, hay momentos que puede no haber detección.

Desafortunadamente, no se dispone del antiguo código de detección ni de datasets de resultados, para realizar una comparativa directa contra el actual detector realizado. Tampoco se dispone de una base de datos con imágenes anotadas que tengan en cuenta las líneas de carril, para determinar cuan efectivo es el detector. Por lo que resulta muy complicado realizar un análisis cuantitativo.

Aun así, se puede realizar un análisis cualitativo, tal y como se ha ido haciendo en las pruebas mostradas en las figuras 18 a la 26.

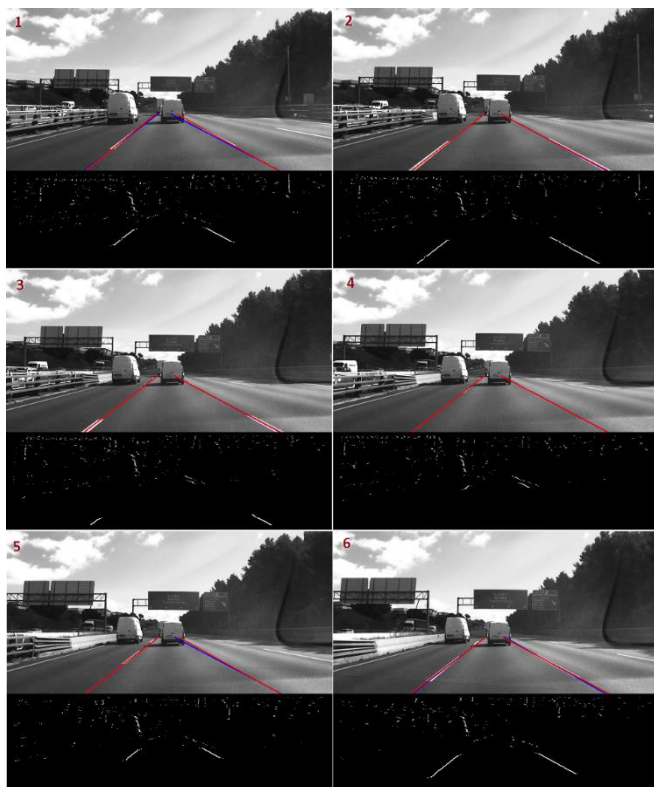


Fig. 19- Secuencia de 6 imágenes que muestran la circulación por un carril en autopista y como se detectan las líneas discontinuas con ayuda del filtro de Kalman.

9. CONCLUSIONES

Detectar las líneas de carril es muy importante porque puede usarse para asistir en la conducción y mejorar la seguridad en las vías, evitando que el vehículo se salga del carril. Además, es un componente indispensable en la conducción autónoma, ya que marca el camino por donde circular el vehículo.

Existen diferentes trabajos en la literatura que abordan la detección de líneas como ya se ha visto, enfocando este trabajo en mejorar [1]. El descriptor de crestas disponible en un código no funcional, ha sido reimplementado en un sistema moderno con librerías actuales. A fin de mejorar la base disponible en [1], se han utilizado diferentes métodos. Se ha aplicado el algoritmo de Hough, de OpenCV, a la imagen de crestas para obtener las posibles rectas. Mediante otro algoritmo, se han seleccionado las líneas de carril candidatas. En general, se han introducido mejoras sobretudo en la lógica de alto nivel, utilizando técnicas de tracking como los filtros de Kalman, que son un gran apoyo al permitir predicciones de líneas ante situaciones desfavorables. Estas mejoras han permitido obtener un resultado robusto como se ha podido observar en los análisis cualitativos realizados.

Además de las mejoras, se han añadido funcionalidades adicionales, que podrían resultar muy útiles como son la detección del tipo de línea de carril (continua o

discontinua) y la detección de cambio de carril, siendo esta primera mejora, importante para determinar si el vehículo puede o no realizar un cambio de carril al saber qué es una línea discontinua. La segunda mejora, resultaría muy útil para saber si el vehículo realiza un cambio de carril, la dirección y de esta manera poder activar o desactivar otros sistemas que ayuden a este propósito.

Finalmente, se ha cumplido el objetivo principal, pero todavía está abierto a mejoras, como pueden ser adaptarlo a cámaras estereoscópicas y aprovechar las funcionalidades adicionales que aportan. Aún queda implementar la detección de curvas, para que pueda responder ante más situaciones. Es por eso que se ha realizado el código adecuadamente para que pueda mejorarse en un futuro y sea fácil de comprender.

AGRADECIMIENTOS

A todo el equipo ADAS del CVC, en especial mención a David Vázquez, Gabriel Villalonga y Antonio López, que me han ayudado con conceptos y funcionalidades adicionales utilizadas para la realización de este proyecto. Y finalmente a mi compañero Jose Escribano, que realizaba otro módulo diferente de detección de espacio libre de carretera, pero nos ayudábamos mutuamente.

BIBLIOGRAFÍA

- [1] López, A., Serrat, J., Saludes, J., Cañero C., Lumbreras F., Graf T. Ridgeness for Detecting Lane Markings.
- [2] Fundación Instituto Tecnológico para la Seguridad del Automóvil, Sistema de Alerta de Cambio Involuntario de Carril [online]: http://www.nextautomotivesafety.com/pdf/cambio_de_carril.pdf.
- [3] Narathip Thongpan and Mahasak Ketcham. The State of the Art in Development a Lane Detection for Embedded Systems Design.
- [4] Aharon Bar Hillel, Ronen Lerner, Dan Levi, Guy Raz. Recent progress in road and lane detection: a survey.
- [5] Jinghong XU, Changsheng CHENG and Yan WAN. The Research of Lane Marker Detection Algorithm Based on Inverse Perspective Mapping.
- [6] López, A., Serrat, J., Saludes, J., Cañero C., Lumbreras F. Robus Lane Lines detection and quantitative assessment.
- [7] López, A., Serrat, J., Saludes, J., Cañero C., Lumbreras F., Graf T. Robust Lane Markings Detection And Road Geometry Computation
- [8] C++ documentation [online]: <http://www.cplusplus.com/doc/>
- [9] Doxygen documentation [online]: <http://www.stack.nl/~dimitri/doxygen/>
- [10] Richard O. Duda and Peter E. Hart. Use of the Hough Transformation To Detect Lines and Curves in Pictures.
- [11] Hough Line Transform, OpenCV v2.4 documentation [online]: http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/hough_lines/hough_lines.html
- [12] Muhammad Asif, Mohd Rizal Arshad, Muhammad Yousuf Irfan Zia and Abid Yahya. An Implementation of Active Contour and Kalman Filter for Road Tracking.
- [13] Kalman filter Wikipedia [online]: https://en.wikipedia.org/wiki/Kalman_filter
- [14] Ramsey Faragher. Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation. IEEE SIGNAL PROCESSING MAGAZINE.

APÉNDICE

A1. PRUEBAS EN CAMPUS UAB (PARTE 2)

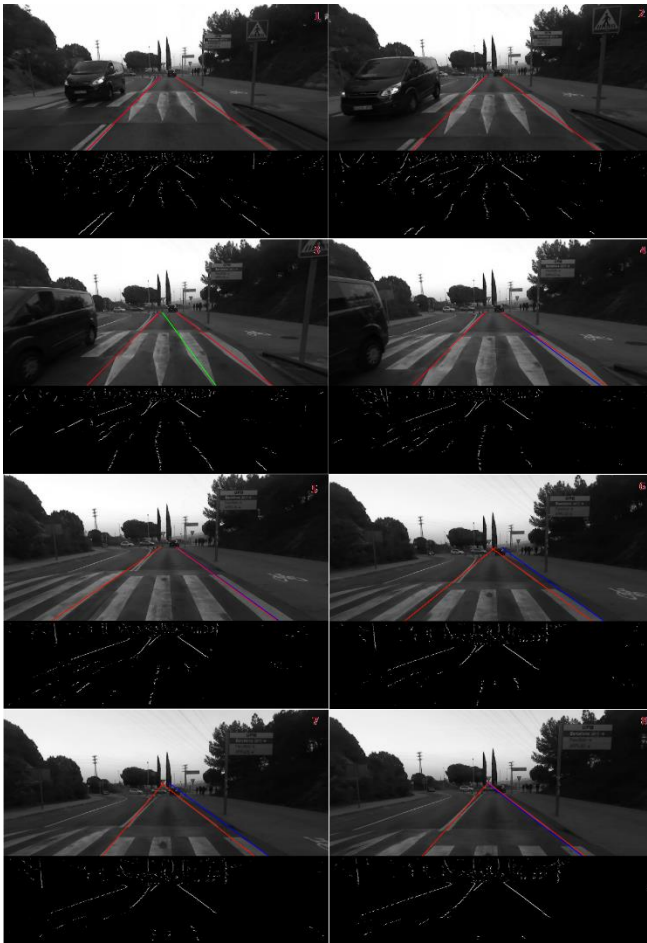


Fig. 20– Secuencia de 8 imágenes que muestra cómo se realiza la detección utilizando los filtros de Kalman y el tracking en una situación desfavorable en el espacio de Hough como es un paso de cebra con resalto. La detección final (líneas rojas) guarda coherencia con el ancho del carril detectado anteriormente.

Para este tercer test, Figura 20, se muestra cómo responde la predicción utilizando los filtros de Kalman, en una situación en que el resultado en el espacio de Hough daba muy malos resultados, el paso de cebra con resalto, este tiene líneas muy gruesas dibujadas y se pierde totalmente las líneas en la imagen de crestas.

En los tres primeros frames, no se observa detección en el espacio de Hough y se usa directamente la línea que predicen los filtros de Kalman, manteniendo un trazo de la línea muy coherente con la imagen y evitando outliers como se puede ver en el frame 3. En los siguientes frames ya se detecta la línea derecha en el espacio de Hough y vuelve a aparecer la línea azul representando el filtro de Kalman ajustándose. Para la izquierda se sigue usando el Kalman ya que no se encuentra línea adecuada todavía en el espacio de Hough, por ese motivo tiene alguna desviación pero se mantiene coherente con el carril.



Fig. 21– Secuencia de 8 imágenes correspondientes al campus de la UAB. Dónde se muestra la detección final en rojo, la predicción del filtro de Kalman en azul y el resultado del espacio de Hough en verde. En esta secuencia se muestra como se utiliza el filtro de Kalman ante la falta de detección del lado derecho y como se evita un outlier detectado en un coche en el frame 7.

Como se puede observar en la Figura 21, este segundo test es la continuación no consecutiva del primer test, ya parte con los filtros de Kalman ajustados (líneas azules) junto a la detección (líneas rojas), nótese que si se observan ambos tipos de línea rojo y azules, es que se están eligiendo las líneas del espacio de Hough (líneas verdes) como resultado final y estas no serán visibles.

En los frames 3 y 4, no aparece la línea de Kalman azul en el lado derecho, debido a que se está seleccionando esta línea como resultado final, ya que si se observa la imagen de crestas, justo debajo de la imagen para ambos frames el lado derecho no detecta línea adecuadamente y el algoritmo elige en este caso el valor predicho por el filtro de Kalman, que en la imagen se observa cómo encaja adecuadamente como una detección correcta.

En el frame 7, se observa un claro outlier producido por el coche del lado derecho, que engañaría al detector en el espacio de Hough como si fuera una línea buena, pero gracias al tracking del punto de fuga y del ancho de carril realizado, esta línea carece de sentido y por lo tanto se elige la línea predicha por el filtro de Kalman.

A2. PRUEBAS EN AUTOPISTA DÍA (PARTE 2)

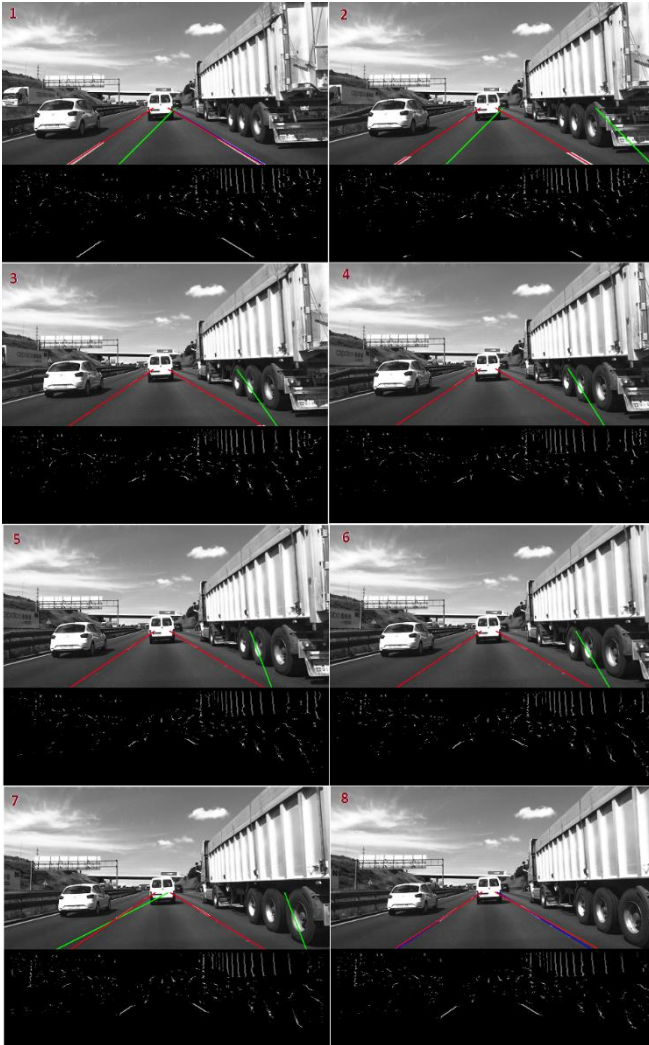


Fig. 22- Secuencia de 8 imágenes que muestra la detección en autopista, ante una situación que provoca muchos outliers, como es el adelantamiento de camiones en autopista.

Este segundo test muestra un caso concreto que se da en la autopista, cuando se realizan adelantamientos a vehículos de gran tamaño como camiones, en la imagen de crestas aparece ruido que provoca outliers en la detección del espacio de Hough, reflejándose en la secuencia de la Figura 22.

En todos los frames aparecen lecturas erróneas en el espacio de Hough, que son las líneas en color verde. Ante estas situaciones, se observa como los filtros de Kalman realizan su trabajo adecuadamente manteniendo la detección final de manera coherente en el margen de las líneas discontinuas del carril.

A3. PRUEBAS EN AUTOPISTA ANOCHECER

El siguiente experimento se realizó en imágenes tomadas de la autopista AP7, al anochecer con la cámara IDS, con una resolución de 960x480 píxeles y un frame rate aproximado de 33.

La condición de visibilidad en las imágenes es realmente baja, ya que había poca luz al anochecer, por lo que las secuencias que se mostrarán en los test tienen el brillo muy aumentado para poderse visualizar adecuadamente. Mediante estos test se pretende mostrar como la imagen de crestas responde bien a condiciones de poca visibilidad y el detector sigue funcionando. No se entrará en detalle ya que el comportamiento es similar al ya visto en las imágenes de día.

Las figuras 23 y 24, muestran 2 ejemplos de secuencias con poca visibilidad y como responde bien el algoritmo de crestas y las detecciones se hacen adecuadamente.

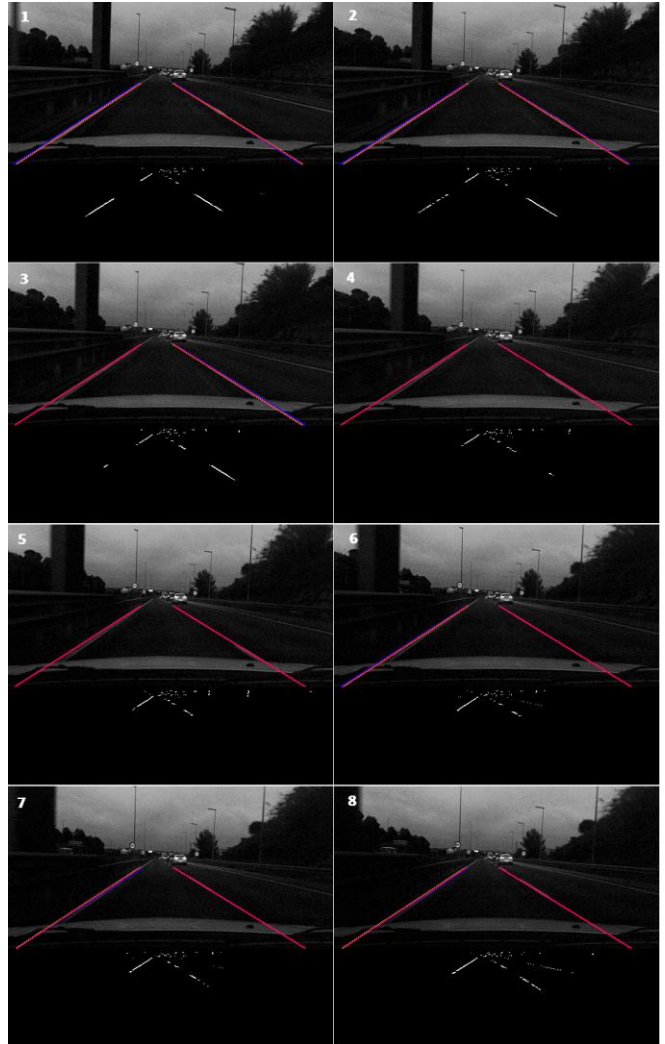


Fig. 23 - Secuencia de 8 imágenes tomadas al anochecer, donde se está pasando por un bajo puente en ese instante. El resultado no se ve afectado y mediante la ayuda de los filtros de Kalman es adecuado.

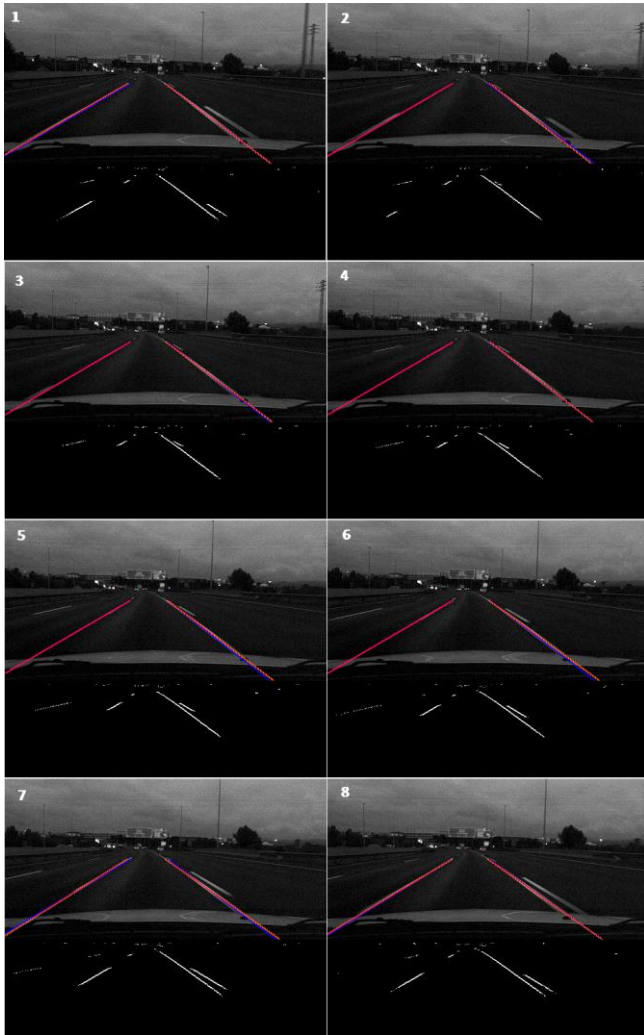


Fig. 24 - Secuencia de 8 imágenes tomadas al anochecer en la autopista AP7, donde aparece una doble línea continua discontinua y la imagen de crestas ve nítidamente esto, por lo que se detectan las líneas adecuadamente.

A4. CAMBIO DE CARRIL (PARTE 2)

En este experimento se muestra cómo funciona el módulo de detección de cambio de carril, mostrando con una flecha en la parte superior central hacia dónde se realiza el cambio.

Los resultados pueden observarse en la figura 25, donde se realiza un cambio de carril hacia la izquierda, traspasando una doble línea, la cual queda detectada la continua en este caso. Conforme se va desplazando el vehículo se va realizando la detección de la línea y su progresión. Cuando finaliza el desplazamiento y se detecta el carril nuevamente con ambas líneas, acaba la detección y deja de mostrarse la flecha que indica el desplazamiento, como puede observarse en el frame 8.

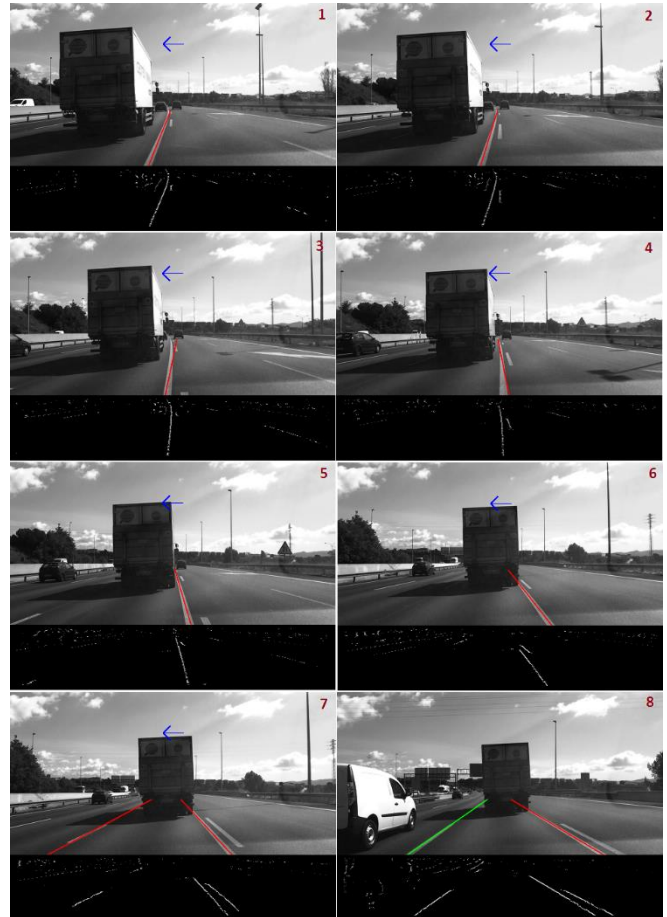


Fig. 25 - Secuencia de 8 imágenes que muestran el cambio de carril y como el módulo detecta que se está efectuando un cambio hacia la izquierda, indicado por la flecha azul.

A5. Detección de tipo de carril



Fig. 26 - Muestra de diferentes imágenes donde puede observarse el tipo de detección realizada en cada una, siendo la línea en rojo línea continua y la línea verde discontinua. En la imagen de crestas puede observarse con mayor calidad si la línea es continua o no.