ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Trabajo Fin de Master

Diseño de un Sistema Avanzado de Asistencia al Conductor (ADAS) basado en Visión Artificial para predicción del Tiempo-de-Colisión (TTC)

AUTOR: **LAENCINA ESCOBAR, RUBÉN**
DIRECTOR: **VERDÚ MONEDERO, RAFAEL**

Septiembre / 2016

Rubén Laencina Escobar

Diseño de un Sistema Avanzado de Asistencia al Conductor (ADAS) basado en Visión Artificial para predicción del Tiempo-de-Colisión (TTC)

Dirigido por: **Prof. VERDÚ MONEDERO RAFAEL**

Departamento de Tecnología de la Información y las Comunicaciones

Escuela Técnica Superior de Ingeniería de Telecomunicación

Proyecto Fin de Máster

Proyecto Desarrollado en el Marco del

PROGRAMA DE MOVILIDAD EUROPEO ERASMUS +

European Commission | Erasmus+

Rubén Laencina Escobar

Design of a Computer Vision based TTC (Time-To-Collision) ADAS (Advanced Driver Assistance System)

Directed by: **Prof. ATHANASSIOS SKODRAS**

University of Patras, Department of Electrical and Computer Engineering

Co-directed by: **Dr. GEORGE SIOGKAS**

Valeo Vision Systems, Research and Development Department

University of Patras, Greece
Valeo Vision Systems, Ireland

Master Thesis Dissertation

Dedicado a mi familia, y a los años por venir.

Abstract

A lo largo de las últimas décadas los accidentes de tráfico han demostrado ser una de las causas más habituales de la pérdida de vidas humanas, contando estas por millones. En consecuencia, la seguridad de los vehículos ha crecido continuamente a lo largo de los años, principalmente gracias a avances en sistemas de seguridad pasivos, como cinturones de seguridad o airbags, incluidos actualmente en prácticamente todos los vehículos comerciales. Sin embargo, los sistemas de seguridad más actuales están diseñados no solo para minimizar las posibles lesiones en caso de accidente, sino para en primera instancia evitar que estos sucedan.

Los Sistemas Avanzados de Asistencia al Conductor, conocidos como ADAS por sus siglas en inglés (Advanced Driver Assistance Systems) son sistemas que como su propio nombre indica proporcionan asistencia al conductor cuando éste se encuentra en carretera, mejorando así su experiencia de conducción, siendo su función principal garantizar la seguridad del vehículo propio y otros vehículos en la carretera, así como la del conductor y peatones o ciclistas. La demanda de Sistemas ADAS ha experimentado un gran auge a lo largo de los últimos años consecuencia del deseo generalizado de construir vehículos y carreteras más seguras con el fin de reducir al mínimo el número de muertes en éstas. Para funcionar de forma fiable, los sistemas ADAS deben ser capaces de reconocer objetos, señales de tráfico, la propia carretera así como cualquier otro tipo vehículos en movimiento en ésta con el fin de poder tomar decisiones en tiempo real, ya sea para advertir al conductor o para directamente actuar en nombre de éste. De este modo, a fin de garantizar sistemas robustos y de confianza, los sistemas ADAS comerciales normalmente combinan diferentes tecnologías como radar, LIDAR, cámaras, o dispositivos de visión nocturna, que aseguren la integridad y seguridad del vehículo, conductor, pasajeros, y peatones. Los sistemas ADAS actuales representan la antesala de los vehículos autónomos del futuro.

El objetivo principal del presente Trabajo de Fin de Máster consiste en el diseño de uno de dichos sistemas Sistema ADAS, y más en concreto de un sistema TTC (Time-To-Collision) capaz de predecir el tiempo restante para producirse un posible impacto entre el vehículo propio y otros presentes en la carretera, a partir de imágenes adquiridas en tiempo real procedentes de una cámara instalada en el interior del vehículo mediante el empleo de técnicas de Visión Artificial. En última instancia, el sistema propuesto podría formar parte real de un sistema anticolisiones CAS (Collision-Avoidance System) capaz de evitar potenciales situaciones de riesgo directo de accidente en es escenarios de tráfico reales donde la integridad de vidas humanas podría verse comprometida.

A tal fin el sistema diseñado propone un modulo de Detección de Vehículos basado en Filtrado de Gabor y clasificación SVM para caracterización y clasificación de vehículos. Posteriormente, seguimiento espacio-temporal de los resultados obtenidos es llevado a cabo mediante Filtrado de Kalman, condición necesaria que permite en última instancia acometer la estimación del Tiempo-de-Colisión (TTC) que este proyecto persigue. Los resultados obtenidos en este punto dependen en gran medida de los resultados obtenidos en las etapas anteriores. Diferentes métodos tanto cuantitativos como cualitativos han sido considerados a lo largo del proyecto para evaluación del sistema propuesto.

Agradecimientos

Mis más sinceros agradecimientos en lo que a este Trabajo de Fin de Máster se refiere van dirigidos sin ningún tipo de distinción a su director y co-director en la Universidad de Patras y Valeo Vision Systems respectivamente, Prof. Athanassios Skodras y Dr. George Siogkas.

Gracias Prof. Skodras por haber creído y confiado en mí como estudiante Erasmus+ en tu universidad para llevar a cabo contigo lo que al final ha resultado ser un reto personal y una bonita experiencia que va más allá de cualquier ámbito académico o universitario. Gracias por haberme abierto las puerta de tu despacho en Patras y haberme tratado desde el principio como a uno más de tus estudiantes. Sin tu amabilidad y apoyo nada de esto habría sido posible.

Y gracias George. Muchas gracias por tu constante ayuda, disponibilidad, y sabio consejo siempre que lo necesité. Gracias por tus esfuerzos por hacer esto posible a pesar de la distancia. Y definitivamente gracias por el tiempo empleado en frente de la pantalla. Tu sincero consejo nos ha permitido llegar con éxito hasta aquí.

En último lugar me gustaría agradecer al Prof. Rafael Verdú Monedero todo su tiempo, esfuerzo y gran facilidad ofrecida desde el primer momento en que contacté con él como director del proyecto en la Universidad Politécnica de Cartagena para que todo esto fuera posible.

Introducción

Este Trabajo de Fin de Máster surge como requerimiento final para la obtención del Título de Máster Universitario en Ingeniería de Telecomunicación ofrecido por la Universidad Politécnica de Cartagena (UPCT), Cartagena, España.

Titulado en castellano *"Diseño de un Sistema Avanzado de Asistencia al Conductor (ADAS) basado en Visión Artificial para predicción del Tiempo-de-Colisión (TTC)"*, y con título original en inglés *"Design of a Computer Vision based TTC (Time-To-Collision) ADAS (Advanced Driver Assistance System)"*, este proyecto ha sido desarrollado en su totalidad en el marco del Programa de Movilidad Europea Erasmus+, concretamente en la Universidad de Patras (UOPA), Patras, Grecia, en el Departamento de Ingeniería Eléctrica e Informática (E&CE), y ha estado dirigido por el Professor Athanassios Skodras. Asimismo, desarrollado en colaboración con Valeo Vision Systems (VVS), Galway, Ireland, este proyecto ha sido co-dirigido por Dr. George Siogkas perteneciente al Departamente de Investigación y Desarrollo.

Como tercera universidad más grande de Grecia, la Universidad de Patras celebró en 2014 su 50 aniversario de funcionamiento académico. Hoy en día goza de reconocimiento como Institución Académica con impacto en todo el mundo, atractiva para miles de estudiantes y un gran número de personal académico y de investigación partícipe activamente en proyectos de última generación, innovación, y excelencia. Formada por 5 escuelas y veinticuatro departamentos, E&CE abarca las áreas educativas e investigación relacionadas con Electricidad, Telecomunicaciones y Tecnologías de la Información, Electrónica y Sistemas Informáticos, y Control Automático. Allí, los intereses actuales del Prof. Skodras se concentran en campos como la codificación de video e imagen, protección IPR mediante marcas de agua digitales, algoritmos basados en transformada rápida, procesamiento de señales digitales en tiempo real, y aplicaciones multimedia.

Por otro lado, Valeo es un proveedor de soluciones tecnológicas en el sector del automóvil socio de fabricantes como BMW, Land Rover o el Grupo Volkswagen a lo largo de todo mundo. Ccon 136 centros de producción, 16 centros de investigación, 34 centros de desarrollo, 15 plataformas de distribución, y en torno a 81.800 empleados en 29 países, el Grupo Valeo se sitúa como proveedor líder mundial de soluciones innovadoras y sistemas avanzados de asistencia a la conducción en el campo de la automoción.

Como empresa tecnológica Valeo Vision Systems se especializa en el diseño y fabricación de sistemas ADAS incluyendo cámaras de visión trasera, y sistemas de localización y visión panorámica 360º. Sus aplicaciones incluyen detección de peatones, y sistemas de alerta de tráfico y aparcamiento automatizado. Y sus productos y líneas de trabajo futuro se centran en el desarrollo de aplicaciones para hacer de la conducción autónoma una realidad. En última instancia VVS persigue mejorar la seguridad y la experiencia de conducción, ofreciendo soluciones tecnológicas intuitivas y de fácil uso e integración con el vehículo y el entorno, lo cual la coloca a la vanguardia de su sector.

En este contexto surge la idea de que da origen a este Trabajo de Fin de Máster, defendido el 29 de Septiembre de 2016 para la obtención del Título de Máster Universitario en Ingeniería de Telecomunicación impartido por la Universidad Politécnica de Cartagena (UPCT), Cartagena, España.

Contenido

Resumen del TFM en Castellano

Contenido

Lista de Figuras

Lista de Tablas

Capítulo I

Sistemas Avanzados de Asistencia al Conductor (ADAS)

1.1 Introducción

Los Sistemas Avanzados de Asistencia al Conductor, conocidos como ADAS por sus siglas en inglés (Advanced Driver Assistance Systems) son sistemas que como su propio nombre indica proporcionan asistencia al conductor cuando éste se encuentra en carretera, mejorando así su experiencia de conducción [1]. Su función principal es garantizar la seguridad del vehículo propio y otros vehículos en la carretera, así como la del conductor y peatones o ciclistas. No obstante, su rango de aplicaciones es muy amplio ya que estos sistemas abarcan desde tareas más simples como la reducción del gasto de combustible del vehículo, hasta la posibilidad de evitar potenciales situaciones de riesgo y colisión en la carretera mediante la actuación directa sobre algún otro sistema del vehículo como la dirección o los frenos. En cualquiera caso, para funcionar de forma fiable, los sistemas ADAS deben ser capaces de reconocer objetos, señales de tráfico, la propia carretera así como cualquier otro tipo vehículos en movimiento en ésta con el fin de poder tomar decisiones ya sea para advertir al conductor o para directamente actuar en nombre de éste.

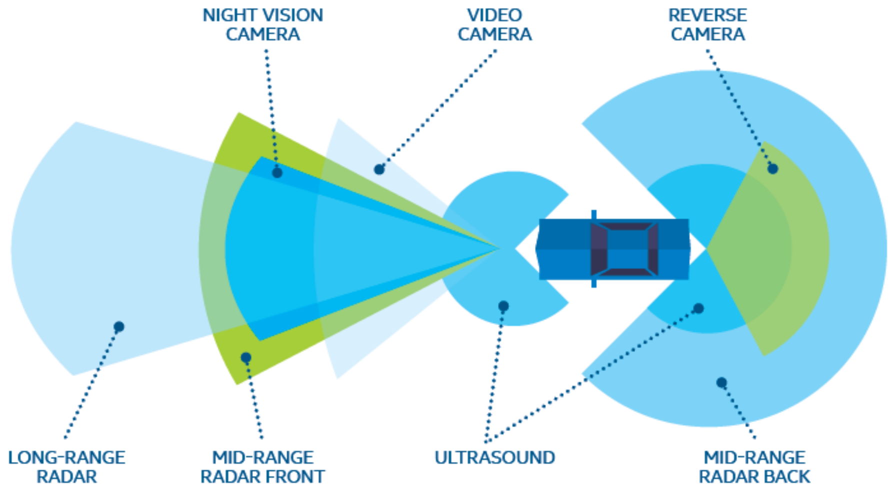


Figura 1.1: Ejemplo del conjunto de sensores de un ADAS

El aumento de la demanda de Sistemas ADAS experimentado en los últimos años en la sociedad encuentra su causa principal en el deseo generalizado de construir vehículos y carreteras más seguras con el fin de reducir el número de muertes en estas, aspectos a su vez ya presentes en la legislación de los países más importantes del mundo. De manera general los sistemas ADAS están o pueden estar compuestos por un lado de los siguientes sensores físicos: radar, LIDAR, basados en ultrasonidos, dispositivos PMD (photonic mixer devices), cámaras y dispositivos de visión nocturna, que dotan al vehículo de capacidad para monitorear en todas direcciones campos tanto cerca como lejos de éste, y por otro lado de algoritmos para la fusión de sensores que aseguran la integridad y seguridad del vehículo, el conductor, pasajeros, y peatones en base a factores tales como el tráfico, el clima, situaciones de riesgo, etc. Además, los sistemas ADAS actuales actúan en tiempo real a través de avisos al conductor o directamente mediante el accionamiento de otros sistemas de control del vehículo, siendo precursores de los vehículos autónomos del futuro.

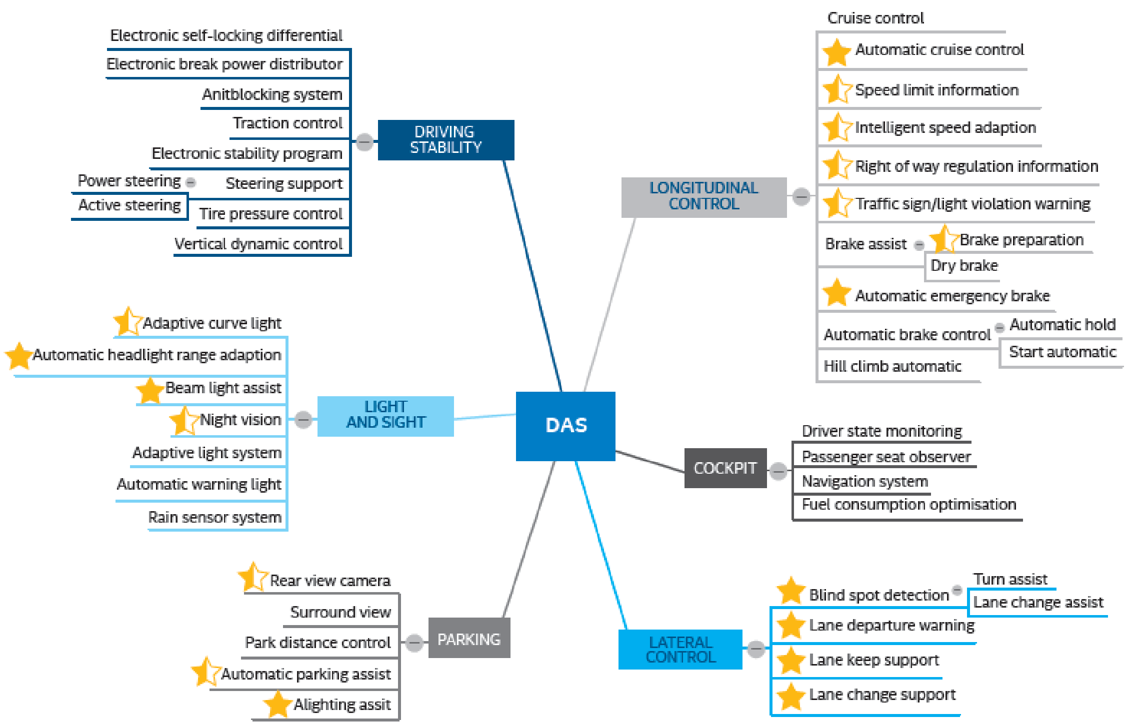


Figura 1.2: Funciones de un sistema DAS y ADAS (estrellas) respectivamente

Son varios los desafíos que presentan diseñar, desarrollar, implementar y poner en funcionamiento un sistema ADAS. De éste se espera tanto ser capaz de reunir información precisa del entorno, así como ser rápido en el procesamiento de ésta, predecir con precisión posibles situaciones en las que el vehículo se puede encontrar involucrado, y en última instancia reaccionar en tiempo real de acuerdo a éstas. Es fundamental además que sea robusto, fiable, y presente bajas tasas de error.

En esta línea, una gran cantidad de esfuerzo y trabajo de investigación ha sido llevado a cabo a lo largo de los últimos años en la industria tecnológica y del automóvil a fin de resolver todos estos problemas y ser capaces de desarrollar la tecnología necesaria para convertir a los sistemas ADAS y la conducción autónoma en una realidad. Asimismo, además de los requisitos funcionales que éstos presentan, los ADAS deben ser capaces de proporcionar altos niveles de seguridad ante posibles agentes externos con intenciones maliciosas cuyo objetivo es poner en peligro el sistema y causar accidentes catastróficos, que al margen de los daños materiales que ocasionan, podrían involucrar la pérdida de vidas humanas . De esta manera, dentro de los próximos años y ya en la actualidad, los Sistemas Avanzados de Asistencia al Conductor deben y son considerados como un requisito fundamental, en el proceso de diseño y fabricación de, como se señaló antes, los vehículos del futuro.

1.2 Objetivos de este TFM

En el contexto de este Trabajo de Fin de Máster, la Visión Artificial o *Computer Vision* es una disciplina científica que incluye métodos para adquirir, procesar, analizar y comprender las imágenes del mundo real con el fin de producir información numérica o simbólica para que puedan ser tratados por un computador. Tal y como los humanos usamos nuestros ojos y cerebros para comprender el mundo que nos rodea, la visión por computador trata de producir el mismo efecto para que las computadoras puedan percibir y comprender una imagen o secuencia de imágenes y actuar según convenga en una determinada situación, comúnmente adquiridas mediante una varias cámaras de video.

Cuando se trata de sistemas ADAS basados en este tipo de tecnología la realidad es que los más actuales vehículos comercial solamente han incluido este tipo de dispositivos en combinación con otros sensores o subsistemas más específicos cuya robustez y fiabilidad está lo suficientemente demostrada como para ser utilizado en escenarios y bajo condiciones reales, donde la seguridad humana se encuentra directamente involucrada, y por lo tanto los riesgo a asumir han de ser los mínimos posibles. No obstante, la tecnología más actual y los avances realizados en este campo durante los últimos años permiten imaginar la posibilidad real de implementar hoy en día uno de estos sistemas ADAS lo suficientemente fiable y robusto como para garantizar la seguridad de los ciudadanos haciendo uso únicamente de sistemas basados en visión artificial. A tal fin dicho sistema debería incluir, si no todos, la mayoría de las los siguientes módulos: reconocimiento de semáforos y señales de tráfico, detección de carretera y vehículos, detección de peatones, visión nocturna, detección de somnolencia del conductor, y detección de puntos ciegos.

De acuerdo con esto, el objetivo principal del presente Trabajo de Fin de Máster consiste en el diseño de uno de dichos sistemas Sistema Avanzados de Asistencia al Conductor ADAS [2] [3] [4], y más en concreto de un sistema TTC (Time-To-Collision) [5] [6] capaz de predecir el tiempo restante para producirse un posible impacto entre el vehículo propio y otros presentes en la carretera, a partir de imágenes adquiridas en tiempo real procedentes de una cámara instalada en el interior del vehículo mediante el empleo de técnicas de Visión Artificial [7] [8]. En última instancia, integrado en un Sistema Anticolisión (Pre-Crash System) o en combinación con un Sistema de Frenado de Emergencia (EBS, Emergency Braking System), el sistema propuesto podría dar lugar a la implementación real de un sistema anticolisiones CAS (Collision-Avoidance System) en última instancia capaz de evitar potenciales situaciones de riesgo directo de accidente de tráfico en escenarios reales de conducción en carretera en los que podría verse involucrada la integridad de vidas humanas.

Todas las posibles variaciones que se pueden dar en un proceso complejo como es el representado por una escena de tráfico en continuo cambio, donde incontables son las variables a tener en cuenta (véase el vehículo propio y otros en la carretera, peatones, ciclistas, motoristas, las condiciones de iluminaciones dependientes del pronóstico meteorológico, señales de tráfico y semáforos, etc) hacen de éste un exigente y complicado proyecto. En consecuencia, diferentes retos asociados al diseño, implementación y funcionamiento de cualquier sistema ADAS son afrontados en el contexto de esta tésis, ya que en última instancia el sistema propuesto ha de presenta una alta velocidad de procesado capaz en definitiva de trabajar y reaccionar en tiempo real, alta precisión de resultados, robustez, fiabilidad y baja tasa de errores.

A tal fin, durante el transcurso de este proyecto, la mayor parte, si no todos los subsistemas que conforman un TTC-ADAS, deberán ser cubiertos. Más en detalle dicho proceso supone: (1) Investigar y desarrollar un algoritmo de detección de vehículos; tarea dividida a su vez en dos etapas: una primera de Generación de Hipótesis (HG) de vehículos candidatos. Para ello se deberán aplicar diferentes técnicas de procesado de imagen tales como detección de bordes, clustering, morfología matemática, filtros de Gabor, etc; y una segunda etapa de Verificación de las Hipótesis (HV) propuestas por la etapa anterior a través de un algoritmo de clasificación de vehículos que permita determinar los vehículos detectados correctamente. Posibles enfoques podrían implicar sistemas de aprendizaje automático (machine learning) como SVM (Support Vector Machines), redes neuronales, etc.

En una segunda (2) etapa de este proyecto se deberá investigar y desarrollar un algoritmo de seguimiento espacio-temporal de los resultados obtenidos en (1). Tales soluciones podrían incluir filtros de partículas o filtros de Kalman entre otras técnicas.

Finalmente, (3) investigación y desarrollo de un algoritmo para la predicción TTC partir de los datos obtenidos en las etapas (1) y/o (2) ha de ser llevado a cabo a fin de considerar este TFM completado con éxito. Los resultados obtenidos en este punto dependerán en gran medida de los resultados obtenidos en las etapas anteriores. Test y optimización de los algoritmos propuestos tanto de manera individual, como a nivel de sistema deberán ser llevados a lo largo del desarrollo del proyecto sobre a cada uno de los módulos diseñados. Métodos cuantitativos y/o cualitativos deberán ser considerados según corresponda.

La siguiente figura recoge una sintetización simplista de las diferentes fases que este TFM presenta.

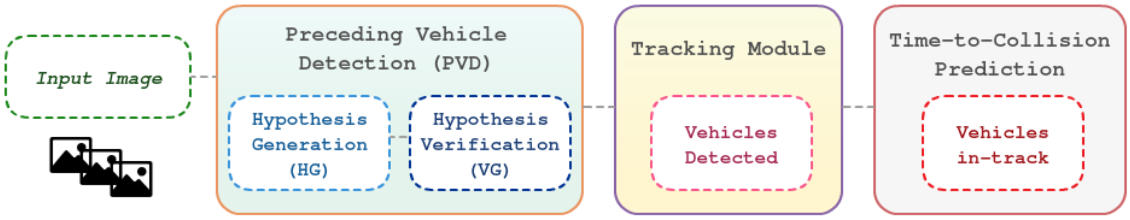


Figure 1.3: Diagrama del TFM

El flujo de trabajo de este proyecto es, por lo tanto y de manera general, muy intuitivo. Todo comienza con las imágenes originales adquiridas en tiempo real por la cámara situada enfocando a la carretera en el interior del vehículo. En ellas, y de manera individual, el módulo de detección de vehículos (PVD) propone en primera instancia una serie de vehículos candidatos a ser detectados, los cuales son verificados posteriormente. Estos resultados son sometidos a un seguimiento a lo largo de sucesivas tramas de imágenes con la intención final de abordar el desafío de predecir el Tiempo-de-Colisión entre dichos vehículos detectados y el vehículo propio.

Así, previa presentación de los algoritmos y principios fundamentales empleados en el desarrollo de este proyecto, el presente informe recoge de manera particular y en detalle a lo largo de los siguientes capítulos el sistema desarrollado, compuesto por los módulos anteriormente mencionados, así como alguno de los experimentos llevados a cabo con tal fin. En última instancia, algunas conclusiones finales y una breve lista de futuras líneas de trabajo son sugeridas.

Véase **Memoria Original del TFM** para acceso al informe completo del presente capítulo - *Chapter I: Advanced Driver Assitance Systems (ADAS).*

Capítulo II

Visión Artificial: Principios & Algoritmos

2.1 Introducción

En contraste con el capítulo anterior, donde los principios de los sistemas ADAS junto con los principales objetivos de este TFM han sido presentados, este Capítulo II recoge una breve y al mismo tiempo detallada presentación de los conceptos más importantes, tanto en relación al procesamiento de imágenes mediante visión artificial como a los algoritmos utilizados en el desarrollo de los módulos y sistemas propuestos en el presente informe.

Algunos de dichos aspectos revisados en las siguientes líneas abarcan el concepto de Aprendizaje Máquina (ML, Machine Learning), y más concretamente SVMs (Support Vector Machine) debido a su importancia clave en la fase de detección de vehículos, así como los Filtros Gabor y el impacto de su correcta parametrización en el éxito de dicha tarea. Posteriormente los Filtros de Kalman son presentados en detalle, representando un papel clave en el módulo de seguimiento desarrollado a fines de completar la predicción TTC que este proyecto persigue. Finalmente el concepto de Flujo Optico (Optic Flow) así como su relevancia en cuanto a la propia estimación del Tiempo-de-Colisión son recogidos.

No obstante, se ha de notar que el cómo cada técnica y/o algoritmo es útil y específicamente aplicado en el contexto de este proyecto es algo que será representado en detalles en los siguientes capítulos III, IV y V respectivamente.

2.2 Aprendizaje Máquina & SVMs

El Aprendizaje Máquina (ML) es una rama de la inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras *aprender*. De forma más concreta, consiste en algoritmos/programas capaces de generalizar comportamientos a partir de una información no estructurada suministrada en forma de ejemplos. Es, por lo tanto, un proceso de inducción del conocimiento que en la última década ha hecho posibles aplicaciones que abarcan desde el reconocimiento de voz práctico, la búsqueda web eficiente, o el mejor entendimiento del genoma humano, hasta constituir la base de los coches autónomos de futuro. Este timo de tecnología y algoritmos se encuentra tan omnipresente hoy día, que es probable que cualquier individuo se aproveche de sus ventajas docenas de veces al día, sin tan siquiera saberlo. Asimismo es considerado por muchos investigadores como la mejor manera de avanzar hacia niveles de inteligencia artificial capaz de simular el humano [9].

En general [10], un problema de aprendizaje máquina considera un conjunto de n muestras de datos los cuales son tratados de determinadas manera a fin de ser capaces de predecir las propiedades de datos desconocidos en relación a dichos conjuntos. Así, un problema típico de aprendizaje máquina puede ser dividido en diferentes categorías en función de sus características, tales como aprendizaje supervisado o no supervisado, y dentro de los primeros el problema en cuestión puede ser considerado de clasificación o de regresión. En cualquier caso, el Aprendizaje Máquina básicamente consiste en tratar de aprender ciertas propiedades de un conjunto de datos, para su posterior aplicación a conjuntos de datos desconocidos. Por ello, una práctica común en lo que a aprendizaje maquina se refiere para evaluar un algoritmo consiste en dividir los datos de los que se dispone en dos conjuntos diferentes, uno llamado conjunto de entrenamiento, el cual es utilizado para las tareas de aprendizaje, y otro conocido como conjunto de evaluación utilizado para como su propio nombre indica evaluar el correcto aprendizaje del algoritmo.

En este contexto, SVMs (Support Vector Machines) consisten en un conjunto de esos algoritmos considerados de aprendizaje supervisado donde, dado un conjunto de ejemplos de entrenamiento (muestras), diferentes comúnmente denominadas clases pueden ser etiquetada para posteriormente entrenar un SVM y construir un modelo que prediga la clase de una nueva muestra. Intuitivamente, un SVM es un modelo que representa a los puntos de muestra en el espacio, separando las clases a 2 espacios lo más amplios posibles mediante un hiperplano de separación definido como el vector entre los 2 puntos, de las 2 clases, más cercanos al que se llama vector soporte. Cuando las nuevas muestras se ponen en correspondencia con dicho modelo, en función de los espacios a los que pertenezcan, pueden ser clasificadas a una o la otra clase.

Veremos en el siguiente capítulo como los SVM constituyen la base de un clasificador vehículo/no-vehículo, esencial para la detección formal de vehículos a partir de las secuencias de videos consideradas en este proyecto.

2.3 Filtros de Gabor

En procesamiento de imagen, un filtro de Gabor, es un filtro lineal utilizado para la detección de bordes. La representación de filtros de Gabor tanto en frecuencia como en orientación son encontradas similares a las del sistema visual humano, siendo posible modelar mediante este tipo de filtros ciertas células simples presentes en la corteza visual del cerebro de determinados mamíferos [11]. Por todo ello, se considera que el análisis de imágenes mediante filtros de Gabor es similar a la percepción en el sistema visual humano, y de ahí que sean ampliamente utilizados en este contexto.

En el dominio espacial, un filtro de Gabor 2D consiste en una función gaussiana modulada por una onda plana sinusoidal. Tal y como son presentados en [12], los filtros de Gabor constituyen un grupo de *wavelets*, capturando cada una de ellos energía a una frecuencia y dirección específica. Estas distribuciones de energía permiten la extracción de diferentes características de manera muy simple en función de dicha frecuencia y orientación de diseño de éstos, lo cual ha permitido que aplicaciones basadas en este tipo de filtros se encuentren ampliamente presentes en la sociedad a día de hoy [13]; algunos de estos ejemplos incluyen el análisis de documentos, la búsqueda y reconocimiento de patrones, la extracción de imágenes de rostros humanos como información biométrica, y entrando más de lleno en el ámbito de este proyecto la detección de vehículos. El siguiente capítulo desvelará cómo estos filtros permiten llevar a cabo dicha labor, y su influencia en el diseño del perseguido sistema TTC.

2.4 Filtros de Kalman

Otro método que muy popular en aplicaciones de visión artificial es el filtrado de Kalman. Un filtro de Kalman es esencialmente un estimador que, en el contexto de este proyecto, puede ser utilizado para predecir la posición de un objeto en futuras tramas de video dado cierto conocimiento de su posición en el momento actual. Funciones de seguimiento mediante el empleo de filtros Kalman proporcionan múltiples ventajas para cualquier aplicación basado en vídeo, ya que permite reducir el área de la trama a analizar en busca de la existencia de un determinado objeto, reduciendo así al mínimo las detecciones de falsos positivos y el tiempo necesario de procesado. Una ventaja adicional ofrecida por la naturaleza misma de los filtros de Kalman, es el efecto de suavizado en los resultados obtenidos lo cual permite mejorar la robustez de estos en presencia de posible ruido que se observa a menudo en videos.

De manera más específica, un filtro de Kalman realiza la estimación de un determinado proceso mediante el empleo de un ciclo de control realimentado, es decir, el filtro estima el estado del proceso bajo estudio en un determinado instante de tiempo, actualizando posteriormente dichos resultados a partir de información proveniente de determinadas mediciones, las cuales como se mencionaba antes pueden presentar cierto ruido. De este modo, las ecuaciones que definen un filtro de Kalman se dividen en dos grupos: las ecuaciones de actualización en tiempo *(time update equations)* y ecuaciones de actualización a partir de medidas *(measurement update equations)*. Basicamente, las primeras son responsables de proyectar hacia adelante (en el tiempo) las estimaciones del estado del proceso siendo estudiado (por ejemplo la posición de un objeto) a fin de obtener una estimación *a priori* del siguiente estado del proceso de manera temporal. Mientras que las segundas son responsables de la actualización de la estimación *a priori* generada, a partir de nueva información obtenida del proceso a partir de mediciones empíricas, para así generar una mejorada y definitiva estimación *a posteriori* [14] [15].

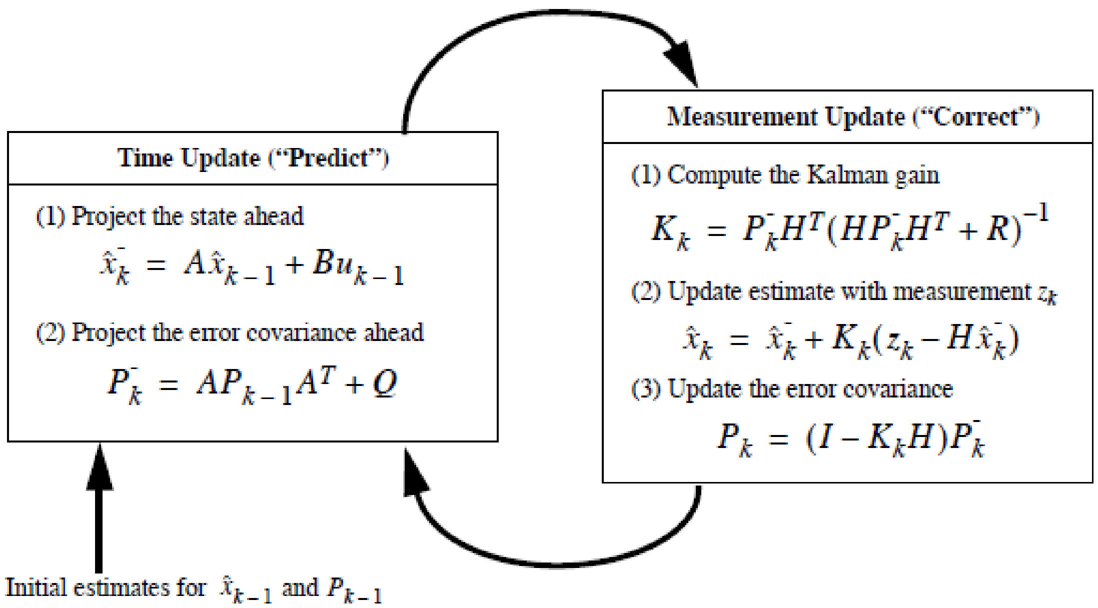


Figura 2.1: Esquema completo del Funcionamiento de un Filtro de Kalman

En cierto modo las ecuaciones de actualización de tiempo también pueden ser considerados como ecuaciones de predicción, mientras que las ecuaciones de actualización por medición pueden ser considerados como ecuaciones correctoras. Siendo de hecho, el algoritmo de estimación final que define un filtro de Kalman, muy semejante a un algoritmo de predicción-corrección para resolver problemas numéricos, tal y como se muestra en la figura anterior, donde además son incluidas las ecuaciones que definen ambas etapas que conforman el proceso.

2.5 Flujo Óptico

El concepto de flujo óptico (optical flow), esencial en las últimas etapas de del presente proyecto para la implementación del sistema Time-to-Colisión aquí propuesto, fue por primera vez introducido por el psicólogo estadounidense James J. Gibson en la 1940 se puede definir como el patrón de movimiento aparente tanto de los objetos y las superficies, como de los bordes en una escena visual causado por el movimiento relativo entre un observador (un ojo o una cámara) y la propia escena [16]. Otra forma de definir el flujo óptico [17] podría ser como el cambio de luz en la imagen, por ejemplo en la retina de un ojo o el sensor de la cámara, debido a un movimiento relativo entre el globo ocular o la cámara y la escena siendo observada

De esta manera, secuencias ordenadas de imágenes permiten la estimación de movimiento tal y como la siguiente ejemplo recoge.

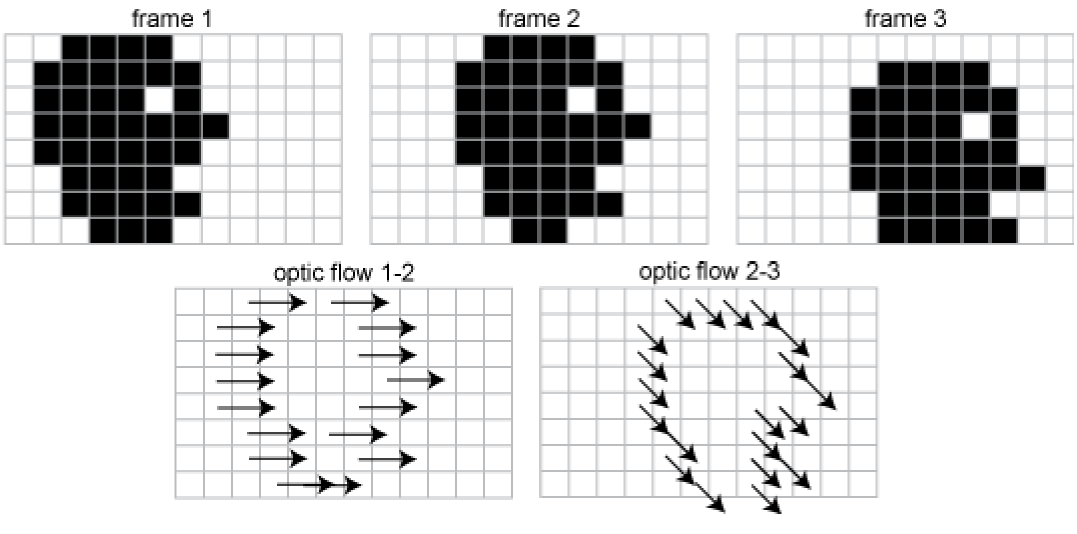


Figura 2.2: Flujo Optico detectado en 3 imagenes consecutives en el tiempo, revela el movimiento de una figura de ejemplo (silueta de una cabeza)

En este ejemplo el flujo óptico es descrito como la correspondencia de píxeles de contorno de la figura representada entre las imágenes 1 y 2, así como entre la 2 y la 3 respectivamente. Los principales desafíos para métodos y algoritmos encargados de estimar este flujo óptico reside en, no sólo considerar los píxeles de contorno del objeto bajo estudio, sino también encontrar la correspondencia espacial (entre tramas consecutivas) para cada pixel en la imagen.

Además se ha de considerar de igual manera el problema de inferir no sólo el movimiento del observador y los objetos en la escena, sino también la estructura de los objetos y el medio ambiente. Como es natural, la conversión de esta capacidad innata en el reino animal a una capacidad llevada a cabo por un ordenador, es uno de los aspectos cruciales en el campo de la visión artificial.

No obstante, los usos y aplicaciones de este tipo de técnicas presentes en la realidad a día de hoy son un hecho, presentando su foco principal en el ámbito de la robótica, y más específicamente en áreas tales como la detección y rastreo de objetos, o la detección de movimiento y funcionaes propias de la navegación del robot. El Capítulo V mostrará como el concepto de flujo óptico y su aplicación práctica representa una de las claves fundamentales del sistema TTC aquí propuesto.

Véase **Memoria Original del TFM** para acceso al informe completo del presente capítulo - *Chapter II: Computer Vision, Principles & Algorithms.*

Capítulo III
Detección de Vehículos (PVD)

3.1 Introducción

Como se ha visto en el Capítulo I, donde se recogen los principales objetivos de este proyecto, considerado en su conjunto, el sistema propuesto en el presente informe puede ser dividido en tres sub-sistemas prácticamente independientes (ver Capítulo 1.2: Objetivos de este TFM) con el fin de satisfacer el reto de diseñar un sistema TTC-ADAS capaz de ser integrado en un sistema real anticolisiones de un vehículo.

Se ha de notar que una robusta y fiable Detección de Vehículos PVD (Preceding Vehicle Detection) en imágenes es una tarea muy desafiante ya que no sólo se ve afectada por el tamaño, la forma, el color, y la situación de los vehículos, sino también por las condiciones de iluminación, tiempo, entorno, así como la propia superficie de la carretera por la que el vehículo transita. Asimismo un sistema de detección de vehículos también debe ser capaz distinguir los vehículos de todos los demás patrones visuales existentes en el mundo, tales como otros objetos de similar aspecto o forma [18] [19].

Figura 3.1: Apariencia normal de una escena de tráfico real; Presencia de vehículos, carretera, señales de tráfico, guarda raíl, paisaje, rayos del sol, sombras, etc.

Normalmente la detección de vehículos en las escenas de tráfico es una tarea dividida en dos etapas bien diferenciadas e igualmente importantes: la primera de ellas suele consistir en una etapa de generación de hipótesis (HG) donde potenciales vehículos presentes en la escena son detectados, y mientras que la segunda etapa sirve para Verificación de dichas Hipótesis (HV) y en consecuencia confirmar la detección propuesta o por otro lado descartarla en última instancia.

De acuerdo con esto, en este capítulo es presentado el primero de dichos sub-sistemas mencionados anteriormente, el cual consiste un algoritmo para la detección de vehículos en imágenes adquiridas en tiempo real mediante una cámara instalada en el interior del vehículo propio enfrentando la carretera delante de éste. El sistema comprende una fase de Generación de Hipótesis (HG) basada en una aproximación a diferentes escalas de detectores de bordes Sobel (horizontales y verticales) en combinación con una fase de segmentación del color rojo de las luces traseras de los vehículos para dar una primera localización de posibles vehículos en la imagen. Debido a limitaciones temporales relacionadas con los plazos en que este proyecto ha sido desarrollado, el primero de los módulos en que este sistema se divide, es decir la fase de Generación de Hipótesis no ha terminado de ser completada en su totalidad. En la Memoria Original de este TFM se puede consultar más en detalle la solución propuesta para este módulo, así como el trabajo desarrollado a pesar de no haberse podido completar e incluir en el sistema TTC propuesto. No obstante, gracias a la modularidad del esquema propuesto, este hecho no compromete en absoluto el desarrollo natural del proyecto, siendo la etapa de Verificación de Hipótesis la protagonista en este sistema de detección de vehículos. Más concretamente, el algoritmo propuesto se basa en un banco de filtros de Gabor para la extracción de características y descripción de dichos vehículos, el cual en combinación con un clasificador SVM debidamente entrenado permite determinar los vehículos correctamente detectados y minimizar los falsos positivos, consiguiendo así a largo plazo un mejor comportamiento del sistema.

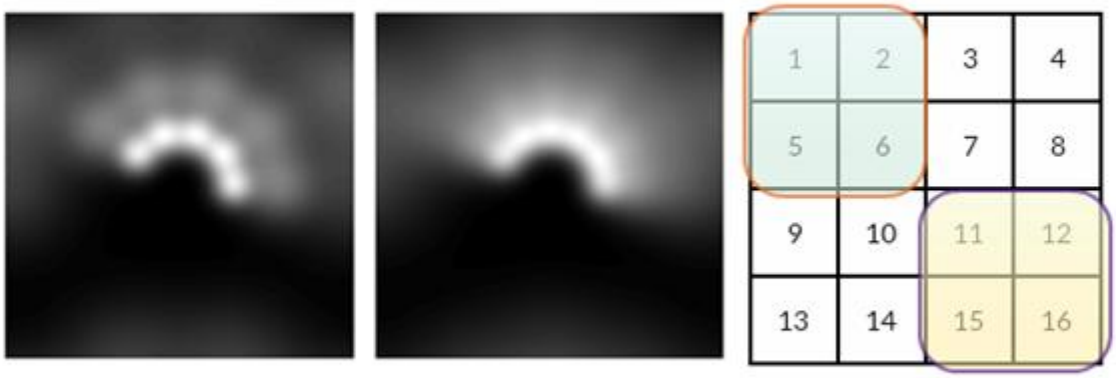
3.2 Sistema Propuesto

Tal y como se ha mencionado, los datos de entrada a este módulo de verificación, son las hipótesis de vehículos generadas en la etapa anterior, siendo responsabilidad de éste clasificarlas definitivamente como vehículos, confirmando así la detección, o como no-vehículos siendo así descartada.

A tal fin, una aproximación basada en filtros de Gabor para la caracterización de dichos vehículos y posteriormente un clasificador SVM es propuesta. Más concretamente, una serie de bancos de filtros de Gabor debidemente parametrizados es utilizada para la caracterización de las clases *vehículo/no-vehiculo* a partir de un conjunto de datos de conocidos, los cuales son utilizados para el entrenamiento del clasificar SVM implementado, encargado en última instancia de discernir entre las detecciones reales y los falsos positivos propuestos por la etapa anterior. Este modulo es presentado en las siguientes líneas.

3.2.1 Extracción de Características mediante Filtros de Gabor

De acuerdo con lo visto en el Capítulo II, los filtros de Gabor básicamente actúan como filtros paso-banda en función de cómo sean definidos. La siguiente figura muestra el espectro en potencia de dos bancos de filtros diferentes, donde las zonas más claras representan respectivamente frecuencia y orientación de éstos. En este proyecto se hace uso de la estrategia descrita en [20] y [21] para la mencionada tarea de extracción de características, la cual es presentada a continuación.



(a) (b) (c)

Figura 3.2: Filtrado de Gabor: (a) Banco de Filtros de Gabor de 3 escalas y 5 orientaciones; (b) Banco de Filtros de Gabor de 4 escalas y 6 orientaciones; (c) Ventanas para la extracción de Características: Primera y última sub-ventana 32x32 a partir de los cuadros 1, 2, 5 y 6, y 11, 12, 15 y 16 respectivamente

Dada una imagen de entrada $I(x, y)$, extracción de características de dicha imagen se realiza mediante la convolución de $I(x, y)$ con un banco de filtros de Gabor, y una fase de post-procesamiento de los resultados obtenidos. En el sistema aquí propuesto son utilizadas características Gabor basadas en momentos, extraídas de varios sub-ventanas de la imagen de entrada.

A tal fin, en primer lugar cada sub-imagen es escalada a un tamaño fijo de 64x64. A continuación, ésta es subdividida en 9 32x32 sub-ventanas superpuestas. Suponiendo que cada sub-imagen consiste en 16 16x16 cuadros (los cuadros 1, 2, 5 y 6 comprenderían la primera sub-ventana 32x32, el 2, 3, 6 y 7 la segunda, 5, 6, 9, y 10 la cuarta, y así sucesivamente (ver Figura 3.2 (c)) hasta la novena sub-ventana. Los filtros de Gabor son aplicados entonces a cada sub-ventana por separado.

El motivo de la extracción de posiblemente información redundante reside en diseñar un sistema lo más robusto posible capaz de compensar posibles errores presentes en la etapa de generación de hipótesis como por ejemplo imágenes que contienen vehículos parcialmente ocultos, o demasiada información del entorno.

En última instancia, la respuesta obtenida por cada filtro de Gabor y sub-ventana, en función de la frecuencia y orientación a la que el filtro estuviese definido, queda representada por su momento de hasta tercer orden: *mean* μ_{ij} *(media)*, *standard deviation* σ_{ij} *(desviación típica)*, y *skewness* κ_{ij} *(medida de la asimetría)* donde i corresponde al filtro i-ésimo y j a la j-ésima sub-ventana). El uso de momentos implica que solo propiedades estadísticas de un grupo de píxeles es tenida en cuenta en el cálculo de los resultados, mientras que información relativa a su posición es esencialmente descartada, lo cual es particularmente útil para compensar errores en la fase de HG como se comentaba anteriormente. De acuerdo con esto y a modo de ejemplo, suponiendo el uso de S = 2 escalas y K = 3 orientaciones en la definición de los filtros de Gabor (S x K filtros), la aplicación de estos sobre cada una de las 9 sub-ventanas definidas, usando μ_{ij}, σ_{ij}, and κ_{ij} como componentes característicos, da como lugar a un vector de tamaño 162, por el cual la imagen queda caracterizada, y que presentaría la siguiente forma:

$$f = [\mu_{11}\ \sigma_{11}\ \kappa_{11}, \mu_{12}\ \sigma_{12}\ \kappa_{12}, \dots\ \mu_{69}\ \sigma_{69}\ \kappa_{69}]$$

Se ha de notar que el uso de tres momentos y no dos se debe a los resultados mucho peores obtenidos solo con dos momentos, lo cual pone de manifiesto la importancia de la información de la asimetría para nuestro problema de detección de vehículos.

3.2.2 Clasificación SVM

El último paso para determinar si las hipótesis generadas, una vez definidas por su vector de características, se corresponden con un vehículo o no, es llevar a cabo una clasificación de estas entre *vehículos* y todo lo demás donde quedarían incluidos carretera, señales de tráfico, paisaje, etc. A tal fin en este proyecto se hace uso del *Toyota Motor Europe (TME) Motorway Dataset* [22] [23], un conjunto de datos/librería compuesta por 28 clips y un total de aproximadamente 27 minutos de video (30000+ imágenes) de escenas de tráfico reales, incluyendo diferentes situaciones y condiciones meteorológicas, carriles en la carretera y curvatura de ésta, así como anotaciones de los vehículo generadas de forma semiautomática a partir de datos del escáner láser.

El TME Motorway Dataset es a su vez dividido en tres conjunto de datos diferentes en función de para qué son utilizados. El primero de ellos se emplea con funciones de entrenamiento del clasificador SVM a implementar. El segundo de ellos se emplea para validar el mejor o peor comportamiento del clasificador, siendo el último de ellos empleado para la evaluación final del mismo, y en definitiva del sistema de detección de vehículos propuesto, en términos de precisión y otras métricas utilizadas a tal fin.

Figura 3.3: Clase *vehiculo*

Figura 3.4: Clase *no-vehiculo*

Se ha de notar que a fin de obtener los mejores resultados posibles en esta etapa de detección de vehículos, es crucial que el clasificador sea entrenado de acuerdo a un conjunto de parámetros óptimo definido por el banco de filtro Gabor (σ, θ, γ...) para la caracterización de ambas clases, así como por un conjunto de datos de entrenamiento de un tamaño suficiente para nuevamente caracterizar con la menor ambigüedad posible ambas clases. De acuerdo con ello, una serie de experimentos presentados a continuación han sido llevados a cabo.

3.2.3 Test y Optimización

Los experimentos y mediciones realizadas se pueden dividir en dos tipos: optimización de parámetros y optimización del tamaño del conjunto de datos de entrenamiento.

Los primeros de ellos consisten esencialmente en la optimización de los parámetros que definen los bancos de filtros de Gabor utilizados para caracterizar las imágenes de entrada al clasificador. Los parámetros a optimizar son en concreto $Ksize$, $Lambda\ (\lambda)$, $Gamma\ (\gamma)$, $Orientation\ (\theta)$ and $Scale\ (\sigma)$. Así, dada una configuración fija del resto de parámetros observamos de manera individual la influencia de cada uno de ellos en la precisión del clasificador a la hora de discernir entre vehículos y no-vehículos. Los resultados obtenidos quedan recogidos en las siguientes tablas:

Tabla 3.1: $Ksize$

Ksize	tp	tn	fp	fn	P	R	g	F	Accuracy (A)
1	74	87.5	12.5	26	0.85549	0.74	0.65778	0.79357	0.8075
3	76	88	12	24	0.86364	0.76	0.67857	0.80851	0.82
5	73	88.5	11.5	27	0.86391	0.73	0.65471	0.79133	0.8075
7	71	86.5	13.5	29	0.84024	0.71	0.62555	0.76965	0.7875
9	74	87.5	12.5	26	0.85549	0.74	0.65778	0.79357	0.8075
11	**91.5**	**95.5**	**4.5**	**8.5**	**0.95313**	**0.915**	**0.8756**	**0.93367**	**0.935**
13	85.5	94.5	5.5	14.5	0.93956	0.855	0.81043	0.89529	0.9
15	90.5	93	7	9.5	0.92821	0.905	0.84579	0.91646	0.9175
17	88	94.5	5.5	12	0.94118	0.88	0.83412	0.90956	0.9125
19	81.5	94.5	5.5	18.5	0.93678	0.815	0.77251	0.87166	0.88
21	80	89.5	10.5	20	0.88398	0.8	0.72398	0.8399	0.8475
23	83	92	8	17	0.91209	0.83	0.76852	0.86911	0.875
25	82	93	7	18	0.92135	0.82	0.76636	0.86772	0.875

Tabla 3.2: $Gamma$, γ

Gamma - γ	tp	tn	fp	fn	P	R	g	F	Accuracy (A)
0.1	92	95	5	8	0.94845	0.92	0.87619	0.93401	0.935
0.3	91.5	95	5	8.5	0.94819	0.915	0.87143	0.9313	0.9325
0.5	91.5	95.5	4.5	8.5	0.95313	0.915	0.8756	0.93367	0.935
0.6	**91.5**	**95.5**	**4.5**	**8.5**	**0.95313**	**0.915**	**0.8756**	**0.93367**	**0.935**
0.7	91.5	95.5	4.5	8.5	0.95313	0.915	0.8756	0.93367	0.935
0.9	91	95	5	9	0.94792	0.91	0.86667	0.92857	0.93
1.5	92	94	6	8	0.93878	0.92	0.86792	0.92929	0.93
4	89.5	93.5	6.5	10.5	0.93229	0.895	0.84038	0.91327	0.915
10	89.5	95	5	10.5	0.94709	0.895	0.85238	0.92031	0.9225

Tabla 3.3: $Lambda, \lambda$

Lambda - λ	tp	tn	fp	fn	P	R	g	F	Accuracy (A)
2.5	88.5	92	8	11.5	0.9171	0.885	0.81944	0.90076	0.9025
5	88	92.5	7.5	12	0.92147	0.88	0.8186	0.90026	0.9025
7.5	89	96	4	11	0.95699	0.89	0.85577	0.92228	0.925
9	89	94.5	5.5	11	0.9418	0.89	0.8436	0.91517	0.9175
9.5	90.5	93.5	6.5	9.5	0.93299	0.905	0.84977	0.91878	0.92
9.8	89.5	95.5	4.5	10.5	0.95213	0.895	0.85646	0.92268	0.925
9.9	90	97	3	10	0.96774	0.9	0.87379	0.93264	0.935
10	**91.5**	**95.5**	**4.5**	**8.5**	**0.95313**	**0.915**	**0.8756**	**0.93367**	**0.935**
10.1	92	95	5	8	0.94845	0.92	0.87619	0.93401	0.935
10.5	86.5	95.5	4.5	13.5	0.95055	0.865	0.82775	0.90576	0.91
11	84	92.5	7.5	16	0.91803	0.84	0.7814	0.87728	0.8825
12.5	74.5	87	13	25.5	0.85143	0.745	0.65929	0.79467	0.8075
15	72.5	85.5	14.5	27.5	0.83333	0.725	0.63319	0.7754	0.79

Tabla 3.4: $Theta, \theta$

Orientation - θ	tp	tn	fp	fn	P	R	g	F	Accuracy (A)
2	90.5	93.5	6.5	9.5	0.93299	0.905	0.84977	0.91878	0.92
3	86	93	7	14	0.92473	0.86	0.80374	0.89119	0.895
4	91.5	94.5	5.5	8.5	0.9433	0.915	0.8673	0.92893	0.93
5	90	95.5	4.5	10	0.95238	0.9	0.86124	0.92545	0.9275
6	**91.5**	**95.5**	**4.5**	**8.5**	**0.95313**	**0.915**	**0.8756**	**0.93367**	**0.935**
7	89	94	6	11	0.93684	0.89	0.83962	0.91282	0.915
8	92	94.5	5.5	8	0.94359	0.92	0.87204	0.93165	0.9325
9	90.5	95	5	9.5	0.94764	0.905	0.8619	0.92583	0.9275
10	90	97	3	10	0.96774	0.9	0.87379	0.93264	0.935

Tabla 3.5: $Sigma, \sigma$

Scale - σ	tp	tn	fp	fn	P	R	g	F	Accuracy (A)
[2, 4, 6, 8]	87.5	95	5	12.5	0.94595	0.875	0.83333	0.90909	0.9125
[4, 6, 8, 10]	88	93.5	6.5	12	0.93122	0.88	0.82629	0.90488	0.9075
[6, 8, 10, 12]	87.5	93.5	6.5	12.5	0.93085	0.875	0.8216	0.90206	0.905
[1, 4, 7, 10]	92.5	94	6	7.5	0.93909	0.925	0.87264	0.93199	0.9325
[1, 3, 5]	87	93.5	6.5	13	0.93048	0.87	0.8169	0.89922	0.9025
[3, 5, 7]	**91.5**	**95**	**5**	**8.5**	**0.94819**	**0.915**	**0.87143**	**0.9313**	**0.9325**
[5, 7, 9]	91.5	94.5	5.5	8.5	0.9433	0.915	0.8673	0.92893	0.93
[1, 5, 9]	89.5	96	4	10.5	0.95722	0.895	0.86058	0.92506	0.9275
[2, 4, 6]	85.5	93.5	6.5	14.5	0.92935	0.855	0.80282	0.89063	0.895
[4, 6, 8]	87.5	94.5	5.5	12.5	0.94086	0.875	0.82938	0.90674	0.91
[6, 8, 10]	88	94	6	12	0.93617	0.88	0.83019	0.90722	0.91
[2, 6, 10]	87	91	9	13	0.90625	0.87	0.79817	0.88776	0.89

Como se puede observar, la configuración optima que se puede extraer de los resultados obtenidos queda compuesta de la siguiente manera: $ksize = 11$, $\lambda =$

10, $\gamma = 0.6, \theta = 6$ y $\sigma = [3, 5, 7]$, siendo la utilizada en la implementación final del sistema.

En cuanto al tamaño óptimo del conjunto de datos utilizado en el entrenamiento del sistema la siguiente tabla recoge los resultados obtenidos.

Tabla 3.6: $Size\ (tamaño)$ conjunto de datos de entrenamiento

Size	tp	tn	fp	fn	P	R	g	F	Accuracy (A)
400	91	95	5	9	0.94792	0.91	0.86667	0.92857	0.93
800	91.5	95	5	8.5	0.94819	0.915	0.87143	0.9313	0.9325
1200	93	94.5	5.5	7	0.94416	0.93	0.88152	0.93703	0.9375
1600	93.5	94	6	6.5	0.9397	0.935	0.88208	0.93734	0.9375
2000	**93.5**	**95.5**	**4.5**	**6.5**	**0.95408**	**0.935**	**0.89474**	**0.94444**	**0.945**
2400	93.5	94	6	6.5	0.9397	0.935	0.88208	0.93734	0.9375

Siendo el tamaño óptimo, en función del pico en la precisión del sistema, 2000 imágenes (1000 vehículos y 1000 no-vehículos). Se ha de notar que los resultados en términos de precisión (Accuracy, A) en relación a los experimentos llevados a cabo son únicamente de interés en términos relativos respecto a los obtenidos tomando otra configuración de parámetros.

3.2.4 Evaluación Final

Una vez que el proceso de optimización del sistema ha sido completado el modulo de detección de vehículos es finalmente evaluado. Para ello un conjunto de datos de evaluación del mismo tamaño que el conjunto de entrenamiento finalmente empleado, es decir 2000 imágenes, es extraído. Así, el comportamiento final del sistema presentado queda recogido a continuación.

Tabla 3.7: Resultados Evaluación Final TME Motorawy Dataset

Dataset	tp	tn	fp	fn	P	R	g	F	Accuracy (A)
TME Motorway	95.5	97	3	4.5	0.96954	0.955	0.92718	0.96222	**96.25%**

Tabla 3.8: Resultados Evaluación Caltech DBs

Dataset	Size (images/frames)	True Positives (TP)	False Negatives (FN)	Detection Rate
Caltech 1999	126	119	7	**94.44%**
Caltech 2001	526	492	34	**93.54%**

Los resultados muestran como, tras el proceso de optimización y haciendo uso de un variado conjunto de evaluación formado por 2000 imágenes, el sistema propuesto arroja tasas de falsos positivos y negativos reducidas, 3% y 4.5% respectivamente, reflejando un comportamiento final caracterizado por una tasa de precisión en la detección de vehículos del 96.25%, un valor considerado satisfactorio en el contexto de esta tesis.

Asimismo, el sistema ha sido además evaluado sobre dos conjuntos de datos conocidos respectivamente como Caltech 1999 y Caltech 2001 [24], formados por 126 y 526 imágenes de vehículos tomados en los parkings de los laboratorios

Caltech y las carreteras de California en condiciones meteorológicas favorables, ideales para la evaluación de nuestro algoritmo. Las observadas tasas de detección de un 94.44% y 93.54% respectivamente ponen de manifiesto el buen comportamiento del sistema, confirmando los resultados obtenidos sobre el TME Motorway Dataset.

3.2.5 Conclusiones

En este capítulo, el primero de los módulos que componen el sistema TTC que esta tesis propone, un modulo para la detección de vehículos presentes en imágenes reales de tráfico, ha sido presentado.

El sistema comprende una fase de Generación de Hipótesis (HG) basada en una aproximación a diferentes escalas de detectores de bordes Sobel (horizontales y verticales) en combinación con una fase de segmentación del color rojo de las luces traseras de los vehículos para dar una primera localización de posibles vehículos en la imagen. Como bien ha sido reflejado, debido a limitaciones temporales relacionadas con los plazos en que este proyecto ha sido desarrollado, este módulo no ha podido ser completado en su totalidad de manera satisfactoria. No obstante, gracias a la modularidad del esquema propuesto, este hecho no compromete en absoluto el desarrollo natural del proyecto, siendo la etapa de Verificación de Hipótesis (HV) la protagonista en este sistema de detección de vehículos. Ésta, basada en un clasificador SVM con núcleo en un conjunto de filtros de Gabor debidamente parametrizados, se encarga de validar las hipótesis generadas en la fase anterior o de descartarlas definitivamente.

En última instancia, un conjunto de test para la optimización de las prestaciones ofrecidas por el clasificador, así como para la evaluación de su comportamiento final, han sido llevados a cabo y presentados, demostrando el más que aceptable funcionamiento del mismo, y en definitiva del sistema de detección de vehículos propuesto.

Véase **Memoria Original del TFM** para acceso al informe completo del presente capítulo - *Chapter III: Preceding Vehicles Detection (PVD).*

Capítulo IV

Módulo de Seguimiento

1.1 Introducción

Como se ha visto en el capítulo anterior, el comportamiento del sistema de detección de vehículos propuesto es correcto para secuencias de imágenes consideradas de manera individual. Sin embargo, en ciertas ocasiones esto podría no ser suficiente al tratar con escenas que cambian de manera dinámica como son las escenas de tráfico. En dichos casos el procesamiento y aprovechamiento de información temporal presente en secuencias de video es crucial para la obtención de los mejores resultados posibles. Dicho de otra manera, el procesamiento de imágenes de manera individual obvia la información intrínsecamente presente en la relación existente entre imágenes consecutivas por el simple hecho de ser parte de un ente mayor como es una secuencia de video.

En este contexto, y a fin de aprovechar dicha información temporal presente en la secuencia de video para mejorar el comportamiento global del sistema, en este capítulo se presenta un modulo de seguimiento basado en filtrado Kalman el cual permite reducir la tasa de detección de falsos positivos comprobando la consistencia de las detecciones propuestas por la fase de Verificación de Hipótesis, disminuir el efecto del ruido presente en la fase de detección ya que proporciona información adicional a tener en cuenta en la generación de las mismas, así como reducir la tasa de pérdida de vehículos bajo seguimiento al ser capaz de estimar el posicionamiento de estos en caso de pérdida esporádica de los mismos o fallo en la fase de detección. En última instancia el correcto funcionamiento del presente módulo es de crucial importancia para la implementación final del sistema TTC que se persigue, el cual será presentado en el capitulo siguiente.

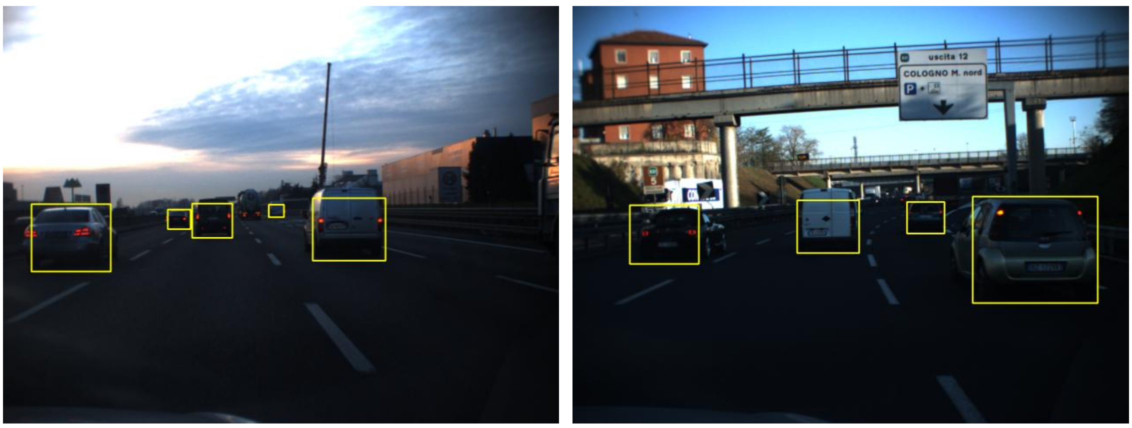


Figura 4.1: Ejemplo de Resultados Etapa PVD

1.2 Consistencia PVD

El modulo de detección de vehículos PVD (Preceding Vehicle Detection) tal y como se ha visto en el capítulo anterior genera una serie de hipótesis sobre la localización de posibles vehículos dentro de la escena analizada. Sin embargo, a fin de diseñar un sistema lo más robusto posible, dichas detecciones no son tomadas como detecciones reales hasta que no presentan consistencia a lo largo del tiempo. Normalmente los falsos positivos no son consistentes en el tiempo, sino más bien errores puntuales en la fase de detección. De este modo dichas falsas detecciones son descartadas, mejorando en consecuencia el comportamiento del sistema.

De manera más concreta, para que un candidato propuesto por la fase PVD sea considerado como detección formal ha de ser detectado, es decir consistente, a lo largo de 3 tramas consecutivas. Dentro de una determinada trama, un vehículo queda caracterizado por una ROI (Región of Interest) en forma de cuadro definida de manera unívoca por el pixel inicial y final que compone el mismo de la siguiente manera $(xmin,\ \ ymin,\ \ xmax,\ \ ymax)$. La consistencia a tener en cuenta ha de ser en consecuencia tanto en tamaño como en posición dentro de la escena, donde el tamaño viene dado en píxeles por la superficie de la ROI, y la posición es considerada como el centro de ésta. Un ejemplo tanto de detecciones consistentes como de la eliminación de un falso positivo es recogido en la siguiente secuencia de imágenes.

Figure 4.2: Ejemplo Práctico Consistencia PVD: Secuencia de 4 tramas consecutivas pertenecientes al TME Motorway Dataset

Como se puede observar en la escena de tráfico recogida en las imágenes, hay 4 vehículos que han sido correctamente detectados y además su detección es consistente en el tiempo (color verde), sin embargo en un determinado momento de la secuencia la fase de detección de vehículos (PVD) propone como candidato a vehículo detectado lo que en realidad es una señal de tráfico (trama 752). La escena se puede observar en mejor detalle en la siguiente figura.

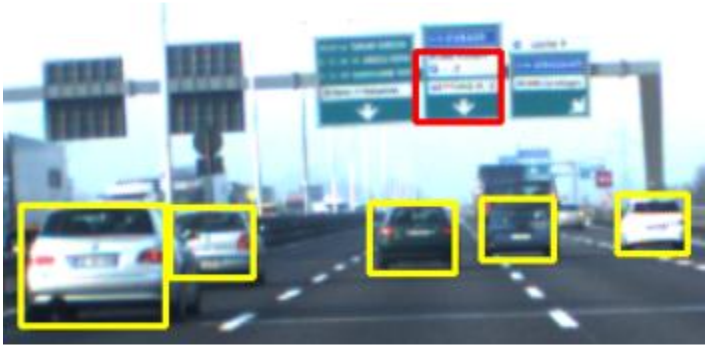


Figure 4.3: Zoom trama 752 revela la presencia de una detección errónea

Tal y como se puede ver, dicha detección no es consistente en el tiempo, y por lo tanto gracias a esta fase no es considerada por el sistema, mejorando las prestaciones de éste al reducir la tasa de falsos positivos. Otros ejemplos de falsos positivos son recogidos en las siguientes imágenes:

Figura 4.4: Detecciones correspondientes a Falsos Positivos (rojo)

1.3 Módulo de Seguimiento Kalman

Una vez un determinado vehículo es detectado de manera formal (consistente) éste pasa a una fase de seguimiento donde tanto su posición y tamaño (intrínsecamente definidas) dentro de la escena son controlados de manera constante por el sistema, además de quedar representado por un identificador único y personal dentro del mismo. Este seguimiento a lo largo del tiempo de los vehículos detectados es de vital importancia de cara a poder realizar una estimación del tiempo de contacto entre el vehículo propio y estos tal y como se verá en el Capítulo V. No obstante, basado principalmente en el empleo de filtros de Kalman, el presente capítulo recoge en detalle el modulo de seguimiento descrito.

De acuerdo a lo visto en el Capítulo 2.4, un filtro de Kalman es esencialmente un estimador simple capaz de predecir la posición de un objeto, vehículo en nuestro caso, en futuras tramas de video dado cierto conocimiento de su posición en el momento actual. De manera más específica, un filtro de Kalman realiza la estimación del estado de un determinado proceso bajo estudio en un determinado instante de tiempo, en nuestro caso la posición de los vehículos detectados en cada trama de video, actualizando posteriormente dichos resultados a partir de información proveniente de determinadas mediciones (fase de PVD). De este modo, nuestro sistema es capaz de predecir la posición/tamaño de los vehículos presentes en una trama de terminada a partir de la posición/tamaño de dichos vehículos en la trama anterior a partir de las *time update equations* que definen el filtro presentadas como se ha dicho en el Capítulo 2.4. Posteriormente, a través de las *measurement update equations* esta estimación *a priori* es actualizada a partir de la información proporcionada por la fase de detección de vehículos PVD para la presenta trama, dando lugar a una mejorada y más completa estimación definitiva de la posición/tamaño de los vehículos considerados por el modulo de seguimiento en dicha trama.

De esta manera, una vez la detección de un vehículo es considerada de confianza este proceso de seguimiento es inicializado, asignándosele como antes se mencionada un identificador único e intransferible que identifica de manera unívoca a dicho vehículo a lo largo de lo que dure la secuencia de vídeo. Y así para cada vehículo formalmente detectado en cada trama.

A partir de que el proceso de inicialización del modulo de seguimiento ha sido completada, para cada nueva trama se lleva a cabo un proceso de comparación entre las detecciones generadas para esa trama y los vehículos bajo seguimiento; y en este punto diferentes escenarios pueden tener lugar. En el caso de que exista coincidencia entre pares, el modulo de seguimiento simplemente ejecuta de manera natural la actualización de posición/tamaño para dicho vehículo. Sin embargo, si una detección presente en la nueva trama no coincide con ninguno de los vehículos bajo seguimiento en ese momento significa que se trata de hecho de una *nueva detección* para la cual se ha de poner en marcha el proceso de inicialización de seguimiento mencionado anteriormente.

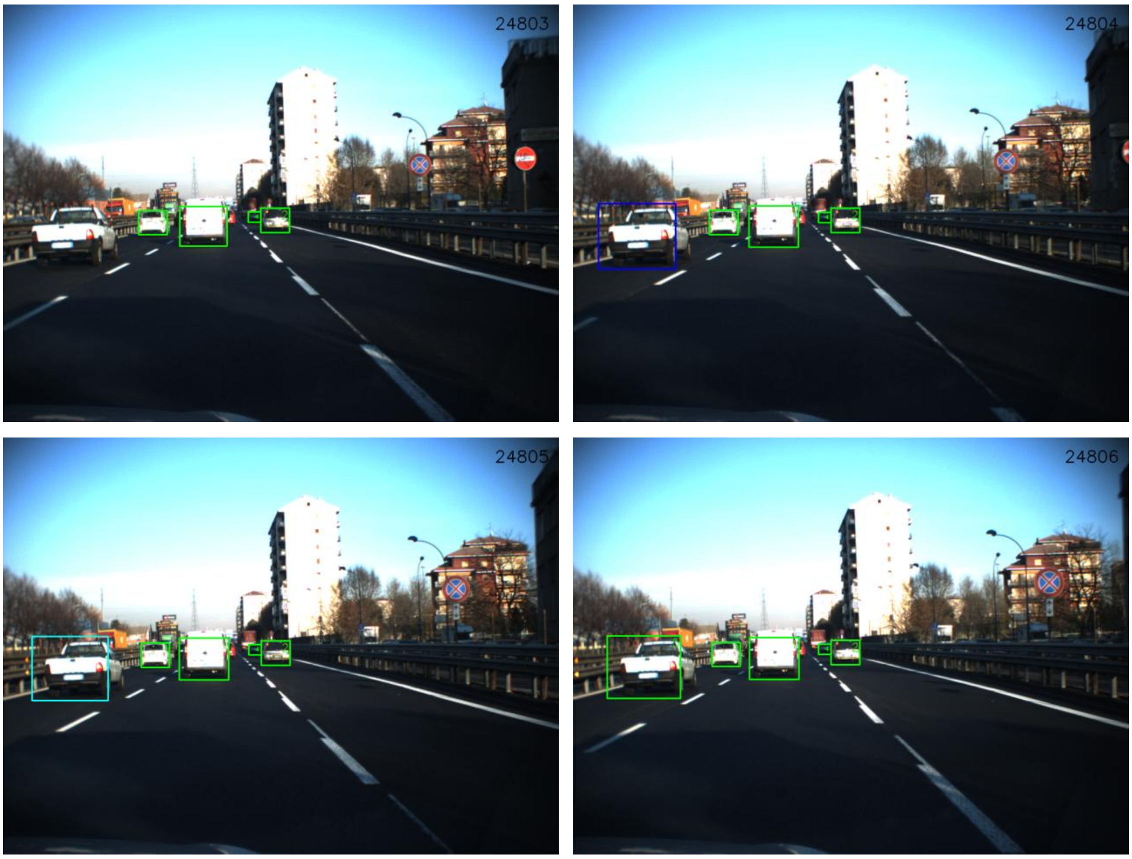


Figura 4.5: Ejemplo de Nueva Vehículo Detectado

Esta figura muestra como en una escena donde 4 vehículos están bajo el control del módulo de seguimiento, uno nuevo entra en escena. Como se ha mencionado en la sección anterior, este nuevo vehículo no es considerado como detección formal hasta que es consistente a lo largo de tres tramas. En este sentido se puede ver como el color de su ROI varía desde el azul oscuro, pasando por azul más claro hasta llegar a verde claro donde pasa a formar parte del modulo de seguimiento al igual que el resto de vehículos en la escena.

1.4 Pérdida de Seguimiento

Una tercera situación puede tener lugar, y este es el caso en el que un vehículo bajo seguimiento no sea detectado en la siguiente trama, hecho que puede deberse a dos razones diferentes: temporal detección errónea de la fase de detección, o pérdida real del vehículo bajo seguimiento bien porque ya no se encuentra dentro de nuestro campo de visión o porque se encuentra a una distancia tan alejada que el sistema no es capaz de reconocerlo.

Se ha de notar que los errores esporádicos presentes en la fase de detección no son persistentes en el tiempo, es decir únicamente bajo circunstancias muy concretas se suceden de manera consecutiva en más de una trama, ya que normalmente tienen lugar debido a, entre otras razones, la obstrucción causada por otro agente externo como puede ser una señal de tráfico o un cambio muy abrupto en la iluminación de la escena tal y como se muestra en la siguiente figura.

Figura 4.6: Pérdida de un vehículo bajo seguimiento

De modo que cuando las razones detrás de la pérdida esporádica del seguimiento de un vehículo, la naturaleza predictiva del propio filtro de Kalman que compone el modulo de seguimiento permite estimar la posición de este vehículo no detectado gracias a la información que conoce de él, y así evitar la pérdida de control sobre éste. No obstante, un vehículo cuya detección es pasada por alto de manera consecutiva a lo largo de diferentes tramas únicamente puede ser debido a la pérdida real del seguimiento del mismo, con independencia del motivo detrás de este hecho (vehículo que toma un desvío en la carretera, otro vehículo en maniobras de adelantamiento, demasiada distancia con respecto al vehículo propio, etc). En dichos casos, un criterio determinado por dos tramas consecutivas siendo no detectado supone la finalización del proceso de seguimiento llevado a cabo sobre tal vehículo. Obviamente, dicho vehículo podría ser de nuevo detectado por el sistema más adelante siempre que satisfaga las condiciones para ello, volviendo a formar parte del sistema de seguimiento de manera natural.

1.5 Conclusiones

Este capítulo revela la importancia crítica del procesamiento de la información temporal presente en secuencias de video, especialmente cuando se trata de escenario que cambian de manera dinámica como es el caso de las escenas de tráfico.

En primera instancia, un algoritmo muy simple para contrastar la consistencia de las detecciones originadas en la fase PVD ha sido presentado, probando a través de ejemplos reales su utilidad para mejorar el comportamiento global del sistema disminuyendo la tasa de falsos positivos considerados por éste.

Asimismo, uno de los algoritmos núcleo en el contexto de esta tésis para seguimiento de vehículos basado en filtrado Kalman ha sido propuesto en detalle. Una vez más a través de casos reales la mejora del sistema gracias a éste ha sido puesta de manifiesto. De manera más específica, la precisión espacial en la detección de vehículos se ve favorecida ya que por cada trama proporciona información adicional a tener en cuenta cuando dichas detecciones son consistentes en el tiempo. Además, más allá de suavizar el posible ruido presenta en la fase de detección, permite reducir la tasa de fallos esporádicos presentes en la fase de detección interpolando la posición de posibles vehículos afectados por estas irregularidades temporales gracias a la naturaleza predictora de los filtros Kalman, lo cual en términos generales incrementa la robustez y fiabilidad del sistema.

Finalmente se ha denotar que el sistema propuesto permite una observación a largo plazo tanto de la posición como del tamaño de los vehículos detectados a lo largo de la secuencia de video, lo cual en última instancia permite describir la trayectoria de los mismos, y asimismo a partir de mayor análisis tanto la distancia como la velocidad relativa de éstos respecto a vehículo propio respectivamente; información necesaria sin la cual el buscado sistema TTC de predicción del tiempo de colisión que esta tesis persigue y el cual es presentado en detalle en el siguiente capítulo no sería realizable.

Véase **Memoria Original del TFM** para acceso al informe completo del presente capítulo - *Chapter IV: Tracking Module.*

Capítulo V
Sistema TTC (Time-To-Collision)

5.1 Introducción

En general el Tiempo-hasta-Colisión o Tiempo-de-Contacto (TTC), es una medida de gran interés en campos que van desde la psicología experimental hasta la robótica [25], la cual dados dos cuerpos en el espacio es una medida del ratio dada por la distancia y la velocidad relativa entre ellos respectivamente. Llevado a un contexto que involucra cámaras de video, el TTC queda definido como el tiempo requerido por un objeto en el mundo real para alcanzar el plano de la cámara, supuesto una velocidad relativa entre ambos fija a lo largo de ese periodo de tiempo. Asimismo, en el plano de la imagen, su cálculo puede considerarse equivalente al estudio del tamaño del determinado objeto bajo consideración, así como del ratio de cambio de este con el tiempo. De esta manera el TTC puede ser obtenido en su totalidad en el plano de la imagen sin la necesidad de realizar mayores medidas experimentales (velocidad, distancia) o de calibración de la cámara. Esta relativa simplicidad en su cálculo hace de la estimación del Tiempo-de-Collisión una medida ideal para su implementación en sistemas que trabajan en tiempo real, como es el propuesto por esta tésis, donde decisiones deben poder ser tomadas en reducidos márgenes de tiempo frente a la posibilidad de situaciones de peligro inminentes.

En este capítulo es finalmente presentado el sistema propuesto por esta tesis para la estimación del Tiempo-de-Colisión a partir de la información captada por una única cámara situada en el interior del vehículo propio, cuya base reside en los sistemas de detección y seguimiento de vehículos respectivamente presentados hasta ahora. Los principios en los que se fundamenta el sistema TTC propuesto son revisados antes de pasar a la descripción detallada del mismo.

5.2 Principios TTC

Como se ha mencionado en la introducción, el Tiempo-de-Colisión puede ser definido como el tiempo que pasará desde que un observador haga contacto con una superficie respecto a la cual existe una velocidad relativa constante a priori desconocida [26]. En consecuencia puede ser definido como la distancia entre dos puntos de la imagen dividida por el ratio de cambio de dicha distancia.

Matemáticamente [27], el tiempo de contacto es una relación distancia-velocidad, y por tanto es normalmente expresado en términos de la velocidad y la distancia del obstáculo considerado a evitar con respecto a la cámara situada en el vehículo propio.

Así, la ecuación clásica para estimar el TTC es:

$$\tau = -\frac{Z}{dZ/_{dt}} = -\frac{1}{\frac{d}{dt}log_e(Z)} \tag{5.1}$$

Donde Z es la distancia entre la cámara y el obstáculo, y dZ/dt la velocidad relativa a la que el objeto se mueve con respecto a la cámara (será negativa si el objeto se acerca a ésta). Sin embargo, con una sola cámara distancia y velocidad no pueden ser conocidas sin información adicional. En dichos casos es posible derivar (5.1), y realizando ciertas asunciones, utilizar el tamaño del obstáculo s como característica clave para estimar el TTC como una relación entre dicho tamaño s de la imagen del objeto, y el ratio de cambio de dicho tamaño con el tiempo ds/dt. Matemáticamente:

$$\tau = \frac{s}{ds/_{dt}} = \frac{1}{\frac{d}{dt}log_e(s)} \tag{5.2}$$

Esto reformula el problema de la estimación del tiempo de contacto a un problema de estimación del tamaño del objeto y del ratio de cambio de éste; enfoque considerado en el desarrollo de esta tesis.

5.3 Sistema Propuesto

Como fue visto en el Capítulo IV, el modulo de seguimiento implementado basado en filtrado Kalman permite una observación a largo plazo tanto de la posición como del tamaño de los vehículos detectados a lo largo de la secuencia de video, los cuales quedan definidos por su ROI.

En este sentido, y gracias al sistema de seguimiento propuesto, el problema del cálculo del TTC reformulado como un problema de estimación del tamaño de los vehículos detectados y del ratio de cambio de este tamaño a lo largo del tiempo es fácilmente planteable ya que como se ha visto dicho modulo nos da acceso al tamaño de estos a lo largo de la secuencia de video. Así, dado el tamaño conocido de un vehículo determinado en una trama concreta s_x , el ratio de cambio de éste con respecto a la trama anterior $x-1$ queda definido de la siguiente manera:

$$Rate\ of\ Change\ (pixels/trama) = s_x - s_{x-1} \tag{5.3}$$

Asimismo, de acuerdo a las definiciones de TTC presentadas con anterioridad, este puede definirse como el tiempo que pasa desde que el plano de cámara alcanza la superficie bajo observación, considerada constante el movimiento relativo entre ambos durante ese tiempo. En el plano de la imagen esto es equivalente a decir que el hipotético contacto entre ambos tendría lugar cuando el objeto que está siendo observado ocupase el total de la imagen. En la práctica, a fin de que el riesgo de impacto sea considerado como real por nuestro sistema, y dado que el fin último del mismo es que éste no llegue a producirse, un umbral o límite ha de ser considerado, el cual en nuestro caso es fijado en un 80% del total del tamaño de adquisición de las imágenes. De este modo, conocido el tamaño de la imagen $S\ (pixels)$, y el ratio al que esta cambia $RoC\ (pixels/trama)$, una primera estimación del Tiempo-hasta-Colisión TTC en tramas viene dado por la siguiente expresión:

$$\mathcal{TTC}\ (tramas) = \frac{S\ (pixels)}{Rate\ of\ Change\ (pixels/trama)} \tag{5.4}$$

Y una vez en este punto, teniendo en cuenta que la frecuencia de adquisición de la cámara considerada en esta tesis es conocida (10 Hz = 10 fps), la estimación del TTC puede ser dada en unidades de tiempo (seg).

$$\boldsymbol{\mathcal{TTC}\ (seg)} = \frac{\boldsymbol{\mathcal{TTC}\ (tramas)}}{\boldsymbol{Frecuencia\ Adq.\ (fps)}} \tag{5.5}$$

La siguiente figura recoge un caso real de estimación del Tiempo-de-Contacto respecto a un vehículo en la carretera en frente nuestro. El límite considerado antes para el cómputo de la estimación es resaltado en rojo; como se ha mencionado antes, representa en el plano de la imagen el tamaño que el vehículo bajo seguimiento debería alcanzar para considerar una potencial colisión con este como inminente.

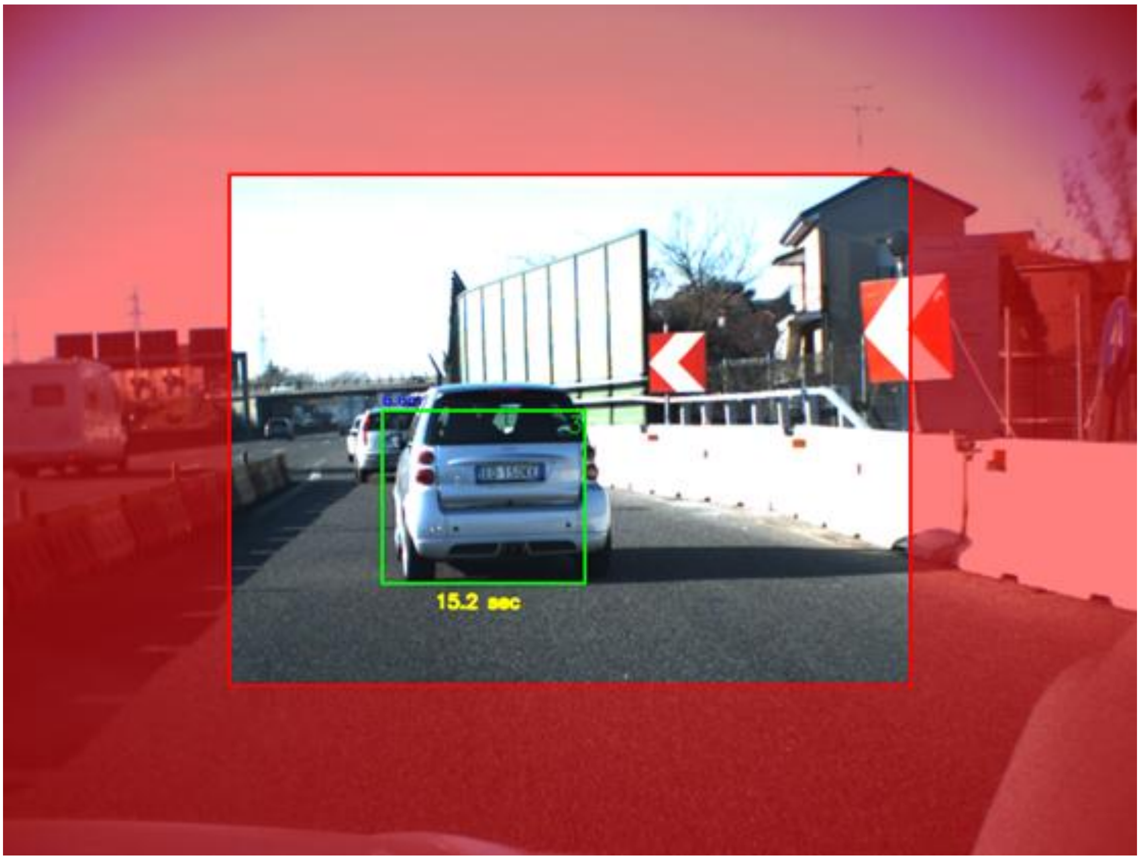


Figura 5.1: Caso real de Estimación del Tiempo-de-Contacto (TTC)

Además, más allá del cálculo del TTC, es posible realizar una estimación a *grosso modo* de la distancia relativa entre ambos vehículos (azul) realizando varias asunciones. Un cálculo preciso de ésta es prácticamente inviable ya que la información contenida en una única trama de video no es suficiente para derivar de manera precisa información relativa a la profundidad en la imagen. No obstante, una buena aproximación para vehículos de un tamaño estándar es posible. Asumiendo la anchura y altura media de un vehículo como aproximadamente 1.8m y 1.5m respectivamente, y conocida el tamaño del vehículo en el plano de la imagen en proporción al tamaño total de la imagen, una estimación de la mencionada distancia entre ambos puede ser obtenida.

En este sentido, y una vez más, todos estos cálculos tanto de distancias como principalmente de tiempo de contacto quedan directamente condicionados por la precisión en la fase de detección de vehículos, ya que de cara al sistema propuesto éstos quedan desde entonces caracterizados por la ROI que los define en ese momento. Dicho esto, se ha de notar igualmente que este tipo de sistemas TTC, si bien cuanto más precisos sean mejor, no es requerida una precisión del 100% en términos reales para considerarlos funcionales. Dado que la idea final que acompaña a estos sistemas es ser integrados en el sistema de seguridad más completo de un vehículo como puede ser un sistema real anticolisiones CAS (Collision-Avoidance System), en última instancia capaz de evitar potenciales situaciones de riesgo directo de accidente de tráfico, siempre serán sistemas que consideren márgenes de seguridad suficientes ya que de ellos podría depender la vida de posibles víctimas. Esto quiere decir que, supuesto el caso real de riesgo de accidente entre el vehículo propio y otro en la carretera, el sistema propuesto no necesitaría conocer con exactitud la distancia o el tiempo de contacto exacto entre estos ya que en definitiva su función es evitar que éste ocurra, y por tanto siempre operará con márgenes de seguridad lo suficientemente amplios como para asegurar la integridad tanto de los vehículos como de las vidas humanas en riesgo.

Figura 5.2: Sistema TTC (Time-To-Collision)

Esta última figura ejemplifica el funcionamiento final del sistema TTc propuesto en diversos escenarios reales de tráfico. Como se puede apreciar en las diferentes imágenes, el sistema es capaz de llevar a cabo la detección de un gran número de vehículos presentes en la misma escena, realizar un seguimiento continuado de los mismos a lo largo del tiempo, y en última instancia realizar una estimación tanto de la distancia relativa de estos respecto al vehículo propio en metros, como del tiempo-de-contacto en segundos, considerando únicamente como sujetos de potencial impacto aquellos vehículos que se aproximan a nosotros (destacados con una *x* de color rojo), y no aquellos que se nos alejan. El sistema TTC propuesto demuestra de esta manera su satisfactoria implementación y funcionamiento.

5.4 Evaluación de la Velocidad de Procesado

Dado que el objetivo final de cualquier ADAS es ser implementado y funcionar en tiempo real, un estudio de la velocidad de procesado de cada módulo que compone el sistema TTC propuesto es llevado cabo en esta sección debido a su gran relevancia.

A efectos de este último análisis son considerados 10 clips de video diferentes, para un total de 100 tramas por cada uno de ellos. Todas las secuencias de video utilizadas pertenecen al TME Motorway Dataset, como ha venido siendo a lo largo de todo el proyecto, representando dos tipos de condiciones meteorológicas diferentes, a la luz del día y con la puesta de sol. Asimismo, las 100 tramas consideradas a propósito del análisis se corresponden con las 100 primeras tramas de cada una de las secuencias tal y como aparecen en el TME dataset. El estudio consiste por tanto en la medición de la duración media de procesamiento para cada uno de las secuencias de video, para posteriormente establecer una velocidad media total del sistema en fps (frames per second). El objetivo final de estos tests consiste en observar el comportamiento de cada uno de los módulos diseñados para concluir en última instancia si una implementación real del sistema propuesto sería en algún case factible. En este sentido el estudio persigue revelar posibles cuellos de botella presentes en el diseño del sistema, que por tanto representarían potenciales líneas de mejora del sistema en un futuro.

En este contexto se ha de notar que el sistema presentado ha sido implementado en su totalidad en el conocido lenguaje de programación Python, más en concreto haciendo uso del entorno de desarrollo Python(x,y),así como OpenCV como librería principal para tareas de procesado de imagen presentes en este proyecto. Optimización de algoritmos y códigos no ha sido especialmente tenida en cuenta, y todos los tests recogidos en este informe han sido llevados a cabo en un ordenador portátil HP Pavilion (64-bit OS y Windows 10) con procesador 1.8GHz Intel® Core™ i3-3217U, y 4 GB de RAM.

Finalmente, los resultados obtenidos son presentados en la siguiente tabla para cada uno de los 10 clips de ideo analizados, considerando las siguientes abreviaturas para mejor comprensión de los mismos: VD (vehículos detectados), HV (verificación de hipótesis), DC (consistencia de la detección), Tracking (módulo de seguimientio), y TTC para el módulo de estimación del tiempo de colisión propiamente dicho.

Tabla 5.1: Resultados Velocidad de Procesado

Video Sequence	VD	HV	DC	Tracking	TTC	time (ms)	fps
daylight_12_1284	14	49	5	91	6	151	**6.62**
daylight_17_24800	8	35	5	82	6	128	**7.81**
daylight_32_5160	13	46	5	92	7	150	**6.66**
daylight_35_8950	5	28	5	78	6	117	**8.54**
daylight_42_520	4	26	4	59	5	94	**10.63**
sunset_12_22610	7	33	5	72	6	116	**8.62**
sunset_16_23650	11	42	5	75	6	128	**7.81**
sunset_16_34200	10	39	5	79	6	129	**7.75**
sunset_46_16900	12	45	5	93	6	149	**6.71**
sunset_46_24670	10	41	5	82	7	135	**7.41**

Para cada una de las diez secuencias de video la tabla de resultados presenta (de izquierda a derecha) por un lado el número de vehículos detectados a lo largo de la secuencia de 100 tramas, por otro lado el tiempo de procesado medio de cada uno de los módulos presentados por trama y en milisegundos, y finalmente el tiempo de procesado medio total del sistema por trama en milisegundo, y en consecuencia una medida del número medio de tramas por segundo (pfs) a la que el sistema es capaz de trabajar.

A continuación los resultados finales obtenidos, media de los resultados individuales obtenidos para las diez secuencias de video, son presentados y analizados en detalle., representando en definitiva el desempeño medio total del sistema propuesto.

		HV	DC	Tracking	TTC
Average Results	ms	38.4	4.9	80.3	6.1
	%	29.6	3.8	61.9	4.7
	time (ms)	129.7 ms			
	fps	7.71 fps			

Tabla 5.2: Resultados Finales Funcionamiento del Sistema

Los resultados obtenidos muestran como el módulo de seguimiento, por otra parte esperado, representa la núcleo principal del sistema TTC suponiendo un 60% del tiempo total de procesado requerido por éste. La fase de comprobación de la consistencia de las detecciones propuestas por HV, como preámbulo del módulo de seguimiento de acuerdo a los resultados representa en ese sentido el algoritmo menos pesado del sistema, a niveles similares del módulo de estimación TTC, funcionamiento el cual sabemos descansa y depende directamente del propio módulo de seguimiento, de ahí que su tiempo de procesado sea prácticamente despreciable a pesar de su evidente importancia crucial en el sistema diseñado. Por otro lado la etapa de verificación de hipótesis como parte del módulo de detección de vehículos propuesto representa la otra gran parte del sistema en cuanto a consumo de recursos se refiere.

Analizando los resultados obtenidos se puede ver como la duración tanto del módulo HV como del de seguimiento está directamente relacionada con el número de vehículos detectados a lo largo de la secuencia, y en definitiva, un ratio medio en la velocidad de procesado de 7.71 fps por parte de nuestro sistema puede ser considerado como un resultado satisfactorio en dirección a la implementación de un sistema capaz de trabajar en tiempo real

5.5 Conclusiones

Este último Capítulo V ha presentado en detalle el módulo de estimación del Tiempo-de-Colisión que en última instancia completa el sistema TTC propuesto por esta tesis. Una vez introducidos los principios por los que cualquier sistema Time-to-collision se rige, la solución propuesta por esta tesis ha sido presentado y mostrada en detalle a través de un extenso conjunto de ejemplos de imágenes de situaciones y escenas de tráfico reales.

Tal y como se ha podido ver, el sistema TTC propuesto es capaz de elaborar con éxito una estimación del tiempo-de-colisión en segundos entre el vehículo propio y cualquier otro presente en la escena de tráfico analizadas, previamente considerado por el módulo de seguimiento implementado en el Capítulo IV. Algunas consideraciones tomadas en cuenta, el sistema también es capaz de dar una estimación de la distancia en metros a los respectivos vehículos bajo seguimiento. De esta manera, los vehículos de más de 50 metros del vehículo propio sólo son rastreados, mientras que los esfuerzos del estimador TTC se centran en los vehículos más cercanos al propio, ya que en cualquier caso son respecto a los cuales existe mayor probabilidad de potencial impacto.

En última instancia este capítulo presenta una evaluación final del sistema TTC propuesto en términos de rendimiento y velocidad de procesado, ya que condición imprescindible para cualquier sistema ADAS es su funcionamiento en tiempo real. A tal fin diferentes tests han sido llevados a cabo, obteniendo una velocidad de procesado media del sistema de 7,71 fps; resultados prometedores para la implementación del mismo bajo condiciones de tiempo real, lo cual nos permite afirmar que en líneas generales los principales objetivos a los que este TFM se ha enfrentado pueden ser considerados como alcanzados satisfactoriamente.

Véase Memoria Original del TFM para acceso al informe completo del presente capítulo - *Chapter V: Time-To-Collision System.*

Capítulo VI
Conclusiones & Trabajo Futuro

6.1 Resumen

Esta Tesis de fin de Máster ha propuesto el diseño de un Sistema Avanzados de Asistencia al Conductor ADAS basado en Visión Artificial, y más en concreto de un sistema TTC (Time-To-Collision) capaz de predecir el tiempo restante para producirse un posible impacto entre el vehículo propio y otros presentes en la carretera, a partir de imágenes adquiridas en tiempo real procedentes de una cámara instalada en el interior del vehículo mediante el empleo de técnicas de Visión Artificial. En última instancia, integrado en un Sistema Anticolisión (Pre-Crash System) o en combinación con un Sistema de Frenado de Emergencia (EBS, Emergency Braking System), el sistema propuesto podría dar lugar a la implementación real de un sistema anticolisiones CAS (Collision-Avoidance System) en última instancia capaz de evitar potenciales situaciones de riesgo directo de accidente de tráfico en escenarios reales de conducción en carretera en los que podría verse involucrada la integridad de vidas humanas.

A tal fin, tres módulos diferentes han sido propuestos; considerados en conjunto dan lugar al sistema TTC introducido. El primero de estos módulos consiste en un algoritmo para la detección de vehículos (Preceding Vehicles Detection (PVD)), el segundo consiste en un módulo de seguimiento de los vehículos detectados en la primera fase (detections tracking), y el último de estos módulos está formado por el estimador TTC (Time-To-Collision) propiamente dicho. Todos ellos han demostrado resultados satisfactorios.

De manera más concreta, el problema de detección de vehículos ha sido considerado como una aproximación en dos etapas. En primer lugar, una aproximación a diferentes escalas de detectores de bordes Sobel (horizontales y verticales) en combinación con una etapa de segmentación del color rojo da lugar a una fase de Generación de Hipótesis que establece en primera instancia posibles localizaciones de vehículos en la escena analizada. Posteriormente, este modulo de detección de vehículos PVD confía principalmente en el empleo de una serie de filtros de Gabor para la extracción de características a diferentes orientaciones y escalas. Estas características guardan la información principal que modela la estructura de un vehículo, atendiendo a diferentes variaciones y de manera robusta frente a la iluminación global de la imagen. De manera más específica, nuestro algoritmo realiza la extracción de características sobre sub-ventanas de la imagen de entrada (no de manera global) y las representa a través de medidas estadísticas como la media o la desviación estándar, para más tarde concluir la Verificación de Hipótesis a través de un clasificador VSM propiamente entrenado. Se ha de notar que limitaciones temporales relacionadas con los plazos en que este proyecto ha sido desarrollado no han permitido concluir en su totalidad la etapa de GH y considerarla como parte activa del sistema propuesto, quedando así como línea de trabajo futuro. No obstante, la etapa de VH por otro lado ha sido extensamente evaluada mediante datasets de acceso público con la intención de obtener las mejores prestaciones del sistema, mostrando en última instancia resultados positivos en términos de precisión en la detección de vehículos.

Posteriormente, un modulo de seguimiento basado en filtrado de Kalman ha sido propuesto, el cual en combinación con el sistema PVD presentado anteriormente ha demostrado mejorar las características del sistema a diferentes niveles. En primera instancia, la precisión espacial en la detección de vehículos se ve favorecida ya que por cada trama proporciona información adicional a tener en cuenta cuando dichas detecciones son consistentes en el tiempo. Además, más allá de suavizar el posible ruido presenta en la fase de detección, permite reducir tanto la tasa de falsos positivos mediante una etapa de verificación de la detección de las consistencia de las detecciones propuestas, como la tasa de fallos esporádicos presentes también en la fase de detección interpolando la posición de posibles vehículos afectados por estas irregularidades temporales gracias a la naturaleza predictora de los filtros Kalman, lo cual en términos generales incrementa la robustez y fiabilidad del sistema. En última instancia el sistema propuesto permite una observación a largo plazo tanto de la posición como del tamaño de los vehículos detectados a lo largo de la secuencia de video, lo cual hace posible describir la trayectoria de los mismos, así como la distancia y la velocidad relativa a estos; información de vital importancia gracias a la cual el buscado sistema TTC de predicción del tiempo de colisión que esta tesis propone ha sido realizable.

Finalmente, a partir de los sistemas presentados anteriormente un estimador TTC (Time-To-Collision) capaz de predecir el tiempo restante hasta producirse un posible impacto entre el vehículo propio y otros en la carretera has sido exitosamente presentado. En la implementación de éste el procesamiento de la información temporal presente en secuencias de video ha jugado un rol de crucial importancia. Más en concreto la solución propuesta se fundamente en el análisis del tamaño de las detecciones de realizadas, así como de su ratio de variación a lo largo del tiempo. Por otra parte, se ha visto que un cálculo preciso de la distancia entre vehículos es prácticamente inviable ya que la información contenida en una única trama de video no es suficiente para derivar de manera precisa información relativa a la profundidad en la imagen. No obstante, consideraciones aparte, una buena aproximación para vehículos de un tamaño estándar ha sido igualmente alcanzada.

Finalmente, el sistema TTC propuesto ha demostrado un comportamiento más que aceptable en términos de rendimiento y velocidad de procesado; resultados prometedores para la implementación del mismo bajo condiciones de tiempo real, lo cual permite afirmar en última instancia que en líneas generales el presente Trabajo de Fin de Máster a resuelto de manera exitosa los principales objetivos y retos para los que fue concebido.

6.2 Líneas de Trabajo Futuro

Al margen del hecho de que los objetivos propuestos desde un inicio por esta tesis pueden ser considerados satisfechos, algunas sugerencias, tanto para mejora del sistema propuesto como en forma de líneas futuras de trabajo, son presentadas a continuación.

La primera de ella, ya mencionada a lo largo de este proyecto, consiste en completar el modulo de Generación de Hipótesis propuesto como parte del sistema de detección de vehículos (PVD) presentado en el Capítulo III, el cual como es explicado en éste, no puedo ser finalizado e incluido como parte del sistema TTC presentado debido a limitaciones de tiempo en relación a los plazos asignados al desarrollo del presente proyecto. No obstante, una solución ha sido propuesta y parte del trabajo ha sido iniciado, por lo que a este respecto es cuestión de tiempo y dedicación la consecución del mismo.

Además, como es presumible, todos los sistemas propuestos en este TFM son susceptibles de mejora, tanto en términos de precisión como de velocidad, teniendo en cuenta la condición de trabajar en tiempo real que presenta nuestro sistema. En este sentido, cualquier mejora tanto a nivel de diseño como de optimización de algoritmos y programación representa una interesante línea de trabajo futura donde seguir dedicando recursos.

Y último pero no por ello menos importante, uno de los objetivos principales comunes a cualquier sistema ADAS es presentar la mayor robustez posible ante cualquier escenario posible. Ante la enorme variedad de situaciones de tráfico que pueden tener lugar en la carretera, no es sin embargo de esperar que la solución aquí propuesta mantenga los mismos niveles de precisión en todas ellas. El sistema propuesto ha sido de hecho entrenado y evaluado bajo dos tipos diferentes de condiciones meteorológicas, a la luz del día (considerando la presencia de sombras en la carretera) y al atardecer enfrentando el sol con un grado de visibilidad más reducido. En esta línea, estudios adicionales podrían llevarse a cabo a fin de diseñar un sistema lo más robusto posible igualmente ante la presencia de otro tipo de condiciones adversas, y por lo tanto más exigentes para el sistema, como podrían ser casos de lluvia, nieve, niebla o conducción en la noche entre otros.

Todo esto en consideración, y tal y como ha sido puesto de manifiesto en diferentes ocasiones a lo largo del presente informe, esta Tesis de fin de Máster ha propuesto el diseño de un Sistema Avanzados de Asistencia al Conductor ADAS basado en Visión Artificial, y más en concreto de un sistema TTC (Time-To-Collision) capaz de predecir el tiempo restante para producirse un posible impacto entre el vehículo propio y otros presentes en la carretera, a partir de imágenes adquiridas en tiempo real procedentes de una cámara instalada en el interior del vehículo mediante el empleo de técnicas de Visión Artificial, con la intención en última instancia de ser integrado en un Sistema Anticolisión (Pre-Crash System), un Sistema de Frenado de Emergencia (EBS, Emergency Braking System) o similares, a fin de dar lugar a la implementación real de un sistema anticolisiones CAS (Collision-Avoidance System) en definitiva capaz de evitar potenciales situaciones de riesgo directo de accidente de tráfico en escenarios reales de conducción en carretera en los que podría verse involucrada la integridad de vidas humanas. De acuerdo con esto el sistema propuesto podría ser considerado e implementado de manera independiente en lo que respecta a la tecnología ADAS, sin embargo el estudio de su comportamiento en combinación con otro tipo de sistemas basados en visión artificial como los presentados en el Capítulo 1.4 (véase sistemas de reconocimiento de señales de tráfico, semáforos y peatones, o de visión nocturna entre otros), al igual que en combinación con información proveniente de otras fuentes basadas en diferentes tecnologías como pueden ser sensores Radar o Lidar, a fin de alcanzar los mejores resultados posibles, representa uno de las líneas de investigación futuras más prometedoras.

Finalmente, el uso de información proveniente de una única cámara situada en el interior del vehículo propio, escenario considerado en esta tesis, representa en sí mismo y al mismo tiempo tanto un desafío como un punto de valor extra añadido al sistema propuesto. En este sentido, futuras aproximaciones podrían considerar el empleo de un mayor número de cámaras migrando así a un sistema en estéreo con todas las implicaciones que ello conlleva. O por el contrario, siendo fieles al esquema de una sola cámara de adquisición de imágenes y video, otras posibilidades podrían ser consideradas. Este es el caso de VALEO Vision Systems, donde actualmente cámaras con gran angular conocidas como *bird-eye cameras* son consideradas en la implementación real de sus sistemas comerciales. Algunas imágenes tomadas con este tipo de cámara son mostradas en la siguiente figura.

Figura 6.1: Imágenes de Valeo Vision Systems: Parte frontal–trasera del vehículo (fila superior); Lateral izquierdo–derecho del vehículo (fila inferior)

Como se puede apreciar la característica más llamativa de este tipo de cámaras frente a las convencionales es que proporcionan un campo de visión de hasta 180º, mucho más amplio que el que las anteriores o las consideradas en el contexto de esta tesis presentan. Es por ello que el empleo de este tipo de cámaras y el aprovechamiento de las ventajas que ellos supone representa en sí mismo el desafío extra de caracterización y compensación de la distorsión que introducen. Así, la consideración de estas o cualquier otro equipamiento más sofisticado en la adquisición de imágenes/video representan interesantes líneas de investigación y desarrollo con las que continuar el trabajo propuesto por esta tesis a lo largo del presente informe.

Véase **Memoria Original del TFM** para acceso al informe completo del presente capítulo - *Chapter VI: Conclusions & Future Work.*

Bibliografía

[1] M. Zhao, "Advanced Driver Assistant System: Threats, Requirements, and Security Solutions", Security & Privacy Research, Intel Labs, 2015. [http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/advanced-driver-assistant-system-paper.pdf] Acceso: Julio 2016.

[2] S. Sivaraman, and M. M. Trivedi, "Looking at vehicles on the road: a survey of vision-based vehicle detection, tracking, and behavior analysis", IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 4, pp. 1773–1795, December 2013.

[3] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: a review", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 5, pp. 694–711, May 2006.

[4] R. A. Hadi, G. Sulong, and L. E. George, "Vehicle detection and tracking techniques: a concise review", Signal & Image Processing: An International Journal (SIPIJ), vol.5, no.1, February 2014.

[5] R. van der Horst, and J. Hogema, "Time-To-Collision and collision avoidance systems", Proceedings of the 6th ICTCT workshop Salzburg, TNO Institute for Human Factors, The Netherlands, 1994.

[6] K. D. Kusano, and H. Gabler, "Method for estimating time to collision at braking in real-world, lead vehicle stopped rear-end crashes for use in pre-crash system design", SAE International, Virginia Tech, December 2011.

[7] M. Betke, E. Haritaoglu, and L. S. Davis, "Real-time multiple vehicle detection and tracking from a moving vehicle", Machine Vision and Applications, vol. 12, no. 2, pp. 69–83, 2000.

[8] D. Ponsa, J. Serrat, and A. M. Lopez, "On-board image-based vehicle detection and tracking", Computer Vision Center and Computer Science Department, University of Barcelona, Transactions of the Institute of Measurement and Control 33, 7 (2011) pp. 783–805.

[9] Andrew Ng, "Machine Learning", Data Science, Stanford Univ., Coursera. [https://en.coursera.org/learn/machine-learning] Acceso: Julio 2016.

[10] "An introduction to machine learning", Scikit Learn. [http://scikit-learn.org/stable/tutorial/basic/tutorial.html] Acceso: Julio 2016.

[11] D. Zhang, A. Wong, M. Indrawan, and G. Lu, "Content-based Image Retrieval Using Gabor Texture Features", Gippsland School of Computing and Information Technology, Monash University, Australia.

[12] P. Moreno, A. Bernardino, and J. Santos-Victor, "Gabor Parameter Selection for Local Feature Detection", IBPRIA, 2nd Iberian Conference on Pattern Recognition and Image Analysis, Estoril, Portugal, June 2005.

[13] A. Nurhadiyatna, A. L. Latifah, and D. Fryantoni, "Gabor Filtering for Feature Extraction in Real Time Vehicle Classification System", Research Center for Informatics, Indonesian Institute of Sciences, Cibinong & Bandung.

[14] G. Bishop, and G. Welch, "An introduction to the kalman filter", Department of Computer Science University of North Carolina, Chapel Hill, SIGGRAPH, Course, vol. 8, 2001.

[15] R. E. Kalman, "A new approach to linear filtering and prediction problems", Research Institute for Advanced Study, Baltimore, Transactions of the ASME, Journal of basic Engineering, vol. 82, no. 1, pp. 35–45, 1960.

[16] S. S. Beauchemin and J. L. Barron, "The computation of optical flow", Department of Computer Science, University of Western Ontario, Canada, ACM Computing Surveys (CSUR), vol. 27, no. 3, pp. 433–466, 1995.

[17] F. Raudies, "Optic flow", Boston University, Boston, MA, USA, Scholarpedia, the peer-reviewed open-access encyclopedia, 2013. [http://www.scholarpedia.org/article/Optic_flow] Acceso: Julio 2016.

[18] N. Khairdoost, S. A. Monadjemi, and K. Jamshidi, "Front and Rear Vehicle Detection using Hypothesis Generation and Verification", Department of Computer Engineering, Faculty of Engineering, University of Isfahan, Signal & Image Processing: An International Journal (SIPIJ), vol.4, no.4, Aug 2013.

[19] Q. B. Truong, and B. R. Lee, "Vehicle Detection using Hypothesis Generation and Verification", Department of Mechanical and Automotive Engineering, School of Mechnical and Automotive Engineering, University of Ulsan, Korea.

[20] Z. Sun, G. Bebis, and R. Miller, "Monocular Pre-crash Vehicle Detection: Features and Classifiers", Computer Vision Lab. Department of Computer Science, University of Nevada, Reno, Technology Department, Ford Motor Company, Dearborn, MI.

[21] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using gabor filters and support vector machines", 14th International Conference on Digital Signal Processing, 2002, vol. 2, pp. 1019–1022.

[22] TME Motorway Dataset. [http://cmp.felk.cvut.cz/data/motorway/] Acceso: Julio 2016.

[23] C. Caraffi, T. Vojír, J. Trefný, and J. Matas, "A system for real-time detection and tracking of vehicles from a single car-mounted camera", TME Motorway Dataset, ITS Conference, pp. 975—982, September 2012.

[24] Caltech Computational Vision: Archive, Cars 1999 & Cars 2001, March 17th, 2005. [http://www.vision.caltech.edu/html-files/archive.html] Acceso: Julio 2016.

[25] S. Pundlik, E. Peli, and G. Luo, "Time to collision and collision risk estimation from local scale and motion", Proceedings of the 7th international conference on Advances in visual computing, vol. part I, pp. 728-737, Schepens Eye Research Institute, Harvard Medical School, Boston, MA, 2011.

[26] G. Alenyà, A. Nègre, and J. L. Crowley, ‶Time To Contact for Obstacle Avoidance″, Institut de Robòtica I Informàtica Industrial, CSIC-UPC, Barcelona, INRIA Grenoble Rhône-Alpes Research Centre, France.

[27] B. K. P. Horn, Y. Fang, and I. Masaki, ‶ Time to Contact Relative to a Planar Surface″, Department of Electrical Engineering and Computer Science, CSAIL, MIT, Cambridge, MA, IEEE Intelligent Vehicles Symposium, pp. 68-74, June 2007.

[UPCT] Universidad Politécnica de Cartagena, España. [http://www.upct.es/]

[UOPA] Universidad de Patras, Grecia. [http://www.upatras.gr/en]

[E&CE] Departamento de Ingeniería Eléctrica e Informática, Universidad de Patras, Grecia. [http://www.ece.upatras.gr/en/]

[VALEO] Valeo Vision Systems, Irlanda. [http://www.valeovision.com/]

Memoria Original del TFM

Master Thesis Dissertation

Design of a Computer Vision based TTC (Time-To-Collision) ADAS (Advanced Driver Assistance System)

RUBÉN LAENCINA ESCOBAR

University of Patras

Department of Electrical and Computer Engineering
Greece, September 2016

Rubén Laencina Escobar

Design of a Computer Vision based TTC (Time-To-Collision) ADAS (Advanced Driver Assistance System)

Directed by: **Prof. ATHANASSIOS SKODRAS**

University of Patras, Department of Electrical and Computer Engineering

Co-directed by: **Dr. GEORGE SIOGKAS**

Valeo Vision Systems, Research and Development Department

University of Patras, Greece
Valeo Vision Systems, Ireland

Master Thesis Dissertatio

To my family, and to the years about to come.

Abstract

During the past few decades, road accidents have proven to be one of the most common causes for the loss of human lives, counting them by millions. Consequently, vehicle safety has steadily improved over the years, mainly because of the advances of passive safety systems (e.g., seat belts, airbags or improvements in their crashworthiness for passengers and pedestrians) that have been included in almost all commercial vehicles. However, today's safety systems are designed to help vehicle passengers not only escape injury during an accident, but actually help prevent the accident in the first place.

Advanced Driver Assistance Systems (ADAS) are systems that provide assistance to the driver and improve driving experience, being its primary function to ensure safety of the vehicle and other vehicles on the road, the driver, and the pedestrians or bikers. And so, demand for ADAS in recent years caused by desire to build safer vehicles and roads in order to reduce the number of road fatalities has been greatly increased. To function reliably, ADAS must be able to recognize objects, signs, road surface, and moving objects on the road and to make decisions whether to warn or act on behalf of a driver in real time. Therefore, in order to guarantee robust and reliable systems, commercial ADAS normally combine different technologies such radar, LIDAR, ultrasonic, photonic mixer device (PMD), cameras, and night vision devices, since it's proven sensor fusion achieves higher system performance than independent systems in order to ensure vehicle, driver, passengers and pedestrians safety. Modern ADAS systems are precursors of the autonomous vehicles of the future.

Furthermore, since technological breakthroughs allowed the incorporation of cheap, low consumption systems with high processing speeds in vehicles, it became apparent that complex computer vision techniques could be used to reliably assist drivers in navigating their vehicles. In this direction, this thesis focuses on the design of a Computer Vision based Advance Driver Assistance System, for the more specific implementation of a Time-To-Collision (TTC) System, able to predict the remaining time to a potential impact taking place between the ego-vehicle and others present on the road, from images acquired in real time from an on-vehicle mounted front camera. Integrated in a Pre-crash System or in combination with an Emergency Breaking System (EBS) the proposed system might lead to the implementation of a real vehicle Collision-Avoidance System (CAS) that would eventually allow to avoiding potential dangerous situations of real risk of traffic accidents on the road within real driving scenarios involving human lives.

To say more, the system presented in this thesis proposes a Preceding Vehicle Detection (PVD) module, more specifically focused in the Hypothesis Verification (HV) stage, based on Gabor Filtering, for target description, in combination with Support Vector Machines (SVM) for vehicle classification purposes, aiming robustness to changes of illumination, weather conditions, and diverse driving environments. Additionally spatiotemporal tracking of detected results is implemented via Kalman Filtering and template-matching techniques, in order to eventually address the main challenge of real-time Time-To-Collision (TTC) estimation this thesis stands for. Several experiments and both quantitative and qualitative methods were considered to evaluate the proposed system.

Acknowledgements

My most sincere appreciations towards this Master Thesis are aimed without any distinction both to its director and co-director, Prof. Athanassios Skodras and Dr. George Siogkas.

Thanks Prof. Skodras for having believed and trusted on me as Erasmus+ student in your university to carry out with you what it turned out to be beautiful challenge and experience that goes beyond the academic and university walls. Thanks for opening the doors of your office in Patras and treating me from the very first beginning as one more of your students. Without your kindness and support none of this would have ever been possible.

And thanks George. Thank you so much for your encouragement, constant assistance, and wise feedback whenever I needed. Thanks for your efforts in making this possible besides the distance. And definitely thanks for the time spent in front of the screen. Your honest advice made us eventually go successfully through all this.

Introduction

This Master Thesis Project arises as final requirement towards the Title of Master in Telecommunications by Technical University of Cartagena (UPCT), Cartagena, Spain.

Titled *"Design of a Computer Vision based TTC (Time-To-Collision) ADAS (Advanced Driver Assistance System)"*, this project was fully developed within the framework of European Commission’s Erasmus+ programme at the University of Patras (UOPA), Patras, Greece, more specifically within the Department of Electrical and Computer Engineering (E&CE), and was directed by Professor Athanassios Skodras. Moreover, carried out in partnership with Valeo Vision Systems (VVS), Galway, Ireland, it was co-directed by Dr. George Siogkas from Research and Development Department.

As third largest university in Greece, the University of Patras celebrated in 2014 its 50th anniversary of academic operation, marked by continuous development and constantly emerging new achievements. Today, the University of Patras enjoys recognition as an Academic Institution with a worldwide impact, attracting thousands of students and a large number of Academic and Research Personnel actively involved in the cutting-of-edge science, innovation and excellence. Consisting of five Schools with twenty-four Departments, E&CE Department covers educational and research areas of Electricity, Telecommunications and Information Technology, Electronics and Computer Systems, and Automatic Control. There, current research interests of Prof. Skodras focus on image and video coding, digital watermarking for IPR protection, fast transform algorithms, real-time digital signal processing, and multimedia applications.

On the other hand, Valeo is an Automotive Supplier, partner to automakers worldwide. As a technology company, Valeo develops innovative products and systems that contribute to the reduction of CO^2 Emissions and to the development of intuitive driving. Valeo Group has 136 Production Sites, 16 Research Centres, 34 Development Centres, 15 Distribution Platforms, Employs 81,800 people across 29 Countries Worldwide.

World’s leading provider of driving assistance systems to automotive manufacturers (e.g., BMW, Land Rover or the Volkswagen Group among others), Valeo Vision Systems specializes in the design and manufacture of advanced Driver Assistance Systems including Rear View Cameras, Surround View and Camera Monitoring Systems. Their applications include Pedestrian Detection, Cross Traffic Alert and Automated Parking. And their next generation products will launch them into the realm of Autonomous Driving applications. VVS aims to improve safety and driving comfort by offering easy access and enhanced visibility around the vehicle, while creating an ergonomic, intuitive relationship with one's environment, thus placing them at the forefront of innovative technologies.

In this context arises the idea that gives birth to the Master Thesis, submitted in total fulfillment of the requirement towards the Master Title in Telecommunications by Technical University of Cartagena (UPCT), and defended on 29th September 2016, Cartagena, Spain.

Contents

List of Figures

List of Tables

Chapter I

Advanced Driver Assistance Systems (ADAS)

1.1 Introduction

Advanced Driver Assistance Systems (ADAS) are systems that provide assistance to the driver and improve driving experience. Its primary function is to ensure safety of the vehicle and other vehicles on the road, the driver, and the pedestrians or bikers. ADAS could be used to save fuel costs by enabling platooning in which vehicles follow each other within close distance, it could warn when a vehicle swerves across the lane, or it could apply emergency brake to avoid collision, etc. To function reliably, ADAS must be able to recognize objects, signs, road surface, and moving objects on the road and to make decisions whether to warn or act on behalf of a driver [1].

Demand for Advanced Driver Assistance Systems in recent years is caused by desire to build safer vehicles and roads in order to reduce the number of road fatalities and by legislation in the leading countries. ADAS are made of the following physical sensors: radar, LIDAR, ultrasonic, photonic mixer device (PMD), cameras, and night vision devices, that allow a vehicle to monitor near and far fields in every direction and of evolving and improving sensor fusion algorithms that ensure vehicle, driver, passenger's, and pedestrian's safety based on factors such as traffic, weather, dangerous conditions, etc. Modern ADAS systems act in real time via warnings to the driver or by actuation of the control systems directly, and are precursors to the autonomous vehicles of the future.

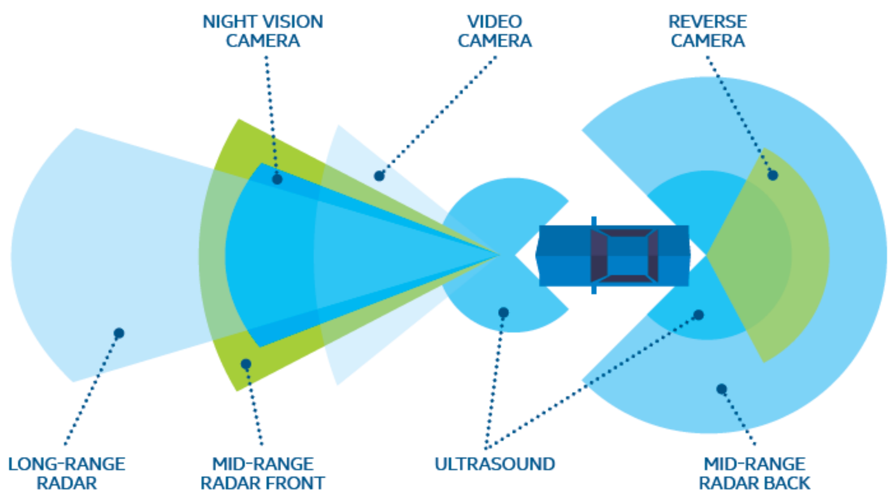


Figure 1.1: Example ADAS Sensors

There are several challenges to design, implement, deploy, and operate ADAS. The system is expected to gather accurate input, be fast in processing data, accurately predict context, and react in real time. And it is required to be robust, reliable, and have low error rates. There has been significant amount of effort and research in the industry to solve all these challenges and to develop the technology that will make ADAS and autonomous driving a reality. In addition to functional requirements, ADAS must be secured from adversaries with malicious intent whose goal is to compromise the system and cause catastrophic accidents with loss of life and damage to property. It has been shown both in academia and automotive industry that control system can be compromised via malicious attacks launched through various means, displaying to the driver wrong warnings or even causing fatality by remotely disabling braking system on a vehicle while it is moving. In addition to protecting the system from criminal actors, there is a bigger threat looming from nation-state sponsored cyber terrorism. In this way, within the next years ADAS should be considered as a fundamental requirement, together with reliability, robustness, performance, and low error rates, in the designing and manufacturing process of, as pointed out before, the vehicles of the future.

The rest of the chapter is organized as follows: first, a review on the background of ADAS systems, including example usage cases and general structure, is presented. To continue, an overview of the road accident statistics worldwide that dictate the need for further predictive action towards safer vehicles takes place, compiling a summary of the most representative ADAS already commercially available. Then, and digging into the scope of this thesis, a more detailed overview of vision-based systems is carried out, together with a demonstration of state-of-the-art of the systems that make up a vision-only ADAS system. Ultimately, the main goals of this thesis are specifically stated; closing up with a presentation of datasets and performance evaluation methods for quantitative and qualitative assessment of the proposed systems considered for the purposes of the thesis.

1.2 ADAS System Background

1.2.1 Example Usage Cases

ADAS system is considered as the advancement from driver assistant system (DAS) [1]. DAS is a system that informs and warns, provides feedback on actions, increases comfort, and reduces workload by actively stabilizing or maneuvering the vehicle. ADAS system is considered as a subset of DASs, with increased use of complex processing algorithms to detect and evaluate the vehicle environment based on data collected via a variety of sensor inputs (see Figure 1.1).

The spectrum of DAS capabilities available in production today are demonstrated in the following Figure 1.2. Some of these capabilities range from Driving Stability of the vehicle to Parking Assistance functionalities. Notice that the capabilities considered as ADAS, such as Automatic Cruise Control (ACC), Automatic Emergency Break (AEB) or Lane Departure Warning (LDW) among others, are highlighted with stars. The ADAS usage cases that require full power of real-time processing and intelligence, like the aforementioned cases, are highlighted with full stars, whereas half-colored star marked usage cases are relatively more rudimentary ADAS cases; these ones include traffic sign/light violation warning, intelligent speed adaptation, night vision, adaptive curve light, or rear view camera.

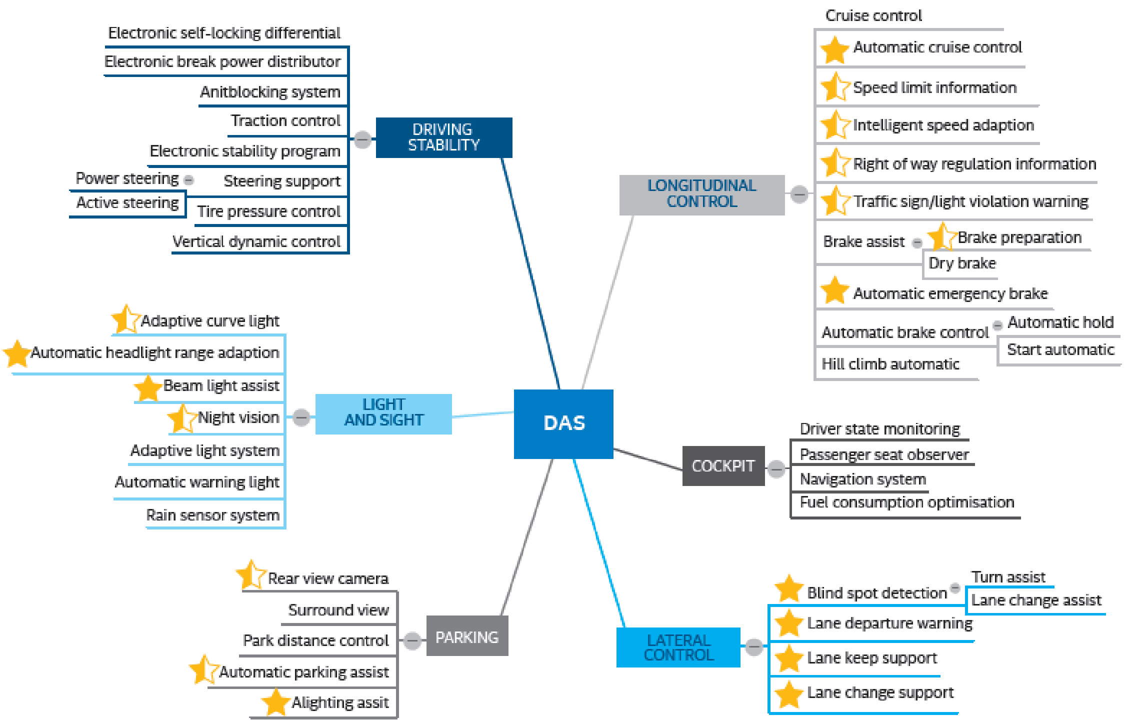


Figure 1.2: Spectrum of DAS and ADAS Functions

1.2.2 General Structure of an ADAS

To support all these aforementioned functionalities it provides, ADAS design at an architecture level must include modules for sensing, processing, intelligence generation, and decision making. Figure 1.3 is a generic view of what the ADAS system might look like. The overall system compromises sensors of various types; a CPU-GPU combination to perform the sensor data processing, object identification, and early sensor fusion; a *Central Brain* CPU for performing sensor fusion from different sensor blocks, object tracking, vehicle control activities to interact with the actuation, and a diagnostics block.

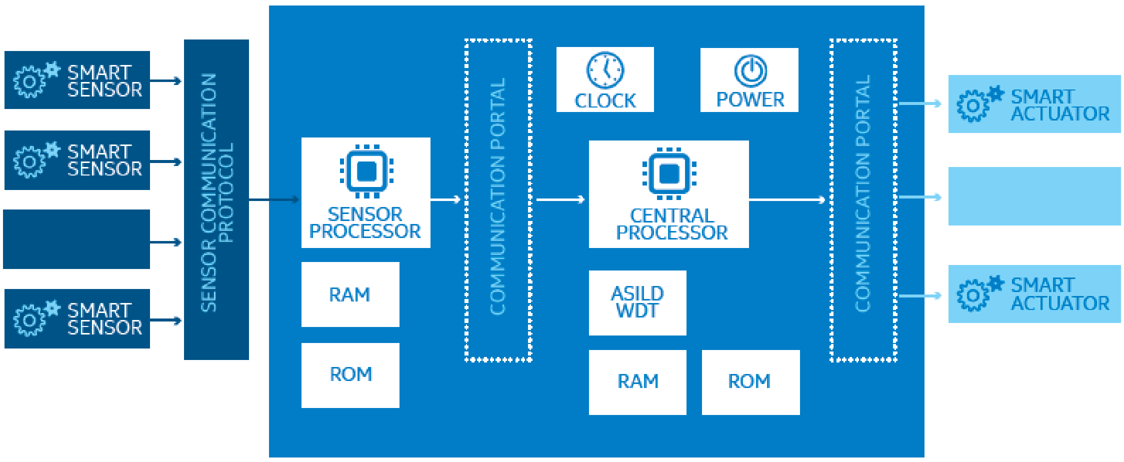


Figure 1.3: Conceptual Hardware Block Diagram for ADAS System

This system is considered as a close-loop control system, where the vehicle control actuation actions are computed based on received data from sensors. And the outcome of the ADAS actuation actions is fed back in the loop as sensor input. All the computing units in ADAS of the vehicular system are generally referred to as electronic control units (ECUs). The sensing and actuation ECUs are relatively resource constrained units, compared with the central processor of ADAS. One of the key advancements in ADAS design is the concept of *sensor fusion*. This is the process by which the internal processing takes input from the multiplicity of external sensors and creates a map of possible impediments around the vehicle. The map then facilitates the computation that creates a series of possible actions and reactions through situational analysis. Figure 1.1 shows an example ADAS-enabled vehicle with a collection of sensors to enable sensor fusion and actions.

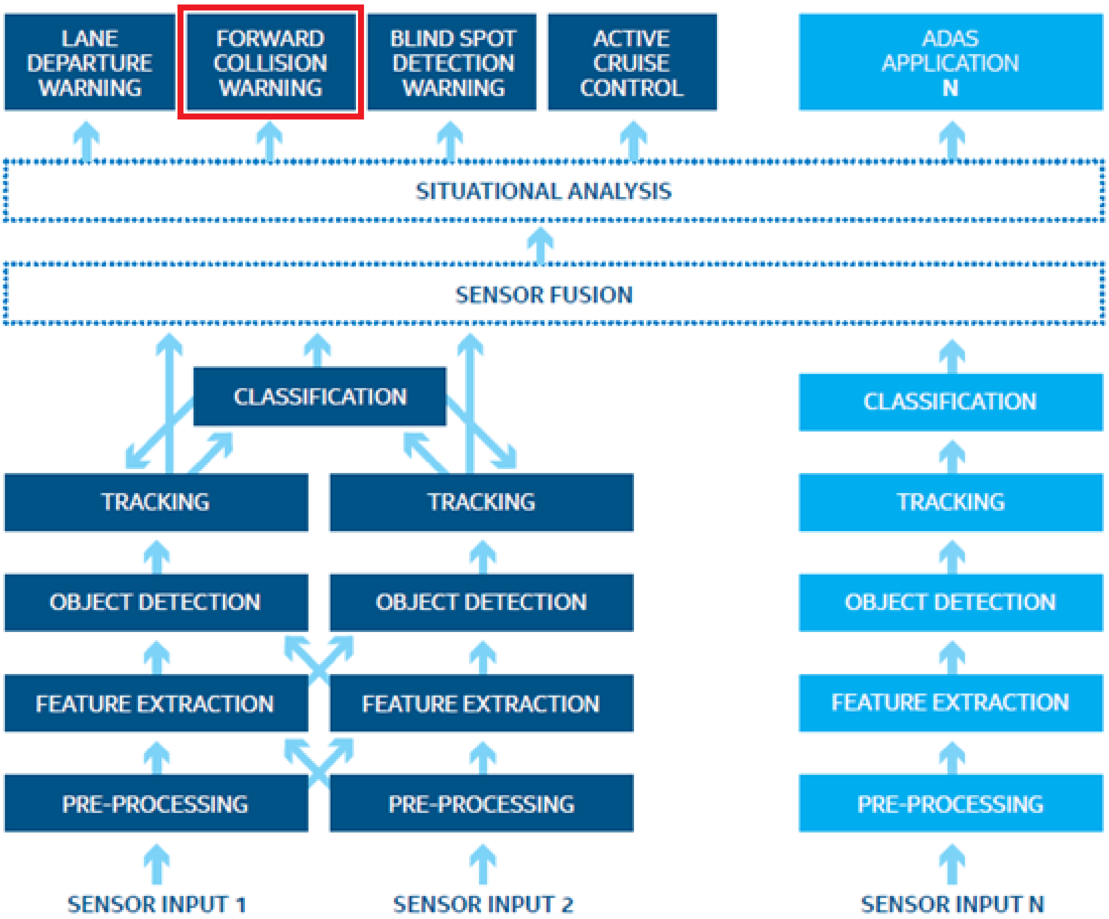


Figure 1.4: Example Sensors Fusion in ADAS

Sensor fusion and situational analysis can be done case by case for different ADAS functions and occurs at multiple levels. It is beneficial to have early fusion (determining conditions as early as possible) and a centralized processing *brain* (improving quality of detection and reducing CPU power consumption). Figure 1.4 demonstrates an approach where sensor fusion occurs in both the sensor processor and the central brain. With this design, the system provides a horizontal architecture that can support multiple ADAS applications in parallel, advancement from vertical systems that only support individual ADAS applications case by case, and therefore represents a closer approach to a real commercial ADAS system. Highlighted in red the Forward Collision Warning (FCW) module presents special relevance in the scope of this thesis; see following sections for further details.

1.3 Need for ADAS Development

As pointed out in [2], during the past few decades, road accidents have proven to be one of the most common causes for the loss of human lives. According to a study by the World Health Organization issued in 2004 [3], road accidents were estimated as the cause for 1.2 million people killed and as many as 50 million people injured worldwide. The seriousness of the situation can be reflected by the fact that in 2000, road accidents were the main cause of death inflicting injuries, resulting to approximately 23% of worldwide injury related deaths (see Figure 1.5), while in 2004 they were the second overall cause of death for people of ages 5-14 and the first cause of death for ages 15-29 years [4].

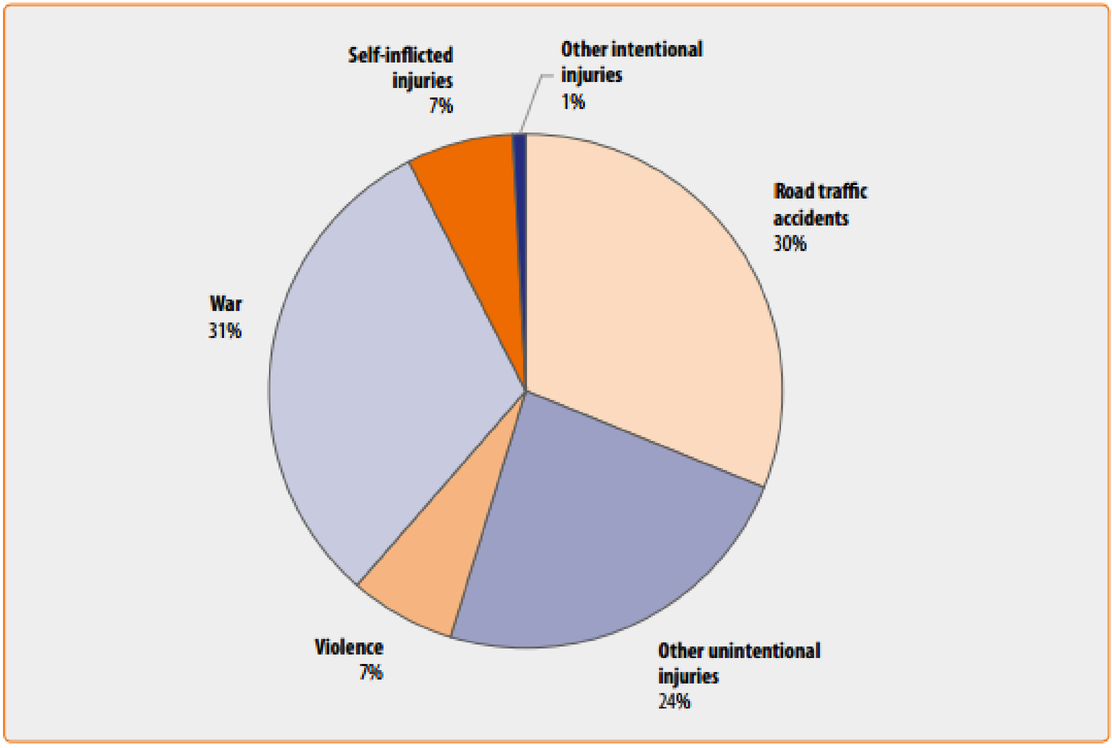


Figure 1.5: Causes of injury deaths among men aged 15–59 years, Eastern Mediterranean Region, 2004

More importantly, the forecasts state that road accident related deaths will raise by 65% between 2000 and 2020 [5] [6]. In fact, the absolute number of road-accident inflicted deaths is estimated to double in the period 2004-2030 [3], as shown in Figure 1.6. In terms of percentages, road accidents are predicted to become the 5th leading death cause by 2030, increasing by a factor of 0.54, from 2.2% to 3.6% of the world's deaths. These dire predictions concerning the rise of road accident related deaths are mainly based on the fact that the total number of vehicles worldwide rises, resulting in a subsequent increase of accidents. However, vehicle safety has improved over the years, mainly because of the advances of passive safety systems that have been included in almost all commercial vehicles. Such systems include seat belts, airbags and various improvements made on the vehicle bodies, improving their crashworthiness for passengers and pedestrians. Improvements in passive safety may have offered a lot so far, but now relative technologies seem to have reached a peak, since they cannot offer solutions to further mitigate deaths caused by accidents inflicted by drivers' mistakes.

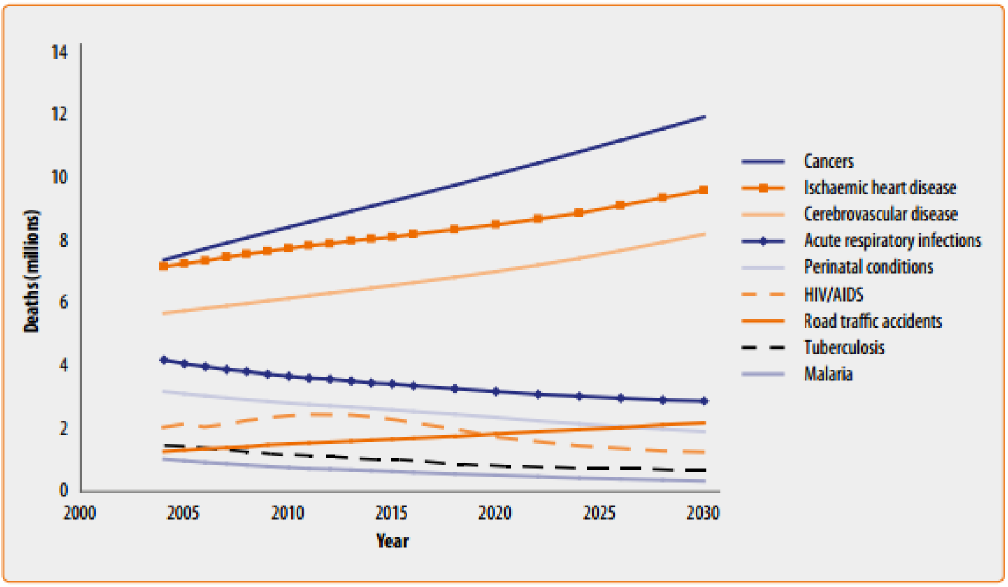


Figure 1.6: Projected global deaths for selected causes, 2004–2030

As demonstrated in Figure 1.7, the reduction in road accident inflicted death has been slowed down over the past years (except from France). In fact, a recent study by the National Highway Traffic Safety Administration (NHTSA) of the U.S.A. revealed that from all the driver-related causes of road accidents, a 41% was due to recognition errors and a 34% due to decision errors [7].

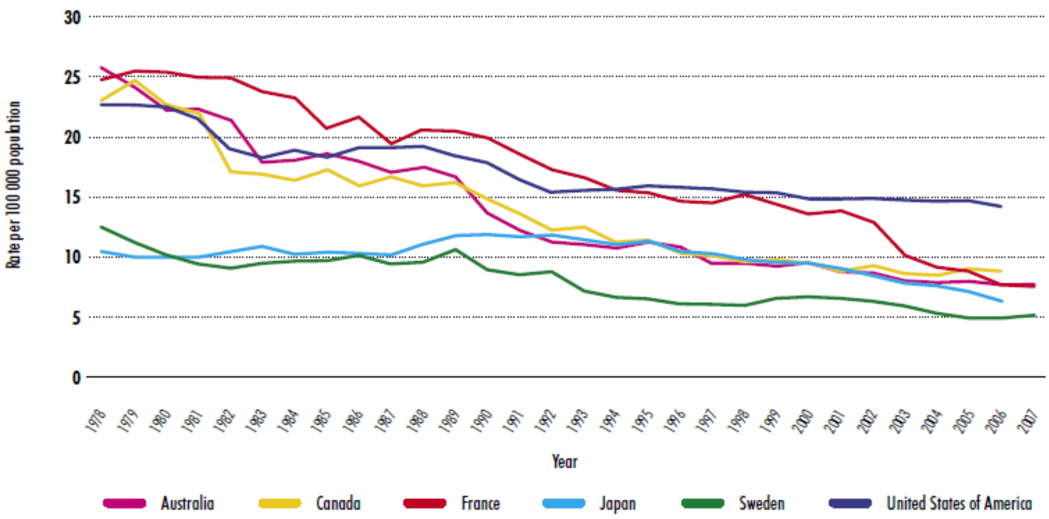


Figure 1.7: Trends in road traffic fatality rates in selected high-income countries

A conclusion that can be drawn from the analysis of road accident related data is that the further reduction of accidents has to rely more on active, state-of-the-art solutions generally known as Advanced Driver Assistance Systems. ADAS are new emerging technologies, primarily developed as automated advisory systems for drivers with a goal to enhance driving safety.

In this context, the European New Car Assessment Programme (Euro NCAP) [8] already announced in the past that starting from 2014, every car manufacturer hoping to achieve a perfect, five-star, score on the Euro NCAP safety ratings would have to equip their vehicle with an Autonomous Emergency Braking (AEB) system.

Table 1.1: Advanced Safety Technologies rewarded by Euro NCAP up to 2016

Advance Safety Technology	Rewarded Manufacturers
Attention Assist	2011 - Ford Driver Alert
	2011 - Mercedes-Benz Attention Assist
Autonomous Emergency Braking	2014 - BMW Pedestrian Warning with City Brake Activation
	2013 - FIAT City Brake Control
	2013 - Mitsubishi Forward Collision Mitigation
	2013 - Skoda Front Assistant
	2012 - Audi Pre Sense Front
	2012 - Audi Pre Sense Front Plus
	2012 - Volkswagen Front Assist
	2011 - Ford Active City Stop
	2011 - Ford Forward Alert
	2011 - Mercedes-Benz Collision Prevention Assist
	2011 - Volkswagen City Emergency Brake
	2010 - Honda Collision Mitigation Brake System
	2010 - Mercedes-Benz PRE-SAFE® Brake
	2010 - Volvo City Safety
Blind Spot Monitoring	2014 - Opel Side Blind Spot Alert (SBSA)
	2011 - Mazda Rear Vehicle Monitoring system (RVM)
	2010 - Audi Side Assist
E-Call (Automatic Emergency Call)	2012 - Ford SYNC Emergency Assistance
	2010 - BMW Assist Advanced eCall
	2010 - Citroën Localized Emergency Call
	2010 - Peugeot Connect SOS
Lane Support Systems	2013 - Skoda Lane Assistant
	2012 - Audi Active Lane Assist
	2012 - Ford Lane Keeping Alert
	2012 - Seat Lane Assist
	2011 - Ford Lane Keeping Aid
	2011 - Infiniti Lane Departure Prevention (LDP)
	2010 - Opel Eye
	2010 - Volkswagen Lane Assist
Pre-crash Systems	2013 - Skoda Crew Protect Assist
	2012 - Audi Pre-Sense Basic
	2012 - Volkswagen Proactive Occupant Protection
	2010 - Mercedes-Benz PRE-SAFE®
Speed Alert Systems	No rewards until the time of writing.
Vision Enhancement Systems	2011 - Opel Adaptive Forward Lighting (AFL)
Other Safety Systems	2013 - Skoda Multi Collision Brake
	2012 - Audi Secondary Collision Brake Assist
	2012 - Ford MyKey
	2012 - Seat Multi Collision Brake
	2012 - Volkswagen Multi Collision Brake

Similar efforts from Euro NCAP since it was first established in 1996, led to the improvement in passive and active safety systems used in vehicles. The corresponding criteria are a good indication of state-of-the-art systems that are supposed to be embedded into commercial vehicles. Some of the future trends are revealed when studying the rewards that Euro NCAP has established since 2009 [9], which give incentives to manufacturers to include advanced safety systems in their vehicles. The advanced safety technologies that have been rewarded since 2009 until today are included in the above Table 1. Nonetheless it seems relevant the absence of rewards from Euro NCAP along the last 2 years (2015 and 2016 up to date) and just 8 within the last 4 years despite the manifested importance of such systems, which are presented in further details in the upcoming sections.

1.4 Computer Vision-Based ADAS

1.4.1 Introduction

Coming closer to the scope of this thesis, when it comes to computer vision systems reality is that current commercial vehicular technology has only included this sort of devices in combination with other sensors, or specific subsystems that are robust enough to be used in real world driving scenarios, where human safety is directly involved, and therefore the risk to be assumed are none.

Nevertheless, up to date technology and advances in this field allow nowadays the realistic possibility of implementing a whole scenario based on vision based systems, robust and reliable enough to be introduced in real life driving situations with all safety guarantees for the citizens. As so, the aforementioned system, where just vision based ADAS considered, in order to fulfill all necessary requirements to be with all guarantees of real implementation in the vehicles today circulating on the roads all around the world must include if not all, most of the following modules: traffic light and sign recognition, vehicle detection, road and lane detection, pedestrian detection, night vision, driver drowsiness detection, and blind spot detection.

The continuous progress in the area of automatic assistance and driver warning methods indicates that we are not very far from developing such reliable ADAS in typical driving conditions. Being indeed the majority of sub-systems present in such systems implemented using vision processing methods.

More precisely, this thesis focuses on the vehicle detection module, with the further intention of implementing a TTC system that could help to avoid situations of direct risk in real driving scenarios. To this purpose, and before presenting in detail in the next chapter the proposed systems here addressed in order to overcome such challenge, a brief review on the state-of-the-art of these kind of vision based systems is presented in the next section, paying special attention not only to the most relevant one in the context of this thesis as Vehicle Detection module is, but also to some other sub-systems that would eventually make up the aforementioned system just based on computer vision techniques.

1.4.2 State-of-the-Art ADAS Sub-systems

Beside the absence of rewards in the ADAS field along the last 3 years by Euro NCAP (see Table 1.1), the summary of state-of-the-art ADAS recently included in commercial vehicles denotes a turn towards heavy usage of computer vision methods in such systems. An increasing number of ADAS rely either on the information merging of several sensors, including cameras, or on the utilization of information derived solely from monocular, or stereoscopic cameras. In fact, from the ADAS presented in Table 1.1 only Automatic Emergency Call does not include a computer vision based possible subsystem. The use of cameras was somehow neglected in the past mainly because computer vision algorithms required very powerful and expensive processors which, combined with the high prices of good quality cameras, raised the price of camera-based ADAS to a level that did not allow their commercialization.

Recently, the increasing processing power of embedded digital signal processing systems combined with the development of cheap, small video cameras with good image quality led ADAS research towards implementation of systems that operate based on information coming only from cameras. Such systems usually concentrate only on one of the problems of driver assistance, since the inclusion of all different problems in one single system is still a quite challenging issue. In this context, the forthcoming sections gather the most prominent examples of vision based subsystems of commercially available ADAS nowadays.

Vehicle Detection

Core of this thesis, Vehicle Detection is the cornerstone of a great amount of Advance Driver Assistance Systems, since it provides information about real possible dangers on live driving scenarios like impending collisions, therefore being used in Autonomous Emergency Braking systems, in pre-crash systems and also in blind-spot detection.

Several systems, including the novel electric Tesla vehicles in their models S and X [10], use alternative techniques to computer vision based to achieve the challenge of vehicle detection, e.g., radar systems in Mercedes Benz Pre-SAFE® Brake and Collision Prevention Assist and Ford Forward Alert, or Light Detection And Ranging (LIDAR) technology like Volvo City Safety, Volkswagen City Emergency Brake or Ford Active City Stop. Radars are chosen due to their good distance measurement when a sufficient radar reflectance of objects in front of the vehicle is present. However, they can be negatively affected by mud, snow (weather conditions) or leaves blocking its "view" and appear problematic when other vehicles cut in to the lane of the ego-vehicle, or when the ego vehicle makes a small radius corner.

Figure 1.8: Honda vision-based Forward Collision System

On the other hand the use of LIDAR technology offers distance measurement both in daytime and nighttime, but their sensors are compromised when stained by mud or snow and they fail to operate in adverse conditions such as fog or heavy rain.

In any case, the aforementioned disadvantages of those two technologies are some of the reasons why vehicle manufacturers have started in the past years using them in synergy with cameras. One of such system is Audi's Pre Sense Front Plus, which combines information from two long range radars that detect obstacles in front of the vehicle, with data from a windscreen-mounted camera to assess the probability of an impending forward collision and warn the driver, or apply the brakes if the threat is imminent. Pure vision-based systems have also been developed [11] like the Honda's Collision Warning (FCW) system presented in Figure 1.9, which integrated with their Mitigation Braking System (CMBS), uses a small camera mounted at the top of the front windshield to detect the presence of vehicles in front of the driver's vehicle, and if the systems determines risk of collision with the detected vehicle it activates audio and visual alerts to warn the driver, and if this one fails to act, the CMBS automatically apply brake pressure.

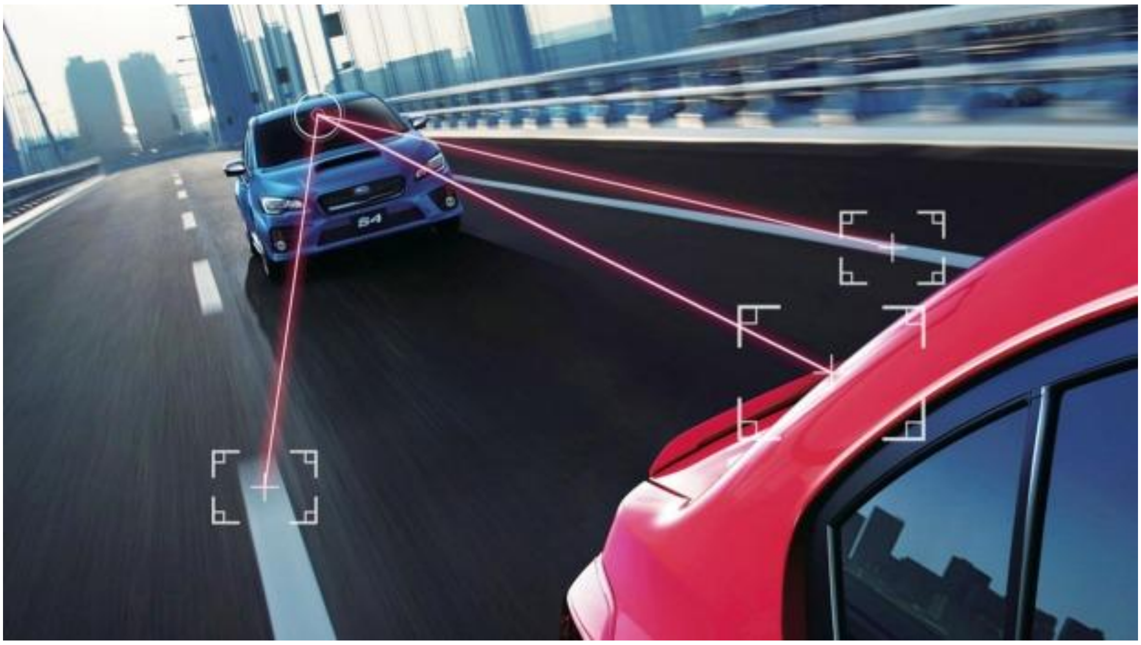


Figure 1.9: Subaru's EyeSight windshield-camera-based collision-avoidance technology

Subaru's EyeSight [12] represents another model case of vehicle collision avoidance system based on computer vision techniques. In Subaru's words, the EyeSight is their *most significant leap in crash prevention* since their previous invention, the Symmetrical All-Wheel Drive, *being an extra set of eyes on the road, and if need be, an extra foot on the brake when you drive.* And as a final example, vision has also been used in Mobileye vehicle detection after-market solution [13], which making use of several modules has impact in different ADAS areas of action such Forward Collision Warning or Headway Monitoring and Warning among others. The difference is that Mobileye has implemented vision-only systems, something that is particularly hard to achieve for all possible driving conditions.

(a) (b)

Figure 1.10: (a) Audi Pre Sense Front Plus system uses two long range radars and a camera to achieve AEB; (b) Mobileye Forward Collision Warning (FCW) module

Traffic Sign Recognition

Traffic signs carry important information about the driving environment and they assist drivers in making correct decisions for the safe navigation of their vehicles. Automatic recognition of traffic signs has been the most widely researched ADAS during the past decades. The need for an efficient traffic sign recognition (TSR) system is very closely related with the high numbers of road accidents that are caused by the driver's inability to focus on the information transmitted by traffic signs. Especially signs that inform about dangers, or speed limits, as well as the signs that prohibit some action are extremely necessary for secure driving. Furthermore, the vast amount of visual and other kinds of information that is brought into the attention of drivers at any given time often distracts their attention from what is really important for their driving reactions, not to mention the great diversity of environmental conditions and driving scenes can deteriorate drivers' visibility and lead to neglecting to focus on Traffic Signs (TS).

(a) (b)

Figure 1.11: TSR Real Systems: (a) Mydriveassist real usage case; (b) Volvo V40 2012, Active TFT Traffic Signs Display

In commercial terms some versions of TSR such as the Opel Eye System (see Table 1.1) or the Mobileye System [13] or myDriveAssist [14] included in some of the BWM series can already be found available in the market. Said this, the technology will have reached maturity only when it will have tackled some rather challenging problems such us overtaking signs and not only detecting and recognizing speed signs, or overcoming the big obstacle of weather and lighting conditions when it comes to the usage and development of such systems for autonomous driving and reliable driver warning assistance. Moreover TSR is a task that can be solved using GPS based technologies in collaboration with precise and dynamically changing maps; simple and efficient solution that leads the major automotive companies to minimize their effort in TSR technologies and research.

Traffic Lights Recognition

Analogously to the TSR, the automation of Traffic Lights Recognition (TLR) will eventually eradicate the human element involved in the detection of the changing lights in driving environments. For the sake of example, Figure 1.13 shows the Audi connect service known as Traffic Light Information Online [15], already incorporated in many of their latest series of vehicles, which predicts exactly when the traffic lights will change and recommends a speed for the driver to adopt in order to reach it before it turns to red.

Figure 1.12: Audi Traffic Light Information Online dashboard

In contrast to the TSR, TLR is actually one of the most neglected research areas in ADAS technology taking into account the large amount of serious accidents caused by traffic light violations compared to those caused by violations of other types of traffic signs. Nevertheless the problem of Traffic light Detection is not a trivial one due to the very dense presence of red and green light sources in cities. Effect that increases significantly the false positive error rate of traffic light recognition systems and consequently the difficulty of developing robust and reliable design.

Night Vision Systems

In recent years, multiple studies have been conducted in the United States as well as Europe and they all seem to paint the same general picture: the majority of driving is done with the aid of daylight, yet a high volume of fatal accidents occur at night. Therefore, it is not surprising that vision based systems are already widely included in commercial vehicles of the main automotive branches all around the world. Just some examples [16] are BMW and Mercedes-Benz which already for several years have been offering night vision as an extra feature of their 7-series and S-Class models respectively, using different approaches.

(a) (b)

Figure 1.13: Night Vision Systems: (a) BMW far-IR Night Vision System; (b) Mercedes-Benz near-IR Night View Assist System

BMW used a passive Far-Infrared (FIR) sensor to stream night vision images in the car monitor, while Mercedes-Benz used a near-IR (NIR) system for the same reason. Both systems are really expensive and this is why they are offered only in top-class vehicles of the two brands. When combined with a module like pedestrian detection or similar, the extra visibility provided by these systems can prove truly life-saving outcomes.

Driver Drowsiness/Gaze Detection

Every year, many car accidents due to driver fatigue and distraction occur around the world and cause many casualties and injuries. Driver face monitoring systems are therefore one of the main approaches for driver fatigue or distraction detection and accident prevention. These sorts of systems basically capture the images from driver face and extract the symptoms of fatigue and distraction from eyes, mouth and head. These symptoms are usually percentage of eyelid closure over time, eyelid distance, eye blink rate, blink speed, gaze direction, eye saccadic movement, yawning, head nodding and head orientation. In consequence the system estimates driver alertness based on extracted symptoms and alarms if needed.

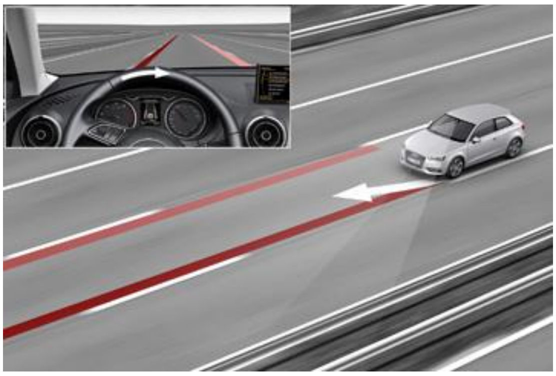
Figure 1.14: Scheme of a regular Driver Gaze Detection System

A real example which perfectly illustrates the importance of such systems in the following years is the Canadian automotive supplier Magna International, which with its vehicle-secured connectivity eye tracking systems, at CES 2016, the global consumer electronics and consumer technology tradeshow that takes place every January in Las Vegas, featured in-cabin imaging and head-up display technologies aimed at reducing distracted driving and improving driver safety, exactly based on the principle above mentioned [17]. However, it must be pointed out that up to date very few vehicles in the market, the system under test in the SAAB 9-3 SportCombi [18] which using a series of increasingly irritated text, voice messages, and vibrations in the driver's seat cushion aims to mitigate two of the biggest causes of accidents as drowsiness and inattention are (see Figure 1.15), have made use of computer vision methods for addressing this challenge; evidence of the very early stage of development in which this kind of systems are nowadays, i.e., far from being mature enough to be widely deployed in commercial vehicles.

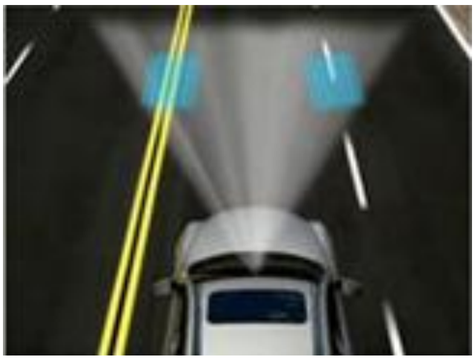
Road Lane Recognition

One of the most mature vision-based technologies for an ADAS is road lane recognition. Its presence is vital in Lane Support Systems, Lane Warning Departure Systems and similar ones, as it provides the basic information needed for such a task. Road lane recognition [2] is targeted mainly on highway driving, or on driving in well-maintained city or country roads, and the technology behind it is fairly simple, since there are several robust algorithms for line finding in an image, the most widely used being Hough transform [19]. Most recent commercially available solutions (see Table 1.1) include the Skoda Lane Assitant or Audi's Active Lane Assist, both intended to help the driver stay within the lane of travel, and assist the driver in cases where lane departure is unintentional among other functionalities.

(a)



(b)



(c)

(d)

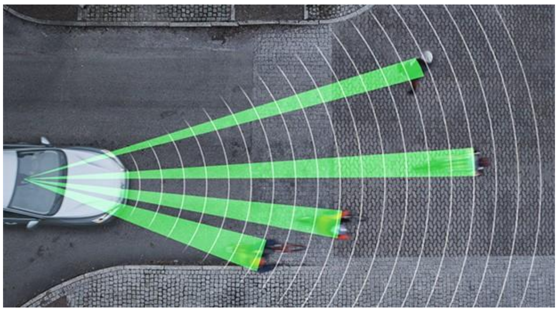
Figure 1.15: RLR Systems: (a) Skoda Lane Assistant; (b) Audi Active Lane Assist; (c) Infiniti Lane Departure Prevention; (d) Opel Eye System

Similarly, the Infiniti Lane Departure Prevention (LDP) module helps drivers return the vehicle to its travelling lane when an unintentional lane departure is likely, or the Opel Eye system which warns the driver if the car is about to veer inadvertently out of the lane in which it is travelling are just some others real cases where this technology is available.

Lane assist systems are invaluable for highway driving, or driving on well-preserved roads. However this is not the general case. These systems still fall short of what is expected by an ADAS in situations of driving in unstructured or badly maintained roads, where road lanes are not visible, or directly non-existent; not to mention driving in unstructured and even unpaved rural roads. In those scenarios Road Lane Recognition systems do not have an alternative solution to the problem of lane keeping, and so they cannot offer useful services to the driver. Such challenges require a more general drivable path detection algorithm, reason why a road detection module would deliver much more reliable information; road detection systems robust in all kinds of environments, at any time of the day and under different weather conditions.

Pedestrian Detection Systems

Another of the most extensively researched areas when it comes to Advance Driver Assistance Systems is pedestrian detection [20] [21]. Pedestrian detection is a key problem in computer vision, with real potential to positively impact quality of life, not only in the automotive field. In recent years, the number of approaches to detecting pedestrians in monocular images has grown steadily. However, the apparition of this technology in commercial vehicle systems was belated, being Volvo one of the very few brands incorporating from 2010 this functionality in one of its commercial cars [22]. The pedestrian detection approach is accomplished with the use of a front-facing camera in combination with a radar sensor that measures the distances to the pedestrians (see Figure 1.17 (a)). Moreover, considering bikers in their latest releases; when bicyclists swerve in front of an automobile heading in the same direction, the setup immediately alerts the driver and applies full brake power. Mobileye [23], already mention as well in previous sections, with its Pedestrian Collision Warning (PCW) (see Figure 1.17 (b)), offers a vision-only commercial solution for this problem.



(a)

(b)

Figure 1.16: PDS Systems: (a) Volvo innovative safety system for Pedestrian Detection; (b) Mobileye Pedestrian Collision Warning (PCW) System

The majority of the subsystems presented in the previous section have been included in commercial vehicles after exhaustive research of their robustness and trustworthiness. As the paradigm of pedestrian detection shows, a system that has been extensively covered by scientific research might take a long time until it reaches commercial maturity. Thus, there are still some systems not mature enough to be considered for inclusion in commercial vehicles. In addition, even the commercially available systems have their limitations; bad weather conditions, limited visibility, sudden illumination changes, dense traffic etc. are only some of the factors that can deteriorate the performance of a vision-based ADAS.

1.5 Aims of this Thesis

1.5.1 Main Goals

Retaking what already stated in previous sections, the scope of this thesis resides in the design of a Computer Vision based Advance Driver Assistance System [24] [25] [26], for the more specific implementation of a so-called Time-To-Collision (TTC) System [27] [28], able to predict the remaining time to a potential impact taking place between the ego-vehicle and others present on the road, from images acquired in real time from an on-vehicle mounted front camera [29] [30]. Integrated in a Pre-crash System or in combination with an Emergency Breaking System (EBS) the proposed system might lead to the implementation of a real vehicle Collision-Avoidance System (CAS) that would eventually allow to avoiding potential dangerous situations of real risk of traffic accidents on the road within real driving scenarios.

All possible variations of a complex process like a driving continuously changing environment where infinite variables take place (e.g., own vehicle, other vehicles on the road, pedestrians, bikers, lighting conditions depending on weather prediction, traffic signs and lights, etc) make of the proposed project a challenging task. Consequently, several challenges addressed to design, implement, deploy, and operate of any ADAS system are faced within this thesis since the system proposed is expected to be fast in processing data, accurately predict context, and react in real time, respecting requirements robustness, reliability, and low error rates.

To this end, during the course of this thesis most, if not all subsystems that make up a TTC-ADAS System should be covered. Process that more specifically involves: (1) investigating and developing a vehicle detection algorithm; task divided itself into two stages: Hypothesis generation (HG) of candidate vehicles and Hypothesis Verification (HV) of the proposed candidates. To this purpose different image processing techniques must be research and applied. Some of them might include popular feature extraction methods such us edge detection, color-based clustering, feature descriptors, mathematical morphology, Gabor filters, etc. Later on a vehicle classification algorithm that allows determining the vehicles correctly detected must be considered; possible approaches may involve machine learning systems like SVM (Support Vector Machines), neural networks, etc.

A second phase (2) of the project involves investigating and developing this time an algorithm for temporal tracking of the results obtained in the Preceding Vehicle Detection (PVD) stage. Such solutions may include particulate filters or Kalman filters among other techniques.

Eventually, (3) a research and development of an algorithm for Time-To-Collision prediction from the data of stages 1 and/or 2 must be accomplished in order for this thesis to be considered successfully addressed. The results obtained in this point will largely depend on the results obtained in the previous stages.

Test and optimization of the proposed algorithms both as modules and as a system must be taken into account along the whole process; quantitative and/or qualitative methods must be considered when appropriate to this end.

A simplistic synthesizing of the scope of this thesis is shown in the following diagram of modules:

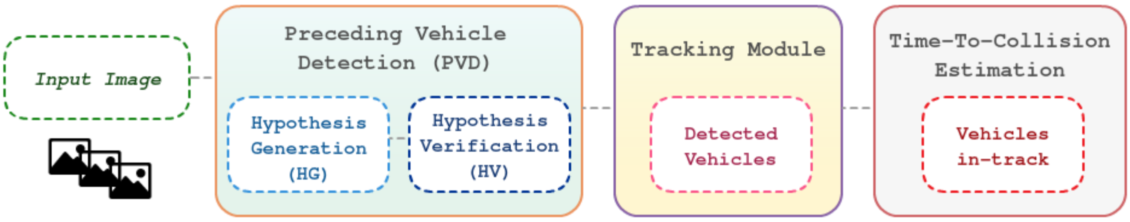


Figure 1.17: Thesis Diagram

The general workflow of this thesis project is therefore quite intuitive. It all begins with the original real-time images acquired from the on-vehicle front-faced mounted camera. From them the Preceding Vehicle Detection (PVD) module proposes a series of vehicle candidate detections. These results are one step further tracked along successive frames by the tracking module toward addressing the Time-to-Collision prediction challenge, carried out by the TTC module. And so forth as the video sequence plays on.

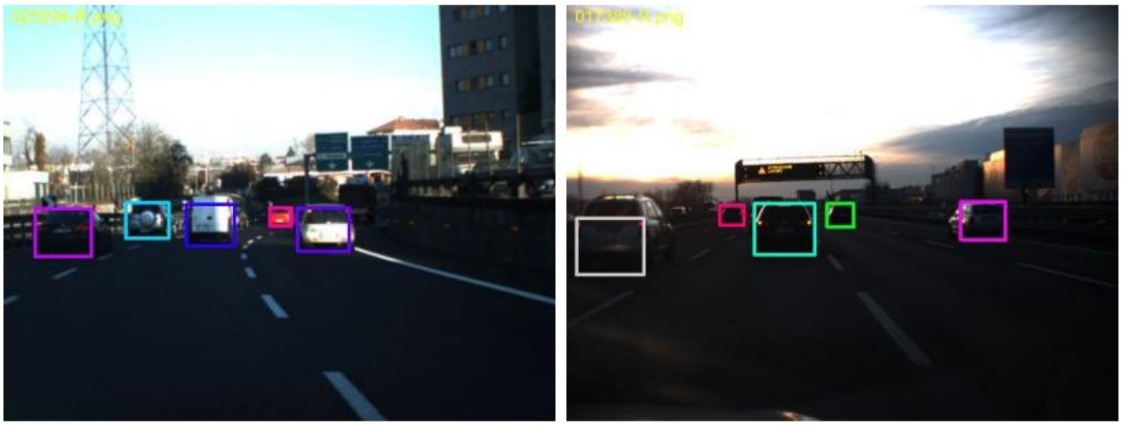
As closing line, the proposed system aims robustness to changes of illumination and weather conditions, as well as to diverse driving environments, always considering the intrinsic real-time constraints of the challenges outlined, and designed for monocular windscreen-mounted cameras in the ego-vehicle.

1.5.2 Datasets Used for System Development & Evaluation

One of the great problems faced during the evaluation of a computer vision system is usually the lack of carefully selected, publicly available, manually annotated datasets with a variety of examples [2]. When the system deals with a real-world problem and demands a great mixture of videos taken under different conditions and frame-based annotation information is required, then the number of man hours that have to be spent on the construction of the dataset is extremely high. The availability of very small publicly available annotated datasets is the most important reason for the absence of common benchmarks for ADAS.

For the purpose of our thesis TME Motorway Dataset [31] [32] due to its characteristics will play the leading key role. The Toyota Motor Europe (TME) Motorway Dataset is a dataset composed by 28 clips for a total of approximately 27 minutes (30000+ frames) with vehicle annotation, where these ones were semi-automatically generated using laser-scanner data. Image sequences were selected from acquisition made in North Italian motorways in December 2011, including variable traffic situations, number of lanes, road curvature, and lighting, covering most of the conditions present in the complete acquisition.

Further detailed the TME Motorway Dataset comprises: (1) image acquisition in stereo, 20 Hz frequency , 1024x768 grayscale losslessly compressed images, 32° horizontal field of view, and bayer coded color information with a checkboard calibration sequence available; (2) Ego-motion estimate (confidential computing method); (3) Laser-scanner generated vehicle annotation and classification (car/truck); and (4) A software evaluation toolkit (C++ source code). Furthermore, the data provided is timestamped, and includes extrinsic calibration.



(a) Long range, variable light (b) Dusk conditions

Figure 1.18: Toyota Motor Europe (TME) Motorway Dataset:
Examples of different weather conditions with vehicle annotations overlaid

The dataset is presented divided in two sub-sets depending on lighting condition, named "daylight" (although with objects casting shadows on the road) and "sunset" (facing the sun or at dusk). For each clip, 5 seconds of preceding acquisition are provided, to allow the algorithm stabilizing before starting the actual performance measurement. Data acquired in cooperation with VisLab (University of Parma, Italy), using the BRAiVE test vehicle.

Complementarily to the TME Dataset, for quantitative evaluation of the static performance of the preceding vehicle detection algorithm developed in this thesis, two datasets of images of the rear part of vehicles provided by Caltech [33], were also used. The first dataset is called Caltech 1999 and comprises 126 images, sized 892x592 pixels, of cars from the rear taken at the parking lots of Caltech with sunny weather. The second dataset, called Caltech 2001, contains 526 pictures of 60 different cars, with resolution 360x240, that have been taken in freeways of southern California with sunny weather as well. Even if they don't exactly represent vehicles in real live driving environments, both datasets are ideal for the evaluation of preceding vehicle algorithms, since they contain a variety of vehicles in front of the ego-vehicle.

1.5.3 Performance Evaluation Measures

In order to evaluate the proposed systems within the scope of this thesis in the most complete and efficient way possible, a combination of both quantitative and qualitative methods will be considered when appropriate. Since depending on the problem and challenge being addressed at each stage of the thesis, some of these methods will point out more interesting features and results, and therefore will lead to deeper and more useful conclusions than other, being sometimes of no relevance at all with regard to the aims of the thesis, and in consequence pointless to be taken into account.

These methods, to more precisely evaluate the performance and quality of the computer vision based algorithms proposed in this thesis, are briefly introduced in the following lines.

Quantitative Measures for Performance Evaluation

Some of the measures most commonly used for assessing the detection quality of all computer vision systems are based on the definition of four basic metrics [2]. Within the scope of an image, when facing a binary classification test, that is to say the problem of classifying the pixels of the image (or part of it) into two different classes (i.e. positive and negative), can be defined as: **True Positive (TP)** the number of pixels classified correctly as belonging to the positive class (also called *sensitivity* measures the proportion of positives that are correctly identified as such); **False Positive (FP)** the number of pixels misclassified as belonging to the positive class; **False Negative (FN)** the number of pixels misclassified as belonging to the negative class; and **True Negative (TN)** the number of pixels classified correctly as belonging to the negative class (also called *specificity* measures the proportion of negatives that are correctly identified as such).

According to these definitions, several metrics for the better characterization of the developed computer vision systems can be established:

$$Correctness\ or\ Precision: \quad P = TP/(TP + FP) \tag{1.1}$$

$$Completeness\ or\ Recall: \quad R = TP/(TP + FN) \tag{1.2}$$

$$Quality: \quad g = TP/(TP + FP + FN) \tag{1.3}$$

$$\boldsymbol{Accuracy}: \quad \boldsymbol{A} = \frac{(\boldsymbol{TP} + \boldsymbol{TN})}{(\boldsymbol{TP} + \boldsymbol{TN} + \boldsymbol{FP} + \boldsymbol{FN})} = \frac{\boldsymbol{T}}{(\boldsymbol{T} + \boldsymbol{F})} \tag{1.4}$$

$$Effectiveness: \quad F = 2PR/(P + R) \tag{1.5}$$

It must be noticed that all the aforementioned measures range from 0 to 1, with 1 denoting the best result and 0 denoting the worst possible result.

Qualitative Performance Evaluation

Apart from quantitative performance evaluation, it is of high importance that a real-world computer vision system is also assessed qualitatively. As a matter of fact, due to the complexity presented by a normal real driving environment, in constant dynamic change, where uncountable variable take place at a time such us ego-vehicle and others on the road, pedestrians, bikers, weather, or traffic lights among others, to build up an annotated dataset comprising all possible variations and challenges here addressed for quantitative evaluation purposes is unfeasible. Thus, an extensive qualitative assessment process is at times more insightful than a quantitative one, and in consequence all systems presented in this thesis have been qualitatively evaluated, paying a special attention to the particular adverse conditions faced in each one.

Processing Speed Performance Assessment

It must be noticed that quality is not the only critical factor for the design of and Advance Driver Assistance System, and ultimately a Time-To-Collision system when it comes to the scope of this thesis. Obviously it is the most important factor, since ultimately such systems involve environments where safety and integrity of human lives is priority. Nonetheless, the processing speed at which these sort of systems are up to behave plays a key role in this context, simply because it is a must-condition that these kind of systems operate in real time so that they can be involved in real time driving environments. Thus, their processing speed must be faster than the frame acquisition rate of video cameras, and in the worst scenario equal to this one.

Given the up-to-date typical camera frame rates, and taking a view on the datasets considered in the scope of this thesis [31], a rate of 10 frames per second (fps) can be considered as the minimum speed at which our TTC-ADAS system should operate in order to be up to real scenarios. It seems as well presumable that depending on the specific details of each application this goal might be either a little lower (i.e. for city driving with low speeds), or even higher when highway driving must be handled. For all these reasons, in order to evaluate real-time constraints a final Time Performance Evaluation of all system developed as part of this thesis is addressed.

1.6 Conclusions

This chapter began introducing ADAS principles, their structure and functionalities. Then, the current worldwide trend for the development of reliable Intelligent Transportation Systems and more specifically Advanced Driver Assistance Systems was presented and justified. Most of the up to date Advance Safety Technologies ADAS based commercially available by recognized manufacturers have been reported. Furthermore, the state-of-the-art systems that utilize only visual information have been briefly examined and a general structure of vision-only ADAS has been proposed. Finally, the main goals and challenges to be addressed by this thesis have been stated, together with datasets used to this end and the metrics used for the evaluation of the solutions proposed.

The remainder of this document is structured as follows. In Chapter II the core Computer Vision algorithms relevant in the scope of this thesis will be presented. Following Sections III, IV and V will respectively depict into detail the systems proposed by this thesis according to its main aims: Preceding Vehicle Detection (PVD), Vehicle Tracking, and Time-to-Collision (TTC) Prediction. Quantitative, qualitative and ultimately Time Performance Evaluation of all modules, together with tests and results carried out to consolidate these stages will be gathered in detail. In last term, Section VI will sum up the whole work by this thesis covered, will address extracted conclusions, and will as well eventually suggest some potential future lines of work to continue what here presented.

Chapter II

Computer Vision: Principles & Algorithms

1.1 Introduction

In contrast to the Chapter I where ADAS principles together with the main goals this thesis aims were presented, Chapter II collects a brief, yet detailed presentation of the computer vision image processing concepts and algorithms used in the development of the modules and systems proposed within the scope of this thesis towards the fulfillment of the main goals addressed.

More specifically, concepts such similarity and template matching, as well as several metrics related to them (e.g., Euclidean distance, square difference and correlation) together with their importance in the computer vision algorithms proposed by this thesis are reviewed and presented with real and illustrative example cases. Furthermore the concept of Machine Learning, and more concretely the supervised learning SVM (Support Vector Machine) algorithms, key factor when it comes to address the crucial dual problem of *vehicle/non-vehicle* classification, is presented. It will be studied as well the importance of Gabor Filters and their correct parameterization in relation to the SVM classification and its success.

In last term, a discrete time Kalman filter for object (vehicles in our case) tracking is presented, at the same time that the importance of motion information in problems related to ADAS is highlighted. The fundamental concept at this point of optical flow is presented, as well as its ultimate and determinant influence in the development of the TTC system this thesis aims for.

How each technique and algorithm is specifically applied and useful in the context of this thesis will be depicted into details in the following Chapters III, IV and V. Furthermore, and just to be mentioned, this entire thesis have been carried out over the well-known programming language Python, within the development environment Python(x,y), and with the use of OpenCV as main library to deal with image processing tasks.

1.2 Similarity Matching

For the purpose of this thesis, where Computer Vision techniques play the main role, dealing with image databases is a must condition imposed by the nature of the aims being addressed. In this line, image databases force us to rethink many of the concepts accounted by other sort of databases. One of these is matching.

As outlined in [34], the fundamental operation in a content-indexed image database should not be matching the query against the images in the database in search of a *target* image that best matches the query. The basic operation in query-by-content will be ranking portions of the database with respect to similarity with the query. Thus, just like the matching is the single most important operation in traditional databases — since it decides, ultimately, which data satisfy the query — so similarity measurement is the single most important operation in image databases; here therefore what kind of similarity measure should be used to address the problem becomes a key factor.

In this context, since the results of the query must be ultimately judged by a human user, it seems natural that the human concept of similarity has been widely studied by experts throughout the last decades towards the definition of a similarity measure. Psychologists have been experimenting for some 70 years trying to define the properties of human similarity judgment. Many of these models succeed in explaining qualitatively some experimental findings, but few of them are in a mathematical form suitable for application to automatic computation.

As shown in [34] as well as in some other relevant bibliography [35] some well know similarity metrics range from very simple ones like Mean Absolute Error (MAE) or Normalized Cross-Correlation (NCC), to some more complex like the Structural Similarity Index (SSIM) which combines luminance, contrast and structural information of the images. In the scope of this thesis the similarity matching metric selected by its simplicity is the Euclidean Distance, which given two images I_a and I_b, described by the sets of $n \in \mathbb{N}$ features $\{a_1, \ldots, a_n\}$ and $\{b_1, \ldots, b_n\}$ respectively, can be mathematically expressed as:

$$Euclidean\ Distance: \quad d_e(I_a, I_b) = \left[\sum_{i=1}^{n} (a_i - b_i)^2\right]^{\frac{1}{2}} \tag{2.1}$$

1.3 Template Matching

1.3.1 Introduction

Often, the detected objects in an image have to be compared to some templates so that their similarity is verified. As so, *Template Matching is* a high-level machine vision technique that identifies the parts on an image that match a predefined template. Advanced template matching algorithms allow as well finding occurrences of the template regardless of their orientation or local brightness [36].

(1) (2) (3)

Figure 2.1: Template Matching Identification: (1) Template image; (2) Input Image; (3) Results of multi-angle matching

Template Matching techniques are flexible and relatively straightforward to use, which makes them one of the most popular methods of object localization. Their applicability is limited mostly by the available computational power, as identification of big and complex templates can be time-consuming.

1.3.2 TM Concept

Template Matching techniques are expected to address the following need: provided a reference image of an object (the *template image*) and an image to be inspected (the *input image*) we want to identify all *input image* locations at which the object from the *template image* is present. Depending on the specific problem at hand, we may (or may not) want to identify the rotated or scaled occurrences.

Shortly put [37], template matching aims finding areas of an image I (input or source image) that match (are similar) to a template image T.

(a) (b) (c)

Figure 2.2: Template Matching Example: (a) Input image; (b) Template image; (c) Input image with Template image location

To identify the matching area, the template image is compared against the source image by sliding it, that is, literally moving the template one pixel at a time (left to right, up to down) over the source image, calculating at each location, a metric that represents how *good* or *bad* the match at that location is, or in other words, how similar the template T is to that particular area of the source image I.

(a) I - T Comparison (sliding) (b) Comparison Results

Figure 2.3: Template Matching Process

For each location of T over I, this metric is stored in the result matrix (R) so that each location (x, y) in R contains the match metric. Therefore the image (b) above is the result R of sliding the template with a metric CCORR_NORMED (see below), where the brightest locations indicate the highest matches. As it can be seen, the location marked by the red circle is probably the one with the highest value, so that location (the rectangle formed by that point as a corner and width and height equal to the template image) is considered the match.

In practice OpenCV presents available 6 different methods (or metrics) to implement the template matching function, and as presumable, all of them present different characteristics and therefore report different results. Thus, according to the nomenclature used before I denotes the input image of size $(W\ x\ H)$, T the template image $(w\ x\ h)$, and R the result matrix $(W - w + 1)\ x\ (H - h + 1)$ the 6 metrics mentioned above are presented in the following lines (with $x' = 0 \dots w - 1$ and $y' = 0 \dots h - 1$):

a. Calculates **Square Difference** between T and I, therefore the best match is represented by a 0 value.

$$SQDIFF: \quad R(x, y) = \sum_{x',y'} \left(T\left(x', y'\right) - I(x + x', y + y')\right)^2 \tag{2.2}$$

$$SQDIFF_{NORMED}: \quad R(x, y) = \frac{\sum_{x',y'} \left(T\left(x', y'\right) - I(x + x', y + y')\right)^2}{\sqrt{\sum_{x',y'} (T(x', y')^2 \cdot \sum_{x',y'} I(x + x', y + y')^2}} \tag{2.3}$$

b. Calculates **Correlation between** T and I, therefore the best match is represented by the highest value.

$$CCORR: \quad R(x, y) = \sum_{x',y'} \left(T\left(x', y'\right) \cdot I(x + x', y + y')\right) \tag{2.4}$$

$$CCORR_{NORMED}: \quad R(x, y) = \frac{\sum_{x',y'} \left(T\left(x', y'\right) \cdot I(x + x', y + y')\right)}{\sqrt{\sum_{x',y'} (T(x', y')^2 \cdot \sum_{x',y'} I(x + x', y + y')^2}} \tag{2.5}$$

c. Uses **Correlation Coefficients**, therefore as before the best match is represented by the highest value.

$$CCOEFF: \quad R(x, y) = \sum_{x',y'} \left(T\left(x', y'\right) \cdot I(x + x', y + y')\right) \tag{2.6}$$

$$CCORR_{NORMED}: \quad R(x, y) = \frac{\sum_{x',y'} \left(T\left(x', y'\right) \cdot I(x + x', y + y')\right)}{\sqrt{\sum_{x',y'} (T(x', y')^2 \cdot \sum_{x',y'} I(x + x', y + y')^2}} \tag{2.7}$$

Where in this case:

$$T'(x',y') = T(x',y') - 1/(w \cdot h) \cdot \sum\nolimits_{x'',y''} T(x'',y'') \tag{2.8}$$

$$I'(x+x',y+y') = I(x+x',y+y') - 1/(w \cdot h) \cdot \sum\nolimits_{x'',y''} I(x+x'',y+y'') \tag{2.9}$$

The NORMED version of algorithms presents the results within the range of $[0,1]$ where 1 means perfect matching (excepting SQDIFF where 0 represents the perfect one) so that by simply setting a threshold the matching can be accepted/discarded with regard to the application being implemented.

1.4 Machine Learning & Support Vector Machines

1.4.1 ML Principles

Machine Learning (ML) is the science of getting computers to act without being explicitly programmed. In the past decade, machine learning has given us self-driving cars, practical speech recognition, effective web search, and a vastly improved understanding of the human genome. Machine learning is so pervasive today that you probably use it dozens of times a day without knowing it. Many researchers also think it is the best way to make progress towards human-level Artificial Intelligence [38].

In general [39], a learning problem considers a set of n samples of data and then tries to predict properties of unknown data. If each sample is more than a single number and, for instance, a multi-dimensional entry, is it said to have several attributes or features. In this way learning problems can be separated in a few large categories: *Supervised learning,* in which the data comes with additional attributes that we want to predict, and *Unsupervised learning,* in which the training data consists of a set of input vectors x without any corresponding target values. The goal in unsupervised problems may be to discover groups of similar examples within the data, where it is called clustering, or to determine the distribution of data within the input space, known as density estimation, or to project the data from a high-dimensional space down to two or three dimensions for the purpose of *visualization*. While on the other hand the problem of supervised learning can be either: *Classification,* where samples belong to two or more classes and we want to learn from already labeled data how to predict the class of unlabeled data. An example of classification problem would be the handwritten digit recognition example, in which the aim is to assign each input vector to one of a finite number of discrete categories. Another way to think of classification is as a discrete (as opposed to continuous) form of supervised learning where one has a limited number of categories and for each of the n samples provided, one is to try to label them with the correct category or class. On the other side, if the desired output consists of one or more continuous variables, then the task is called *regression.* An example of a regression problem would be the prediction of the length of a salmon as a function of its age and weight.

Shortly, machine learning is about learning some properties of a data set and applying them to new data. This is why a common practice in machine learning to evaluate an algorithm is to split the data at hand into two sets, one called the *training set* on which we learn data properties and another one called the *testing* set on which we test these properties.

1.4.2 Support Vector Machines (SVMs)

Further explained, supervised learning is the machine learning task of inferring a function from supervised training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which is called a classifier (if the output is discrete, see classification) or a regression function (if the output is continuous, see regression). The inferred function should predict the correct output value for any valid input object. This requires the learning algorithm to generalize from the training data to unseen situations in a *reasonable* way.

A wide range of supervised learning algorithms is available, each with its strengths and weaknesses. There is no single learning algorithm that works best on all supervised learning problems. Due to the characteristic of the problem faced in this thesis and reviews made in previous researches like [40] where SVMs are for instance compared to Neural Networks (NN) with vehicle detection purposes, Support Vector Machines (SVMs) seems to present the best results toward this end, and consequently is the solution chosen to be implemented within this thesis.

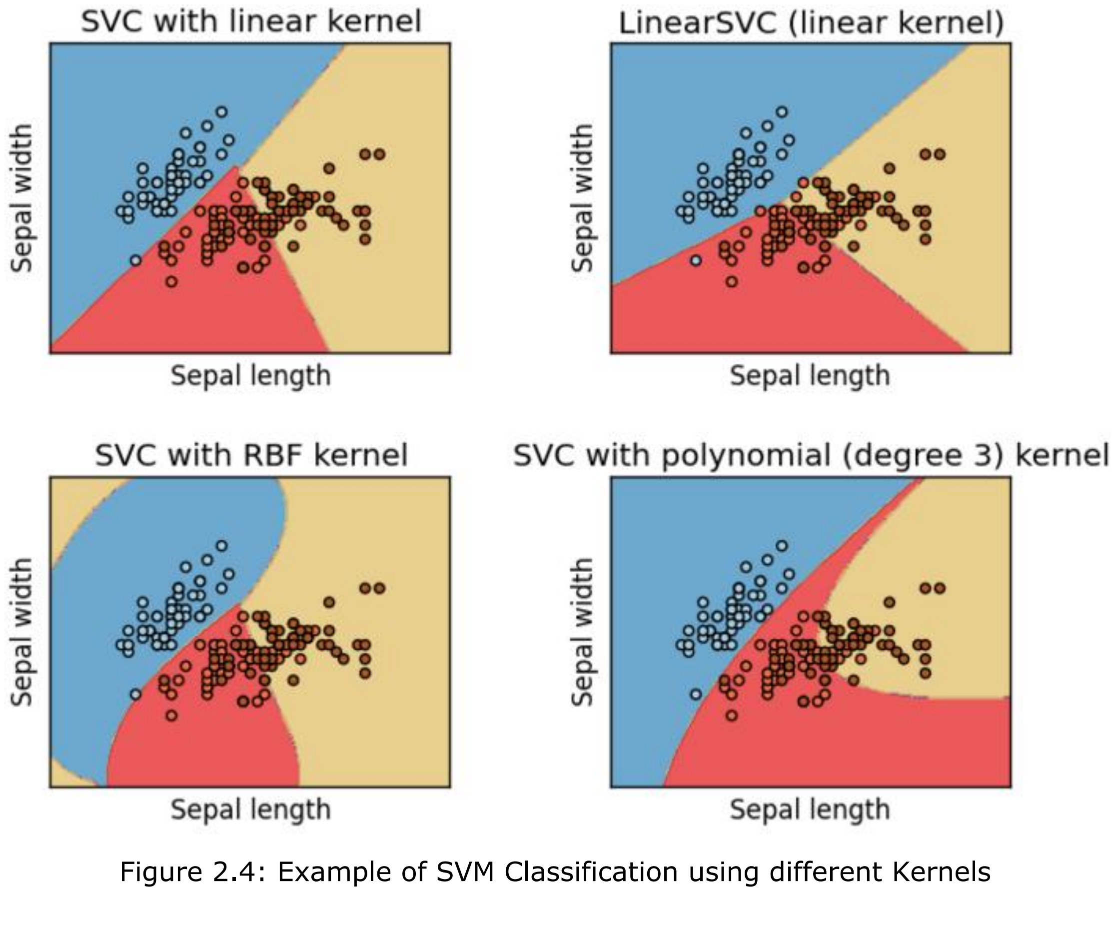


Figure 2.4: Example of SVM Classification using different Kernels

VMs are a set of supervised learning methods used for *classification, regression* and *outliers detection* [41]. On one hand, some of the advantages of support vector machines are: effective in high dimensional spaces and in cases where number of dimensions is greater than the number of samples. More precisely uses a subset of training points in the decision function (called support vectors), so it is also memory efficient, and versatile, different Kernel Functions can be specified for the decision function, both common kernels already well defined (e.g., linear, polynomial, rbf and sigmoid) and specified custom kernels. On the other hand, the disadvantages of support vector machines include poor performances if the number of features is much greater than the number of samples, and the fact that SVMs do not directly provide probability estimates, these are rather calculated using an expensive five-fold cross-validation.

1.5 Discrete Time Kalman Filter for Object Tracking

1.5.1 Introduction

Another method that is very popular in computer vision applications is Kalman filtering. A Kalman filter is essentially an estimator that, in this context, can be used to predict the position of an object in future frames of a video. Kalman tracking can provide multiple advantages for any video-based computer vision application, since it can reduce the area of the frame that is scanned for the existence of an object thus minimizing false positive detections and reducing the time needed for the process. An added advantage offered by the very nature of the Kalman filter, is the smoothing effect that can improve the robustness of the tracking result to the presence of noise that is often observed in videos.

1.5.2 Definition of the discrete process model

For the purposes of this thesis, only the discrete time Kalman filter [42] [43] will be considered. The Kalman filter addresses the general problem of trying to estimate the state $x \in R^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation, with a measurement $z \in R^m$ that is:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (2.10)$$

$$z_k = Hx_k + v_{k-1} \quad (2.11)$$

Where x_k and z_k are vectors representing respectively the model state and measurement at the discrete time step k, while A, B and H are the transition, control and measurement matrices of the model. The $n\,x\,n$ transition matrix A connects the previous process state (at time $k-1$) to the current state (at time k). The $n\,x\,l$ matrix B relates the optional control input $u \in R^l$ to the state x. Finally, H is a $m\,x\,n$ matrix that relates the process state to the measurement, z_k. In practice all these matrices might change at each time step, though they are generally assumed to be constant.

The random variables w_k and v_k represent the process and measurement noise (respectively). They are assumed to be statically independent (of each other), white, zero mean Gaussian. Their probability distributions can be denoted as:

$$p(w) \sim N(0, Q) \quad (2.12)$$

$$p(v) \sim N(0, R) \quad (2.13)$$

In practice, the *process noise covariance Q and measurement noise covariance R* matrices might change as well with each time step or measurement, however in most studies they are also assumed constant.

1.5.3 Algorithm of the Discrete Kalman Filter

More specifically, the Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of measurements (could contain noise). As such, the equations for the Kalman filter fall into two groups: *time update* equations and *measurement update* equations. The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the *a priori* estimates for the next time step. The measurement update equations are responsible for the feedback, that is, for incorporating a new measurement into the a priori estimate to obtain an improved *a posteriori* estimate.

The time update equations can also be thought of as *predictor* equations, while the measurement update equations can be thought of as *corrector* equations. Indeed the final estimation algorithm resembles that of a *predictor-corrector* algorithm for solving numerical problems as shown below.

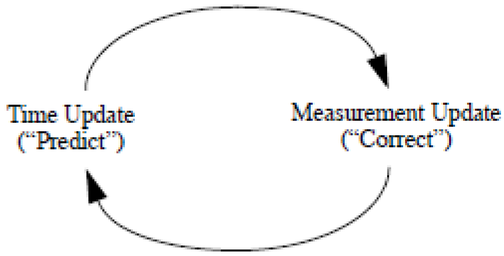


Figure 2.5: The ongoing discrete Kalman filter cycle

As previously said, the *time update* projects the current state estimate ahead in time, and the *measurement update* adjusts the projected estimate by an actual measurement at that time. Moreover, being $\hat{x}_k^- \in R^n$ the *a priori* state estimate and $\hat{x}_k \in R^n$ the *a posteriori* estimate at time k, given the measurement z_k, the *a priori* and *a posteriori* estimate errors can be defined as:

$$e_k^- \equiv x_k - \hat{x}_k^- \tag{2.14}$$

$$e_k \equiv x_k - \hat{x}_k^- \tag{2.15}$$

Leading to the following *a priori* and *a posteriori* estimate error covariance matrices, where E denotes the variance.

$$P_k^- = E[e_k^-(e_k^-)^T] \tag{2.16}$$

$$P_k = E[e_k^-(e_k^-)^T] \tag{2.17}$$

Thus, the *time update equations* of the discrete Kalman filter can be defined:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \tag{2.18}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{2.19}$$

And the *measurement update equations* as:

$$K_k = P_k^- H^T (KP_k^- H^T + R)^{-1} \tag{2.20}$$

$$\hat{x}_k = A\hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{2.21}$$

$$P_k = (I - K_k H)P_k^- \tag{2.22}$$

The first task during the measurement update is to compute the Kalman gain, K_k. The next step is to actually measure the process to obtain z_k, and then to generate an *a posteriori* state estimate by incorporating the measurement as in (2.21). The final step is to obtain an *a posteriori* error covariance estimate via (2.22).

After each time and measurement update pair, the process is repeated with the previous *a posteriori* estimates used to project or predict the new *a priori* estimates. This recursive nature is one of the very appealing features of the Kalman filter, it makes practical implementations much more feasible than for example other similar filters. The next figure offers a complete picture of the operation of the filter, combining the high-level diagram of Figure 2.5 with the above equations.

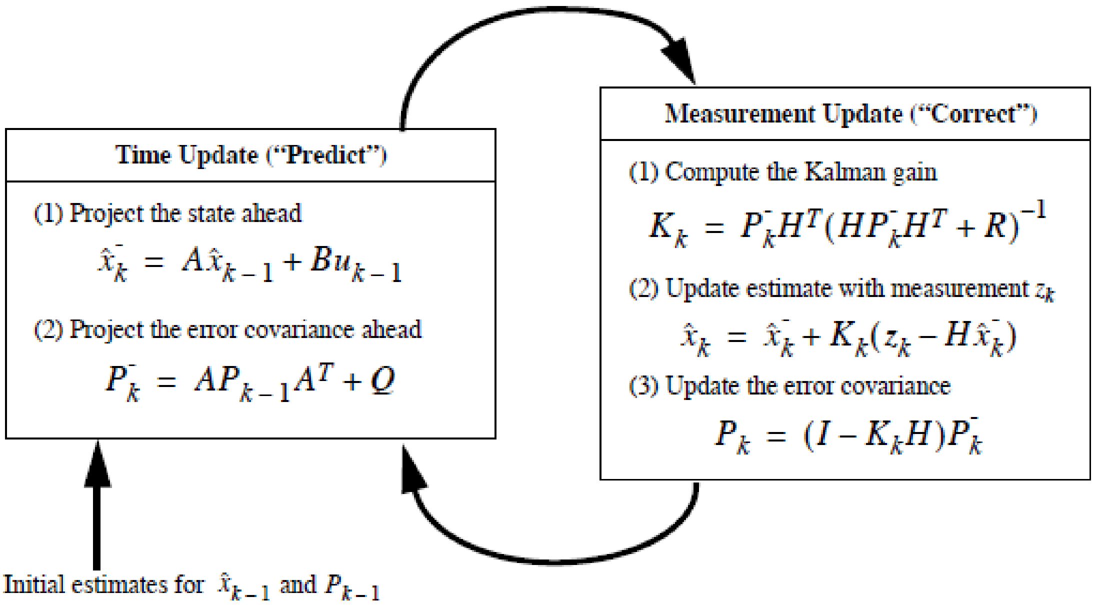


Figure 2.6: A complete picture of the operation of the Kalman filter

In real implementations of the filter, the measurement noise covariance R is usually measured prior to operation of the filter. Measuring the measurement error covariance R is generally practical (possible) because we need to be able to measure the process anyway (while operating the filter) so we should generally be able to take some off-line sample measurements in order to determine the variance of the measurement noise.

The determination of the process noise covariance Q is generally more difficult as we typically do not have the ability to directly observe the process we are estimating. Sometimes a relatively simple (poor) process model can produce acceptable results if one *injects* enough uncertainty into the process via the selection of Q. Certainly in this case one would hope that the process measurements are reliable.

1.6 Gabor Filters

1.6.1 Introduction

In image processing, a Gabor filter, is a linear filter used for edge detection. Frequency and orientation representations of Gabor filters are similar to those of the human visual system, and they have been found to be particularly appropriate for texture representation and discrimination. In addition, simple cells in the visual cortex of mammalian brains can be modeled by Gabor functions. Thus, image analysis with Gabor filters is thought to be similar to perception in the human visual system.

In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave. As presented in [44], Gabor filters are a group of wavelets, with each wavelet capturing energy at a specific frequency and a specific direction. Expanding a signal using this basis provides a localized frequency description, therefore capturing local features/energy of the signal. Texture features can then be extracted from this group of energy distributions. The scale (frequency) and orientation tunable property of Gabor filter makes it especially useful for texture analysis. Experimental evidence on human and mammalian vision supports the notion of spatial-frequency (multi-scale) analysis that maximizes the simultaneous localization of energy in both spatial and frequency domains. The motivation to use Gabor functions is mostly biological, since Gaborlike receptive fields have been found in the visual cortex of primates [45]. Gabor functions act as low-level oriented edge and texture discriminators and are sensitive to different frequencies and scale information. These facts raised considerable interest and motivated researchers to extensively exploit the properties of Gabor functions.

The applications of these kinds of filters are widely present in our society. As outlined in [46], Gabor filter is usually used for biometric recognition, such as face recognition or fingerprint recognition. For example, Gabor filter have been used in real applications for extracting face image as biometric data. In pattern recognition and vehicle detection field, Gabor filter has been widely used for feature extraction. Moreover in document image processing, Gabor features are ideal for identifying the script of a word in a multilingual document. Gabor filters with different frequencies and with orientations in different directions have been used to localize and extract text-only regions from complex document images (both gray and colour), since text is rich in high frequency components, whereas pictures are relatively smooth in nature. It has also been used in pattern analysis applications. For example, it has been used to study the directionality distribution inside the porous spongy trabecular bone in the spine. Finally, the Gabor space is very useful in image processing applications such as optical character recognition and iris recognition. Relations between activations for a specific spatial location are very distinctive between objects in an image. Furthermore, important activations can be extracted from the Gabor space in order to create a sparse object representation [47].

1.6.2 The Gabor Function

Mathematically [45], a 2D Gabor function, g, is the product of a 2D Gaussian and a complex exponential function. The general expression is given by the following equation:

$$g_{\theta,\lambda,\sigma_1,\sigma_2}(x,y) = exp\{-1/2\,(x,y)M(x,y)^T\}\,exp\left\{\frac{j\pi}{\lambda}(x\cos\theta + y\sin\theta)\right\} \qquad (2.23)$$

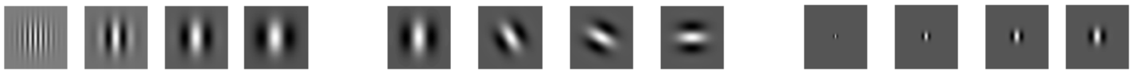
Where $M = diag(\sigma_1^{-2}, \sigma_2^{-2})$ Some examples of Gabor functions are shown in Figure 2.7. The parameter θ represents the orientation, λ is the wavelength, and σ_1 and σ_2 represent scale at orthogonal directions. When the Gaussian part is symmetric, we obtain the isotropic Gabor function:

$$g_{\theta,\lambda,\sigma}(x,y) = exp\left\{-\frac{x^2+y^2}{2\sigma^2}\right\}\,exp\left\{\frac{j\pi}{\lambda}(x\cos\theta + y\sin\theta)\right\} \qquad (2.24)$$

However, with this parameterization the Gabor function does not scale uniformly, when σ changes. It is preferable to use a parameter $\gamma = \lambda/\sigma$ instead of λ so that a change in σ corresponds to a true scale change in the Gabor function. Also, it is convenient to apply a 90 degrees counterclockwise rotation to Eq. (anterior), such that θ expresses the orthogonal direction to the Gabor function edges. Therefore, in the remainder of the paper we will use the following definition for the Gabor functions:

$$g_{\theta,\lambda,\sigma}(x,y) = exp\left\{-\frac{x^2+y^2}{2\sigma^2}\right\}\,exp\left\{\frac{j\pi}{\gamma\sigma}(x\cos\theta + y\sin\theta)\right\} \qquad (2.25)$$

By selectively changing each of the Gabor function parameters, we can *tune* the filter to particular patterns arising in the images. Figure 2.7 illustrates the variation of parameters (γ, θ, σ) in the shape of the Gabor function.



(a) (b) (c)

Figure 2.7: Examples of Gabor functions. Real part of Gabor function for: (a) $\gamma = \{1/2, 3/2, 5/2, 7/2\}$; (b) $\theta = \{0, \pi/6, \pi/3, \pi/2\}$; (c) $\sigma = \{4, 8, 12, 16\}$

1.6.3 Gabor Response

By convolving a Gabor function with image patterns $I(x,y)$, we can evaluate their similarity. With $*$ representing convolution, the Gabor response at point $(x_0,\ y_0)$ can be defined as:

$$G_{\theta,\gamma,\sigma}(x_0,y_0) = \left(I * g_{\theta,\gamma,\sigma}\right)(x_0,y_0) = \int I(x,y) g_{\theta,\gamma,\sigma}(x_0 - x,\ y_0 - y) dxdy \qquad (2.26)$$

The Gabor response obtained from the previous Equation (2.26) can emphasize basically three types of characteristics in the image: edge-oriented characteristics, texture-oriented characteristics and a combination of both. In order to emphasize different types of image characteristics, we must vary the parameters σ, θ and γ of the Gabor function. The variation of θ changes the sensitivity to edge and texture orientations. The variation of σ will change the *scale* at which we are viewing the world, and the variation of γ the sensitivity to high/low frequencies.

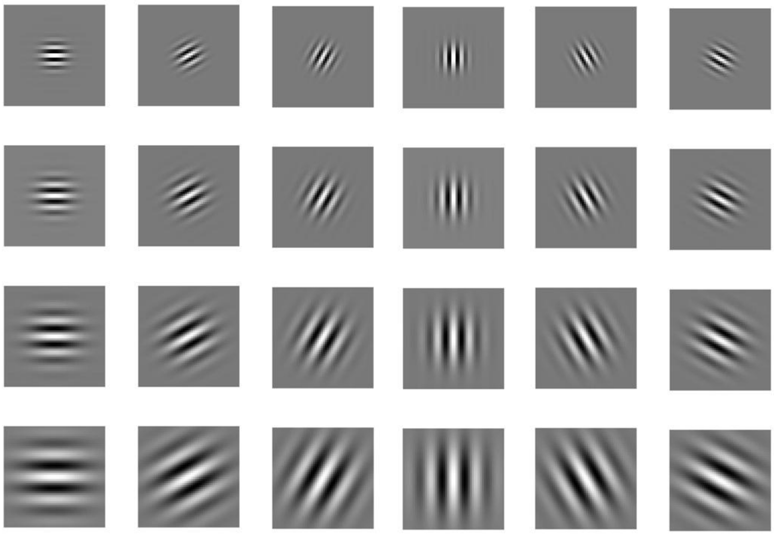


Figure 2.8: Gabor Kernel using 4 Scales (vertical) and 6 orientations (horizontal)

The previous figure shows an example of Gabor Kernel using 4 Scales and 6 orientations as presented in [46]. In the scope of this thesis we would like to find the most adequate combinations of σ, θ and γ towards our particular objective of vehicle detection.

1.7 Optical Flow

The concept of Optical flow or optic flow, which will be essential in the latest stages of this thesis for the implementation of the Time-to-Collision system here proposed, was for the first time introduced by the American psychologist James J. Gibson in the 1940s to describe the visual stimulus provided to animals moving through the world, and can be defined as the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene [48] [49] . Another way to define Optic flow [50] could be as the change of structured light in the image, e.g. on the retina or the camera's sensor, due to a relative motion between the eyeball or camera and the scene.

Thus, sequences of ordered images allow the estimation of motion as either instantaneous image velocities or discrete image displacements as shown in the example below.

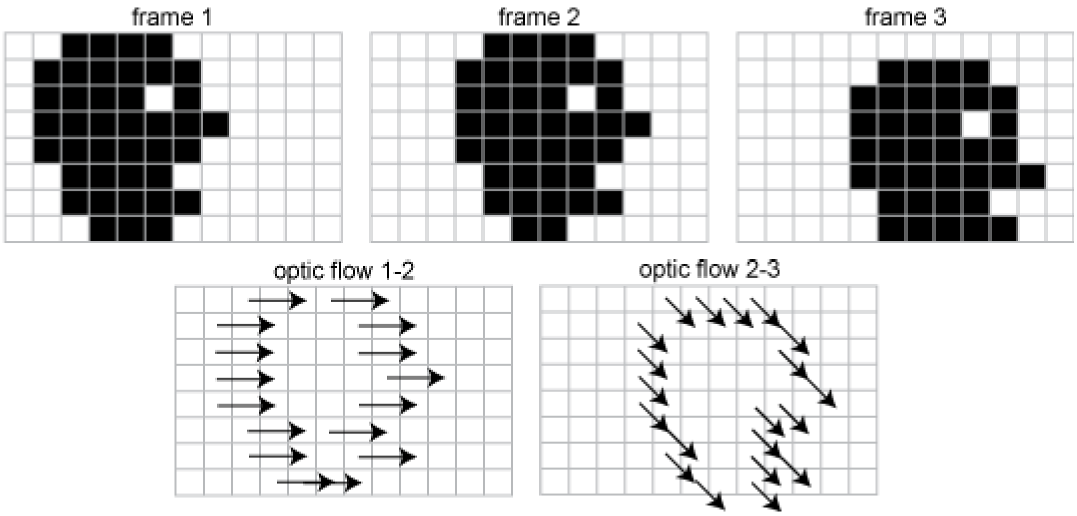


Figure 2.9: Detection of the optical flow in 3 temporally-consecutive images, showing the movement of an example shape (silhouette of a head)

In this example the optic flow is depicted as the correspondence of contour pixels between frame 1 and 2 as well as frame 2 and 3. For methods estimating flow, the challenge resides in, not only to consider the contour pixels, but also to find the spatial correspondence (between consecutive frames) for each pixel in the image.

For the sake of better explanation, [51] imagine a point (u, v) in an image captured by a camera at time t (as depicted in figure 2 above). The pixel has intensity $I(u, v, t)$. Assume that the pixel is translated by $(\Delta u, \Delta v)$ in time Δt to $I(u + \Delta u, v + \Delta v, t + \Delta t)$ in the next frame. Since the pixel is the same in both images and have the same intensity:

$$I(u, v, t) = I(u + \Delta u, v + \Delta v, t + \Delta t) \tag{2.27}$$

Being this the core assumption of optical flow - brightness constancy: the brightness or intensity of a pixel remains the same in a small region given that Δu, Δv, Δt are small. This is the basis of the optical flow constraint equation and is depicted in the following Figure 2.10.

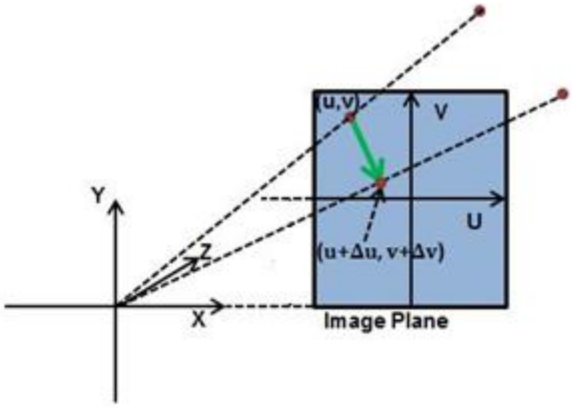


Figure 2.10: Optical flow vector (in green) from point (u, v) in frame x to point $(u + \Delta u, v + \Delta v, t + \Delta t)$ in frame $x + 1$

In the mathematical context, there are two main classes of optical flow algorithms, and multiple methods, to address this challenge: sparse and dense algorithms. Sparse optical flow algorithms compute the aforementioned displacement vectors for distinct points (called corners) while dense optical flow algorithms compute these for every pixel in an image. As presumable, there is a trade-off between processing speed of such algorithms and their accuracy. Optical flow algorithms that offer dense, accurate results usually tend to have prohibiting computational complexity for real-time applications. On the other hand, faster algorithms that produce sparse optical flow results cannot be trusted to provide an accurate approximation that could be used for tasks like motion-based segmentation of objects.

In any case, the real uses and applications of optical flow in real life are a mere fact. Motion estimation and video compression have developed as a major aspect of optical flow research. Optical flow was used by robotics researchers in many areas such as: object detection and tracking, image dominant plane extraction, movement detection, robot navigation and visual odometry. Optical flow information has been recognized as being useful for controlling micro air vehicles.

The application of optical flow includes the problem of inferring not only the motion of the observer and objects in the scene, but also the structure of objects and the environment. Since awareness of motion and the generation of mental maps of the structure of our environment are critical components of animal (and human) vision, the conversion of this innate ability to a computer capability is similarly crucial in the field of machine vision.

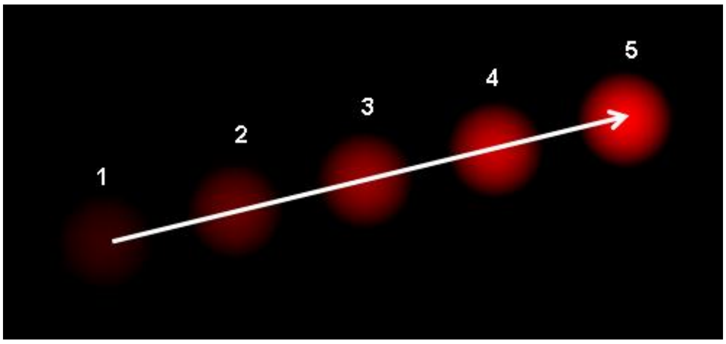


Figure 2.11: Optical flow vector of a moving object in a 5-frames video sequence

Consider a five-frame clip of a ball moving from the bottom left of a field of vision, to the top right. Motion estimation techniques can determine that on a two dimensional plane the ball is moving up and to the right and vectors describing this motion can be extracted from the sequence of frames. For the purposes of video compression (e.g., MPEG), the sequence is now described as well as it needs to be. However, in the field of machine vision, the question of whether the ball is moving to the right or if the observer is moving to the left is unknowable yet critical information. Not even if a static, patterned background were present in the five frames, could we confidently state that the ball was moving to the right, because the pattern might have an infinite distance to the observer.

Chapter V will show how the concept of optical flow becomes the core stone of the TTC system proposed in this thesis.

1.8 Conclusions

In this chapter a brief, yet detailed presentation of all computer vision image processing concepts and algorithms used throughout this thesis is featured.

Concepts such similarity and template matching, together with several related matrix like Euclidean distance or square difference and correlation, are shown. Furthermore the concept of Machine Learning, and supervised learning SVM algorithms, key factor when it comes to address the crucial dual problem of *vehicle/non-vehicle* classification is presented. It was studied as well the impact of Gabor Filters and their correct parameterization in this process. In last term, a discrete time Kalman filter for vehicles tracking is collected, at the same time that the importance of motion information in problems related to ADAS systems is highlighted. Particular emphasis is put into the concept of optical flow due to its ultimate and determinant influence in the development of the TTC system this thesis aims for.

Chapter III
Preceding Vehicles Detection (PVD)

3.1 Introduction

As seen in Section 1.5 where the main goals of this thesis are collected, considered as a whole, this thesis can be divided into three sub-systems in order to fulfill the challenge of designing a TTC-ADAS System for forward collision avoidance; in this chapter, the first one of those sub-systems, an algorithm for Preceding Vehicles Detection (PVD), is presented in details.

Normally, a PVD system is itself composed of two different stages of Hypothesis Generation (HG) of vehicle candidates to be detected, and Hypothesis Verification (HV) where those candidates are verified. In the following sections the solutions employed in these two stages are described.

More precisely the HG stage is based on prior-knowledge information methods, such us edge detectors and red-color segmentation, while the HV proposed algorithm is based on Gabor Filtering for feature extraction and target description, in combination with SVM classification for discrimination of true detections and minimization of false positives, which in the long term leads to a better behavior of the system. As preamble of the proposed modules, a small analysis on the importance and challenges of such a system is presented. To conclude the chapter, a series of experiments carried out for the enhancement of the PVD module presented are depicted as well.

3.2 PVD Scenario/Review

Robust and reliable vehicle detection in images is the critical step for these systems and self-guided vehicles as well as traffic controllers. This is a very challenging task since it is not only affected by the size, shape, color, and pose of vehicles, but also by lighting conditions, weather, dynamic environments and the surface of different roads. A vehicle detection system must also distinguish vehicles from all other visual patterns which exist in the world, such as similar looking rectangular objects [52] [53].

Vehicle detection in traffic scenes is an important issue in driver assistance systems and self-guided vehicles that includes two stages of Hypothesis Generation (HG) and Hypothesis Verification (HV). Both stages are important and challenging. In the first stage, potential vehicles are hypothesized and in the second stage, all hypotheses are verified and classified into vehicle and non-vehicle classes.

Various HG methods have been suggested in the literature and can be classified in three basic categories [25]: (1) knowledge-based, (2) stereo-based and (3) motion-based. Knowledge-based methods employ information about color and vehicle shape as well as general information about the context such as shadow, symmetry, horizontal/vertical edges, color, texture, and vehicle lights [54]. Stereo-based approaches usually employ the Inverse Perspective Mapping (IPM) to estimate the locations of people, vehicles and obstacles in the images. Motion-based approaches detect object such as people, vehicles and obstacles using optical flow. However, generating a displacement vector for each pixel is a time-consuming task and impractical for real time systems. To attack this problem, discrete methods employ the image features such as color blobs or local intensity minima and maxima. In the HV stage, correctness of hypotheses are verified and sorted into vehicle and non-vehicle classes. The HV approaches can be divided into two categories: (1) template-based and (2) appearance-based. The template-based methods employ the predefined patterns of the vehicle class and perform correlation between the template and the image. In the appearance-based methods, the characteristics of the vehicle appearance are learned from a set of training images which capture the variability in the vehicle class. Usually, the variability of the non-vehicle class is also modeled to improve performance. To begin, each training image is presented by a set of global or local features. Then, the decision boundary between vehicle and non-vehicle classes is learned either by training a classifier (e.g. Support Vector Machine, Adaboost and Neural Network) or by modeling the probability distribution of the features in each class (e.g. employing the Bayes rule assuming Gaussian distributions). For a more extended and detailed review on PVD, see [24-25] [52-53].

When it comes to this thesis, in the Hypothesis Generation stage a multi-scale approach based on vertical and horizontal edge map in combination with red-rear-light segmentation is used to create potential regions where vehicles may be present. Later on, in the HV stage all hypotheses are verified by using a set of Gabor filters for feature extraction purposes, and a support vector machine (SVM) for classification. Gabor filters provide a mechanism to extract line and edge information by tuning orientation and changing scales. Then, a linear Support Vector Machines (SVMs) classifier is train based on a large dataset of real vehicles and non-vehicles in real driving environments in order to determine true detections by the algorithm. To improve the performance of the PVD module a study in details of Gabor bank of filters configuration for feature extraction is accomplished. While results in the HG stage are insufficient due to time constrains, results of the proposed HV stage showed good classification accuracy of more than 97% correct classification on realistic on-road vehicle dataset images and also it has better classification accuracy in comparison with other approaches. Both stages, leading to the PVD system aimed are presented in detail within the following sections.

3.3 Hypothesis Generation (HG)

As see in the previous chapters, the input data of our vision system consists of image sequences taken from a camera mounted inside our car, just behind the windshield. The images show the environment in front of the car: the road, other cars, bridges, and trees next to the road. The primary task of the system is to distinguish the cars from other stationary and moving objects in the images and recognize them as cars. This is a challenging task, because the continuously changing landscape along the road and the various lighting conditions that depend on the time of day and weather are not known in advance. In addition the recognition of vehicles that suddenly enter the scene is difficult [29].

The approach proposed in this thesis is cored in knowledge-based methods. To hypothesize possible vehicle locations in an image, prior knowledge about rear vehicle view appearance seems feasible, since rear vehicle views contain lots of horizontal and vertical structures, such as rear-window or bumpers, as well as other characteristics such as red tail lights common to every vehicle. Based on this observation, a similar procedure to the one presented in [40] is applied to hypothesize candidate vehicle locations. First, interesting horizontal and vertical structures are identified by applying horizontal and vertical Sobel edge detectors, picking up later on the most promising horizontal and vertical structures by extracting the horizontal and vertical profiles of the edge images and performing some further analysis to identify the strongest peaks (e.g., last row of Figure 3.1).

However, despite its effectiveness, this process depends on a number of parameters that affect system performance and robustness. For example, we need to decide the thresholds for the edge detection step, the thresholds for choosing the most important vertical and horizontal edges, and the thresholds for choosing the best maxima (i.e., peaks) in the profile images. A set of parameter values might work well under certain conditions, however, they might fail in other situations. The problem is even more severe for on-road vehicle detection since the dynamic range of the acquired images is much bigger than that of an indoor vision system.

To deal with this issue, as presented in [40] a multi-scale approach which combines sub-sampling with smoothing to hypothesize possible vehicle locations more robustly is developed. Assuming that the input image is f, let set $f(K) = f$. The representation of $f(K)$ at a coarser level $f(K-1)$ is defined by a reduction operator.

Figure 3.1: Multi-scale hypothesis generation

Within the previous figure [40], the size of the images increases by rows from up to bottom. Moreover in the first column the images have been obtained by applying low pass filtering at different scales; second column: vertical edge maps; third column: horizontal edge maps; fourth column: vertical and horizontal profiles. All images have been scaled back to highest resolution for illustration purposes.

In our case the size of the input images from the TME Motorway Dataset is $1024x768$, therefore our three levels of detail are: $f^k(1024x768)$, $f^{k-1}(512x384)$, and $f^{k-2}(256x192)$. At each level, we process the image by applying the following steps: (1) low pass filtering (e.g., first column of Fig. 3.1), (2) vertical edge detection (e.g., second column of Fig. 3.1), and vertical profile computation of the edge image (e.g., last column of Fig. 3.1), (3) horizontal edge detection (e.g., third column of Fig. 3.1), and horizontal profile computation of the edge image (e.g., last column of Fig. 3.1), (4) local maxima and minima detection (e.g., peaks and valleys) of the two profiles. The peaks and valleys of the profiles provide strong information about the presence of a vehicle in the image.

Starting from the coarsest level of detail $f(K-2)$, first we find all the local maxima at that level. Although the resulted low resolution images have lost fine details, important vertical and horizontal structures are mostly preserved (e.g., first row of Fig. 3). Once we have found the maxima at the coarsest level, we trace them down to the next finer level $f(K-1)$. The results from $f(K-1)$ are finally traced down to level $f(K)$ where the final hypotheses are generated.

The proposed multi-scale approach improves system robustness by making the hypothesis generation step less sensitive to the choice of parameters. Forming the first hypotheses at the lowest level of detail is very useful since this level contains only the most salient structural features. Besides improving robustness, the multi-scale scheme speeds up the whole process since the low resolution images have much simpler structure as illustrated in Fig. 3.1 (i.e., candidate vehicle locations can be found faster and easier).

It should be noted as well that due to the complexity of the scenes, some false peaks are expected to be found. In order to get rid of them and reduce the rate of false hypothesis, once these have been generated at $f(K)$ level, a segmentation of red color is applied, so that according to the presence/non-presence of this within the generated hypothesis these can be considered as formal HG or discarded, since the presence of this color red is highly tight to the existence of a vehicle, symbolizing its tail lights.

3.3.1 Time Constraints

Time constraints in the planning of this master thesis, which represents a load of 21 ECTS towards the title of Master in Telecommunications by the Technical University of Cartagena (UPCT), a priori is meant to be completed within a university semester, that is to say around 5 months, didn’t make possible a full development of the HG module here presented..

While the natural workflow of the system goes from the Preceding Vehicle Detection module, to the Time-to-Collision estimation, passing through the tracking module, within the scope of this thesis the Time-To-Collision (TTC) estimation was the cornerstone and therefore priority, thus together to the tracking module necessarily linked to this one. Consequently, in the timeline set for the project these ones were the first ones to be developed, remaining for the last stage the PVD module. Moreover, in the going backwards timeline HV module was places before the HG module which eventually was the last one to be considered.

This backward setup was feasible thanks to the time-stamped characteristics of the TME Motorway Dataset, partially considered within the scope of this thesis because of this reason. This time-stamped information present in the dataset facilitates evaluation tasks of our system. However in our case, it eventually moreover allowed to simulate the output of our Hypothesis Generation stage, reason why the fact of not having totally completed our HG algorithm has no impact in the natural development of the whole project. As a matter of fact, the HV stage, tracking module and TTC estimator were successfully implemented, and evaluated showing indeed successful results.

Nevertheless, beside not having gotten in this stage results accurate enough to be considered within the whole scope of the thesis, some research work and enquires have been done towards this end, and an early solution has been proposed, so with this regard it's all about dedicating the necessary resources to be implemented and fused as part of the whole system, remaining this way as a future line of work. On the other side, the HV solution proposed in this thesis is completely presented in the following section.

3.4 Hypothesis Verification (HV)

As previously stated, the framework in this stage is feature extraction from the hypotheses generated in the HG stage, and classification of them into vehicle and non-vehicle classes. Therefore the performance of this stage is directly dependent to employing a classifier that is well trained by the appropriate features.

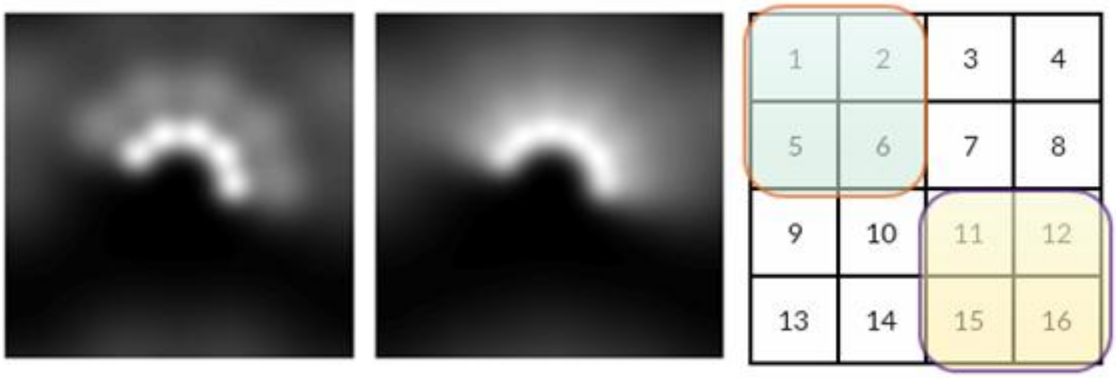
Figure 3.2: Normal appearance of a real driving scene within TME Dataset; Presence of vehicles, road, signs, guardrail, landscape, sunlight, shadows, etc.

For achieving this purpose, we propose an approach based on Gabor filter for feature extraction plus SVM classification. Gabor features are extracted from an image dataset as the primitive features for vehicle detection task since they have shown good results in this field and ultimately linear SVM for vehicle/non-vehicle classification is employed. [40] [45]

3.4.1 Gabor Features Extraction

In this section we describe our Gabor feature extraction procedure. As seen in Section 2.7, Gabor filters basically act as local band-pass filters. Figures 3.3 (a) and (b) show the power spectra of two Gabor filter banks (the light areas indicate spatial frequencies and wave orientation). In this paper, we use the approach strategy described in [40] and [55], and presented straight away.

Given an input image $I(x, y)$, Gabor feature extraction is performed by convolving $I(x, y)$ with a Gabor filter bank. Although the raw responses of the Gabor filters could be used directly as features, some kind of post-processing is usually applied (e.g., Gabor-energy features, thresholded Gabor features, and moments based on Gabor features). In this paper, we use Gabor features based on moments, extracted from several sub-windows of the input image.



(a) (b) (c)

Figure 3.3: Gabor Filter: (a) GF bank with 3 scales and 5 orientations; (b) GF bank with 4 scales and 6 orientations; (c) Feature extraction sub-windows: first and last $32x32$ sub-window from patches 1, 2, 5 and 6, and 11, 12, 15 and 16 respectively

In the context of this thesis, the input to this feature extraction sub-system are the hypothesized vehicle sub-images obtained in the HG phase (like the ones presented in Section 3.4.2). First each sub-image is scaled to a fixed size which is $64x64$. Then, it is subdivided into 9 overlapping $32x32$ sub-windows. Assuming that each sub-image consists of 16 $16x16$ patches (patches 1, 2, 5 and 6 comprise the first $32x32$ sub-window, 2, 3, 6 and 7 the second, 5, 6, 9, and 10 the fourth, and so on till the 9th subwindow (see Figure 3.3 (c)). The Gabor filters are then applied on each sub-window separately.

The motivation for extracting possibly redundant Gabor features from several overlapping sub-windows is to compensate for errors in the hypothesis generation step (e.g., sub-images containing partially extracted vehicles or background information), making feature extraction more robust. After applying Gabor filters on the image with different orientation at different scale, we obtain an array of magnitudes:

$$E(m,n) = \sum_x \sum_y |G_{mn}(x,y)|\,, m = 0,1,\dots,M-1; n = 0,1,\dots,N-1 \tag{3.1}$$

These magnitudes of the Gabor filter responses, representing the energy content at different scale and orientation of the image, are ultimately collected from each sub-window and represented by three moments: the *mean* μ_{ij}, the standard *deviation* σ_{ij}, and the *skewness* κ_{ij} (i.e., i corresponds to the i-th filter and j to the j-th sub-window). Using moments implies that only the statistical properties of a group pixels is taken into consideration, while position information is essentially discarded. This is particularly useful to compensate for errors in the hypothesis generation step (i.e., errors in the extraction of the sub-images). Suppose we are using S = 2 scales and K = 3 orientations (i.e., S x K filters). Applying the filter bank on each of the 9 sub-windows, using μ_{ij}, σ_{ij}, and κ_{ij} as the feature components, yields a feature vector of size 162, having the following form:

$$f = [\mu_{11}\ \sigma_{11}\ \kappa_{11}, \mu_{12}\ \sigma_{12}\ \kappa_{12}, \dots\ \mu_{69}\ \sigma_{69}\ \kappa_{69}] \tag{3.2}$$

Where the k^{th} standardized moment may be generalized as [56]:

$$\hat{\mu}_k = \frac{\mu_k}{\sigma^k} = \frac{\mathrm{E}[(\mathrm{X}-\mu)^{\mathrm{k}}]}{(\mathrm{E}[(\mathrm{X}-\mu)^{\mathrm{k}}])^{\mathrm{k}/2}} \tag{3.3}$$

The use of three moments instead of using the first two moments only is due to much worst results obtained using just two moments, which implies that the skewness information is very important for our problem of preceding vehicle detection.

3.4.2 SVM Classification

For SVM Classification purposes the TME Motorway Dataset presented in Section 1.5.2 is divided in three parts comprising around one third of the whole dataset each one: Training Data, Validation Dataset, and Test Data. As their names indicate, the first sub-dataset is used for training purposes of the SVM classifier implemented, while the Validation dataset is employed for validation of the classifier behavior, and the Test Dataset to measure the final behavior of the classifier in terms of accuracy, etc. In this paper, accuracy as presented in Chapter I is the evaluation key since is the ratio of the number of data recognized correctly by the algorithm against the entire data set.

More specifically that is to say, we employ linear SVM for vehicle/non-vehicle classification which is trained with the Training dataset. Then Validation dataset is used for validation of the better/worse behavior of the classifier. And finally the classifier is tested making use of the Test dataset obtaining the classification accuracy of the proposed HV stage proposed, among other metrics of its behavior*.*

Following figures collect random examples of images belonging to both classes *vehicle* and *non-vehicle* used with training purposes of the aforementioned SVM classifier.

Figure 3.4: *Vehicle* class

Figure 3.5: *Non-Vehicle* class

In order to get the better results the SVM classifier must be trained regarding the Optimum Parameters and a Training Dataset as complete as possible. To this end, a series of experiment have been carried out. All of them are presented in the next section.

3.4.3 Experiments Results and Comparison

In the scope of this thesis we would like to find the most adequate combinations of parameters (σ, θ, γ…) when featuring extraction towards our particular objective of vehicle detection.

To this end, several series of different test have been carried out. Most specifically these tests aim to maximize the behavior of the classifier in terms of accuracy by optimizing the parameters by which the Gabor Filter used, for featuring extraction, is defined, as well as by optimizing the size and composition of the Training Dataset used to train the classifier. Both approaches are presented straight away.

GF Parameter Optimization

As introduced in Section 2.7, in its most simplistic way Gabor filter can be defined as band-pass filter formed by a Gaussian kernel function in the two dimensional spatial domain. In mathematical terms a Gabor filter is a sinusoidal plane wave that is formulated as follow:

$$G(x,y) = exp\left\{-\frac{x^2 + \gamma^2 {y'}^2}{2\sigma^2}\right\} exp\{\, i2\pi \frac{x'}{\lambda} + \varphi\} \tag{3.4}$$

Where:

$$y' = x\cos\theta + y\sin\theta \tag{3.5}$$

$$x' = -x\sin\theta + y\cos\theta \tag{3.6}$$

G is the Gabor filter kernel at spatial coordinate (x, y) with a certain dimension, and the other parameters described as: waves orientation θ, phase offset φ, standard deviation σ, aspect ratio γ, and wavelength of sinusoidal factor λ.

In this context we proceed to measure how each parameter impact the behavior of the SVM classification in terms of accuracy. And the procedure to accomplish this stage is always the same: given a set configuration for each parameter, the parameter under test is gradually modified while comparing results in order to see which configuration shows the highest performance. Additionally to those parameters, in order to compute the Gabor Filter, OpenCV considers an extra parameter called $Ksize$, representing the size of the kernel of the Gabor Filter being implemented. Therefore this parameter must be optimized as well. Moreover, all test are run using a Training Dataset of 800 random images (400 vehicles and 400 non-vehicles), plus a Validation Dataset of another 400 random images (200 vehicles and 200 non-vehicles). Besides *Accuracy (A)*, results included comprise rates of true positives, true negatives, false positives and false negatives: tp, tn, fp, fn (in %) and precision, recall, quality and effectiveness: P, R, g and F as presented in Chapter I (in scale [0, 1] being 1 the best result and 0 the worst possible). As so, the tests carried out and their correspondent outcomes are sum up in the figures bellow.

First parameters to be tested are $Ksize$, $Lambda$ and $Gamma$ since their impact in the whole behavior of the system will be less powerful than the caused by the orientations and scales chosen to implement. This is due to the size of the feature vector for each image is directly dependent on the scales and orientations implemented, as seen in Section 3.4.1, and therefore the response of the classifier in terms of time, which is a restriction we must take into seriously account since the system must be able to run in real time. On the other hand the feature vector's size is not affected by $Ksize$, $Lambda$ and $Gamma$. The results obtained for these parameters are collected in the following charts.

Table 3.1: $Ksize$

Ksize	tp	tn	fp	fn	P	R	g	F	Accuracy (A)
1	74	87.5	12.5	26	0.85549	0.74	0.65778	0.79357	0.8075
3	76	88	12	24	0.86364	0.76	0.67857	0.80851	0.82
5	73	88.5	11.5	27	0.86391	0.73	0.65471	0.79133	0.8075
7	71	86.5	13.5	29	0.84024	0.71	0.62555	0.76965	0.7875
9	74	87.5	12.5	26	0.85549	0.74	0.65778	0.79357	0.8075
11	**91.5**	**95.5**	**4.5**	**8.5**	**0.95313**	**0.915**	**0.8756**	**0.93367**	**0.935**
13	85.5	94.5	5.5	14.5	0.93956	0.855	0.81043	0.89529	0.9
15	90.5	93	7	9.5	0.92821	0.905	0.84579	0.91646	0.9175
17	88	94.5	5.5	12	0.94118	0.88	0.83412	0.90956	0.9125
19	81.5	94.5	5.5	18.5	0.93678	0.815	0.77251	0.87166	0.88
21	80	89.5	10.5	20	0.88398	0.8	0.72398	0.8399	0.8475
23	83	92	8	17	0.91209	0.83	0.76852	0.86911	0.875
25	82	93	7	18	0.92135	0.82	0.76636	0.86772	0.875

Table 3.2: $Lambda, \lambda$

Lambda - λ		tp	tn	fp	fn	P	R	g	F		Accuracy (A)
2.5		88.5	92	8	11.5	0.9171	0.885	0.81944	0.90076		0.9025
5		88	92.5	7.5	12	0.92147	0.88	0.8186	0.90026		0.9025
7.5		89	96	4	11	0.95699	0.89	0.85577	0.92228		0.925
9		89	94.5	5.5	11	0.9418	0.89	0.8436	0.91517		0.9175
9.5		90.5	93.5	6.5	9.5	0.93299	0.905	0.84977	0.91878		0.92
9.8		89.5	95.5	4.5	10.5	0.95213	0.895	0.85646	0.92268		0.925
9.9		90	97	3	10	0.96774	0.9	0.87379	0.93264		0.935
10		**91.5**	**95.5**	**4.5**	**8.5**	**0.95313**	**0.915**	**0.8756**	**0.93367**		**0.935**
10.1		92	95	5	8	0.94845	0.92	0.87619	0.93401		0.935
10.5		86.5	95.5	4.5	13.5	0.95055	0.865	0.82775	0.90576		0.91
11		84	92.5	7.5	16	0.91803	0.84	0.7814	0.87728		0.8825
12.5		74.5	87	13	25.5	0.85143	0.745	0.65929	0.79467		0.8075
15		72.5	85.5	14.5	27.5	0.83333	0.725	0.63319	0.7754		0.79

Table 3.3: $Gamma, \gamma$

Gamma - γ		tp	tn	fp	fn	P	R	g	F		Accuracy (A)
0.1		92	95	5	8	0.94845	0.92	0.87619	0.93401		0.935
0.3		91.5	95	5	8.5	0.94819	0.915	0.87143	0.9313		0.9325
0.5		91.5	95.5	4.5	8.5	0.95313	0.915	0.8756	0.93367		0.935
0.6		**91.5**	**95.5**	**4.5**	**8.5**	**0.95313**	**0.915**	**0.8756**	**0.93367**		**0.935**
0.7		91.5	95.5	4.5	8.5	0.95313	0.915	0.8756	0.93367		0.935
0.9		91	95	5	9	0.94792	0.91	0.86667	0.92857		0.93
1.5		92	94	6	8	0.93878	0.92	0.86792	0.92929		0.93
4		89.5	93.5	6.5	10.5	0.93229	0.895	0.84038	0.91327		0.915
10		89.5	95	5	10.5	0.94709	0.895	0.85238	0.92031		0.9225

As it can be seen, while the accuracy of the system is only affected in a 1% by $gamma$, when it comes to $ksize$ and $lambda$, it oscillates around a 15%, which is a quite considering value. See highlighted in yellow the best value for each case: $ksize\ =\ 11,\ \lambda = 10$ and $\gamma = 0.6$, obtaining in all of them an accuracy, $A = 93.5\%$.

Regarding to the orientations θ and scales σ to be used for optimizing results, these are the tests carried out:

Table 3.4: $Theta, \theta$

Orientation - θ		tp	tn	fp	fn	P	R	g	F		Accuracy (A)
2		90.5	93.5	6.5	9.5	0.93299	0.905	0.84977	0.91878		0.92
3		86	93	7	14	0.92473	0.86	0.80374	0.89119		0.895
4		91.5	94.5	5.5	8.5	0.9433	0.915	0.8673	0.92893		0.93
5		90	95.5	4.5	10	0.95238	0.9	0.86124	0.92545		0.9275
6		**91.5**	**95.5**	**4.5**	**8.5**	**0.95313**	**0.915**	**0.8756**	**0.93367**		**0.935**
7		89	94	6	11	0.93684	0.89	0.83962	0.91282		0.915
8		92	94.5	5.5	8	0.94359	0.92	0.87204	0.93165		0.9325
9		90.5	95	5	9.5	0.94764	0.905	0.8619	0.92583		0.9275
10		90	97	3	10	0.96774	0.9	0.87379	0.93264		0.935

Even with a difference of almost 5% between the best and worst results, in the first case of $theta$ tests carried out show better accuracies for even orientations (4, 6, 8 and 10), getting the best results for $\theta = 6$ and $\theta = 10$. However as stated before the processing time of the system increases with the amount of orientations used, that is to say that the more orientations used the slower the system behaves. In consequence the optimum value in case of θ would be to use 6 orientations.

Table 3.5: $Sigma$, σ (1)

Scale - σ	tp	tn	fp	fn	P	R	g	F	Accuracy (A)
[1, 3, 5, 7]	**91.5**	**95.5**	**4.5**	**8.5**	**0.95313**	**0.915**	**0.8756**	**0.93367**	**0.935**
[3, 5, 7, 9]	91	95	5	9	0.94792	0.91	0.86667	0.92857	0.93
[5, 7, 9, 11]	91.5	93.5	6.5	8.5	0.93367	0.915	0.85915	0.92424	0.925
[7, 9, 11, 13]	90.5	91	9	9.5	0.90955	0.905	0.83028	0.90727	0.9075
[9, 11, 13, 15]	85.5	92.5	7.5	14.5	0.91935	0.855	0.79535	0.88601	0.89
[11, 13, 15, 17]	83	91	9	17	0.90217	0.83	0.76147	0.86458	0.87
[13, 15, 17, 19]	82.5	90	10	17.5	0.89189	0.825	0.75	0.85714	0.8625
[15, 17, 19, 21]	83.5	90	10	16.5	0.89305	0.835	0.75909	0.86305	0.8675
[17, 19, 21, 23]	87.5	90.5	9.5	12.5	0.90206	0.875	0.79909	0.88832	0.89
[19, 21, 23, 25]	86.5	92	8	13.5	0.91534	0.865	0.80093	0.88946	0.8925
[21, 23, 25, 27]	87	92	8	13	0.91579	0.87	0.80556	0.89231	0.895
[23, 25, 27, 29]	86.5	91.5	8.5	13.5	0.91053	0.865	0.79724	0.88718	0.89

It must be noticed that at this stage of the process global results are not relevant, but relative results in comparison to those obtained making use of different configurations. When it comes to the scales to be used, following the approach in [55] and presented above, we decide to start from a configuration of 4 scales. It can be clearly seen in the results that best results are obtained for low scale values. In order to precise more at this point further tests are carried out.

Table 3.6: $Sigma$, σ (2)

Scale - σ	tp	tn	fp	fn	P	R	g	F	Accuracy (A)
[2, 4, 6, 8]	87.5	95	5	12.5	0.94595	0.875	0.83333	0.90909	0.9125
[4, 6, 8, 10]	88	93.5	6.5	12	0.93122	0.88	0.82629	0.90488	0.9075
[6, 8, 10, 12]	87.5	93.5	6.5	12.5	0.93085	0.875	0.8216	0.90206	0.905
[1, 4, 7, 10]	92.5	94	6	7.5	0.93909	0.925	0.87264	0.93199	0.9325
[1, 3, 5]	87	93.5	6.5	13	0.93048	0.87	0.8169	0.89922	0.9025
[3, 5, 7]	**91.5**	**95**	**5**	**8.5**	**0.94819**	**0.915**	**0.87143**	**0.9313**	**0.9325**
[5, 7, 9]	91.5	94.5	5.5	8.5	0.9433	0.915	0.8673	0.92893	0.93
[1, 5, 9]	89.5	96	4	10.5	0.95722	0.895	0.86058	0.92506	0.9275
[2, 4, 6]	85.5	93.5	6.5	14.5	0.92935	0.855	0.80282	0.89063	0.895
[4, 6, 8]	87.5	94.5	5.5	12.5	0.94086	0.875	0.82938	0.90674	0.91
[6, 8, 10]	88	94	6	12	0.93617	0.88	0.83019	0.90722	0.91
[2, 6, 10]	87	91	9	13	0.90625	0.87	0.79817	0.88776	0.89

New test show the accuracy result previously obtained with a configuration of 4 scales [1, 3, 5, 7] is the highest one achieved. However, very close result have been obtained A=93.25%, only 0.25% inferior to the best results, and this time with a configuration of just 3 scales. Again, the real time constrains this project presents leads us to a configuration of 3 scales [3, 5, 7] as the optimum one to be implemented in our bank of Gabor Filters for feature extraction purposes.

Training Dataset Optimization

Once the parameter optimization stage is done the next step is optimizing the training dataset to be used in the implementation of our SVM classifier. To that end a second set of 400 random images (200 vehicles and 200 non-vehicles) is extracted from the Validation Dataset in order to gradually test training datasets of different sizes in order to see which configuration achieves the best results once again in terms of accuracy. Tests carried out in this stage are presented together with their results in the following chart:

Table 3.7: *Size* Training Dataset

Size	tp	tn	fp	fn	P	R	g	F	Accuracy (A)
400	91	95	5	9	0.94792	0.91	0.86667	0.92857	0.93
800	91.5	95	5	8.5	0.94819	0.915	0.87143	0.9313	0.9325
1200	93	94.5	5.5	7	0.94416	0.93	0.88152	0.93703	0.9375
1600	93.5	94	6	6.5	0.9397	0.935	0.88208	0.93734	0.9375
2000	**93.5**	**95.5**	**4.5**	**6.5**	**0.95408**	**0.935**	**0.89474**	**0.94444**	**0.945**
2400	93.5	94	6	6.5	0.9397	0.935	0.88208	0.93734	0.9375

Results show Accuracy increases gradually with the size of the training dataset, reaching its highest peak at a value of 94.5% for a dataset size of two thousand images (half vehicle, half non-vehicles), decreasing again while the size parameter keeps growing. As so, 2000 images is the final size of the dataset employed to train our SVM classifier.

3.4.4 Final Evaluation

Once the system parameter optimization has been accomplished the Preceding Vehicle Detection algorithm is eventually evaluated. To this purpose from the evaluation sub-dataset a set of same size that the final training dataset, that is to say 2000 photos, is extracted. Then with a group of 1000 photos of vehicle and non-vehicle we proceed to evaluate the PVD module behavior.

Table 3.8: Final TME Motorway Evaluation Results

Dataset	tp	tn	fp	fn	P	R	g	F	Accuracy (A)
TME Motorway	95.5	97	3	4.5	0.96954	0.955	0.92718	0.96222	**96.25%**

Table 3.9: Caltech DBs Evaluation Results

Dataset	Size (images/frames)	True Positives (TP)	False Negatives (FN)	Detection Rate
Caltech 1999	126	119	7	**94.44%**
Caltech 2001	526	492	34	**93.54%**

Results show how our system, after the optimization process, and over an diverse evaluation dataset of 2000 images, behaves at low rates of false positive and negative detections, showing an overall behavior of 96.25% of Accuracy; results successfully considered in the scope of this thesis.

Furthermore, for a more complete evaluation the system proposed is tested in 2 more different datasets called *Caltech 1999* and *Caltech 2001*, both presented in Chapter I*.* Even if they don't exactly represent vehicles in real live driving environments, both datasets are ideal for the evaluation of preceding vehicle algorithms, since they contain a variety of vehicles in front of the ego-vehicle. Moreover, Caltech DBs dataset has been used in other researches to evaluate different systems for vehicle detection such as [57] or [58] which makes it interesting for comparing purposes. However, while in all those cases a whole PVD system has been evaluated in our case, as we have seen only the HV phase have been successfully implemented, and therefore until our system is completed establishing comparisons wouldn't bring up more information. Nevertheless for evaluation purposes of our systems this datasets are ideal. As so, respective detection rates of 94.44% and 93.54% reveal a successful behavior of our PVD algorithm, confirming previous results on the TME Motorway Dataset.

3.5 Conclusions

In this chapter, the first module of the TTC system this thesis aims for, a Preceding Vehicle Detection (PVD) system, has been proposed. The system comprises a Hypothesis Generation and Verification stage, based on a multi-scale Sobel vertical and horizontal edge detector combined with a phase of red tail light segmentation in first place, followed by a verification stage based on a Support Vector Machine Classifier which finds each core in a bank of Gabor Filters previously parameterized for feature extraction of the input images. Time constraints didn't make possible a full development of the HG module and therefore the complete system, however thanks to the characteristics of the Dataset considered in the scope of this thesis this fact has no impact in the natural development of the whole project. As a matter of fact, the HV stage was successfully implemented, and evaluated; several test carried out in order to optimize its behavior as well as others to evaluate its final behavior have indeed shown successful results by this first module of our TTC system.

Chapter IV
Tracking Module

4.1 Introduction

As seen in the previous section, the Hypothesis Generation and Verification stages work certainly successfully for static images within the Preceding Vehicles Detection algorithm. More precisely, simply what it happens is that once at a time, and in order, a frame from the video sequence under test is taken and processed, giving ultimately the vehicle detections candidates found in that frame. And so forth with the following frames. However, this might not always be sufficient for dealing with dynamically changing scenes such as a driving environment. In this way, the process of temporal information present in video streams is being dismissed. In other words, processing frames individually obvious the intrinsic information present in the relation of consecutive frames, for the mere fact of being part of a bigger common whole which is the video stream. That is to say that theoretically every frame in a video sequence is totally independent, while in practice there is some information that can be derived from the relation between consecutive frames. Thus, the incorporation of a stage to take advantage of the temporal continuity of video data would no doubts enhance the behavior and accuracy of the system.

In this context, this chapter proposes a tracking algorithm based on a Kaman Filter which allows the suppression of false negative detections, performs smoothing of the detection noise and provides information for the temporal association between frames by interpolating positions during sporadic erroneous or missed detections. More specifically, the reduction of false negative rates in the PVD stage is achieved by checking the persistency and accuracy of the HV candidates in time. This has the effect of removing HV candidates that do not appear for more than a couple of frames, thus improving the precision of the system. Furthermore, the spatial accuracy over time is improved, as it provides additional information for the most appropriate candidates in each frame which are also consistent in time. And ultimately it also allows reducing the rate of lost vehicles in tracking estimating their position when sporadically the PVD stage could fail in their detection, improving in this way the whole behavior and features of the proposed system.

4.2 PVD Consistency

At the end of the Preceding Vehicle Detection (PVD) stage (see Chapter III) a series of candidate vehicle detections has been extrapolated from the image information within the frame processed. As a mere of fact, this series of probable vehicle detected acts as the only input of this stage of the system proposed.

Figure 4.1: PVD stage output in example frames

However, for the understanding of the system in its whole scope these candidate vehicle detections are not considered yet as former detections by the system. Given that the relative movement of the preceding vehicles between two successive frames is very small (in practice, as pointed out in the (chapter of datasets) the frame acquisition of the dataset considered in this thesis is $10\,Hz$, which means a frame every $0.1\,sec$), seems acceptable to presume that two consecutive frames will look almost the same, both for the human eye and in computational terms. Thus, in order to consider a detection candidate as former detection and start to be tracked by the system, this candidate detection being consistent in time at least for three consecutive frames is established as must condition.

To this aim, and in order to initialize the tracking procedure, the position and size of the detected preceding vehicle are stored and monitored for several consecutive frames, to examine the reliability of the detection. More precisely the detection candidates proposed by a frame are stored and compared with the two following frames, according to two considerations: the position and size of candidate within the whole frame being processed. Taking into account that the false positives are not persistent in time, i.e. very rarely the same false positive appears in more than one frame, detections with a certain consistency along several consecutive frames ensure true vehicle detection.

And so, as stated before, since the variation from two consecutive frames is almost non-existent, is acceptable to presume that if a candidate is detected in a position (x, y) and with a size (w, h) in the frame k (keeping the nomenclature of Chapter II), its position and size in the next frame $k + 1$ will be almost the same.

For this purposed the Euclidean distance, presented as well in Chapter II, between the position of the candidate vehicle in both frames is taken into account, as well as comparison between the size of this in both frames respectively.

Thus, given the boundary boxes of the vehicle candidate for the frame k and $k + 1$:

$$candidate_k = (xmin_k,\ ymin_k,\ xmax_k,\ ymax_k) \tag{4.1}$$

$$candidate_{k+1} = (xmin_{k+1},\ ymin_{k+1},\ xmax_{k+1},\ ymax_{k+1}) \tag{4.2}$$

The position (x, y) and the size (w, h) of the candidate can be estimated as the follows:

$$position_k = (xmin_k + \frac{xmax_k - xmin_k}{2}, ymin_k + \frac{ymax_k - ymin_k}{2}) \tag{4.3}$$

$$(w, h)_k = (xmax_k - xmin_k, ymax_k - ymin_k) \tag{4.4}$$

$$size_k = w_k * h_k \tag{4.5}$$

$$position_{k+1} = (xmin_{k+1} + \frac{xmax_{k+1} - xmin_{k+1}}{2}, ymin_{k+1} + \frac{ymax_{k+1} - ymin_{k+1}}{2}) \tag{4.6}$$

$$(w, h)_{k+1} = (xmax_{k+1} - xmin_{k+1}, ymax_{k+1} - ymin_{k+1}) \tag{4.7}$$

$$size_{k+1} = w_{k+1} * h_{k+1} \tag{4.8}$$

And therefore according to (2.1):

$$d_e(position_k,\ position_{k+1}) \leq thres_{position} \tag{4.9}$$

$$|size_k - size_{k+1}| \leq thres_{size} \tag{4.10}$$

Considering both detection as consistent always that the Euclidean Distance and the size difference are under a set thresholds, and the detection candidate as *true preceding vehicle,* becoming a tracking target, as long as that consistency takes place, as stated before, along three consecutive frames.

The example presented below illustrates how this module allows reducing false positive rate from PVD Stage.

Figure 4.2: PVD Consistency 4-frames-sequence example

Figure shows a sequence of 4 consecutive frames from one of the TME Motorway Dataset video sequences, where PVD module has been applied. It can be seen how in frame 752 a false positive candidate (red) has successfully passed the HV stage in PVD module. However it is obvious this is due to a sporadic wrong detection that is not consistent in time, while all five other vehicles detections are consistent along the sequence.

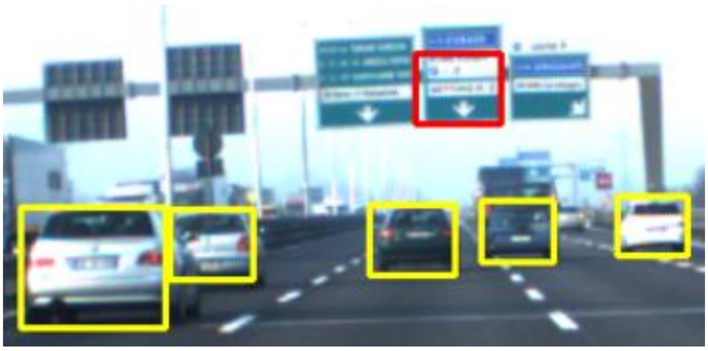


Figure 4.3: Zoom-in frame 752 clearly reveals the presence of a wrong detection

Thanks to this stage sporadic false positive detection like the one shown by the example are dismissed, improving the behavior of the system. Other examples of false positive detections can be seen in red within the following images:

Figure 4.4: False Positive Detections

4.3 Kalman Filter Tracker

Based on control theory Kalman filter is one of the most well-known methods of object tracking that exist [59] [60] [61]. This method is able to predict the status of a linear system with Gaussian distribution using a recursive algorithm. Kalman filter determines the object position in the next frame based on the motion type of the moving object in the previous frames so that the presence probability of the target object at that point is maximum.

In the context of this thesis, and retaking the above mentioned, given that the relative movement of preceding vehicles between two successive frames is virtually non-existent, the linear discrete-time Kalman filter reviewed in Chapter II represents an optimal choice over some other more complex techniques such as the extended Kalman filter or particle filter, which in theory are more demanding computationally speaking, in order to address the problem of modeling its movement.

As already described, the Kalman filter is a recursive, adaptive technique that estimates the state of a dynamic system from a series of noisy measurements, presenting a fast and straightforward implementation, requiring only for this purpose the tracking estimate of the previous frame. More precisely, it estimates the state of a process by recursively updating the system dynamics in two phases: the *time update* phase and the *measurement update* phase. The *time update* equations project forward in time the tracking estimates (state vector and state error covariance matrix) of the previous frame in order to obtain *a-priori* predictions for the current frame. The predicted position is the expected measurement, given all the previous measurements and the mechanics of the system. The state transition matrix, which captures the mechanics of the system, is derived from the theory of motion under constant acceleration. On the other hand, the *measurement update* equations (denoted also as correction equations) incorporate the new measurement into the *a-priori* prediction in order to obtain an improved, final estimate. Furthermore a Kalman gain serves at reflecting the importance of the prediction to the current measurement which, in turn, depends on a measurement noise covariance matrix R. Thus, the Kalman gain is used to update the predictions, given a current measurement. See Chapter II for a deeper understanding of Kalman Filter theory.

In this thesis, the Kalman filter method, as already many times pointed out before, is used to track the vehicles that reached the state of *detected* or *true preceding vehicle*. In the previous step, vehicles were detected, staying characterized within the context of the image sequence by the coordinates of their boundary boxes $(xmin,\ \ ymin,\ \ xmax,\ \ ymax)$. At this stage of the system those coordinates are used as the parameters of the state vector of the Kalman filter, in order to address the tracking purposes. Then the state vector X is denoted as:

$$X = (xmin,\quad ymin,\quad xmax,\quad ymax)^T \tag{4.11}$$

The prediction of the state vector $\hat{X}_{k+1}$ could be estimated by the state transition equation:

$$X(k) = A(k|k-1)X(k-1) + w(k-1) \tag{4.12}$$

The measurement equation is expressed as follows:

$$Z(k) = C(k)X(k) + r(k) \tag{4.13}$$

Where $A(k|k-1)$ and $C(k)$ are the state transition and measurement matrix, respectively, which can be defined as:

$$A(k|k-1) = \begin{bmatrix} 1 & 0 & 0 & 0 & dt & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & dt & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & dt & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & dt \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.14}$$

$$C(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.15}$$

With $dt = 0.1$, defined as the difference between consecutive video frames (frame sampling time at a frequency of 10Hz). And where $w(k)$ and $r(k)$ are assumed to be independent, zero-means and white Gaussian noise with their own covariance matrices W_k and R_k. The measurement noise covariance parameter R is used to determine the sensitivity of the tracker to updates, by assigning weighting between the current measurements and the previous estimates. In our specific case, knowing are working frame rate is $10\,fps$, a value of $dt = 0.1$ ensures a good tradeoff between responsiveness and smoothness during noisy measurements.

So at this point, in order to initialize the tracking procedure, the consistency of the position and size of the detected preceding vehicle have been monitored for several consecutive frames, as seen in the previous section, until the detection is considered reliable. Once this fact is proven, the Kalman filter is initialized, and a unique ID is assigned to each vehicle detected, in order to keep track of them along the rest of the video sequence.

It must be noticed that normally in one frame there will be present more than just one vehicle, so actually, what we get as output from the PVD stage is a list of detected vehicle in the current frame (see Figure 4.3). Obviously, the initialization process mentioned is applied to every one of the vehicle detected.

In last term, after the initialization of the tracking procedure is completed, for each new frame, a template matching stage takes place in order to evaluate if the PVD candidates of the new frames match with any of the vehicles already being in track; and here several different scenarios must be considered. In case the match exists the actualization of the tracking state performing the *measurement update* stage is naturally applied. If on the contrary the detection doesn’t match any of the vehicles already being tracked that means a new detection took place and therefore the before mentioned initialization process is applied to this vehicle.

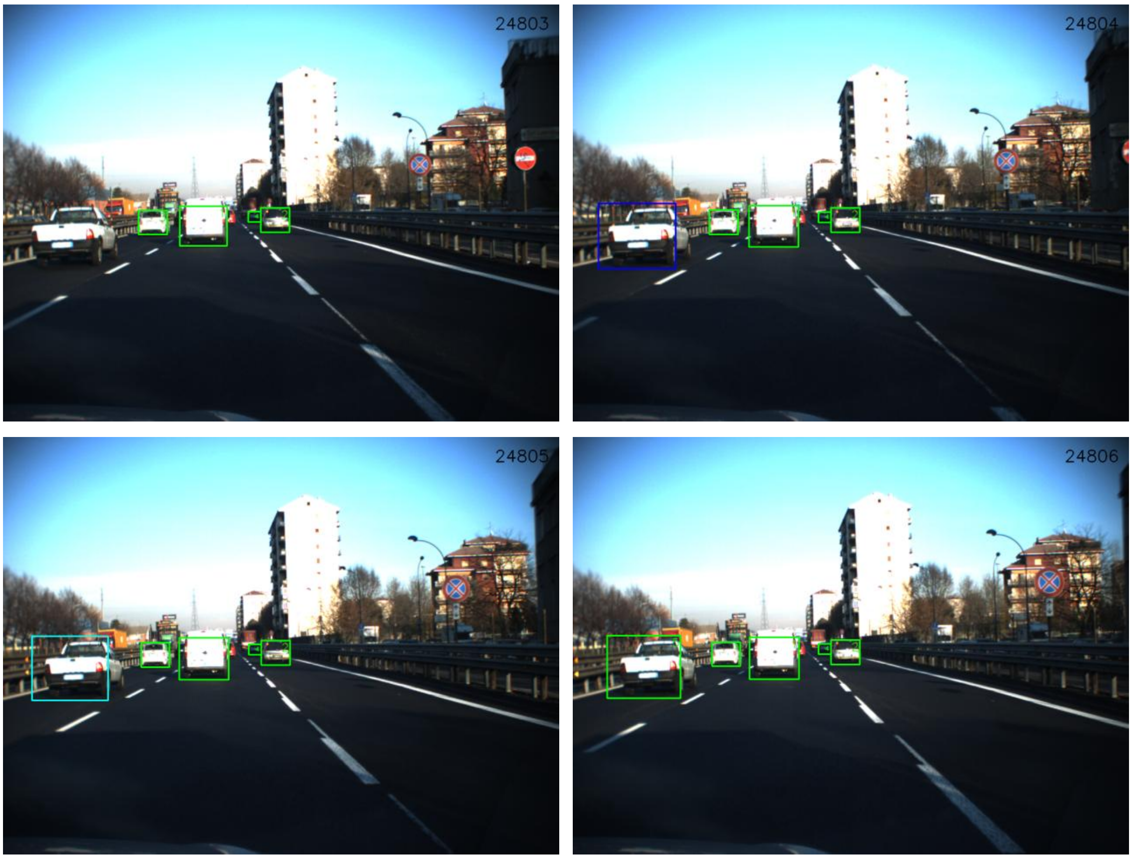


Figure 4.5: New Detection Real Case

This figure shows the real case of how in a scene where 4 vehicles are being tracked a new one overtaking us enters into it. As stated before, this new vehicle in our field of view is not considered as a true detection until is consistent along three consecutive frames. In this way it can be seen how the color of its boundary box change gradually from the dark blue of its first appearance, to clear blue and eventually clear green once it is considered as *true detection*; becoming part as well of the tracking module from that moment on.

4.4 Lost Tracking

Ultimately a third option could also take place, and this is the case when a vehicle being tracked is not detected in the upcoming frame. This fact may be due to two different reasons: temporary missed or wrong detections or a real loss of track of the vehicle.

It must be noticed that miss or wrong detections are not persistent in time, i.e. very rarely a detection is missed in more than one frame, since normally these take place because of the obstruction caused by other external agent like, among others, a traffic signs or a sudden change in illumination like in the scene shown straight away.

Figure 4.6: Vehicle Loss of Track

When these are the reasons behind the sporadic loss of track of a vehicle, the predicting nature of the Kalman filter allows the system to estimate the position of these miss-detected vehicles thanks to the tracking information and in this way keep track of them. In contrast, a vehicle being miss-detected consistently along several consecutive frames must only be due to the real lost of track of the vehicle, independently to the reason, e.g., vehicle that get out of or sight because is overtaken by us, because it takes a detour on the road, is obstructed permanently by other vehicle closer to us, or simply because it gets enough far away from us to be detected by the system. In these cases, a criterion of two consecutive frames of non-detection leads the system to consider the target as lost, consequently terminating the tracking procedure of such vehicle. Of course that vehicle could later on fulfill again the criterions to be detected and therefore be tracked again, but always this time as a new detection plus tracking event.

As seen, the proposed system for preceding vehicle tracking permits to overcome the irregularities due to temporary missed or wrong detections that could take place, thus being the overall accuracy and robustness of the system improved.

It must be noticed as well that a long term observation of the position can yield a description of the trajectory of the target vehicle in an image sequence, while its size and derivative of size can yield a description of the distance and the relative to the ego-vehicle velocity, respectively; in any case interesting and useful data for potential further applications like the TTC implementation this thesis aims.

4.5 Conclusions

This chapter reveals the critical importance of processing temporal information present in video streams, especially when it comes to dynamically changing scenes as driving environments are. In first term a simple algorithm to test detection consistency in PVD stage has been presented, proving through empirical examples its helpfulness to improve the whole behavior of the system by dismissing sporadic false positive detections, which directly impacts the accuracy of this one. Furthermore, a core algorithm in the scope of this thesis for vehicle tracking based on Kalman filter has been presented in detail. Once again through examples extracted from real driving scenarios the improvement of the system behavior has been shown. More specifically, the spatial accuracy over time is improved, as it provides additional information in each frame for the detections consistent along this. It performs smoothing of the detection noise and allows reducing the rate of sporadic miss-detection providing information for the temporal association between frames by interpolating positions when these sporadic events take place. Ultimately, as pointed out in the previous section, a long term observation of the position of the detected vehicles throughout a given video frame can yield a description of their trajectory within the image sequence, while at the same time its size and derivative of size can yield a description of the distance and the relative to the ego-vehicle velocity, respectively; crucial information of high usefulness toward the next stage of this thesis where the sought Time-To-Collision system is eventually presented.

Chapter V
TTC (Time-To-Collision) System

5.1 Introduction

Time-to-Collision or Time-to-Contact (TTC) is a quantity of interest to many fields, ranging from experimental psychology to robotics [62]. Generally, TTC for two bodies in space is the ratio of the distance between them and their relative speed. In the context of video cameras, TTC can be defined as the time required for an object in the real world to reach the camera plane, assuming that the relative speed remains fixed during that time period. While TTC is the ratio of distance and speed, using a pinhole camera model, it becomes equivalent to the computation of the ratio of an object's size on an imaging plane to its time derivative. It has been suggested that analogous processing takes place in the human visual system while performing tasks involving TTC computation, such as avoiding collisions or catching a moving object. One of the chief advantages of formulating TTC in terms of object dilation over time is that TTC can thus be obtained entirely from image based data, without having to actually measure physical quantities such as distance and velocity. Consequently, the need for complicated and computationally expensive camera calibration processes, 3D reconstruction of the scene, and camera ego-motion estimation is eliminated. Due to its relative simplicity and computational efficiency, the idea of TTC estimation is ideally suited for real-time systems, where quick decisions have to be made in the face of impending collisions. For this reason, computer-vision based TTC estimation approaches can be useful for obstacle avoidance and collision detection by vehicles or individuals.

In this chapter our vision based approach for computing distance information from a windshield monocular camera system is presented. TTC principles are first review in the next section, as preamble to the driving algorithm we have implemented which is presented straight away.

5.2 Time-To-Collision Principles

Time-to-Collision can be defined as the time that an observer will take to make contact with a surface under unknown constant relative velocity [63]. TTC can be estimated as the distance between two image points divided by the rate of change in that distance. As pointed out before, the result is a form of relative distance to the object in temporal units that does not require camera calibration, 3D reconstruction or depth estimation. And indeed, part of the attraction of TTC is that the calculation relies only on image measurements and does not require knowledge of the structure of the environment or the size of shape obstacles or camera calibration. Moreover, TTC naturally encodes the dynamics of the motion of the observer. As a consequence, TTC can be used to construct motion reflexes for collision avoidance and local navigation, provided that a fast, reliable measure can be made of distance in the image, and valid also for dynamic environments.

In other words, the time to contact (TTC) is defined as the time that would elapse before the center of projection (COP) of our camera reaches the surface being viewed if the current relative motion between the camera and the surface were to continue without change. Mathematically [64], Time-to-Contact is essentially a ratio of distance to velocity, and therefore is usually expressed in terms of the speed and the distance of the considered obstacle to overcome with respect to the camera. The classical equation to compute the TTC is:

$$\tau = -\frac{Z}{dZ/_{dt}} = -\frac{1}{\frac{d}{dt}log_e(Z)} \tag{5.1}$$

Where Z is the distance between the camera and the obstacle, and dZ/dt the velocity at which the object is moving relative to the camera (which will be negative if the object is approaching the camera). However, with a monocular camera only, distance and velocity cannot be recovered without additional information. In those cases it's possible to derive (5.1) by using a characteristic size of the obstacle in the image and making several assumptions.

Consider a simple situation where the camera is approaching an elongated planar object lying perpendicular to the optical axis, with the direction of translational motion along the optical axis. If the (linear) size of the object is S and the size of its image is s, then, from the perspective projection equation, we have $(s/f) = (S/Z)$, that is, $sZ = fS$. Differentiating w.r.t time yields:

$$s\frac{dZ}{dt} + Z\frac{ds}{dt} = 0 \tag{5.2}$$

Together with (5.1), this shows that the TTC is equal to the ratio of the size s of the image of the object to the rate of change of the size, that is:

$$\tau = \frac{s}{ds/_{dt}} = \frac{1}{\frac{d}{dt}log_e(s)} \tag{5.3}$$

Where, as mentioned above, s is the size (or the scale) of the object in the image and ds/dt the time derivative of this scale. Finally it is convenient to use the inter-frame interval as the unit of time and express the TTC as a multiple of that interval. In any case, this equation seems to be more appropriate as the size of the object is something can be obtained directly in the image space. This reformulates the problem as a problem of estimating the obstacle size, as well as the rate of change of size.

Note once again that the TTC does not rely on the absolute size of the object in the image sequence, but in the relative change in scale from one frame to another. As a consequence, the TTC computation is not dependent on camera optics or the object size, only is dependent on the depth distance and the camera velocity. Furthermore, the time varying image is sampled at regular intervals and the time derivative of size is estimated using the difference between sizes of the images of the object in two frames. Therefore, high accuracy is needed in measuring the size of the image in order to obtain accurate estimates of the TTC when it is large compared to the inter-frame interval. For example, when the time to collision is 100 frames, then an image of size 100 pixels changes by only 1 pixel from frame to frame, and so, to achieve even 10% error in the TTC one would have to measure the size of the image with an accuracy of better than 1/10 of a pixel. The tolerance for measurement error becomes even smaller when the object is further away and the TTC larger, and in general, providing a fast, reliable distance measurement for TTC is a challenging task.

In our case, the vehicles being tracked and towards we are interested in calculating the Time-To-Collision are characterized by their boundary box within the image field. As so, for estimating purposes the size of the vehicles is taken as the size of their boundary box, and therefore the rate of change of theses ones along consecutive frames is what allows us in last term to implement the TTC. Is in consequence understandable that the PVD stage, responsible of the detection of the vehicles (giving a boundary box to each detection) plays a crucial role in the accuracy and successful development of the TTC system here addressed.

5.3 System Proposed

As previously stated, the tracking module implemented in Chapter IV allows a long term observation of the position and size of vehicles formally detected in the road ahead of us. This information in last terms can yield a description of the trajectory, the distance and the relative to the ego-vehicle velocity, of the target vehicle in track; data that turns out to be the key for the Time-To-Collision estimation purposes of this thesis.

More specifically, and according to the previous section where TTC principles have been presented, the problem of estimating the time to collision between the ego-vehicle and others present on the road becomes a problem of estimating the size s of the vehicle being tracked within the image frame on one hand, and the rate of change of that size with respect to consecutive frames in the other hand (see Equation 5.4). This approach is easily feasible thanks to the tracking module since, as already seen, it keeps track of the size of the vehicle under tracking in every frame and its previous one, what allows us to directly calculate the rate of change by comparing them.

$$Rate\ of\ Change\ (pixels/frame) = s_x - s_{x-1} \tag{5.4}$$

Where x is the current frame and $x-1$ the previous one. Furthermore, as aforementioned, TTC can be defined as the time that would elapse before the center of projection of the camera reaches the surface being viewed if the current relative motion between the camera and the surface were to continue without change. That is to say in other words, that technically, the potential collision our system aims to avoid would eventually happen when the vehicle under risk of collision reaches the on-vehicle mounted camera center of projection, what in image terms that the mentioned vehicle covers the whole image. In practice a threshold must be set in order to consider the risk as real, and so in our case in order to give an estimation of the Time-To-Collision, 80% of the size of the image is taken as the limit to consider a collision as imminent. Thus, since size of the image (pixels), and the rate of change of the size of the vehicle under track (pixels/frame) are known, a direct rule of three gives us our first Time-To-Collision estimation in terms of frames.

$$TTC\ (frames) = \frac{S\ (pixels)}{Rate\ of\ Change\ (pixels/frame)} \tag{5.5}$$

Once here, and taken into account that the frequency of acquisition of the camera considered is also known (10 Hz = 10 fps), the TTC estimation can be given in time units (sec).

$$\boldsymbol{TTC\ (sec) = \frac{TTC\ (frames)}{Acq.\ Frequency\ (fps)}} \tag{5.6}$$

The following figure shows a real case where the TTC estimation towards a vehicle on the road is given (yellow). The thresholds considered to compute the estimation are highlighted in red; as mentioned before, in image terms they represent the size the vehicle under track must reach in order to consider the collision as imminent. Size, and not position, despite the fact that for representation purposes the boundary limits have been placed centered within the image.

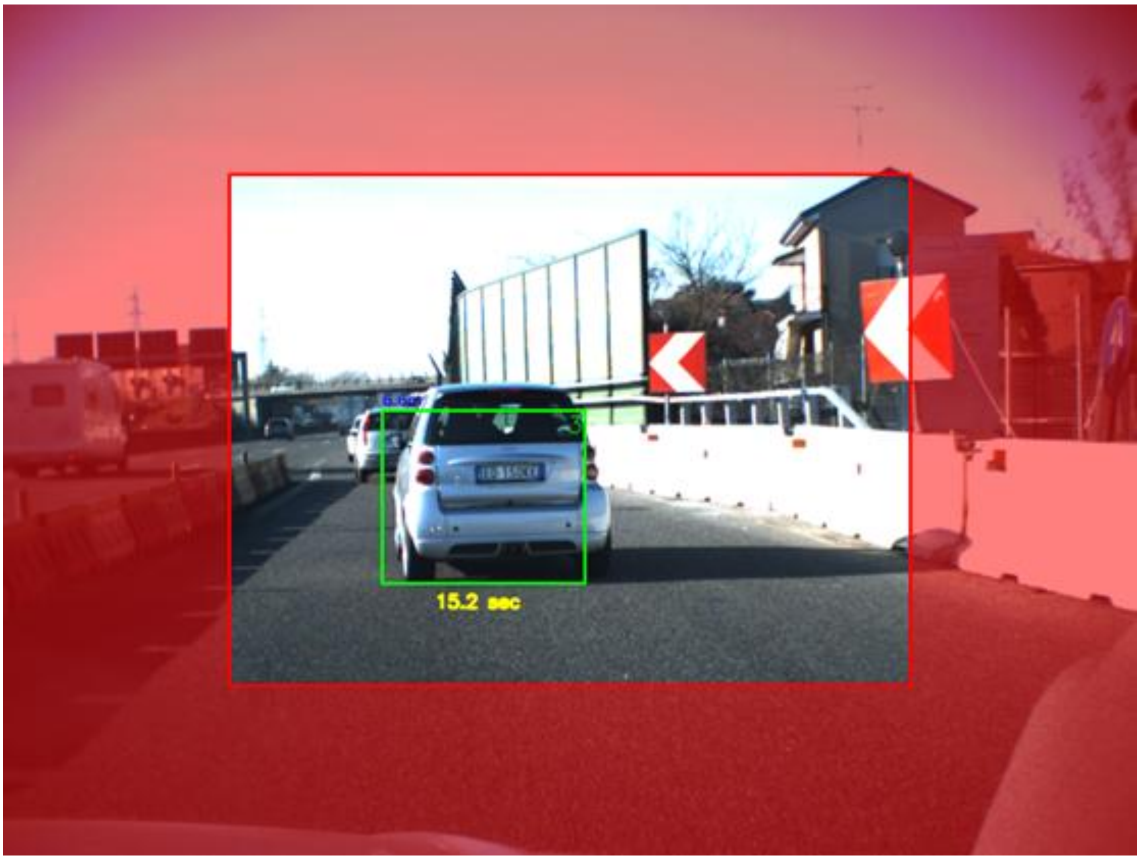


Figure 5.1: Time-To-Collision Estimation Real Case

Furthermore the relative distance from the vehicle in track to the ego-vehicle can also be estimated (blue). A precise distance calculation is practically infeasible, as the information contained in a single video frame is not sufficient to derive precise depth information. However, a good approximation for the distance of cars of typical size can be achieved like in [2]. Assuming that the average vehicle width and height are approximately 1.8m and 1.5m respectively and given the size of the detected preceding vehicle in the image (as a proportion of the vehicle's size in pixels to the total image size in pixels) we can estimate the desired distance.

Again, as aforementioned, these calculations rely directly on the accuracy in the PVD stage when detecting the vehicles in the traffic scene and giving a boundary box for them, therefore these are at anytime close approximations and hardly exact values. However, as a matter of fact, TTC systems don't strictly require 100% accuracy when giving their estimations since, as pointed out many times throughout this thesis, the final idea for these sorts of systems is to work integrated within a higher and more complete vehicle safety system (separately are valueless) which eventually aims to ensure the safety of the vehicle and other vehicles on the road, the driver, and the pedestrians or bikers. As it is presumable, this kind of systems, where human lives are involved, always consider safety ranges of operation, and here is what resides that not necessity of total accuracy when TTC estimating. Imaging a situation of high possibilities of collision between the ego-vehicle and other in a close distance, the TTC system doesn't need to know the exact distance and therefore time-to-collision towards it, because its goal is to impede the accident, and therefore according to ego-speed, the TTC estimation, and further information, the safety system would take into action, either acting directly on the braking system or direction system, with enough time in advance so that the integrity of the vehicles and human beings involved in the scene are ensured.

Figure 5.2: Time-To-Collision System

Figure 5.2 exemplifies in different driving scenarios the final behavior of the proposed Time-to-Collision operating system. This one is eventually able to perform the detection of a high number of vehicles within the scene, keep track of them and elaborate both, a distance and a TTC estimation towards them, considering only as potential subject of collision those vehicles getting closer to the ego-vehicle (stressed out with a red-x when applying), and not those getting further away. The system demonstrates this way a successful behavior and implementation.

5.4 Time Performance Evaluation

As the goal of every ADAS is to be implementable for real time processing purposes, a discussion on the processing speed of each module that build up the proposed TTC system is considered due to its high importance as closing evaluation test.

For the purposes of this last analysis, ten different clips of video for a total of 100 frames each one of them are considered. All clips of video belong to the TME Motorway Dataset and represent both daylight and sunset weather conditions, and the 100 frames considered for the evaluation test correspond to the first 100 frames of the sequence as it appears in the TME dataset. The mean processing time values required by the system are then considered within the study. The main aim is to observe the behavior of each one of the modules that build up the proposed TTC system and conclude if an eventual real-time implementation would be feasible. In this way the study aims to pinpoint possible bottlenecks of the system that could object of improvement in future researches. With this regard, it must be noticed that all parts of the system have been implemented in the well-known programming language Python, within the development environment Python(x,y), and with the use of OpenCV as main library to deal with image processing tasks. No code optimization has been extensively considered, and the tests were run on a HP Pavilion (64-bit OS and Windows 10) 1.8GHz Intel® Core™ i3-3217U processor, with 4 GB RAM. The response time performance of the system according to its modules is presented in the next table, where: VD (vehicles detected), HV (hypothesis verification), DC (3-frames Detection Consistence phase), Tracking Module and Time-To-Collision estimation.

Table 5.1: Time Performance Evaluation Results

Video Sequence	VD	HV	DC	Tracking	TTC	time (ms)	fps
daylight_12_1284	14	49	5	91	6	151	**6.62**
daylight_17_24800	8	35	5	82	6	128	**7.81**
daylight_32_5160	13	46	5	92	7	150	**6.66**
daylight_35_8950	5	28	5	78	6	117	**8.54**
daylight_42_520	4	26	4	59	5	94	**10.63**
sunset_12_22610	7	33	5	72	6	116	**8.62**
sunset_16_23650	11	42	5	75	6	128	**7.81**
sunset_16_34200	10	39	5	79	6	129	**7.75**
sunset_46_16900	12	45	5	93	6	149	**6.71**
sunset_46_24670	10	41	5	82	7	135	**7.41**

The name of the video sequence follows the pattern *weather-condition_number-of-video-sequence_first-frame-of-video-sequence*, and for each one of the ten considered for the Time Performance Test the table of Results represent (from left to right) on one hand the vehicles detected along the 100-

frames sequence, the average processing time per frame in milliseconds required by each module, and finally both, the total average processing *time/frame (ms)* required by the system and eventually the ratio fps the system performed in average.

Straight away are shown the final average result for the ten video sequences, therefore representing the average system performance.

		HV	DC	Tracking	TTC
Average Results	ms	38.4	4.9	80.3	6.1
	%	29.6	3.8	61.9	4.7
	time (ms)	129.7 ms			
	fps	7.71 fps			

Table 5.2: System Performance Results

The results obtained show how the tracking module, as expected, represents the main core of the TTC system requiring 60% of the total processing time per frame. The 3-frames phase for detection consistence, as preamble of the tracking module represents the lightest algorithm, quite at the same level that the TTC estimation, which as stated, relies directly on the tracking module, and in consequence its time processing consumption is negligible, despite its importance towards the system. On the other hand, the Hypothesis Verification stage as part of the Preceding Vehicle Detection module represents the other relevant module in resources consumption. By analyzing the previous table of results, it can be seen together with the tracking module, how their behavior are related to the number of vehicles detected in the sequence; something reasonable taking into account the more vehicles detected the more operation the PVD module must consider, as well as the more vehicles the tracking module must track. Thus, as main bottleneck of the system it is reasonable considering dedicating more resources in future work to debug and improve as much as possible the tracking module. Nevertheless, with a rate of 10 fps as goal to consider real-time implementation of the system, this shows an average processing rate of 7.71 fps, which can be considered as successful result for this first version of the proposed Time-To-Collision system. Table 5.1 shows how the lowest rates are given by the video sequences were more vehicles are detected, with a rate of 6.6 in average, while on the other side there is even one video sequence that reached a higher rate than the aforementioned 10 fps, more precisely 10.63 fps, which eventually shows the real-time implementation is feasible.

5.5 Conclusions

Chapter V has presented in detail the Time-To-Collision Estimator module that builds eventually up the TTC system presented by this thesis. Once introduced the principles by which any Time-to-Collision system is defined, the solution proposed by this thesis has been presented and shown in detail through extensive real example and images taken from the TME Motorway Dataset used along this thesis for implementation and evaluation purposes.

As seen, the TTC system proposed is eventually able to successfully give a time-to-collision estimation in seconds between the ego-vehicle and any other present on the driving scenario and being monitored by the tracking module implemented in Chapter IV. Some considerations taken into account, the system is also able to give an estimation of distance in meters to the respective vehicles being tracked. In this way, vehicles further than 50 meters from the ego vehicles are only tracked, while TTC efforts are focused on the closest vehicles to the ego-vehicle, since in any case they represent the subjects toward potential impact is higher.

In last term this chapter presents a final evaluation of the proposed TTC system in terms of Time Performance since a must condition for any ADAS system is real-time operation. To this end several test have been carried out, eventually showing an average processing time for the system of 7.71 fps; promising results for real-time implementation, letting us state that in general lines the main goals here addressed by this thesis can be considered both, at a quantitative and qualitative level, successfully achieved.

Chapter VI

Conclusions & Future Work

6.1 Summary

This thesis has set out the design of a Computer Vision based Advance Driver Assistance System, and more specifically the design of a Time-To-Collision (TTC) System able to predict the remaining time to a potential impact taking place between the ego-vehicle and others present on the road, from images acquired in real time from an on-vehicle mounted front camera. As seen, integrated in a Pre-crash System or in combination with an Emergency Breaking System (EBS) the proposed system might lead to the implementation of a real vehicle Collision-Avoidance System (CAS) that would eventually allow to avoiding potential dangerous situations of real risk of traffic accidents on the road within real driving scenarios.

To this end, a 3-modules system has been proposed, with focusing areas on Preceding Vehicles Detection (PVD), Detections Tracking, and Time-To-Collision Estimation respectively, showing all of them successful results.

The problem of on-road vehicle detection has been considered itself as 2-stages tackle. Firstly, an edge detection plus red-color segmentation approach has been considered for the Hypothesis Generation phase. Then, the PVD module mainly relies on the idea of using Gabor Filter banks to extract edge and line features at different scales and orientations. These features encode the coarse structure of a vehicle, can handle within-class variations, and are not very sensitive to global illumination. More specifically, in our approach Gabor features were extracted from sub-windows of the input image and were represented using statistical measures (i.e., mean, standard deviation, and skewness), to later on conclude the Hypothesis Verification stage through a SVMs Classifier properly trained. Time constraints didn't allow to eventually get in the HG stage results accurate enough to be considered within the whole scope of the thesis, remaining this way as a future line of work. However, the HV stage has been extensively tested and evaluated on publicly available datasets towards the optimum behavior of the system, what eventually turned out into positive outcomes in terms of accuracy when detecting preceding vehicles.

Furthermore, a tracking module cored in Kalman filtering has been proposed, which combined with the previous PVD module has shown enhancing of the spatiotemporal coherence of the detection results, as well as a reduction in the rate of false negative detections, by checking the spatial consistency these ones compared to previous frames. Moreover this module performs smoothing of the detection noise and allows reducing the rate of sporadic miss-detection providing information for the temporal association between frames by interpolating positions when these sporadic events take place. Ultimately, it carries out a long term observation of the position and size of the detected vehicles throughout a given video frame what can yield a description of their trajectory as well as distance and the relative to ego-vehicle velocity, respectively within the image sequence; information of crucial importance towards the eventual Time-To-Collision estimation the system provides.

In this context, and as proven, precise distance calculation is practically infeasible, as the information contained in a single video frame is not sufficient to derive precise depth information. However -considerations took into account- a good approximation for the distance of cars of typical size has been achieved. What is more important, without further external information than the considered in this thesis, that is to say the video sequence captured from the on-vehicle mounted camera, Time-To-Collision estimations rely directly on PVD module’s accuracy. As so, processing temporal information present in the video streams played a role of crucial importance towards this end, which eventually was successfully fulfilled and demonstrated within great variety of dynamic changing driving environments scenes.

In last term, since a must condition for any ADAS system is real-time operation, all modules individually presented in previous along this paper, as well as the proposed TTC system as a hole, were evaluated in terms of Time Performance, eventually showing an average processing time for the system of 7.71 fps; promising results for real-time implementation, letting us state that both, at a quantitative and qualitative level, in general lines the main goals here addressed by this master thesis can be considered successfully achieved.

6.2 Future Lines of Work

Regarding the fact that the goals from the beginning addressed by this master thesis can be considered satisfied some suggestion of possible improvements and direct lines of future work can be extracted from the research proposed.

This first one, already mentioned along this paper, involves the completion of the Hypothesis Generation stage of the Preceding Vehicles Detection (PVD) module presented in Chapter III, which afore stated due to time constraints of this project couldn’t be completely fulfilled. Nevertheless, some early enquires have been done towards this stage, and a solution has been proposed, so with this regard it’s all about dedicating the necessary resources to be implemented and fused as part of the whole system.

Furthermore, as presumable, all system proposed are susceptible of improvement, both in terms of accuracy and speed, taking into account a must condition for them is a real time operation. In this line, any enhancement toward any of the modules proposed as well as to the algorithms and programming optimization represents an interesting line of future work to keep focusing on.

Last but not least, the ultimate goal that is common in all ADAS implementations is to produce systems that are robust in all possible conditions. In the face of the enormous variety of possible traffic situations, it is, however, not to be expected that the approach described here can lead to perfect detection accuracy under all circumstances. The proposed system has been indeed evaluated under two different weather conditions, according to the main dataset used in the scope of this thesis which is divided in two sub-sets depending on lighting condition, named “daylight” (although with objects casting shadows on the road) and “sunset” (facing the sun or at dusk). Therefore, further studies might be devoted to the issue of how the performance of the whole driver assistance system could operate more robustly under different adverse and more demanding conditions such us cases of rainfall, snow, or night driving.

All this considered, this thesis has presented and evaluated a Time-To-Collision vision based system that could potentially be included in a complete ADAS, and more precisely integrated in a Pre-crash Vehicle Safety System ore used in combination with an Emergency Breaking System (EBS) and/or systems alike leading towards the eventual implementation of a real vehicle Collision-Avoidance System (CAS) that would eventually allow to avoiding potential dangerous situations of real risk of traffic accidents on the road within real driving scenarios involving human lives integrity. As so, the system proposed by this thesis could be considered as a stand-alone solution within the field of ADAS technology, however studying its behavior in combination with other vision based systems like those presented in Chapter 1.4 (e.g., traffic light and sign recognition systems (TLR and TSR), road and lane detection (RD and LD), pedestrian detection (PD), night vision (NV), driver drowsiness detection (DDD), and blind spot detection (BSD)), as well as fusing information from different sources such as Radar or Lidar systems in order to achieve the best possible results, represents one of the most potential and promising points of future research.

Figure 6.1: Front – Back (upper row), Left – Right (bottom row)

Finally, the usage of information coming only from a monocular on-board camera, scenario considered within this thesis, represents a challenge itself at the same time that adds extra value to the proposed systems as highlighted along this paper. In this line, future work may consider a migration to stereo systems by putting into scene a second camera and all what involves when approaching solutions. Or, being faithful to the single camera scenario, other possibilities could take place. This is the case of VALEO Vision Systems, where currently so-called bird-eye cameras with a wide angle of vision are considered within their systems.

Some images taken with this kind of cameras are shown in Figure 6.1. As it can be seen, the most prominent characteristic of this sort of cameras with regard to conventional ones resides in the field of view these ones offer, which goes up to 180º, in any case much more than the one present in the conventional ones or those considered within the scope of this thesis. According to this, an extra challenge that the use of these cameras involves in order to take advantage of the possibilities they provide resides in compensating the distortion they introduce. Thus, considering this or different more sophisticated equipments in the image/video acquisition represents future potential researching lines to keep focusing on in order to continue the work proposed within this paper.

Bibliography

[1] M. Zhao, "Advanced Driver Assistant System: Threats, Requirements, and Security Solutions", Security & Privacy Research, Intel Labs, 2015. [http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/advanced-driver-assistant-system-paper.pdf] Access: July 2016.

[2] G. Siogkas, "Advanced Driver Assistance Systems with Computer Vision Methods", Doctoral Dissertation, Electrical and Computer Engineering Department, University of Patras, June 2013.

[3] C. Mathers, D. M. Fat, and J. T. Boerma, "The Global Burden of Disease: 2004 Update", World Health Organization, 2008. [http://www.who.int/healthinfo/global_burden_disease/GBD_report_2004update_full.pdf] Access: July 2016.

[4] "Global Status Report On Road Safety: Time for Action", Geneva, World Health Organization, 2009. [http://www.who.int/violence_injury_prevention/road_safety_status/2009/en/] Access: July 2016.

[5] E. Kopits, and M. Cropper, "Traffic fatalities and economic growth: Accident Analysis & Prevention", vol. 37, no. 1, pp. 169-178, 2005.

[6] C. J. L. Murray, and A. D. Lopez, "The global burden of disease and injury series: a comprehensive assessment of mortality and disability from diseases, injuries, and risk factors in 1990 and projected to 2020", Harvard University, Boston, MA, USA, and World Health Organization, Geneva, Switzerland, 1996.

[7] "National Motor Vehicle Crash Causation Survey: Report to Congress", U.S. Department of Transportation, National Highway Traffic Safety Administration, July 2008.

[8] European New Car Assessment Programme (Euro NCAP), for safer cars. [http://www.euroncap.com/en] Access: July 2016.

[9] Euro NCAP Advanced Rewards. [http://www.euroncap.com/en/ratings-rewards/euro-ncap-advanced-rewards/] Access: July 2016.

[10] Tesla Motors, Premium Electric Vehicles, Model S & Model X, United States. [https://www.teslamotors.com/models],[https://www.teslamotors.com/modelx] Access: July 2016.

[11] American Honda Motor Company, Honda Safety Systems. [http://automobiles.honda.com/safety/] Access: July 2016.

[12] Subaru Starlink™, Confidence in Motion, EyeSight Driver Assist Technology. [http://www.subaru.com/engineering/eyesight.html] Access: July 2016.

[13] Mobileye: Our Vision, Your Safety™, Vehicle Detection Applications. [http://www.mobileye.com/technology/applications/vehicle-detection/] Access: July 2016.

[14] Robert Bosch GmbH, myDriveAssist Mobile Application. [https://play.google.com/store/apps/details?id=com.bosch.mydriveassist&hl=en] Access: July 2016.

[15] Audi Online Traffic Light Information: intelligent drive in smart cities. [http://www.audi.com/com/brand/en/vorsprung_durch_technik/content/2014/06/traffic-light-information-online.html] Access: July 2016.

[16] J. Briggs, "How In-dash Night-vision Systems Work", Tech. Automotive Gadgets, HowStuffWorks, InfoSpace LLC. [http://electronics.howstuffworks.com/gadgets/automotive/in-dash-night-vision-system.htm] Access: July 2016.

[17] "The connectivity of things: Connected cars on the roll", Injection Moulding Asia. [http://www.injectionmouldingasia.com/feb2016/news6.html] Access: July 2016.

[18] "Saab Tech: Keep Drivers Awake and Focused", SAAB 9-3 SportCombi, Saab Planet. [http://www.saabplanet.com/saab-tech-keep-drivers-awake-and-focused/] Access: July 2016.

[19] R. O. Duda, and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures", Communications of the ACM, vol. 15, no. 1, pp. 11–15, 1972.

[20] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection: An Evaluation of the State of the Art", IEEE Transactions on Pattern Analysis and Machine Intelligence.

[21] R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Ten Years of Pedestrian Detection, What Have We Learned?", Max Planck Institut for Informatics, Saarbrücken, Germany.

[22] Volvo Car Group's international car range, "Volvo unveils innovative safety system: Pedestrian Detection with Full Auto Brake debuts on the all-new S60", Volvo Car USA. [https://www.media.volvocars.com/us/en-us/media/pressreleases/31773] Access: July 2016.

[23] Mobileye: Our Vision, Your Safety™, Pedestrian Detection Applications. [http://www.mobileye.com/technology/applications/pedestrian-detection/] Access: July 2016.

[24] S. Sivaraman, and M. M. Trivedi, "Looking at vehicles on the road: a survey of vision-based vehicle detection, tracking, and behavior analysis", IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 4, pp. 1773–1795, December 2013.

[25] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: a review", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 5, pp. 694–711, May 2006.

[26] R. A. Hadi, G. Sulong, and L. E. George, "Vehicle detection and tracking techniques: a concise review", Signal & Image Processing: An International Journal (SIPIJ), vol.5, no.1, February 2014.

[27] R. van der Horst, and J. Hogema, "Time-To-Collision and collision avoidance systems", Proceedings of the 6th ICTCT workshop Salzburg, TNO Institute for Human Factors, The Netherlands, 1994.

[28] K. D. Kusano, and H. Gabler, "Method for estimating time to collision at braking in real-world, lead vehicle stopped rear-end crashes for use in pre-crash system design", SAE International, Virginia Tech, December 2011.

[29] M. Betke, E. Haritaoglu, and L. S. Davis, "Real-time multiple vehicle detection and tracking from a moving vehicle", Machine Vision and Applications, vol. 12, no. 2, pp. 69–83, 2000.

[30] D. Ponsa, J. Serrat, and A. M. Lopez, "On-board image-based vehicle detection and tracking", Computer Vision Center and Computer Science Department, University of Barcelona, Transactions of the Institute of Measurement and Control 33, 7 (2011) pp. 783–805.

[31] TME Motorway Dataset. [http://cmp.felk.cvut.cz/data/motorway/] Access: July 2016.

[32] C. Caraffi, T. Vojír, J. Trefný, and J. Matas, "A system for real-time detection and tracking of vehicles from a single car-mounted camera", TME Motorway Dataset, ITS Conference, pp. 975—982, September 2012.

[33] Caltech Computational Vision: Archive, Cars 1999 & Cars 2001, March 17th, 2005. [http://www.vision.caltech.edu/html-files/archive.html] Access: July 2016.

[34] S. Santini, and R. Jain, "Similarity Matching", Visual Computing Lab University of California, San Diego.

[35] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Measurement to Structural Similarity", IEEE Transactions on Image Processing, vol. 13, no. 1, Januray 2004.

[36] Adaptative Vision, Machine Vision Guide, Template Matching. [http://docs.adaptive-vision.com/current/studio/machine_vision_guide/TemplateMatching.html] Access: July 2016.

[37] OpenCV Documentation, Image Processing, Template Matching. [http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html] Access: July 2016.

[38] Andrew Ng, "Machine Learning", Data Science, Stanford Univ., Coursera. [https://en.coursera.org/learn/machine-learning] Access: July 2016.

[39] "An introduction to machine learning", Scikit Learn. [http://scikit-learn.org/stable/tutorial/basic/tutorial.html] Access: July 2016.

[40] Z. Sun, G. Bebis, and R. Miller, "Monocular Pre-crash Vehicle Detection: Features and Classifiers", Computer Vision Lab. Department of Computer Science, University of Nevada, Reno, Technology Department, Ford Motor Company, Dearborn, MI.

[41] "Support Vector Machines (SVM)", Scikit Learn. [http://scikit-learn.org/stable/modules/svm.html#svm-kernels] Access: July 2016.

[42] G. Bishop, and G. Welch, "An introduction to the kalman filter", Department of Computer Science University of North Carolina, Chapel Hill, SIGGRAPH, Course, vol. 8, 2001.

[43] R. E. Kalman, "A new approach to linear filtering and prediction problems", Research Institute for Advanced Study, Baltimore, Transactions of the ASME, Journal of basic Engineering, vol. 82, no. 1, pp. 35–45, 1960.

[44] D. Zhang, A. Wong, M. Indrawan, and G. Lu, "Content-based Image Retrieval Using Gabor Texture Features", Gippsland School of Computing and Information Technology, Monash University, Australia.

[45] P. Moreno, A. Bernardino, and J. Santos-Victor, "Gabor Parameter Selection for Local Feature Detection", IBPRIA, 2nd Iberian Conference on Pattern Recognition and Image Analysis, Estoril, Portugal, June 2005.

[46] A. Nurhadiyatna, A. L. Latifah, and D. Fryantoni, "Gabor Filtering for Feature Extraction in Real Time Vehicle Classification System", Research Center for Informatics, Indonesian Institute of Sciences, Cibinong & Bandung.

[47] "Gabor filter: Applications of 2-D Gabor filters in image processing", Wikipedia, the free encyclopedia. [https://en.wikipedia.org/wiki/Gabor_filter] Access: July 2016.

[48] S. S. Beauchemin and J. L. Barron, "The computation of optical flow", Department of Computer Science, University of Western Ontario, Canada, ACM Computing Surveys (CSUR), vol. 27, no. 3, pp. 433–466, 1995.

[49] "Optical Flow", Wikipedia, the free encyclopedia. [https://en.wikipedia.org/wiki/Optical_flow] Access: July 2016.

[50] F. Raudies, "Optic flow", Boston University, Boston, MA, USA, Scholarpedia, the peer-reviewed open-access encyclopedia, 2013. [http://www.scholarpedia.org/article/Optic_flow] Access: July 2016.

[51] "Visual Obstacle Avoidance using Optical Flow on the Android-powered HTC EVO for Safe Navigation of the iRobot Create", Teyvonia Thomas, Aspiring Roboticist. [http://teyvoniathomas.com/index.php/projects/55-opticalflow.html] Access: July 2016.

[52] N. Khairdoost, S. A. Monadjemi, and K. Jamshidi, "Front and Rear Vehicle Detection using Hypothesis Generation and Verification", Department of Computer Engineering, Faculty of Engineering, University of Isfahan, Signal & Image Processing: An International Journal (SIPIJ), vol.4, no.4, Aug 2013.

[53] Q. B. Truong, and B. R. Lee, "Vehicle Detection using Hypothesis Generation and Verification", Department of Mechanical and Automotive Engineering, School of Mechnical and Automotive Engineering, University of Ulsan, Korea.

[54] E. Skodras, G. Siogkas, E. Dermatas, and N. Fakotakis, "Rear lights vehicle detection for collision avoidance", 19th International Conference on Systems, Signals and Image Processing (IWSSIP), 2012, pp. 134–137.

[55] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using gabor filters and support vector machines", 14th International Conference on Digital Signal Processing, 2002, vol. 2, pp. 1019–1022.

[56] "Standardized moment", Wikipedia, the free encyclopedia. [https://en.wikipedia.org/wiki/Standardized_moment] Access: July 2016.

[57] C.-C. R. Wang, and J.-J. J. Lien, "Automatic Vehicle Detection Using Local Features-A Statistical Approach", IEEE Transactions on Intelligent Transportation Systems, vol. 9, no. 1, pp. 83 –96, Mar 2008.

[58] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning", IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. II–264, 2003.

[59] M. Sigari, M. R. Pourshahabi, M. Soryani, and M. Fathy, "A Review on Driver Face Monitoring Systems for Fatigue and Distraction Detection", International Journal of Advanced Science and Technology, vol.64, pp.73-100, 2014.

[60] S. Siang, and T. Bräunl, "A reliability point and kalman filter-based vehicle tracking technique", International Conference on Intelligent Systems (ICIS 2012), vol. 1, Malaysia, Planetary Scientific Research Centre, pp. 134-138, 2012.

[61] J. Guo, J. Wang, X. Guo, C. Yu, and X. Sun, "Preceding Vehicle Detection and Tracking Adaptive to Illumination Variation in Night Traffic Scenes Based on Relevance Analysis", Article, Sensors, ISSN 1424-8220, pp. 15325-15347, 2014.

[62] S. Pundlik, E. Peli, and G. Luo, "Time to collision and collision risk estimation from local scale and motion", Proceedings of the 7th international conference on Advances in visual computing, vol. part I, pp. 728-737, Schepens Eye Research Institute, Harvard Medical School, Boston, MA, 2011.

[63] G. Alenyà, A. Nègre, and J. L. Crowley, '’Time To Contact for Obstacle Avoidance’’, Institut de Robòtica I Informàtica Industrial, CSIC-UPC, Barcelona, INRIA Grenoble Rhône-Alpes Research Centre, France.

[64] B. K. P. Horn, Y. Fang, and I. Masaki, '’ Time to Contact Relative to a Planar Surface’’, Department of Electrical Engineering and Computer Science, CSAIL, MIT, Cambridge, MA, IEEE Intelligent Vehicles Symposium, pp. 68-74, June 2007.

[UPCT] Technical University of Cartagena, Spain. [http://www.upct.es/]

[UOPA] University of Patras, Creece. [http://www.upatras.gr/en]

[E&CE] Department of Electrical & Computer Engineering, University of Patras, Greece [http://www.ece.upatras.gr/en/]

[VALEO] Valeo Vision Systems, Ireland. [http://www.valeovision.com/]