

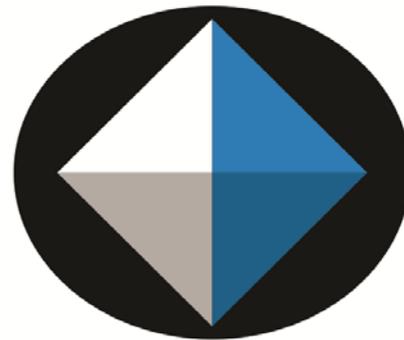


UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Instituto Universitario de Microelectrónica Aplicada



TECNOBIT S.L.U.
Departamento de Desarrollo Tecnológico

Máster en Tecnologías de Telecomunicación



Trabajo Fin de Máster

Actualización y mejora del Algoritmo de Detección de Vehículos
incorporados al Sistema Optrónico CENTINELA

Autor: Carlos Tarajano Beracoechea

Tutor: Gustavo I. Marrero Callicó

Fecha: Julio 2012

	<p>www.iuma.ulpgc.es Campus Universitario de Tafira 35017 Las Palmas de Gran Canaria</p>	<p>www.tecnobit.es Santa Leonor 65 28037 Madrid</p>	
---	---	--	---



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Instituto Universitario de Microelectrónica Aplicada



TECNOBIT S.L.U.
Departamento de Desarrollo Tecnológico

Máster en Tecnologías de Telecomunicación



Trabajo Fin de Máster

Actualización y mejora del Algoritmo de Detección de Vehículos
incorporados al Sistema Optrónico CENTINELA

HOJA DE FIRMAS

Alumno: Carlos Tarajano Beracoechea

Fdo.:

Tutor: Gustavo I. Marrero Callicó

Fdo.:

Fecha: Julio 2012

	www.iuma.ulpgc.es Campus Universitario de Tafira 35017 Las Palmas de Gran Canaria	www.tecnobit.es Santa Leonor 65 28037 Madrid	
---	---	--	---

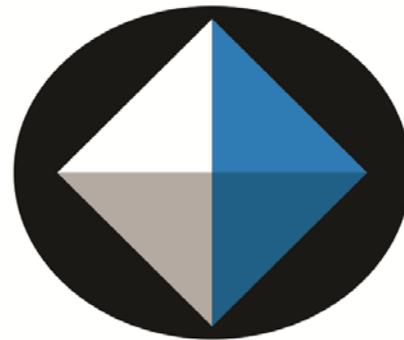


UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
 Instituto Universitario de Microelectrónica Aplicada



TECNOBIT S.L.U.
 Departamento de Desarrollo Tecnológico

Máster en Tecnologías de Telecomunicación



Trabajo Fin de Máster

Actualización y mejora del Algoritmo de Detección de Vehículos
 incorporados al Sistema Optrónico CENTINELA

HOJA DE EVALUACIÓN

Calificación:

Presidente:

Fdo.:

Secretario:

Fdo.:

Vocal:

Fdo.:

Fecha: Septiembre 2012

	<p>www.iuma.ulpgc.es Campus Universitario de Tafira 35017 Las Palmas de Gran Canaria</p>	<p>www.tecnobit.es Santa Leonor 65 28037 Madrid</p>	
---	--	---	---

ÍNDICE

1	Introducción y objetivos.....	15
1.1	Introducción.....	15
1.2	Objetivos.....	16
2	Estado del Arte	19
2.1	Detección de vehículos	19
2.2	Sensores activos Vs sensores pasivos	19
2.3	La detección de vehículos mediante visión artificial.....	20
2.3.1	Basados en conocimiento previo	21
2.3.2	Basados en sistemas de visión estéreo.....	23
2.3.3	Basados en movimiento.....	24
2.4	Conclusiones	28
3	Sistema de detección de vehículos desarrollado.....	29
3.1	Análisis del algoritmo actual frente al estado del arte	29
3.2	Elección de la Plataforma de desarrollo	30
3.3	Diseño e implementación	31
3.3.1	Detección dinámica	31
3.3.2	Detección Estática.....	37
3.3.3	Combinación de estrategias utilizada.....	38
3.4	Descripción del algoritmo	39
3.5	Conclusiones	41
4	Pruebas y resultados obtenidos	43



4.1	Secuencias de prueba	43
4.2	Pruebas realizadas	50
4.3	Resultados obtenidos	53
4.3.1	Compensación de movimiento global.....	53
4.3.2	Ecuación.....	53
4.3.3	Filtro post-ecuación	54
4.3.4	Detección de contornos	54
4.3.5	Operaciones Morfológicas	55
4.4	Comparación con otras estrategias	55
5	Conclusiones y Líneas Futuras	59
5.1	Conclusiones	59
5.2	Líneas Futuras.....	59
6	Bibliografía.....	61



Índice de Figuras

Figura 1: Sistema Centinela.....	15
Figura 2: Sistema de seguimiento mono-objetivo STT	16
Figura 3: Algoritmo Detección de Vehículos en el sistema CENTINELA	29
Figura 4: Algoritmo CCODA para detección de vehículos estáticos	29
Figura 5: Fotogramas originales y magnitudes de los vectores de movimiento.....	32
Figura 6: Lukas-Kanade con MB=16.....	34
Figura 7: LUT de valor de cambio.....	35
Figura 8: Proceso de ecualización	35
Figura 9: Detección de bordes	36
Figura 10: Operaciones Morfológicas aplicadas.....	37
Figura 11: Detección de vehículos por contornos	38
Figura 12: Resultado de combinar ambas técnicas.....	39
Figura 13: Algoritmo de detección de vehículos mejorado	40
Figura 14: Resultado del algoritmo final	41
Figura 15: Porcentaje de aciertos según iteración	53
Figura 16: Máscara de ecualización.....	54
Figura 17: Ejemplos de contornos que no se han cerrado	55
Figura 18: Comparativa entre algoritmos (parte 1)	56
Figura 19: Comparativa entre algoritmos (parte2)	56
Figura 20: Diferencias entre los algoritmos de detección.....	57



Índice de Tablas

Tabla 1: Ejecución del Flujo Óptico.....	33
Tabla 2: Tipos de cerrado	36
Tabla 3: Resultados de las iteraciones.....	51

Glosario

CCD	Charge-Couple Device
CCODA	Closed Contour Object Detection Algorithm
DSI	División de Sistemas Integrados
FLIR	Forward Looking Infrared
GPS	Global Positioning System
IPM	Inverse Perspective Mapping
IR	Infrared – Infrarrojo
IUMA	Instituto Universitario de Microelectrónica Aplicada
KDE	Kernel Density Estimation
LASER	Light Amplification by Stimulated Emission of Radiation
LIDAR / LADAR	Light Detection and Ranging
LOC	Local Orientation Coding
LRF	Laser Range Finder
LUT	Look Up Table
MTT	Multiple Target Tracking
OF	Optical Flow - Flujo Óptico
STT	Single Target Tracking
ULPGC	Universidad de Las Palmas de Gran Canaria



1 INTRODUCCIÓN Y OBJETIVOS

Este Trabajo de Fin de Máster ha sido realizado a través de una colaboración entre el Departamento de Desarrollo Tecnológico de la empresa TECNOBIT S.L.U. y la División de Diseño de Sistemas Integrados (DSI) del Instituto Universitario de Microelectrónica Aplicada (IUMA).

1.1 INTRODUCCIÓN

Tecnobit ha desarrollado un Sistema Avanzado de Vigilancia Móvil llamado CENTINELA [1]. Tal y como se muestra en la Figura 1, CENTINELA está formado por un sistema de sensores electro-ópticos (a) montado sobre una plataforma de dos ejes (b). El sistema está controlado a través de una consola (c) y un panel de control con botones y joystick (d).



Figura 1: Sistema Centinela

El sistema electro-óptico incluye una cámara de infrarrojo de barrido frontal (FLIR) de 3 a 5 μm de alta resolución, una cámara CCD y un medidor de distancias láser (LRF).

El sistema CENTINELA se encuentra actualmente en uso por el cuerpo de Infantes de Marina de la Armada Española y por la Guardia Civil, principalmente en misiones de vigilancia y control. Cuenta con una aplicación de cartografía y GPS que le permite etiquetar objetivos en un mapa táctico y dispone de un sistema de seguimiento básico.

Un sistema de seguimiento es aquél que es capaz de detectar, seguir (e incluso determinar la posición) e identificar un objetivo [2]. Los sistemas de seguimiento pueden emplear un amplio abanico de sensores, siendo los más comunes los de radar, infrarrojos, láser, sonar y equipos de radio [3]. Los sistemas de seguimiento se pueden clasificar en dos categorías:

1. Los capaces de realizar el seguimiento a un solo objetivo (Single Tracking Target - STT)
2. Los capaces de realizar el seguimiento a varios objetivos simultáneamente (Multiple Tracking Target - MTT).

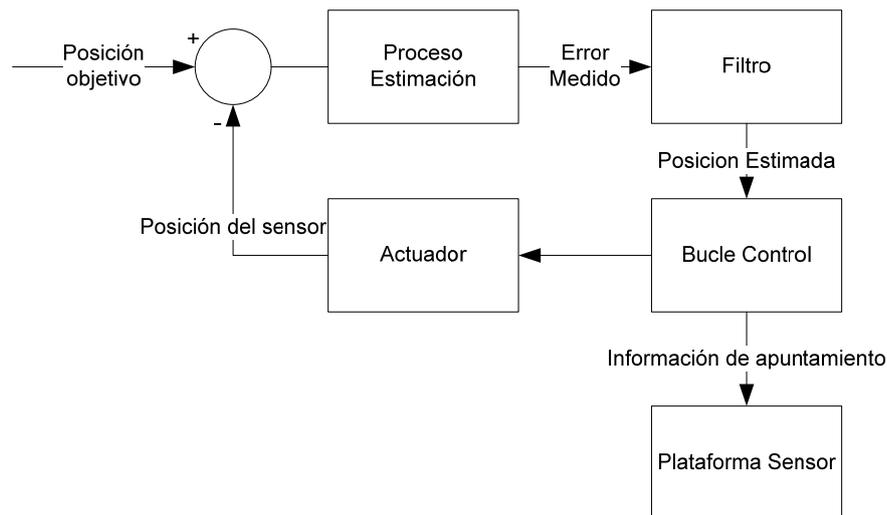


Figura 2: Sistema de seguimiento mono-objetivo STT

Los sistemas STT trabajan normalmente en un bucle cerrado sobre el objetivo de forma que se mantiene el objetivo siempre dentro de la línea de visión del sensor. En la Figura 2 se observa el bucle cerrado típico y tradicional de los STT. En primer lugar se encuentra el sensor en estado de observación hasta que se detecta el objetivo. En ese momento se procede a estimar la posición del objetivo en el próximo paso. Esta se pasa por el filtro que elimina espurios y el bucle de control, en base a la información que obtiene de la plataforma sobre la posición de apuntamiento del sensor, le envía las nuevas coordenadas de apuntamiento. Una vez finalizado el bucle, se verifica dónde se encuentra el objetivo y se empieza de nuevo [4].

Una de las partes más críticas del bucle de seguimiento es aquella que determinar la posición inicial y el estado del objetivo: la etapa de detección. Una de las etapas de detección más estudiada es la detección de objetos artificiales, especialmente la relacionada con detección de vehículos tanto para sistemas de ayuda, vigilancia, seguridad, militares o médicos.

1.2 OBJETIVOS

Tras varias pruebas de uso del sistema de seguimiento CENTINELA, surge la necesidad de actualizar el detector de vehículos por uno más avanzado que consiga un ratio de aciertos superior. De esta necesidad surge el presente Trabajo de Fin de Máster con el objetivo de desarrollar un algoritmo de detección de vehículos que cumpla con los siguientes requisitos:

- Las imágenes de entrada son en formato luminancia.
- Ha de poder utilizarse tanto para el sensor FLIR como para el sensor CCD.
- La cámara puede encontrarse en movimiento en el momento de solicitar la detección.
- Se han de etiquetar los vehículos detectados, otra etapa del proceso se encargará de elegir el objeto al cual realizar el seguimiento.
- El algoritmo debe detectar todo tipo de vehículos: (aéreos, marítimos y terrestres).
- Las situaciones en las que se puede solicitar la detección son de todo tipo: vehículos en entornos urbanos, rurales, cielos y mar bajo cualquier condición climatológica, etc.

Para cubrir esta necesidad se van a estudiar los diferentes algoritmos de detección de vehículos disponibles en la literatura técnica y científica. De este estudio se extraerán las conclusiones oportunas y se diseñará un nuevo algoritmo con el fin de cumplir con los requisitos antes expuestos. Una vez implementado el algoritmo se probará la bondad de este con un conjunto de imágenes de test de la biblioteca de TecnoBit seleccionadas con el fin de cubrir la mayor cantidad de casos posibles.

2 ESTADO DEL ARTE

2.1 DETECCIÓN DE VEHÍCULOS

Los sistemas de detección de vehículos basados en sistemas de visión artificial se han estado estudiando intensivamente durante más de 20 años. El auge de este campo está motivado principalmente para el desarrollo de sistemas de ayuda a la conducción y al conductor, con el fin de reducir el riesgo de pérdida de vidas humanas [5].

Esto hace que los principales desarrollos estén enfocados en sistemas fijos de vigilancia (para la contabilización, alarma por accidente, etc.) o en sistemas a bordo de los propios vehículos con el fin último de obtener vehículos totalmente autónomos [6].

El incremento en la potencia de procesamiento y el desarrollo de nuevas técnicas de procesado de imagen para visión artificial de los últimos años ha provocado un cambio sustancial en este campo. En la actualidad existen numerosos esfuerzos por parte de gobiernos, fabricantes de automóviles, institutos de investigación, y otras empresas, trabajando en algunos casos de forma conjunta o con soluciones totalmente divergentes.

En el campo de los vehículos autónomos, Europa ha liderado los esfuerzos con el proyecto PROMETHEUS durante muchos años, englobando tanto a fabricantes de vehículos e institutos de investigación de 19 países, llegando a fabricar varios vehículos prototipos y sistemas demostradores.

Actualmente Google ha conseguido desarrollar el primer vehículo autónomo con licencia para circular en el estado de Nevada. Esto ha supuesto un punto de inflexión muy importante en cuanto al comienzo del desarrollo de la legislación necesaria para hacer llegar a las carreteras los vehículos autónomos.

2.2 SENSORES ACTIVOS VS SENSORES PASIVOS

El método más eficaz para detectar vehículos es el uso de sensores activos [7] como pueden ser los radares, los dispositivos láser o los sistemas acústicos. Todos ellos se distinguen por emitir una señal y medir el tiempo en que tarda en volver dicha señal. Un radar [8] emite un pulso electromagnético; un LIDAR (*Light Detection and Ranging*) [9] un pulso óptico y un sistema acústico un pulso sonoro [10]. La principal ventaja de estos sistemas es que no requieren una gran capacidad de cómputo, ya que para medir el tiempo en que tarda en retornar el pulso no se requiere realizar gran cantidad de operaciones matemáticas.

Un sistema radar es capaz de localizar un vehículo a más de 150 metros, sin importar las condiciones climáticas. Un LIDAR es mucho más barato de fabricar aunque no funciona tan bien en condiciones adversas (lluvia o niebla), en cambio es mucho más preciso con buenas condiciones.

Los sistemas activos ofrecen muy buenos resultados pero presentan un gran inconveniente. En una situación de tráfico normal, en la que pueden haber decenas (por no decir cientos) de vehículos circulando por la misma vía, evitar las interferencias de unos sistemas con otros puede ser tremendamente conflictivo. Esto hace que el uso de sistemas activos haya sido desechado para su uso en vehículos autónomos.

Con respecto a su uso en otros sistemas de detección de vehículos, normalmente tampoco se utilizan ya que, por ejemplo, en sistemas de uso militar, un sensor activo al emitir, puede delatar la posición al



enemigo. En sistemas de vigilancia tampoco son muy usados ya que pueden ser fácilmente interferidos.

Por último, los sensores activos suelen ser mucho más costosos que los sensores pasivos. El coste de las cámaras ha disminuido drásticamente con lo que su uso se ha popularizado enormemente. Además, su bajo coste permite desarrollar soluciones que utilicen más de un sensor con lo que su cobertura y resolución llega a límites que con otro tipo de sensores no se pueden alcanzar: por ejemplo, con un radar, realizar la identificación de un vehículo es imposible, cosa que es bastante más asequible usando una cámara de vídeo.

Por estas razones, el desarrollo tecnológico de sensores activos no ha visto la inversión de la que han disfrutado los sensores pasivos. Los sensores pasivos son aquellos que sólo procesan la señal recibida y no emiten ningún tipo de energía. Dentro de esta categoría entran los sensores ópticos y los sensores infrarrojos [7].

2.3 LA DETECCIÓN DE VEHÍCULOS MEDIANTE VISIÓN ARTIFICIAL

El coste computacional dedicado al procesamiento de imágenes es muy alto, por tanto, para sistemas en tiempo real, se buscan soluciones intermedias, que encuentren algún tipo de reducción de la carga computacional del proceso al tiempo que permitan obtener buenos resultados.

Las técnicas de detección de vehículos tienen dos sub-conjuntos de técnicas complementarias: las de Generación de Hipótesis, que estiman en que zona de la imagen puede haber un vehículo; y las de Verificación de la Hipótesis, en las que se comprueba que el objeto encontrado es realmente un vehículo.

Los algoritmos de Generación de Hipótesis se pueden clasificar en los siguientes:

- **Basados en conocimiento previo.**
- **Basados en sistemas de visión estéreo.**
- **Basados en movimiento.**

Existen fundamentalmente dos clases de métodos para realizar la verificación:

- **Basados en plantillas:** estos métodos usan patrones predefinidos según la clase de vehículo a detectar y realizan una correlación entre el objeto encontrado y el patrón predefinido.
- **Basados en apariencia:** este tipo de métodos usa un conjunto de imágenes de entrenamiento con las que se pretende capturar la variabilidad en la apariencia de los vehículos a detectar.

La verificación, por lo tanto, es ejecutar un conjunto de pruebas sobre el resultado obtenido por la propia detección del vehículo para corroborar que el objetivo detectado es correcto. Este tipo de verificaciones son útiles siempre y cuando se sepa con certeza qué tipo de vehículo concreto es el que se quiere detectar. Por ello, resulta idóneo disponer de un método de detección robusto que no necesite verificación, o que al menos el número de falsos aciertos sea muy bajo.

Los sistemas en que la incertidumbre sobre el tipo de objetivo es alta (militares y seguridad principalmente) son complicados de implementar por varios aspectos: hay casos en los que no se tienen fotografías de los vehículos enemigos, o por otro lado, el número de tipos de vehículos a



detectar es muy elevado (un sistema de vigilancia que pueda intervenir en situaciones con vehículos aéreos, terrestres y navales al mismo tiempo).

Por ello, este trabajo se centra en analizar los diferentes métodos de Generación de Hipótesis.

2.3.1 BASADOS EN CONOCIMIENTO PREVIO

Este tipo de métodos se basan en emplear un conocimiento previo para suponer la situación del vehículo en la imagen. Existen a su vez diferentes tipos:

2.3.1.1 Simetría

Una de las principales características de los objetos realizados por el hombre es su simetría. Por ello, la búsqueda de objetos simétricos es una de las técnicas más usadas para la detección de vehículos [11]. La imagen de un vehículo tomada desde una posición frontal o posterior es, por lo general, simétrica. Esta es una pista usada en varios estudios sobre detección de movimiento [12] [13].

2.3.1.2 Color

Aunque pocos son los sistemas que usan la información del color para la detección de vehículos, la información del color suele ser usada para la detección de obstáculos, carreteras y vías. Varios estudios recientes han empleado diferentes técnicas para poder extraer o segmentar el vehículo del fondo de la imagen [14]. Crisman et al. [15] usaron dos cámaras para extender el rango dinámico de forma que una cámara tenía el iris muy abierto y la otra casi cerrado. De esta forma, crearon un espacio de color en seis dimensiones y pudieron distinguir si un píxel dado formaba parte de la carretera o no.

2.3.1.3 Sombras

Mori y Charkai [16] fueron los primeros en utilizar la información de las sombras como un indicio de la presencia de vehículos. Detectaron que la intensidad de la zona que se encuentra justo por debajo de un vehículo es ligeramente más oscura que otras zonas del asfalto. Después de algunos intentos [17], se llegó a la conclusión de que el principal problema de este método es la determinación de los umbrales, ya que estos dependían de la iluminación general de la imagen, y esta de las condiciones climáticas. Aunque existen trabajos que proponen métodos para determinar dichos umbrales [18], ninguno de ellos es concluyente.

2.3.1.4 Esquinas

Varios estudios explotan el hecho de que los vehículos, en general tienen una forma rectangular con cuatro esquinas. Bertozzi [19] propone un método en el que busca todas las esquinas de una imagen y posteriormente asocia las esquinas entre sí con el objetivo de encontrar las cuatro esquinas del vehículo.

2.3.1.5 Detección de bordes

Normalmente los vehículos suelen contener estructuras que destacan sobre el fondo y ofrecen grandes contrastes en sus bordes, especialmente en las esquinas. Además los vehículos suelen contener estructuras verticales y horizontales como las ventanillas, los paragolpes, las puertas, etc., lo que puede llevar a realizar una buena detección mediante un estudio morfológico de los bordes generados por algún algoritmo de detección de bordes.

La detección de contornos [20] se realiza mediante búsqueda de discontinuidades en la imagen, es decir, de píxeles en los cuales el nivel de gris varíe bruscamente respecto al de sus vecinos. Las dos

técnicas más conocidas para la detección de puntos de contorno son la búsqueda de puntos de elevado gradiente (utilizando la primera derivada) o de cruces por cero de la laplaciana (segunda derivada).

2.3.1.5.1 Métodos de primera derivada

Una vez aplicada a la imagen un operador derivativo, como los de Sobel [10], Roberts [11], etc., se realiza una clasificación de los píxeles de la imagen según si son o no pertenecientes a los contornos. Para realizar esta clasificación, existen varias técnicas, que podemos clasificar básicamente en:

- **Aplicación de un umbral a la imagen de gradiente.** De esta forma se seleccionan únicamente los puntos de gradiente superior a un valor determinado. Aunque esta técnica, al tratarse de una simple umbralización, es muy sencilla, tiene el inconveniente de que no busca los máximos del gradiente, sino sólo los puntos de gradiente elevado. Por lo tanto, se eliminarán contornos más débiles, que sin embargo pueden ser muy importantes.
- **Operador de Canny.** Es una de las técnicas más utilizadas en detección de contornos y se basa en la idea de la búsqueda de los máximos del gradiente. En primer lugar, se realiza un filtrado de la imagen original mediante la aplicación de un filtro gaussiano, con lo cual se consigue minimizar el efecto del ruido sobre el proceso de detección. A continuación, comienza la fase de detección de máximos de gradiente. En el caso unidimensional, basta con realizar una búsqueda de los máximos absolutos, pues en ese caso sería imposible detectar contornos enteros, que corresponden a crestas y no a picos aislados. La búsqueda de los máximos ha de realizarse por lo tanto en la dirección perpendicular al contorno, que no es más que la propia dirección del gradiente. La técnica consiste en realizar la detección de los máximos del gradiente en cada dirección

2.3.1.5.2 Métodos de segunda derivada

Este método se basa en detectar los cruces por cero de la derivada segunda de la imagen original. Los puntos correspondientes a bordes de la imagen original corresponden a puntos en los que la laplaciana cambia de signo, esto es, a cruces por cero de la laplaciana. La detección de estos cruces por cero es relativamente sencilla, y da una estimación muy exacta de la posición de los contornos, además se trata de una detección muy fiable. Los contornos obtenidos son curvas de nivel de la imagen (las correspondientes al nivel cero), y por tanto siempre son cerrados o desaparecen por el borde de la imagen, por lo que nunca aparecen segmentos aislados. Esta propiedad es extremadamente importante cuando lo que se desea es segmentar la imagen. Las desventajas principales que presenta la laplaciana son dos. La primera desventaja surge a la hora de detectar bordes, ya que son muy sensibles al ruido, por lo que se suele recurrir a técnicas combinadas de umbralización/detección. La segunda desventaja surge a la hora de detectar los cruces por cero pues, en general, los píxeles en la laplaciana no poseen exactamente el valor cero, así que los cruces por cero se detectan cuando dos píxeles contiguos poseen valores de signo diferente. En este caso, el contorno se encuentra realmente en algún lugar entre los dos píxeles, pero no en ninguno de ellos; este inconveniente se resuelve generalmente suponiendo que el contorno se encuentra en aquel de los píxeles cuyo valor de laplaciana está más cercano a cero, con lo cual se toma la opción menos mala.

Estas técnicas de detección de bordes han sido ampliamente utilizadas a la hora de realizar detección de vehículos: Matthews [21] usó una detección de bordes vertical para encontrar la parte izquierda y derecha de un vehículo. Un método similar pretende encontrar la parte superior e inferior del vehículo. Goerick [22] propuso un método llamado *Local Orientation Coding* (LOC) para extraer información de los bordes. El método consistía en cadenas de código binario representando la



dirección mediante la variación de niveles de gris en los píxeles de la vecindad. Existen múltiples ejemplos de métodos que usan la detección de bordes de una u otra forma para determinar la posición del vehículo dentro de la imagen [23] [24] [25].

2.3.1.6 Texturas

La presencia de vehículos en una imagen causa cambios en la intensidad local. Debido a la homogeneidad de los vehículos, los cambios de intensidad siguen ciertos patrones de texturas [26]. Esta información sobre las texturas puede servir para centrar y disminuir el área de búsqueda. La entropía ha sido usada como el primer indicador para la detección de texturas. Se divide la imagen en regiones y sólo se procesan aquellas con alto nivel entrópico.

Se han usado otros métodos de segmentación como el uso de matrices co-ocurrentes [27]. De ellas se extraen varios estadísticos, siendo los más críticos los de energía, contraste, entropía y correlación. El método es más preciso que usando sólo la entropía, pero por contra requiere de un cómputo mucho más intensivo.

2.3.1.7 Luces de los vehículos

La mayoría de los métodos anteriores no obtienen buenos resultados en ambientes de baja iluminación o nocturnos. En este tipo de situaciones, una característica muy clara de un vehículo es el empleo de luces de iluminación de la carretera. Cucchiara y Piccardi [28] han usado análisis morfológico para detectar pares de luces.

2.3.2 BASADOS EN SISTEMAS DE VISIÓN ESTÉREO

Existen dos clases de métodos para detectar vehículos que usen sistemas de visión estereoscópica: uno usa mapas de disparidad, mientras que el otro usa una transformación de anti-perspectiva [29]:

2.3.2.1 Mapas de disparidad

A la diferencia entre los correspondientes píxeles de la imagen derecha e izquierda se le llama disparidad. Las disparidades de todos los puntos de la imagen forman el mapa de disparidad. Si los parámetros de la plataforma estéreo son conocidos, el mapa de disparidad se puede convertir en un mapa 3D de la escena. Obtener el mapa de disparidad es muy intensivo computacionalmente ya que conlleva resolver el problema de correspondencia para cada píxel.

Una vez obtenido el mapa de disparidad se puede obtener el histograma de disparidad. Si hay algo que sobresalga (como un obstáculo o un vehículo), esto se refleja en el histograma mediante un pico en el intervalo del histograma correspondiente.

Franke y Kutzbach [30] argumentaron que resolver el problema de correspondencia es muy costoso computacionalmente y además los mapas de disparidad no son lo suficientemente densos. Por ello propusieron un extractor de características locales para resolver el problema de correspondencia más rápidamente. De esta forma, la búsqueda de píxeles correspondientes se reduce a un simple test que buscan si dos píxeles forman parte de la misma categoría o no. El número de picos en el histograma de disparidad vuelve a determinar el número de vehículos detectados.

2.3.2.2 Mapeado de perspectiva inversa (IPM)

El término “mapeado de la perspectiva inversa” no es totalmente correcto [29], ya que matemáticamente no es posible. Mejor dicho, denota una inversión bajo la restricción adicional de que los puntos mapeados invertidos residen sobre un plano horizontal.



El método consiste en predecir lo que se debería ver en la imagen izquierda a través de la imagen derecha. Usando IPM, se transforman todos los puntos de la imagen izquierda en coordenadas, se re proyectan otra vez sobre la imagen derecha y se compara [31]. De esta forma, se pueden detectar contornos de objetos sobre el plano del fondo.

2.3.3 BASADOS EN MOVIMIENTO

Los métodos anteriormente resumidos, se basan fundamentalmente en detectar características espaciales. La otra forma de detección es mediante el uso de técnicas que detecten características del movimiento de los vehículos. De los métodos se puede extraer tres categorías:

1. Basados en extracción del fondo.
2. Diferencia temporal.
3. Flujo óptico.

Se detallan a continuación cada uno de estos métodos.

2.3.3.1 Basados en extracción del fondo

La extracción del fondo es una técnica muy popular, especialmente en situaciones de fondo estático. Esta técnica pretende detectar qué zona constituye el fondo y qué objetos no pertenecen a él. Para ello, obtiene una imagen de fondo a la que denomina como referencia y señala como objeto a cualquier píxel que no pertenezca al fondo.

La obtención de la imagen de referencia o “fondo modelo” es la clave para el correcto funcionamiento del método. El fondo modelo ha de ser la representación de la escena sin objetos en movimiento y con actualizaciones frecuentes para adaptarse a las condiciones cambiantes de iluminación.

Las variaciones más recientes proponen estimar el fondo modelo a partir de un número reducido de fotogramas. Las diferencias de rendimiento y funcionamiento entre una técnica y otra son debidas a diferentes aproximaciones en la estimación del fondo modelo. Estas técnicas van desde las orientadas a rendimiento en velocidad o memoria, a las más sofisticadas en las que la precisión prima por encima de todo. Las podemos dividir en las siguientes categorías:

1. **Media gaussiana en ejecución** [32]: en este método se calcula el fondo modelo a partir de calcular la Función de Densidad de Probabilidad (PDF) gaussiana para cada píxel para los últimos n fotogramas. Para comprobar si un píxel es fondo o no se establece una función umbral.
2. **Filtro de mediana temporal**: varios estudios argumentan que se pueden usar otras formas de media con mejores resultados. Lo y Velastin proponen usar un filtro de mediana temporal [33].
3. **Mezcla de gaussianas** [34] [35]: con el transcurso del tiempo, puede ocurrir un cambio en la geometría de la escena. Los métodos revisados pueden adaptarse más o menos rápidamente. De todas formas, cambios en los objetos del fondo pueden no ser permanentes y pueden cambiar más rápido que la tasa de actualización del fondo (por ejemplo, el movimiento de las ramas de un árbol). Una posible solución a este problema es que se permita varios valores en el fondo modelo para un mismo píxel dado: una hoja de un árbol, la rama, o la casa tras el árbol. Para ello, se describe la probabilidad de observar el valor de un cierto píxel mediante una mezcla de gaussianas.



4. **Estimación de la densidad del kernel:** Elgammal [36] propone un modelo de distribución del fondo basado en una Estimación de la Densidad del Kernel (KDE) en el *buffer* de los últimos n valores de fondo.
5. **Aproximación de la densidad del kernel secuencial** [37]: el empleo de técnicas de desplazamiento de la media del vector por medio del gradiente ascendente se ha empezado a utilizar en problemas de reconocimiento de patrones. Se han propuesto diversas optimizaciones para mitigar este problema, con el inconveniente de un alto coste computacional.
6. **Variaciones de imágenes coocurrentes:** Seki [38] va más allá de las medias cronológicas y estima qué bloques vecinos de píxeles sufren cambios similares con el paso del tiempo.
7. **Auto-fondos (*eigenbackgrounds*):** esta aproximación propuesta por Oliver [39] está basada en una descomposición en auto-vectores aplicándola a toda la imagen. Este dominio espacial tan extendido permite explorar correlaciones espaciales y evitar efectos no deseados por el fragmentado de los bloques.

Existen otras técnicas que permiten discriminar objetos del fondo en movimiento (como nubes, olas del mar, etc.) que usan técnicas más avanzadas y que no pueden ser clasificadas dentro de ninguna de estas categorías [40] [41] [42].

2.3.3.2 Diferencia Temporal

Este método usa las diferencias entre píxeles de dos o tres fotogramas consecutivos para extraer las zonas en las que se producen movimientos. La diferencia temporal se adapta muy bien a entornos muy dinámicos, pero ofrece, en cambio, un rendimiento pobre para extraer todos los píxeles relevantes. Por ejemplo, para un objeto grande con textura homogénea, los píxeles que se detectarían como en movimiento son los que están en los bordes. Lipton [43] propone un método para detectar objetos en movimiento en secuencias de vídeo en tiempo real hallando la diferencia absoluta para cada píxel y usando una función umbral para determinar los cambios.

2.3.3.3 Flujo óptico

El flujo óptico juega un papel importante en la estimación y descripción del movimiento, por lo cual es comúnmente utilizado en tareas de detección, segmentación y seguimiento de objetos móviles en una escena a partir de un conjunto de imágenes. En este trabajo se seleccionó una estrategia de flujo óptico, para estimar el movimiento relativo de la imagen, ya que puede ser implementada con bajo coste computacional.

2.3.3.3.1 Definición de flujo óptico

El flujo óptico puede ser definido como el movimiento aparente de los patrones de intensidad en una imagen. La palabra aparente indica que el movimiento espacial de los objetos (campo de movimiento) puede coincidir o no con el flujo estimado. No obstante, en situaciones en las cuales el movimiento de los objetos implica un movimiento de sus patrones de intensidad en el plano imagen, el flujo óptico puede ser directamente relacionado con el movimiento de los objetos en la escena.

La mayoría de las técnicas existentes para la estimación del flujo óptico se puede clasificar en 4 categorías:

1. Las basadas en gradientes espacio-temporales.
2. Las basadas en comparación de regiones.

3. Las basadas en fase.
4. Las basadas en energía.

En todas las estrategias de estimación de flujo óptico se parte de la hipótesis de que los niveles de gris permanecen constantes ante movimientos espaciales en un tiempo dado. Dicha hipótesis da lugar a la ecuación general de flujo óptico (1), donde $I(x, y, t)$ corresponde a la intensidad en niveles de gris del píxel (x, y) de la imagen I en el instante t

$$I(x, y, t) = I(x + dx, y + dy, t + t) \quad (1)$$

Expandiendo (1) en series de Taylor sobre el punto (x, y, t) se obtiene que:

$$I(x, y, t) = I(x, y, t) + dx \frac{\partial I}{\partial x} + dy \frac{\partial I}{\partial y} + dt \frac{\partial I}{\partial t} + \varepsilon \quad (2)$$

donde ε contiene la información de las derivadas de orden superior. Si se asume ε despreciable, la ecuación de flujo óptico puede reescribirse como:

$$I_x u + I_y v + I_t = 0 \quad (3)$$

donde (u, v) , con $u = dx/dt$ y $v = dy/dt$, corresponde al vector de flujo óptico e, I_x e I_y son las derivadas parciales horizontal y vertical de la imagen, respectivamente. Para cada píxel (x, y) de la imagen, en el tiempo t , puede plantearse la ecuación (3), sin embargo no existe una única solución para esta ecuación. Diferentes restricciones pueden emplearse para estimar el flujo óptico en la imagen. Horn y Shunck [44], restringen el flujo óptico en la imagen a variar suavemente, por lo que la ecuación (3) es minimizada junto a un término de regularización que penaliza los cambios abruptos del flujo. Lucas y Kanade (LK) [45] proponen un método alternativo que se describe a continuación el cual puede ser implementado de forma más eficiente que el propuesto por Horn. Las técnicas propuestas se basan en gradientes espacio- temporales pues minimizan la ecuación (3).

2.3.3.3.2 Método de Lucas y Kanade

Este método asume que el flujo óptico es constante sobre una región. Sea R una región de la imagen y (u, v) su vector de flujo óptico asociado, entonces (3) se cumple para cada píxel de la región, es decir:

$$I_x(p_i)u + I_y(p_i)v + I_t(p_i) = 0, \forall p_i \in \mathbb{R} \quad (4)$$

siendo p_i el conjunto de píxeles de la imagen. Organizando el conjunto de ecuaciones en forma matricial:

$$\underbrace{\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_N) & I_y(p_N) \end{bmatrix}}_A \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_d = - \underbrace{\begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ \vdots \\ I_t(p_N) \end{bmatrix}}_b \quad (5)$$

donde la matriz 'A' contiene las derivadas espaciales de la imagen, el vector 'd' corresponde al vector de flujo óptico (u, v) y el vector 'b' contiene las derivadas temporales de la imagen. Pre-multiplicando (5) por la transpuesta de A para asegurar que sea cuadrada y por lo tanto invertible, se tiene:



$$A^T A d = A^T b \quad (6)$$

donde el vector de flujo óptico es encontrado como:

$$d = (A^T A)^{-1} A^T b \quad (7)$$

El cálculo del flujo óptico implica la inversión de la matriz:

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_y I_x & \sum I_y I_y \end{bmatrix} \quad (8)$$

por la cual la solución existe si la matriz $A^T A$ es invertible y bien condicionada. Shi y Tomassi [46] definen las propiedades que debe cumplir una región para que el flujo óptico sea estimado apropiadamente utilizando la técnica de LK. Sean λ_1 y λ_2 los valores propios de la matriz $A^T A$ para cierta región R de la imagen, entonces se debe cumplir que:

1. $\min(\lambda_1, \lambda_2) > \lambda_{min} \in \mathbb{R}^+$, lo cual garantiza que $A^T A$ es invertible y la región no es ruidosa.
2. $\lambda_1/\lambda_2 < \tau$, donde τ es un umbral prefijado que garantiza que $A^T A$ está bien condicionada y no presenta bordes en una sola dirección.

Bajo estas 2 condiciones el flujo óptico puede ser apropiadamente estimado. Puede comprobarse que la solución propuesta por LK corresponde también a la solución de la técnica de comparación de regiones definida como:

$$\min_{(u,v)} \sum_{(x,y) \in \mathbb{R}} [I(x, y, t + 1) - I(x + u, y + v, t)]^2 \quad (9)$$

En la práctica existen ciertos factores que pueden inducir errores en la estimación. Entre ellos se encuentran la variación temporal de los niveles de gris sobre la región, grandes desplazamientos de la región entre las imágenes consecutivas e incoherencia de movimiento. El primero es independiente de la técnica de LK pero los otros 2 factores pueden ser controlados seleccionando un tamaño apropiado para la región R . Una región pequeña en comparación al tamaño del objeto garantiza una consistencia en el movimiento de las intensidades de gris, sin embargo si el objeto se desplaza rápidamente, éste puede salir de la región, lo que produce un error en la estimación del flujo óptico. Por tal motivo existe un compromiso en el tamaño de la región R , el cual puede ser manejado con una implementación piramidal que estime secuencialmente el flujo en diferentes escalas. En principio el flujo es estimado sobre imágenes en una escala baja para permitir grandes desplazamientos, posteriormente la escala se reduce para realizar una estimación más precisa y evitar inconsistencias de movimiento. El tamaño de la región se mantiene fijo sobre todas las escalas.

2.3.3.3 Uso del Flujo Óptico en detección de vehículos

Existen diferentes métodos que se aprovechan del Flujo Óptico para detectar vehículos. La mayoría de ellos calcula las derivadas espaciales y temporales de los perfiles. Meyer [47] calcula el campo de vectores de desplazamiento para inicializar un algoritmo de seguimiento basado en contornos. Fleet [48] compara el rendimiento de varios métodos para recuperar el Flujo Óptico. Giachetti [49] desarrolla algunos de los diferentes métodos tanto de primer como de segundo orden para usarlos con una secuencia de un vehículo en una carretera, llegando a ser capaz de re-mapear los puntos correspondientes a dos fotogramas consecutivos. Kruger [50] estima el Flujo Óptico mediante una

aproximación local a las derivadas espacio-temporales, conociendo el movimiento propio de la cámara llega a ser capaz de detectar objetos estáticos y en movimiento.

Todas estas aproximaciones usan una estimación del Flujo Óptico “densa” en la que se calcula el vector de movimiento para cada píxel con un alto coste computacional, haciéndola impracticable para aplicaciones de tiempo real. Weng [51] y Smith [52] calculan el flujo óptico para solo ciertos puntos significativos de la imagen, con lo que consiguen una aceleración en el computo, reduciendo también la cantidad de ruido presente.

El problema fundamental de los métodos que usan Flujo Óptico es su elevado coste computacional. Se han presentado alternativas y simplificaciones que permiten acelerar el proceso con una pérdida aceptable de precisión [53] [54].

2.4 CONCLUSIONES

La detección de vehículos es un tema de gran actualidad, especialmente desde los últimos impulsos por parte de los grandes fabricantes de vehículos y gobiernos en aras de mejorar la seguridad en carretera. Otra área en la que se están realizando numerosos esfuerzos es en el de la seguridad, vigilancia y defensa, como mecanismo de alarma o de guiado de armas inteligentes.

Según la aplicación del sistema que incorpore la detección de vehículos, será conveniente usar la técnica más apropiada en función de los requerimientos de dicha aplicación. Mientras que las técnicas de extracción de fondos son muy empleadas en cámaras fijas (funciones de vigilancia), las técnicas de detección por estudio del flujo óptico o por estudio de las características de la imagen son más útiles para vehículos en movimiento.

Para este trabajo se va a intentar combinar lo mejor de cada una de ellas y buscar una solución que pueda cumplir con los requisitos enunciados en el primer capítulo.



3 SISTEMA DE DETECCIÓN DE VEHÍCULOS DESARROLLADO

Para el diseño de un nuevo algoritmo de detección de vehículos para el sistema CENTINELA se han llevado a cabo las siguientes tareas:

1. Análisis del algoritmo actual frente al estado del arte.
2. Elección de la plataforma de desarrollo.
3. Diseño e Implementación de un nuevo algoritmo de detección de vehículos.
4. Validación del Algoritmo.
5. Documentación.

3.1 ANÁLISIS DEL ALGORITMO ACTUAL FRENTE AL ESTADO DEL ARTE

En primer lugar, se analizó el algoritmo de detección de vehículos ya existente en el sistema CENTINELA.

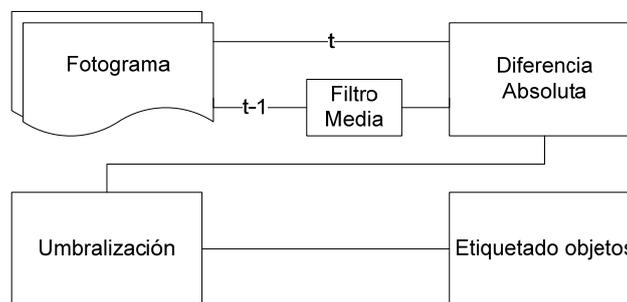


Figura 3: Algoritmo Detección de Vehículos en el sistema CENTINELA

En la Figura 3 se muestra el diagrama de flujo de la primera versión del algoritmo de detección de vehículos de CENTINELA. Esta primera versión se basa en un algoritmo de detección por medio de la comparación de dos fotogramas de tiempo $t-1$ y t . Al primero de ellos se le realiza un filtrado de mediana y se calcula la diferencia absoluta del fotograma resultante con el segundo. A la imagen resultante se le aplica un proceso de umbralización con lo que se obtiene una imagen binaria con los objetos. Por último se etiquetan los objetos detectados y se pasa a la siguiente etapa.

A este algoritmo se le detectó una debilidad en cuanto a que no era capaz de detectar vehículos que estuvieran quietos o con movimiento sub-píxel. Por ello, se diseñó un nuevo algoritmo de detección de objetos estáticos (*Closed Contour Object Detection Algorithm - CCODA*) [55] con el fin de subsanar este problema.

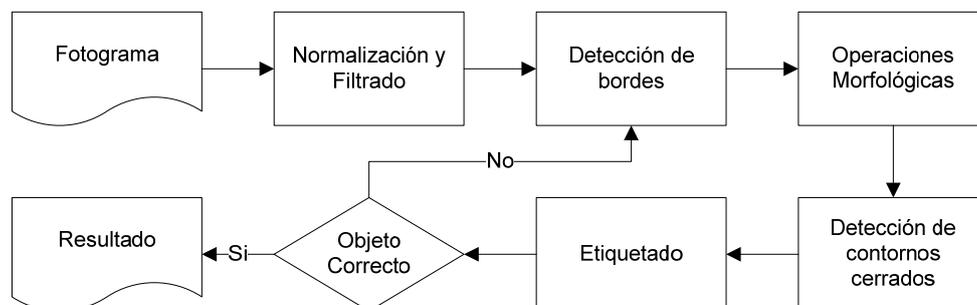


Figura 4: Algoritmo CCODA para detección de vehículos estáticos

En la Figura 4 se muestra el algoritmo CCODA [55] que procesa la imagen en primer lugar con un filtro paso bajo con variación de coeficientes según la distancia al centro. A la imagen filtrada se le aplica una detección de bordes según Canny [56]. Posteriormente se realiza una dilatación binaria con el fin de terminar de cerrar los bordes y se detectan aquellos bordes cerrados sobre sí mismos. Por último se etiquetan y se verifica que los objetos detectados sean correctos.

Por tanto, tenemos dos estrategias de detección. Una que se puede etiquetar como dinámica, capaz de detectar vehículos en movimiento, y una segunda estrategia estática. En primer lugar se ejecuta la detección dinámica y en caso de que no detecte ningún vehículo, se ejecuta la estática.

De este estudio se obtuvo como resultado el trabajo presentado en la asignatura del presente Máster "Sistemas Reconfigurables para Multimedia". El trabajo consistió en una evaluación del algoritmo, la prueba de algunas modificaciones y el estudio de los resultados obtenidos. Como conclusión del trabajo se obtuvo que para la fase dinámica, el pre-procesado de la secuencia de entrada mejoraba los resultados de la detección, mientras que para la fase estática, el detector de bordes de Canny es el que mejor resultados obtiene.

Una vez finalizado el estudio del algoritmo existente en CENTINELA, se procede a comparar el algoritmo desarrollado con el estado del arte actual. Este análisis surge del capítulo 2 de este Trabajo de Fin de Máster y se alcanzan las siguientes conclusiones:

- El análisis morfológico/color de la imagen es costoso computacionalmente y puede no llegar a ser posible ejecutarlo en tiempo real. Además el análisis del color es inviable para imágenes IR.
- La detección de bordes ofrece una buena solución de compromiso para detectar artefactos que tengan un buen contraste con el fondo de la imagen.
- No se pueden usar métodos de detección por comparación con plantillas o librerías ya que el vehículo a detectar puede no estar incluido en éstas.
- Los sistemas de visión estéreo ofrecen muy buenos resultados pero no son aplicables a este problema en concreto.
- Las técnicas de extracción de fondo son claras candidatas pero tienen una debilidad, y es que necesitan cierta estabilidad previa antes de comenzar la detección.
- Las técnicas de diferencia temporal sólo detectan movimiento en los bordes del objeto (si este es uniforme) y siempre que el objeto tenga suficiente contraste con el fondo.
- La determinación del flujo óptico de la imagen es costosa computacionalmente.

Partiendo del algoritmo antes explicado y de lo aprendido tras el análisis, se pretende optimizar el algoritmo de detección de vehículos mediante la inclusión de nuevas estrategias que mejoren el rendimiento del mismo.

3.2 ELECCIÓN DE LA PLATAFORMA DE DESARROLLO

Como plataforma para poder implementar cualquiera de las soluciones y probar su rendimiento se ha decidido trabajar con MATLAB (*Matrix Laboratory*) [57]. MATLAB es un entorno de programación para el desarrollo de algoritmos, el análisis de datos, la visualización y el cálculo numérico. Con MATLAB, se pueden solucionar problemas de diseño de cálculo técnico más rápidamente que con lenguajes de programación tradicionales tales como C, C++ o Fortran.

MATLAB es un lenguaje de programación técnico-científico que básicamente trabaja con variables vectoriales y matriciales. Es fácil de utilizar debido a que contiene varias cajas de herramientas con

funciones incorporadas (*Toolbox* de procesamiento de señales, herramientas de C empotrado, matemática aritmética de punto fijo, etc.).

Para el desarrollo de este Trabajo de Fin de Máster se ha utilizado una herramienta especialmente útil dentro de MATLAB: la *Image Processing Toolbox*, que proporciona un conjunto completo de algoritmos de referencia-estándar y herramientas gráficas para el procesamiento, el análisis y la visualización de imágenes y el desarrollo de algoritmos. Esta herramienta incorpora funciones para realizar mejoras de imágenes, enfoque de imágenes borrosas, detección de funciones, reducción de ruidos, segmentación de imágenes, transformaciones geométricas y registro de imágenes, entre otras muchas utilidades.

La elección de estas herramientas es por su alta versatilidad y su corto plazo de desarrollo. Si para el trabajo se hubiesen elegido otras herramientas (tipo C++, Java, Fortran, etc.), los tiempos de desarrollo e implementación hubiesen sido mucho más extensos, con lo que no se hubiese podido implementar diferentes algoritmos y el trabajo no hubiese sido tan completo a nivel de investigación del diseño. Como principal inconveniente, la utilización de MATLAB como herramienta de prueba implica que se aumentan los tiempos de cómputo, ya que con otras herramientas de bajo nivel se hubiese podido tener una visión más realista de lo que tardaría en ejecutarse el algoritmo en la plataforma final.

Por esta razón, el uso de MATLAB impide que se pueda verificar la condición de tiempo real. Esta decisión se ha tomado teniendo presente que el algoritmo que finalmente se diseñe, pueda ser optimizado para trabajar en plataformas de alto rendimiento computacional en donde sí se pueda cumplir con este requisito.

3.3 DISEÑO E IMPLEMENTACIÓN

De entre las diferentes técnicas estudiadas, dos son las que a priori parecen más prometedoras: la extracción del fondo y el uso del Flujo Óptico. La primera de las dos técnicas no requiere un uso intensivo de la CPU, mientras que la segunda sí supone un alto coste computacional. En cambio, la extracción no parece apta para todos los entornos, cosa que no sucede con el Flujo Óptico.

Analizando más en profundidad la técnica de la extracción del fondo de la imagen se realizan búsquedas para encontrar algún artículo que explique si es posible eliminar fondos con movimiento global de la cámara de forma continua o con situaciones cambiantes. No encontrando nada, se decide desechar esta técnica.

3.3.1 DETECCIÓN DINÁMICA

Como se ha explicado anteriormente, el algoritmo de detección consta de dos etapas, en primer lugar, se estudiará la parte dinámica del mismo.

3.3.1.1 Flujo Óptico

Finalmente, se decide implementar como solución a estudiar el Flujo Óptico. Como se explico anteriormente, el Flujo Óptico no es una técnica en sí misma, sino que es el flujo calculado entre dos fotogramas y que, usando las técnicas actuales, lo que existe son estimadores del mismo.

Se implementan cuatro de los estimadores más frecuentemente utilizados:

1. Horn-Schunck [44]
2. Lucas-Kanade [45]
3. Brox [58]
4. Sand [59]

Y se obtienen para cada uno de ellos dos matrices con las dos coordenadas de los vectores de movimiento para cada pixel de la imagen U y V . Las representamos en forma de magnitud tal y como se muestra en la Figura 5.

En la Figura 5 se muestra en (a) y (b) los fotogramas originales, en (c) el resultado para Horn Schunk;

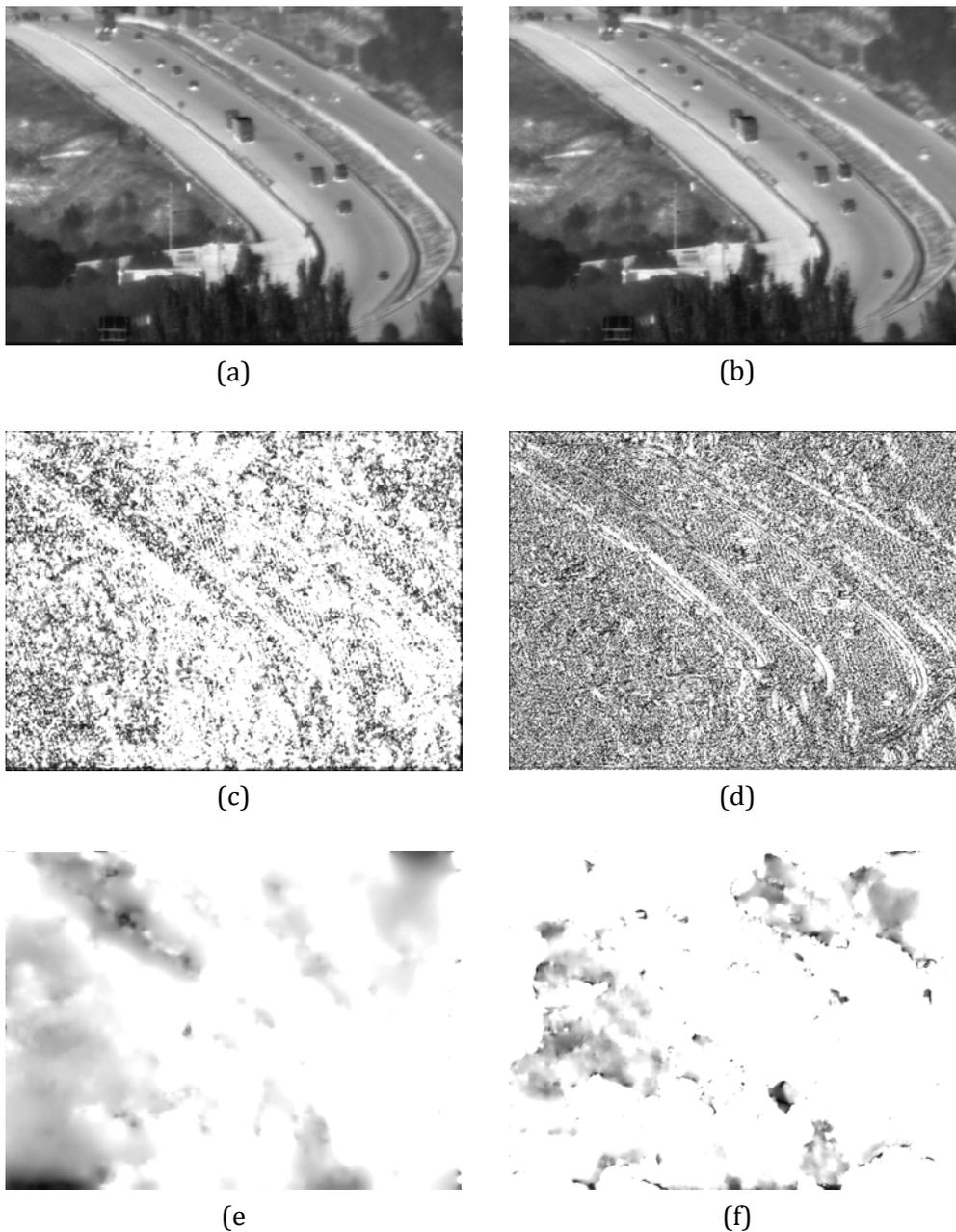


Figura 5: Fotogramas originales y magnitudes de los vectores de movimiento

en (d) para Lucas-Kanade; en (e) para Brox y en (f) para Sand. Los tiempos de ejecución se muestran en la tabla 1:

Técnica	H-S	L-K	Brox	Sand
Tiempo	2,213 s.	52,088 s.	622,521 s.	642,56 s.

Tabla 1: Ejecución del Flujo Óptico

Los dos primeros algoritmos no son iterativos, mientras que los dos últimos sí. De ahí la diferencia en cuanto al ruido presente en la imagen y en cuanto a los tiempos de ejecución.

Se ha probado con diferentes secuencias y variando los parámetros de configuración de cada algoritmo y se ha observado que el que mejor relación entre la calidad de los vectores y la velocidad de computo es el de Lucas-Kanade. A partir de este momento, se usará este algoritmo como el estimador de Flujo Óptico.

3.3.1.2 Información contenida en los vectores de movimiento

Una vez calculado el Flujo Óptico para varias secuencias, hay que extraer la información de las mismas. El primer paso es extraer el vector de movimiento global y de esta forma tratar de agrupar aquellas zonas de la imagen en las que los píxeles tienen un vector de movimiento diferente.

Tras realizar varios intentos, usando varias herramientas estadísticas (cálculo de la media, mediana, desviación estándar y PCA) se evidencia que los vectores tienen mucho ruido y que no se obtiene ninguna información relevante.

En el trabajo de curso de la asignatura de Sistemas Reconfigurables para Multimedia, se descubrió que mediante un pre-tratamiento de la secuencia de imágenes, se podía mejorar mucho la calidad de los resultados para la detección dinámica.

Después de probar varios tipos de filtrado (mediana, gaussiano, de disco y de alta frecuencia) se apreció que el pre-procesado perjudicaba el rendimiento del estimador de Flujo Óptico. Lo que sí es de destacar es el hecho de que usar una compensación de movimiento global mejora de forma sustancial el rendimiento.

Para este trabajo se ha supuesto que el Sistema de Seguimiento de CENTINELA tiene un subsistema de cálculo de movimiento global de la imagen que usa tanto la información obtenida del acelerómetro de la plataforma como de la propia imagen.

A partir de conocer el vector de movimiento global, se realiza un desplazamiento global ($shift_x$, $shift_y$) de los píxeles del fotograma en tiempo t .

$$I_{shift}(x, y) = I_t(x + shift_x, y + shift_y) \quad (10)$$

Y procedemos a calcular de nuevo el Flujo Óptico para la pareja de imágenes I_{shift} e I_{t-1} . Los resultados obtenidos contienen menos ruido y por tanto son mejores, pero no son lo suficientemente buenos como para que se pueda extraer algo de información sin un procesado excesivo.

Jugando con los parámetros del Flujo Óptico, se decide prescindir de las pirámides jerárquicas y se establece un tamaño de macrobloque (MB) de 16 píxeles. De esta forma aceleramos el proceso y



dividimos la imagen en bloques de 16 por 16 píxeles sobre los que aplicaremos el proceso en vez de hacerlo a nivel de píxel.

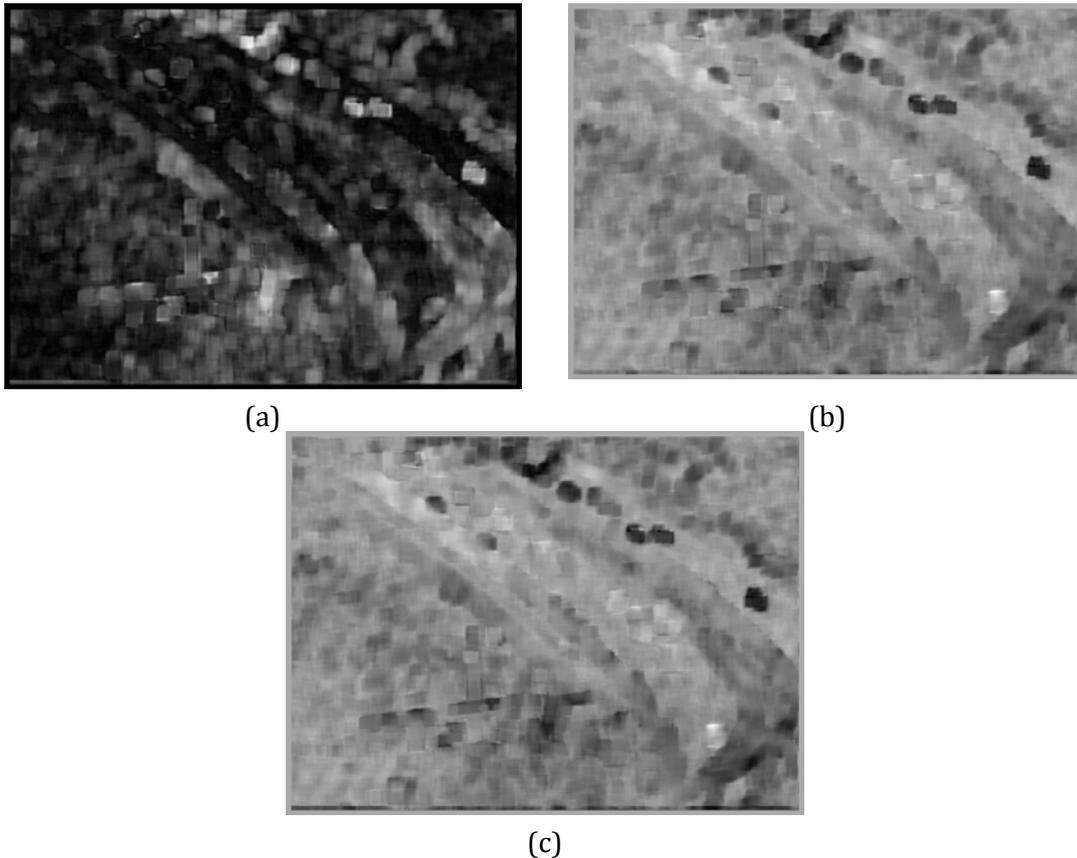


Figura 6: Lukas-Kanade con MB=16

En la Figura 6 (a) se muestra el resultado de calcular la magnitud de $U+V$. En (b) normalizando previamente U y V , y en (c) la suma de U y V normalizadas. Debido a que no hay mucha diferencia entre la suma con respecto al cálculo de la magnitud, seguiremos trabajando con la suma normalizada ya que reduce el tiempo de cálculo.

Como se aprecia en la Figura 6 (c), ya se puede observar cierta mejoría frente a lo obtenido en la Figura 5 (d). Aún así, todavía no es lo suficientemente sencillo obtener información de cuáles pueden ser los vehículos contenidos en la imagen. Aunque sí es cierto que se observan cierto grupo de píxeles con un color o muy oscuro o muy claro: esto se corresponde con píxeles con movimiento muy diferente al resto.

3.3.1.3 Procesado de los vectores de movimiento

Del proceso explicado en el apartado anterior tenemos una matriz (M) resultante de la siguiente operación:

$$M = norm_0^{255}(norm(U) + norm(V)) \quad (11)$$

donde se normalizan a 1 las matrices U y V obtenidas por el estimador de Flujo Óptico se suman y el resultado se normaliza a 255. Los valores extremos (0 y 255) simbolizan grandes cantidades de movimiento frente al movimiento medio (valor 128). A esta matriz se le aplicó una detección de

bordes por Canny y pero los resultados no resultaron ser muy buenos: se obtuvo mucho ruido y no se llegaron a detectar bien los bordes de interés

Se plantearon varias estrategias y después de probar varias combinaciones de éstas (reflejadas en el capítulo de resultados) se estimó que la mejor estrategia era usar una ecualización lineal de los valores de forma que los resultados de los extremos fueran más extremos y los cercanos al valor medio, se acercan más al valor medio.

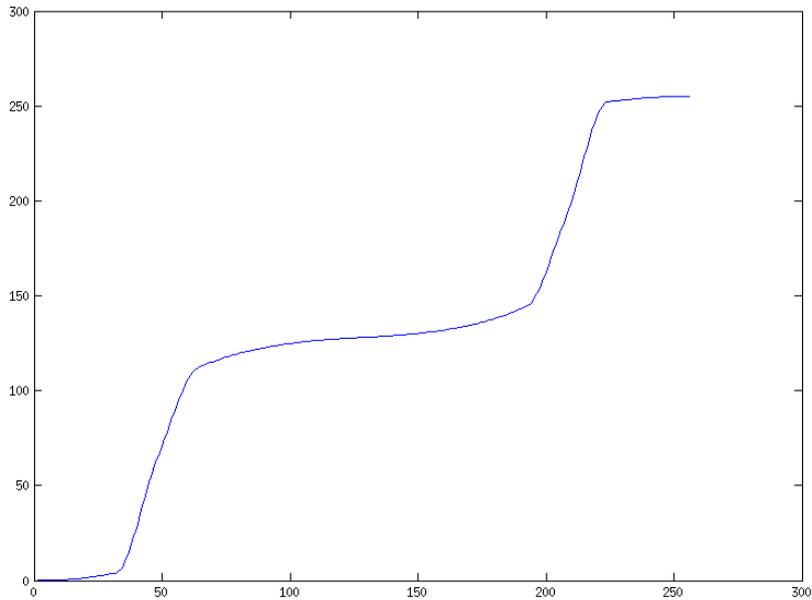
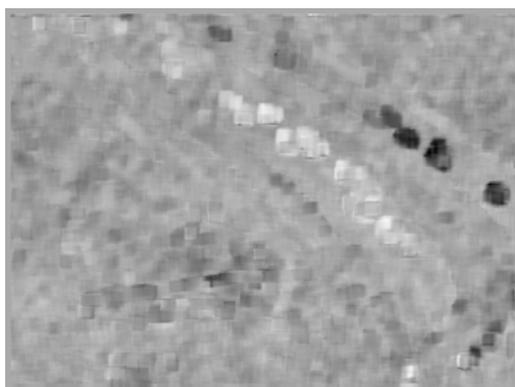


Figura 7: LUT de valor de cambio

La ecualización se realizó empleando una Look Up Table LUT que se representa en la Figura 7, donde el eje horizontal muestra los valores de la imagen original, y el eje vertical muestra los valores que tendrá la imagen resultado.



(a) Antes de la ecualización



(b) Después de la ecualización

Figura 8: Proceso de ecualización

Al resultado mostrado en la Figura 8 (a) se le aplica un filtro gaussiano para terminar de eliminar posibles ruidos. Una vez filtrado, se ejecuta una detección de bordes de tipo Canny con un algoritmo propio para detectar el umbral de detección. La determinación de dicho umbral es fundamental ya que varía enormemente los resultados y afecta al funcionamiento global del sistema. Las diferentes pruebas realizadas se encuentran reflejadas en el capítulo de resultados. En la Figura 8 (b) se muestra el resultado tras la ecualización.

3.3.1.4 Operaciones Morfológicas

Una vez realizada la detección de bordes y obtenido el resultado mostrado en la Figura 9, se pretenden identificar aquellos contornos cerrados que se corresponderán con zonas con movimiento homogéneo. Para ello, se realiza una dilatación y un cerrado de bordes. Se han probado con varios tipos de máscaras de dilatación y cerrado tal y como se muestra en la Tabla 2.

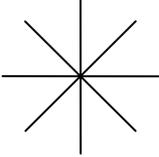
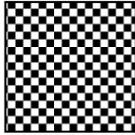
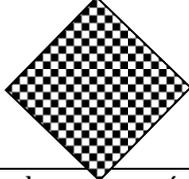
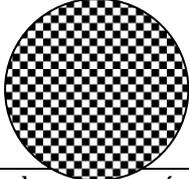
			
Cerrado mediante líneas	Cerrado con máscara cuadrada	Cerrado con máscara en diamante	Cerrado con máscara en disco

Tabla 2: Tipos de cerrado

Al resultado mostrado en la Figura 9 se le aplica una detección de contornos cerrados y se limpia la imagen. En la Figura 10(a) se muestra el resultado tras la dilatación; en (b) tras el cerrado de bordes y en (c) el rellenado de huecos.



Figura 9: Detección de bordes

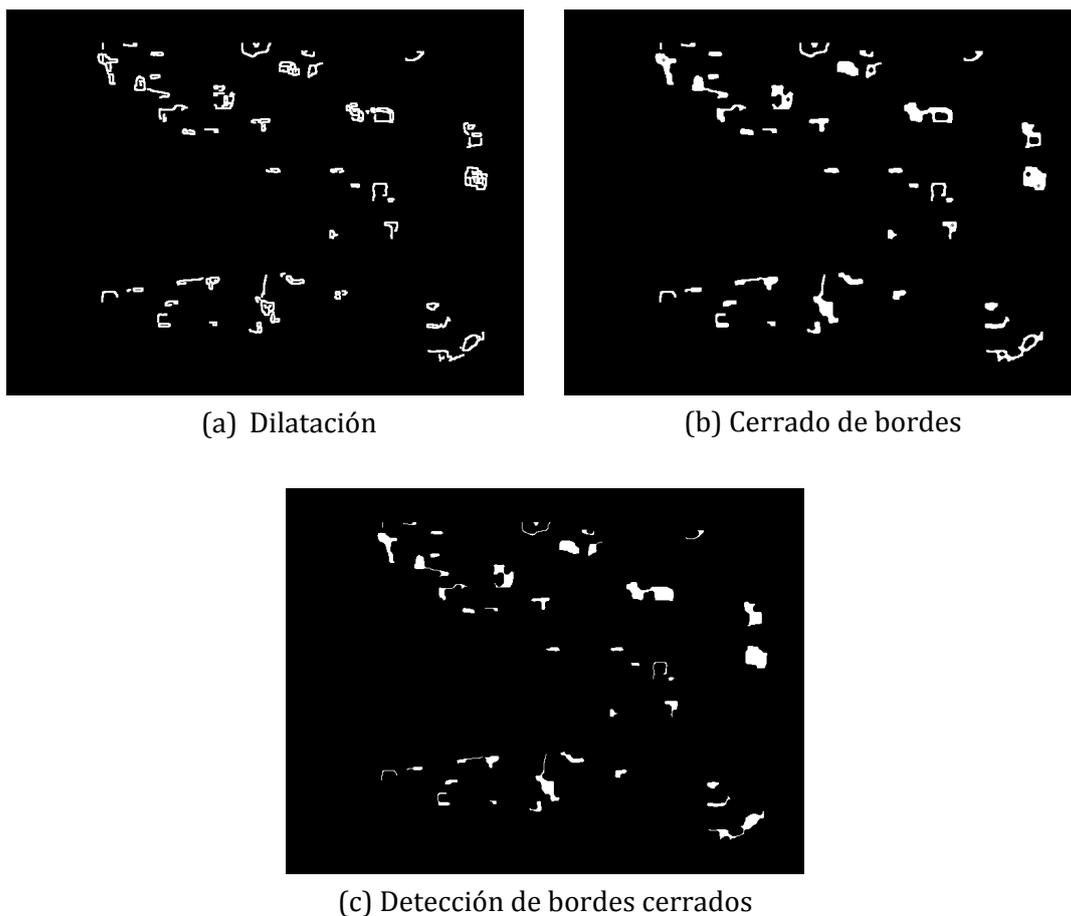


Figura 10: Operaciones Morfológicas aplicadas

3.3.2 DETECCIÓN ESTÁTICA

En principio, si el detector dinámico ha sido capaz de localizar zonas con movimiento, la detección estática no tendría por qué ejecutarse. Pero tal y como se ha visto en el estado del arte, el uso de una detección de vehículos por contorno puede dar buenos resultados, y por eso se decidió ver qué resultados arrojaría una detección con un solo fotograma y si estos resultados serían comparables a los de la detección dinámica.

Se aplica una detección de bordes partiendo del trabajo realizado en [55] en el que se filtra la imagen, se detectan los bordes y posteriormente se realizan una serie de operaciones morfológicas.

Al contrario que en [55], no nos interesa focalizar los esfuerzos en el centro de la imagen, con lo que el filtro paso bajo para los bordes de la imagen es desechado. El detector de contornos sigue siendo Canny, y al igual que para la detección dinámica, la umbralización es fundamental para obtener un buen resultado.

Por último se realizan operaciones morfológicas para la detección de contornos cerrados. en este caso no se va a realizar ninguna dilatación, porque las imágenes, ya de por si muy ruidosas, al dilatarse no se está ayudando a cerrar bordes, sino a incluir otros bordes que no debieran de estar presentes. Se aplica por tanto una operación de cerrado lineal con cuatro líneas (vertical, horizontal, y las dos diagonales). Por último se rellenan los huecos y se limpia la imagen.

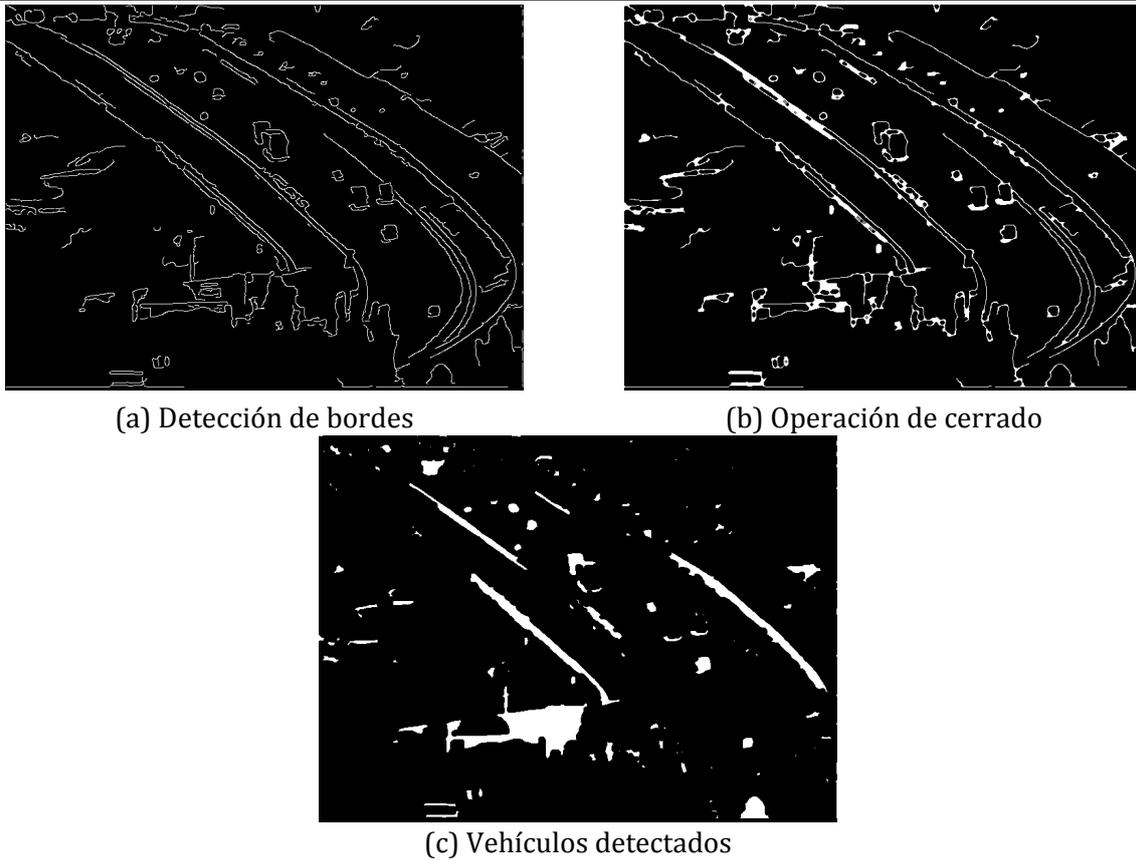


Figura 11: Detección de vehículos por contornos

En la Figura 11 se observa en primer lugar, en la imagen (a), el resultado de la detección de contornos por Canny, en la imagen (b) el resultado de la operación morfológica de cerrado y por último, en (c) los objetos detectados. Hay que tener en cuenta que muchos de ellos pueden pertenecer al fondo aunque no tengan altas frecuencias. Además en situaciones de observación de una carretera, los límites de ésta son una gran fuente de falsos positivos debido al contraste entre lo que no es carretera y lo que sí es carretera (tal y como se puede observar en la imagen (c) con grandes rayas blancas diagonales).

3.3.3 COMBINACIÓN DE ESTRATEGIAS UTILIZADA

Llegados a este punto en que cada método obtiene resultados diferentes, cabe plantearse si se puede obtener alguna información al conjugar ambos resultados.

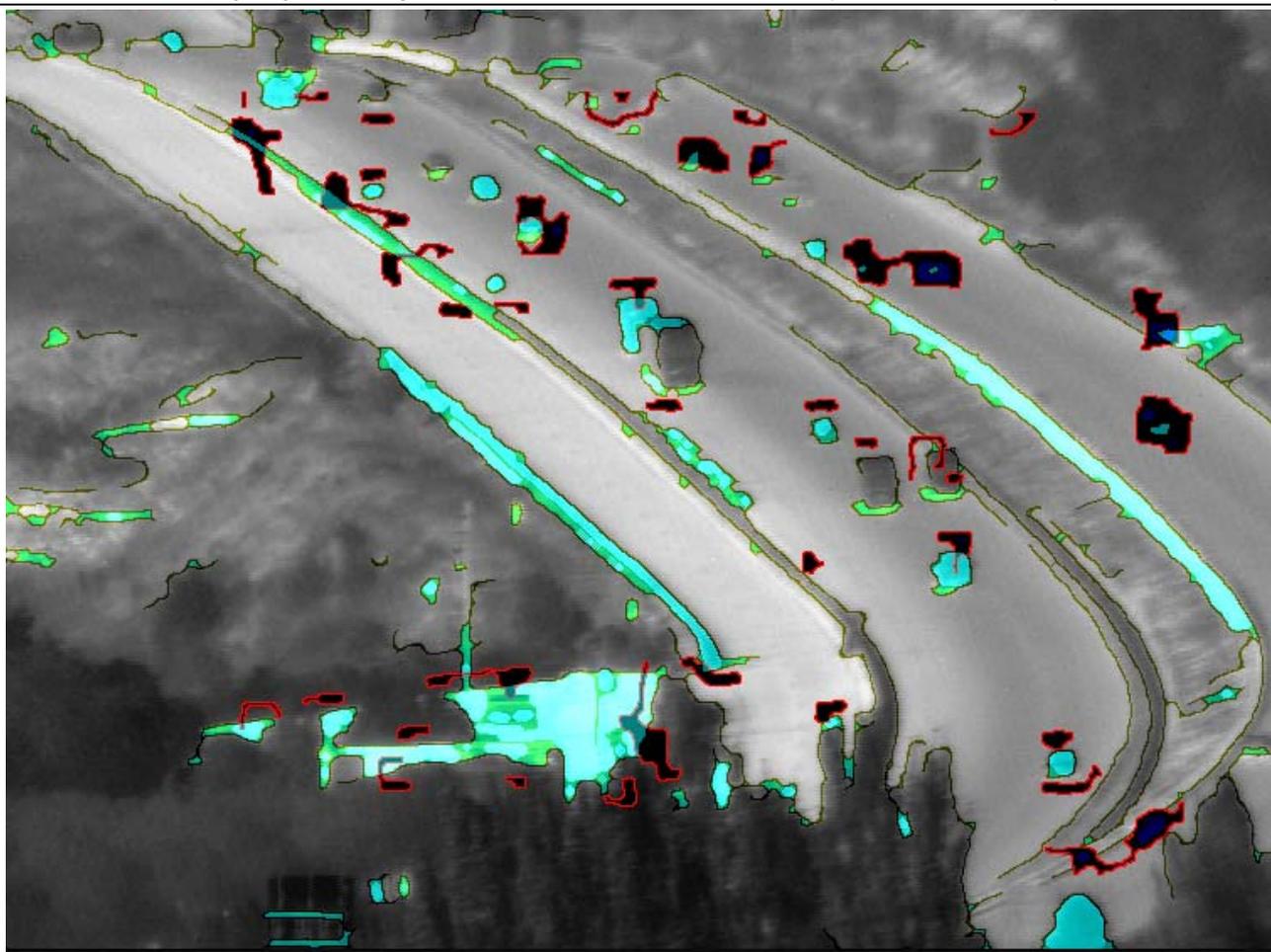


Figura 12: Resultado de combinar ambas técnicas

En la Figura 12 se muestra lo que ocurre al conjugar ambos resultados. Las zonas en azul claro son objetos detectados por el sistema estático, y en oscuro están los objetos en movimiento, es decir, detectados por el sistema dinámico. Como se puede apreciar, las zonas donde confluye el movimiento con el estático pertenecen a objetos artificiales.

3.4 DESCRIPCIÓN DEL ALGORITMO

El algoritmo se ha ido diseñando a medida que se iban obteniendo resultados y usando éstos como retroalimentación para afinar y obtener el mejor resultado posible. Finalmente se ha optado por una solución de compromiso que arroja buenos resultados para casi todas las secuencias de prueba. Para la evaluación del algoritmo se han utilizado 30 secuencias de pruebas de las cuales solo se pueden mostrar 20 en este trabajo por motivos de seguridad. Las secuencias serán presentadas en el capítulo de pruebas.

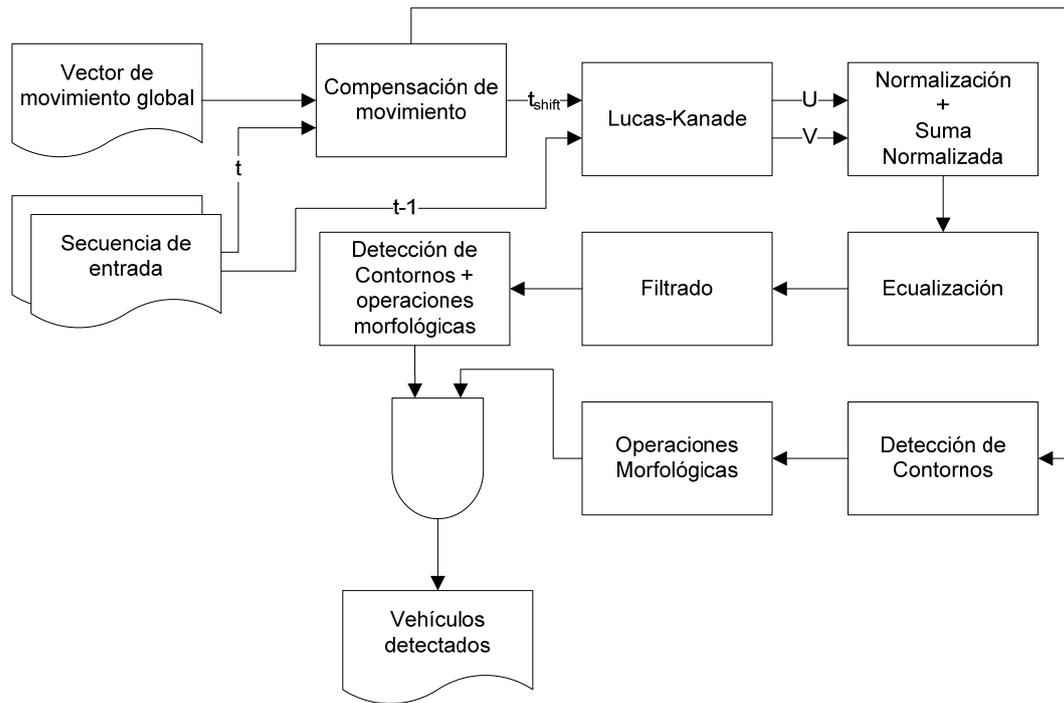


Figura 13: Algoritmo de detección de vehículos mejorado

En el capítulo de pruebas y resultados se detalla los diferentes resultados obtenidos y cómo se ha ido ajustando el funcionamiento del algoritmo hasta el diseño final del mismo.

El diseño final es el mostrado en la Figura 13. Como secuencia de entrada se disponen de dos matrices de luminancia con 8 bits de precisión. Las matrices se corresponden con los valores de luminancia de los fotogramas en tiempo t y $t-1$. Al fotograma t se le aplica una compensación de movimiento según el vector de movimiento global. Al resultado de esta transformación se le conoce por t_{shift} y servirá de entrada tanto para el estimador de Flujo Óptico como para la detección estática.

Con $t-1$ y t_{shift} se alimenta al estimador de Lucas-Kanade y se obtienen dos matrices (U y V) con los vectores de movimiento correspondientes al Flujo Óptico de la secuencia. Se normalizan dichos vectores a 1 y se suman ambas matrices para volver a normalizar a 255. Con esta última normalización se pierde precisión, pero resulta más fácil interpretar visualmente la imagen generada con las intensidades de los vectores de movimiento, además de facilitar la ecualización posterior.

$$M = norm_0^{255}(norm(U) + norm(V)) \quad (12)$$

Una vez obtenida la matriz M dada por la ecuación (12), se ecualiza el resultado medio de la LUT correspondiente a la Figura 7 y se detectan los contornos. Esta operación permite detectar en qué zonas de la imagen se está produciendo movimiento y en cuáles no.

Con los contornos detectados en una imagen binaria, se aplica una operación morfológica de cerrado mediante cuatro líneas a 0, 90, 45 y -45 grados, rellenando los contornos cerrados para detectar las manchas resultantes. Dichas manchas se corresponden con posibles vehículos.

Una vez finalizada la fase de detección dinámica, se realiza la fase de detección estática, consistente en realizar las operaciones de detección de contornos, búsqueda y cerrado de contornos cerrados (mediante la misma operación morfológica que en la fase dinámica) y detección de posibles vehículos.

Por último, con los resultados de ambas fases, se realiza una multiplicación entre ambas matrices binarias para identificar a los posibles vehículos.



Figura 14: Resultado del algoritmo final

En la Figura 14 se muestra el resultado obtenido por el algoritmo final, destacando en color negro los posibles vehículos detectados. Como se explicó en el estado del arte, el algoritmo final pertenece a los del tipo Generadores de Hipótesis, con lo que para resolver si lo detectado es realmente un vehículo o no, tendría que aplicársele un Verificador de Hipótesis, aspecto no contemplado en este Trabajo Fin de Máster.

3.5 CONCLUSIONES

En este capítulo se ha resumido el trabajo realizado para llegar a obtener un detector de vehículos relativamente robusto. Se ha estudiado el problema planteado a partir del algoritmo existente en el sistema CENTINELA y se han analizado las diferentes técnicas disponibles en la literatura para la detección de vehículos.

Una vez realizados dichos análisis se han desechado aquellas técnicas que no pueden ser aplicadas al problema planteado y a partir de las técnicas que resultan validas en este entorno, se ha planteado una solución innovadora en cuanto a que combina el uso del Flujo Óptico con la detección de contornos.

Se ha ido ajustando el funcionamiento del algoritmo mediante la combinación y el ajuste de varias operaciones (filtrado, ecualización, detección de contornos, y operaciones morfológicas) hasta llegar a una solución que obtiene buenos resultados para la mayoría de las secuencias de prueba, extremo que será presentado en el siguiente capítulo.

4 PRUEBAS Y RESULTADOS OBTENIDOS

Como se ha comentado en el capítulo de diseño e implementación, se han ido probando las diferentes versiones del algoritmo sobre un conjunto de secuencias de prueba extraídas de la biblioteca de TecnoBit. A medida que se han ido obteniendo los resultados se han comparado con los ya obtenidos y se han anotado los resultados y qué versión del algoritmo funciona mejor en cada caso.

4.1 SECUENCIAS DE PRUEBA

Se presentan en este apartado 20 de las 30 secuencias de prueba utilizadas en este TFM. A cada secuencia se le ha asignado un identificador y se ha incluido el primer fotograma de cada secuencia junto con una breve descripción de su contenido y/o características más importantes.

ID	Primer fotograma	Comentario
1		Pájaro volando sobre la línea del horizonte
2		Carretera con tránsito medio. Carriles bien identificados

3		Carretera con tránsito medio. Carriles bien identificados
4		Carretera con tránsito medio. Carriles con mala identificación. Mucha estructura artificial. Algunos vehículos se confunden con el fondo
5		Carretera con tránsito medio. Carriles con mala identificación. Mucha estructura artificial. Algunos vehículos se confunden con el fondo

6		<p>Carretera con tránsito medio. Carriles bien identificados. Algunas sombras pueden conducir a engaño</p>
7		<p>Carretera con tránsito medio. Carriles bien identificados</p>
8		<p>Avión en maniobra de despegue. Fondo con muchos objetos artificiales</p>

9		<p>Avión en maniobra de despegue sobre fondo homogéneo</p>
10		<p>Fondo con muchas construcciones que destacan. Vehículos casi indistinguibles sobre la carretera en el inferior de la imagen</p>
11		<p>Fondo con muchas construcciones que destacan. Vehículos casi indistinguibles sobre carretera en el inferior de la imagen</p>

12		Helicóptero de tamaño pequeño sobre fondo homogéneo. Imagen en IR
13		UAV de tamaño medio sobre fondo homogéneo y algo ruidoso. Imagen en IR
14		Avioneta de gran tamaño sobre fondo homogéneo. Imagen en IR

15		Fondo con muchas construcciones que destacan. Vehículos casi indistinguibles sobre carretera en el inferior de la imagen. Imagen en IR
16		Fondo con muchas construcciones que destacan. Vehículos casi indistinguibles sobre carretera. Imagen en IR
17		Carretera con tránsito medio. Carriles bien identificados. Imagen en IR

18		Carretera con tránsito medio. Carriles bien identificados. Imagen en IR
19		Carretera con tránsito medio. Carriles bien identificados. Tamaño de los vehículos muy pequeños. Imagen en IR
20		Avión en posición de aterrizaje en el centro de la escena camuflado por estructuras humanas. Imagen en IR

Con estas imágenes se cubren la mayoría de los casos que se pueden encontrar en las situaciones reales donde este sistema tiene aplicación. Hay que notar que se usaron 10 imágenes adicionales de prueba, pero por motivos de seguridad no se pueden reproducir en este documento.

4.2 PRUEBAS REALIZADAS

Como ya se ha indicado con anterioridad, el algoritmo se ha ido ajustando a medida que se han ido obteniendo resultados, siendo por tanto un proceso iterativo en el que, partiendo de una versión inicial del algoritmo, se fueron modificando bloques funcionales o parámetros (uno sólo en cada iteración) y registrando y analizando los resultados, comprendiendo como afectaban los cambios al rendimiento.

Para ello, se ha ejecutado en cada iteración el algoritmo sobre todas las imágenes de prueba. Una vez obtenidos los vehículos detectados, se ha comparado con los resultados obtenidos en iteraciones previas y se ha anotado la evaluación de los resultados.

Para evaluar los resultados se ha observado cuántos de los vehículos se han detectado, cuántos han sido falsos positivos y finalmente se ha realizado una evaluación subjetiva de si el algoritmo ha acertado en su detección o no. Esta medición binaria (bien o mal) junto con el análisis de por qué detecta mejor o peor ha sido la guía que ha dirigido la evolución del trabajo.

Para decidir si una detección es buena o mala, superponiendo en la imagen varias capas (contornos y objetos encontrados de ambos tipos de detecciones y objeto final detectado) se observa cuántos de los objetos finales se corresponden con vehículos y cuáles no son vehículos. Si los falsos positivos son de área pequeña o en número reducido (menos de un tercio de los vehículos detectados) y se detectan más del 60 por ciento de los vehículos contenidos en la secuencia, se da por bueno. En caso contrario, por malo.

Esta evaluación se repite para cada una de las treinta secuencias en cada iteración. Además, se observan también los resultados obtenidos por cada bloque funcional para entender mejor lo sucedido. Por último, se compara el resultado global con el de la mejor iteración anterior, y si el cambio mejora el algoritmo, se adopta.

En la Tabla 3 se recogen las 14 iteraciones más significativas de las más de 40 iteraciones realizadas para cada imagen. El significado de cada columna es el siguiente:

- **It:** número de iteración.
- **Ecu:** tipo de equalización empleada. A indica suave, B indica fuerte.
- **Filtrado:** tipo de filtro empleado y su tamaño.
- **Umbral Contorno:** valor del multiplicador del umbral.
- **Estrategia de Cerrado:** Operación Morfológica usada para cerrar los contornos.
- **Calidad Detección Dinámica:** número de pruebas válidas para el total de 30 secuencias.
- **Calidad Detección Estática:** número de pruebas válidas para el total de 30 secuencias.
- **Calidad Detección Global:** número de pruebas válidas para el total de 30 secuencias.
- **Comentario:** Aspectos más relevantes a destacar.



It	Ecu	Filtrado	Umbral Contorno	Estrategia Cerrado	Calidad Detección Dinámica	Calidad Detección Estática	Calidad Detección Global	Comentario
1	A	Gauss [3]	1	Square[3]	15/30	14/30	12/30	Primera versión del algoritmo. El bajo valor del umbral empeora la calidad de la detección de contornos
2	A	Sin filtro	2,5	Square[3]	12/30	19/30	8/30	El no utilizar filtro hace que aumenten las altas frecuencias en la matriz de vectores de movimiento empeorando el resultado global. El uso del multiplicador mejora la detección de contornos
3	Sin ecu	Gauss [3]	2,5	Square[3]	9/30	19/30	15/30	Empeora notablemente el resultado. La ecualización consigue diferenciar las zonas de movimiento de las estáticas
4	A	Media [5]	2,5	Square[3]	18/30	19/30	15/30	El uso de filtros de mayor tamaño elimina muchas de las altas frecuencias no deseadas.
5	A	Gauss [5]	2,5	Square[3]	19/30	19/30	16/30	Es el que mejor resultado obtiene de los tres tipos de filtros probados
6	A	Mediana [5]	2,5	Square[3]	17/30	19/30	14/30	Resultados muy similares al filtro de media
7	A	Gauss [5]	2,5	Lines [4]	22/30	22/30	19/30	Mejora en el cierre de contornos cerrados
8	A	Gauss [5]	2,5	Square [5]	19/30	17/30	14/30	Las líneas de contornos se vuelven más gruesas, se añade ruido y objetos no deseados al resultado final
9	A	Gauss [5]	2,5	Diamond [5]	15/30	18/30	11/30	Este tipo de estrategia perjudica ya que posibles objetos colindantes los unifica y los muestra como uno solo
10	A	Gauss [5]	2,5	Disk [5]	14/30	16/30	9/30	Ocurre lo mismo que en la It 9
11	B	Gauss [5]	2,5	Lines [4]	25/30	22/30	21/30	La nueva LUT mejora los resultados obtenidos, eliminando ruido
12	B	Gauss [5]	2	Lines [4]	22/30	17/30	14/30	Los bordes se acentúan y muchos de los contornos que se deberían cerrar, no terminan de cerrarse
13	B	Gauss [5]	4	Lines [4]	22/30	18/30	17/30	Los bordes se suavizan demasiado y muchos objetos no se detectan
14	B	Gauss [5]	2,2	Lines [4]	27/30	26/30	24/30	Mejor versión obtenida

Tabla 3: Resultados de las iteraciones

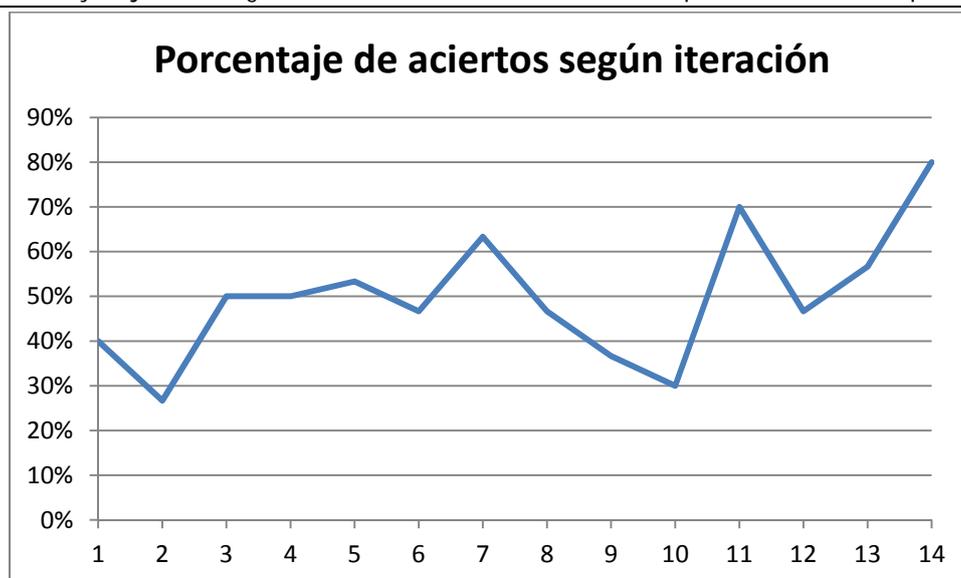


Figura 15: Porcentaje de aciertos según iteración

En la Figura 15 se muestra la tendencia de los aciertos a medida que se iba afinando el algoritmo. En total se han analizado más de 6000 imágenes resultantes de cada etapa del proceso (vectores del flujo óptico, filtrado, detección de contornos, etc.) y se ha llegado a este resultado mediante un procedimiento de prueba y error guiado por la calidad de los resultados obtenidos.

4.3 RESULTADOS OBTENIDOS

A continuación se exponen los resultados tras analizar los resultados obtenidos al ejecutar las distintas simulaciones con la variación de los parámetros que afectan de forma más directa a los resultados.

4.3.1 COMPENSACIÓN DE MOVIMIENTO GLOBAL

La detección del vector de movimiento global es fundamental, la fiabilidad de este vector es crítica para poder realizar una correcta estimación del flujo óptico. Se realizaron varias pruebas con vectores de movimiento con error aleatorio empeorando sensiblemente el resultado global en la detección de vehículos. El vector de movimiento empleado para cada secuencia viene proporcionado por el estimador de movimiento global del CENTINELA que combina la información proporcionada por un acelerómetro y un estimador software basado en la correlación de ambas imágenes.

4.3.2 ECUALIZACIÓN

Se usaron varias máscaras de ecualización ajustadas por métodos lineales, Spline o Hermitte. La que mejor resultado obtuvo con bastante diferencia fue la que se muestra en la Figura 16.

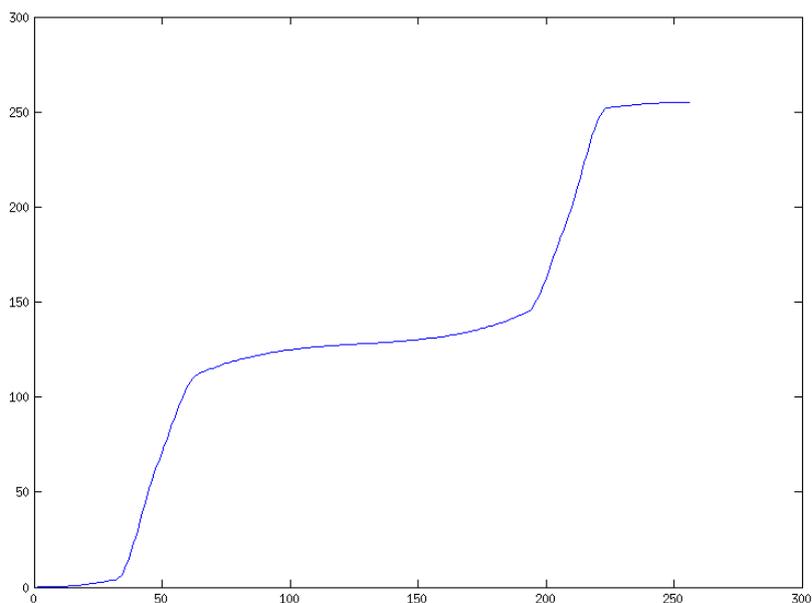


Figura 16: Máscara de ecualización

Hasta cierto punto, es conveniente que se divida en tres zonas de valores, 0, 128 y 255 con cambios pronunciados entre estos tres valores, pero no demasiado abruptos porque podría llegar a generar ruido de alta frecuencia.

4.3.3 FILTRO POST-ECUALIZACIÓN

Se han probado diferentes tipos de filtro: gaussiano, media, mediana o sin filtrado.

El no filtrar el resultado de la ecualización provoca que las altas frecuencias que la ecualización haya generado se propaguen a lo largo del algoritmo en forma de nuevos contornos y posibles falsos aciertos.

Cuando la máscara del filtro es pequeña (3x3) no se llega a suavizar lo suficiente, provocando que se detecten demasiado contornos y haciendo que al ejecutar el cerrado de contornos se cierren objetos que no lo son.

De entre los filtros usados, el que mejor comportamiento ha demostrado es el de un filtro gaussiano con una máscara de 5x5 píxeles.

4.3.4 DETECCIÓN DE CONTORNOS

Esta fase, junto con la estimación del Flujo Óptico, es la fase más crítica del sistema: si se detectan demasiados contornos, demasiado pocos o los que no debieran ser contornos, el sistema no funciona en absoluto.

El parámetro más crítico de este apartado es la determinación del umbral. Si bien en la versión implementada del detector de Canny, el cálculo del umbral es automático, con la posibilidad de incluir un parámetro que lo controle, este no es perfecto. A lo largo de las distintas ejecuciones se ha

detectado que el multiplicador que va bien para unas secuencias, no funciona bien para otras. Al final el valor usado es el que maximiza el número de vehículos detectados para todas las secuencias.

4.3.5 OPERACIONES MORFOLÓGICAS

Si antes se decía que la detección de contornos es crítica, la detección de los objetos de contornos cerrados no lo es menos. Se ha visto a lo largo de las pruebas que muchos de los vehículos (tanto en las imágenes de bordes estáticas como dinámicas) se quedaban sin detectar ya que faltaba un píxel para terminar de cerrar el contorno



Figura 17: Ejemplos de contornos que no se han cerrado

En la Figura 17 se muestran varios ejemplos de objetos después de haber pasado por la operación de cerrado y que no han terminado de cerrar todos su píxeles y por tanto en la etapa de detección de contornos cerrados no serán etiquetados.

Se han probado varias estrategias para el correcto cerrado:

- Dilatación.
- Máscara de diamante de 3 y 5 píxeles de radio.
- Máscara circular de 3 y 5 píxeles de radio.
- Máscara cuadrada de 3 y 5 píxeles de lado.
- Líneas a 0°, 90°, 45° y -45° de 3, 4 y 5 píxeles de largo.

Todas las estrategias de máscara y la dilatación ofrecían pobres resultados ya que terminaban por introducir mucho ruido al conectar contornos que no tenían que ser conectados. La mejor estrategia resulto ser la de cerrado mediante líneas a diferentes ángulos con 4 píxeles de lado.

Es importante destacar que una mejor estrategia de cerrado mejorará el rendimiento general del algoritmo.

4.4 COMPARACIÓN CON OTRAS ESTRATEGIAS

En el presente trabajo no se ha podido comparar el rendimiento del algoritmo diseñado, frente a otros algoritmos de los mencionados en la bibliografía debido a que no se dispone del código de algoritmos de detección alternativos. Al tratarse de algoritmos de gran potencial comercial, los desarrolladores de estos algoritmos no los ponen en dominio público.

Sí se ha comparado en cambio con el algoritmo original disponible en el sistema oprónico CENTINELA y se ha detectado una mejoría o igualdad para todas las secuencias de prueba. El rendimiento del algoritmo en cuanto a tiempo de cómputo no ha podido ser medido ya que ambos se ejecutan sobre distintas plataformas.

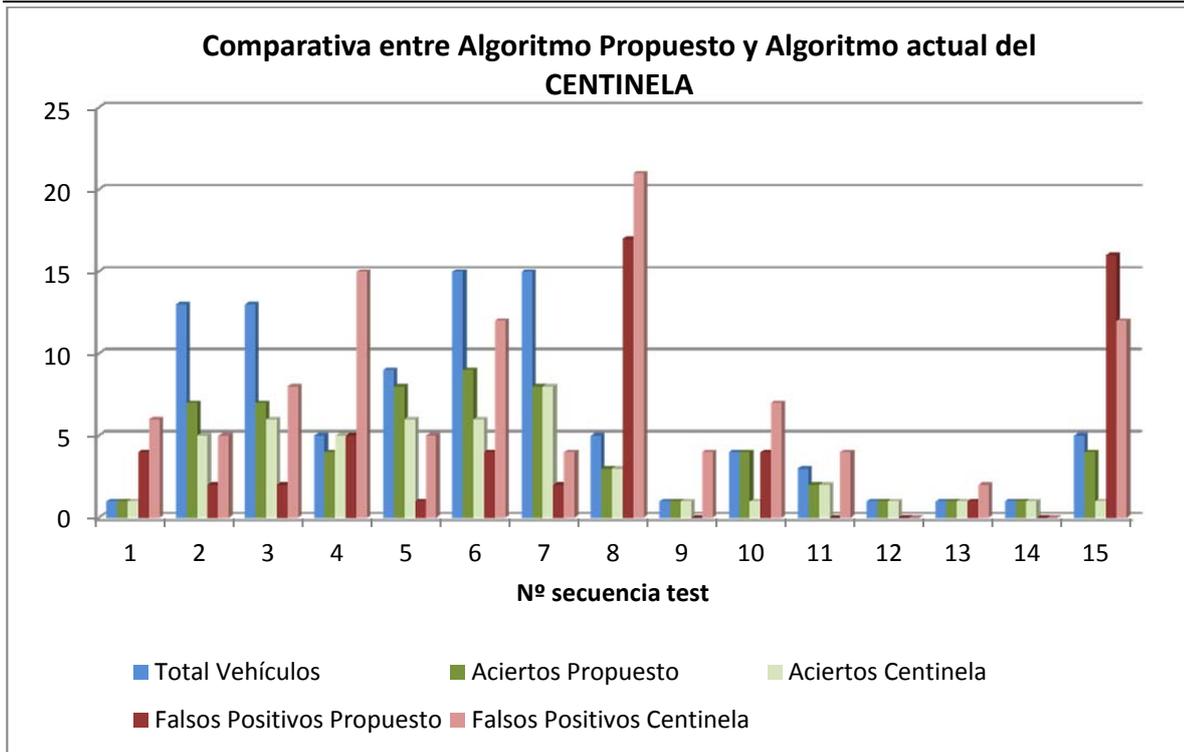


Figura 18: Comparativa entre algoritmos (parte 1)

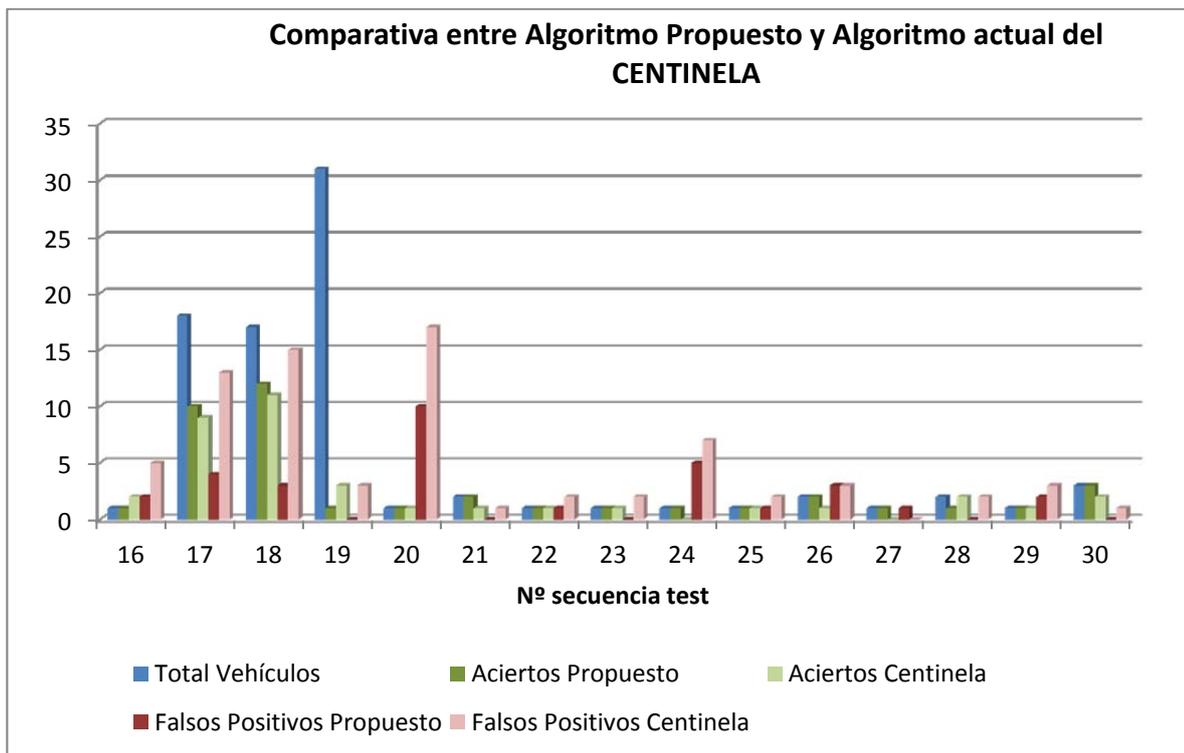


Figura 19: Comparativa entre algoritmos (parte2)

En la Figura 18 y en la Figura 19 se muestra una comparativa entre los resultados obtenidos por el algoritmo original del CENTINELA y el algoritmo propuesto. La comparativa se ha realizado para cada una de las 30 secuencias de prueba y en ella se muestran el número total de vehículos de la secuencia



Título: Actualización y mejora del Algoritmo de Detección de Vehículos incorporados al Sistema Optrónico CENTINELA
(azul), el número de vehículos señalados por el algoritmo propuesto (verde oscuro), por el algoritmo original (verde claro), el número de elementos señalados como vehículos y que no lo son por el algoritmo propuesto (rojo) y los falsos positivos por el algoritmo del CENTINELA (rojo pálido).

En ellas se aprecia como el algoritmo propuesto obtiene un rendimiento superior en la mayoría de las secuencias de test, detectando más vehículos y obteniendo un menor número de falsos positivos. Gracias a esta comparativa se puede afirmar que el algoritmo propuesto mejora el original en un 20% en cuanto al número de detecciones y reduce el número de falsos positivos en un 50%.



Figura 20: Diferencias entre los algoritmos de detección

En la Figura 20 se muestra una imagen reveladora acerca de la diferencia entre el algoritmo original y el algoritmo propuesto, apreciándose mucho ruido en el primero (a), y mucho más definida la detección de vehículos en el algoritmo propuesto (b).

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 CONCLUSIONES

El objetivo de este trabajo era la de estudiar los algoritmos actuales de detección de vehículos, analizar el algoritmo de detección de vehículos del sistema CENTINELA; y diseñar una mejora que cumpliera con los requisitos mencionados en el primer capítulo.

Se ha realizado un estudio intensivo del estado del arte en este área de la técnica, identificando las principales estrategias para la detección de vehículos. De estas técnicas se han extraído sus usos, sus fortalezas y sus debilidades y a partir de las necesidades requeridas por los casos de uso de CENTINELA, se ha propuesto un nuevo algoritmo con prestaciones mejoradas.

Este algoritmo cumple con los requisitos iniciales, excepto el de que pueda ejecutarse en tiempo real; usando técnicas contrastadas y con una tasa de aciertos aceptable.

La principal innovación propuesta del algoritmo es el uso combinado de las técnicas de detección de contornos con la estimación del flujo óptico. En la literatura consultada, en ningún caso se hace uso del estudio del gradiente de los vectores de movimiento siendo esta aportación novedosa en el campo de algoritmos de detección de vehículos.

5.2 LÍNEAS FUTURAS

Tras haber realizado este trabajo, queda claro que hay mucho camino por recorrer y muchas cosas a mejorar. Los resultados obtenidos son buenos pero se podrían optimizar, ya que se han probado un número limitado de soluciones y optimizaciones. Los principales bloques funcionales a mejorar son el estimador de flujo óptico, el detector de contornos y la detección de contornos cerrados.

El estimador de Flujo Óptico empleado, si bien cumple con su misión, puede mejorarse en precisión y tiempo de ejecución. Una futura línea de investigación puede ser la de usar estimadores jerárquicos o la implementación de un estimador en sistemas de procesamiento paralelo.

El detector de contornos depende altamente de la función de umbralización. Esto hace que un cierto umbral sea muy bueno para un tipo de secuencia, para para otro tipo no funcione correctamente. Un estimador automático de umbrales en función de la imagen de entrada sería una interesante continuación de este trabajo.

Por último, el cerrado de líneas es un campo con muchas posibilidades. A través del estudio de los diferentes resultados obtenidos, se ha hecho patente que, en más de una ocasión, el vehículo no ha sido detectado por culpa de uno o dos píxeles abiertos, o por el contrario, contornos que no debieran haberse cerrado sí lo fueron. Encontrar la solución de estos problemas mejoraría de forma importante el rendimiento del algoritmo.



6 BIBLIOGRAFÍA

- [1] TECNOBIT, «Brouchure sistema oprónico CENTINELA,» [En línea]. Available: http://www.tecnobit.es/es/c/document_library/get_file?uuid=03aaf814-fb36-4e90-a3e0-4d0ab989039e&groupId=10156. [Último acceso: 27 08 2012].
- [2] S. Blackman y R. Popoli, Design and Analisis of Modern Tracking Systems, Artech House, 1999.
- [3] Y. Bar-Shalom y X.-R. Li, Multitarget-Multisensor Tracking: Principles and Techniques, YBS Publishing, 1995.
- [4] M. Masten y L. Stockum, «Acquisition, Tracking and Pointing,» de *SPIE vol. 2739*, 1996.
- [5] W. Jones, «Building Safer Cars,» *IEEE Spectrum*, vol. 39, 2002.
- [6] U. Franke, From Door to Door - Principles and Applications of Computer Vision for Driver Assistan Systems, 2001: Butterworth Heinemann.
- [7] M. Herbert, «Active and Passive Range Sensing for Robotics,» de *Proc. IEEE Int'l Conf. Robotics and Automation*, 2000.
- [8] S. Park, T. Kim, S. Kang y k. Heon, «A Novel Signal Processing Technique for Vehicle Detection Radar,» *IEEE MTT-S Int'l Microwave Symp Digest*, 2003.
- [9] C. Wang, C. Thorpe y A. Suppe, «Ladar-Based Detection and Tracking of Moving Objects from a Ground Vehicle at High Speeds,» de *Proc. IEEE Intelligent Vehicles Symp*, 2003.
- [10] R. ChellappaÇ, G. Qian y Q. Zheng, «Vehicle Detection and Tracking Using Acoustic and Video Sensor,» de *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Processing*, 2004.
- [11] G. Marola, «Using Simmetry for Detecting and Locating Objects in a Picture,» *Computer Vision, Graphics and Image Processing*, 1989.
- [12] A. Kuehnle, «Simmetry-Based Recognition for Vehicle Rears,» *Pattern Recognition Letters*, 1991.
- [13] T. Zielke, M. Brauckmann y W. v. Seelen, «Intensity and Edge-Based Symmetry Detection with an Application to Car-Following,» *CVGIP: Image Understanding*, vol. 58, 1993.
- [14] S. Buluswar y B. Draper, «Color Machine Vision for Autonomous Vehicle,» de *Int'l J. Eng. Applications of Artificial Intelligence*, 1998.
- [15] J. Crisman y C. Thorpe, «Color Vision for Road Following,» de *Proc. SPIE Conf. Mobile Robots*, 1988.
- [16] H. Mori y N. Charkai, «Shadow and Rhythm as Sign Patterns of Obstacle Detection,» de *Proc. Int'l Symp. Industrial Electronics*, 1993.



- [17] E. Dickmanns, «The Seeing Passenger Car 'Vamors-P',» de *Proc. Int'l Symp. Intelligent Vehicles*, 1994.
- [18] C. Tzomakas y W. Seelen, «Vehicle Detection in Traffic Scenes Using Shadows,» 1998.
- [19] J. Bertozzi, A. Brooggi y S. Castelluccio, «A Real-Time Oriented System for Vehicle Detection,» *J. Systems Architecture*, 1997.
- [20] M. Sharifi, M. Fathy y m. T. Mahmoudi, «A classified and comparative study of edge detection algorithms,» de *Proc. Int'l Conf. on Information Technology: Coding and Computing*, 2002.
- [21] N. Matthews, P. An, D. Charnley y C. Harris, «Vehicle,» *Control Eng. Practice*, vol. 4, 1996.
- [22] C. Goerick, N. Detlev y M. Werner, «Artificial Neural,» *Pattern Recognition Letters*, vol. 17, 1996.
- [23] M. Betke, E. Haritaglu y L. Davis, «Real-Time Multiple Vehicle Detection and Tracking from a Moving Vehicle,» *Machine Vision and Applications*, vol. 12, nº 2, 2000.
- [24] N. Srinivasa, «A Vision-Based Vehicle Detection and Tracking Method for Forward Collision Warning,» de *Proc. IEEE Intelligent Vehicle Symp.*, 2002.
- [25] Z. Sun, R. Miller, G. Bebis y D. DiMeo, «A Real-Time Precrash Vehicle Detection System,» de *Proc. IEEE Int'l Workshop Application of Computer Vision*, 2002.
- [26] T. Kalinke, C. Tzomakas y W. v. Seelen, «A Texture-Based Object Detection and an Adaptive Model-Based Classification,» de *Proc. IEEE Int'l Conf. Intelligent Vehicles*, 1998.
- [27] R. Haralick, B. Shanmugam y I. Dinstein, «Texture Features for Image Classification,» de *IEEE Trans. System, Man, and Cybernetics*, 1973.
- [28] R. Cucchiara y M. Piccardi, «Vehicle Detection under Day and Night Illumination,» de *Proc. Int'l ICSC Symp. Intelligent Industrial Automation*, 1999.
- [29] H. Mallot, H. Bulthoff, J. Little y S. Bohrer, «Inverse Perspective Mapping Simplifies Optical Flow Computation and Obstacle Detection,» *Biological Cybernetics*, vol. 64, nº 3, 1991.
- [30] U. Franke y I. Kutzbach, «Fast Stereo Based Object Detection for Stop and Go Traffic,» *Intelligent Vehicles*, 1996.
- [31] G. Zhao y S. Yuta, «Obstacle Detection by Vision System for Autonomous Vehicle,» *Intelligent Vehicles*, 1993.
- [32] J. W. T. H. J. M. G. O. B. R. S. R. D. Koller, «Towards Robust Automatic Traffic Scene Analysis in Real-time,» de *Proc ICPR*, 1994.
- [33] B. Lo y S. Velastin, «Automatic congestion detection system for underground platforms,» de *Proc. ISIMP 2001*, 2001.



- [34] C. Stauffer y W. Grimson, «Adaptative background mixture models for real-time tracking,» de *IEEE CVPR*, 1999.
- [35] D.-S. Lee, «Effective Gaussian Mixture Learning for Video Background Subtraction,» *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, nº 5, 2005.
- [36] D. H. L. D. A. Elgammal, «Non parametric model for background subtraction,» de *ECCV*, 2000.
- [37] D. Comaniciu y P. Meer, «Mean shift: a robust approach toward feature space analysis,» *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, nº 5, 2002.
- [38] T. W. H. F. K. S. M. Seki, «Background subtraction based on cooccurrence of image variations,» de *Proc. of CVPR*, 2003.
- [39] B. R. A. P. N.M. Oliver, «A Bayesian computer vision system for modeling human interactions,» *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, nº 8, 2000.
- [40] O. Javed, K. Shafique y M. Shah, «A Hierarchical Approach to Robust Background Subtraction using Color and Gradient Information,» de *Proc. of the Workshop on Motion and Video Computing*, 2002.
- [41] B. Shoushtarian y H. E. Bez, «A practical adaptive approach for dynamic background subtraction using an invariant colour model and object tracking,» *Pattern Recognition Letters*, vol. 26, 2005.
- [42] J. Zhong y S. Sclaroff, «Segmenting Foreground Objects from a Dynamic Textured Background via a Robust Kalman Filter,» de *Proc. ICCV*, 2003.
- [43] A. Lipton, H. Fujiyoshi y R. Patil, «Moving Target Classification and tracking from real-time video,» de *Proc. IEEE Workshop Applications of Computer Vision*, 1998.
- [44] B. Horn y B. Schunck, «Determining optical flow,» vol. 17, 1981.
- [45] B. Lucas y T. Kanade, «An Iterative Image Registration Technique with an Application to Stereo Vision,» de *Proc. Imaging Understanding Workshop*, 1981.
- [46] J. S. a. C. Tomasi, «Good features to track,» de *IEEE Computer Society Conference*, 1994.
- [47] D. Meyer, J. Denzler y H. Niemann, «Model based extraction of articulated objects in image sequences for gait analysis,» de *Proc. IEEE Int'l Conf. Image Processing*, 1998.
- [48] D. Fleet, J. Barron y S. Beauchemin, «Performance of optical flow techniques,» *International Journal on Computer Vision*, vol. 12, nº 1, 1994.
- [49] A. Giachetti, M. Campani y V. Torre, «The Use of Optical Flow for Road Navigation,» *IEEE Trans. Robotics and Automation*, vol. 14, nº 1, 1998.
- [50] W. Kruger, W. Enkelmann y S. Rossle, «Real-Time Estimation and Tracking of Optical Flow Vectors



- [51] J. Weng, N. Ahuja y T. Huang, «Matching Two Perspective Views,» *IEEE Trans. Pattern Analysis and Machine intelligence*, vol. 14, 1992.
- [52] S. Smith y J. Brady, «Real-Time Motion Segmentation and Shape Tracking,» *ASSET-2*, vol. 17, 1995.
- [53] D. Koller, N. Heinze y H. Nagel, «Algorithmic Characterization of Vehicle Trajectories from Image Sequence by Motion Verbs,» de *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 1991.
- [54] B. Heisele y W. Ritter, «Obstacle Detection Based on Color Blob Flow,» de *Proc. IEEE Intelligent Vehicle Symp.*, 1995.
- [55] K. Tarajano y L. Monterde, «Closed Contours Objects Detection Algorithm for static vehicle detection in FLIR images,» de *DCIS*, 2010.
- [56] J. Canny, «A Computational Approach To Edge Detection,» *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, nº 6, 1986.
- [57] Mathworks, «Matlab,» [En línea]. Available: <http://www.mathworks.es/products/matlab/>.
- [58] T. Brox, A. Bruhn, N. Papenbergh y J. Weickert, «High accuracy optical flow estimation based on a theory for warping,» de *European Conference on Computer Vision*, 2004.
- [59] P. Sand, Long-range video motion estimation using point trajectories, PhD thesis, Robotics, Vision, and Sensor Networks Group Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2006.



Upgrade of a Vehicle Detection System based on the CENTINELA

Karlos Tarajano¹
Technological Development Unit
TECNOBIT S.L.U.
Madrid
carlos.tarajano@tecnobit.es

Gustavo I. Marrero Callicó²
DSI – IUMA
Las Palmas de Gran Canaria University
Las Palmas de Gran Canaria
gustavo@iuma.ulpgc.es

Abstract—in this work, a novel vehicle detection algorithm has been designed in order to provide a cue to the CENTINELA's tracking system. Mixtures of techniques are used in this work as Optical Flow, Canny edge detection and others. The results outperform the original algorithm in terms of vehicle detection.

Keywords: vehicle detection, optical flow, canny, edge detection, movement vector

I. INTRODUCTION

TECNOBIT SLU is a lead Spanish Defense company. One of its main business areas is the optronics line. Tecnobit has developed a movable observation and surveillance system for tactical operations named CENTINELA. As it is shown in Figure. 1, CENTINELA is composed of several electro-optical sensor fitted in a protective case (a) and on a pan & tilt platform (b). The complete system is controlled via a portable console composed of a ruggedized PC (c), a joystick panel (d) and a connection box (e). CENTINELA's Tracking System is presently outdated and the need to upgrade this system was the original reason of this work.

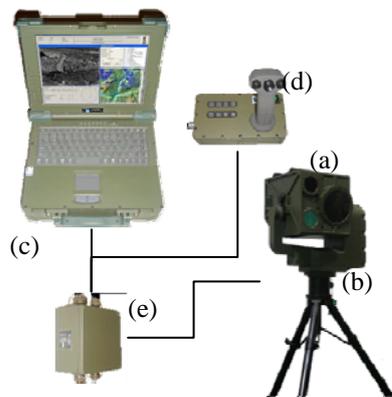


Figure 1. TECNOBIT's CENTINELA System

II. STATE OF THE ART

Tracking systems has a very wide range of application, and is widely studied in many different areas [1]. Tracking systems

can involve many different kinds of sensors [2]. In this work, the sensor used was a FLIR camera.

In an active tracking system [3], the process can be split in four stages: engage, prediction, aiming and evaluation. In the first stage, the object to track has to be locked in order to generate a signature. Then, a prediction is made and once it is finished, command the platform to aim at the objective. After that, a new search for the signature is done. The tracking-error is a new input for the predictor that starts the cycle again. This work studies and upgrades the detection stage.

The study of the state of the art on detection for tracking systems was focused on the techniques developed for artificial vision systems (visible or infrared cameras) in order to detect vehicles that are inside the field of vision of the camera.

The majority of the methods reported in the literature follow two basic steps:

- a) **Hypothesis Generation (HG)**, where the locations of possible vehicles in an image are hypothesized
- b) **Hypothesis Verification (HV)**, where tests are performed to verify the presence of vehicles in an image.

The second step HV is not considered due the future use of the system on military situations where unknown vehicles should be detected.

Inside of the HG step, a lot of different techniques had flourished. They can be classified into three main categories:

- a) Knowledge-Based methods
- b) Stereo Vision-Based methods
- c) Motion-Based methods

In the first category, the methods try to verify some assumption such as the vehicles should have some symmetry, vertical or horizontal edges, location of shadows or some kind of texture [4].

In the second one, there are two main kinds of methods: based on a disparity map or based in an Inverse Perspective Mapping [5].

In the last category, the methods compute more than one frame. Through the differences that exist between frames, the methods can detect the vehicles. Three sub-categories arise:

- a) Background Subtraction [6]

¹TECNOBIT S.L.U. - Technological Development Unit
www.tecnobit.es

²Integrated System Division – Applied Microelectronics Institute
Las Palmas de Gran Canaria University
www.iuma.ulpgc.es

- b) Temporal Difference [7]
- c) Optical Flow [8]

III. SOLUTION DEVELOPED

The current algorithm for vehicle detection that is found in CENTINELA is based on two stages: dynamic and static detection. In the first stage applies a temporal difference between two frames. If no vehicle is detected, the second stage is run. The static stage performs a closed edge detection.

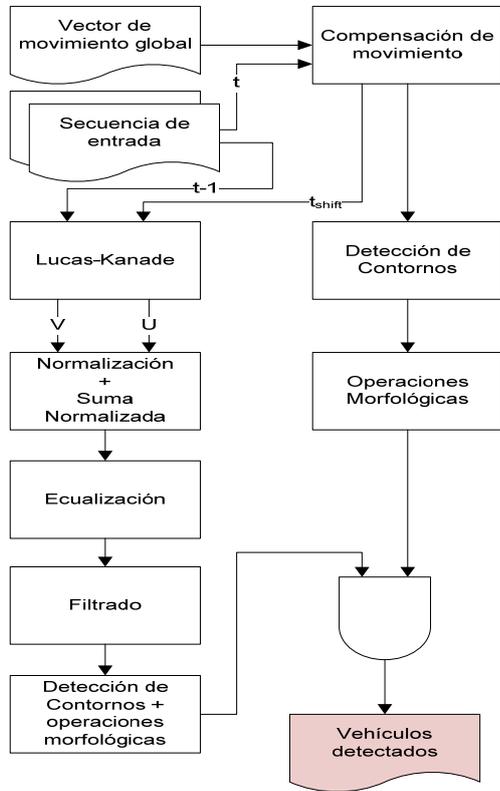


Figure 2. New Algorithm Flow Chart

This method does not obtain an outstanding performance for all possible situations that the system could be involved and so a new improved solution is needed.

The algorithm was designed in an iterative way. A first version of the algorithm was implemented and validated using all the test sequences. The results were studied and several modifications on the algorithm was applied at every iteration.. The result on all iterations was back-annotated.

The final version of the algorithm is shown in Figure 2. Two consecutive frames are needed, where the newest one is compensated by the global compensator. The Optical Flow estimator (Lucas-Kanade method) is executed. After normalizing the vectors and the sum of the two components, an equalization of the matrix is performed. The result is filtered with a Gaussian filter. In this stage, the Canny edge detector is applied to the filtered matrix. The resultant binary image is closed by a morphological close lines function and the blobs are labeled.

At the same time, a Canny edge detection is performed on the newest frame of the test sequence. Then the edges are closed by the same function and the blobs are also labeled. The blobs that are in both images are the detected vehicles.

IV. RESULTS

30 test sequences were used to test the performance of the different iterations of the algorithm. In Figure 3 the first frames of the first two sequences are shown.

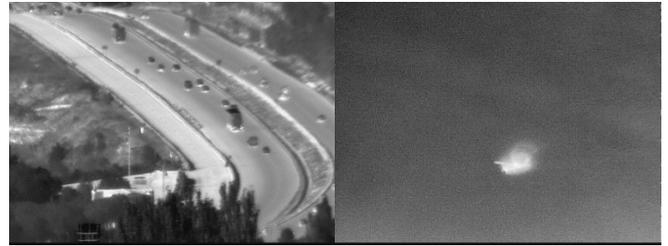


Figure 3. Two of the test sequences used

A total of 40 iterations were executed in order to reach the final version of the algorithm. In every iteration, the quality of the detection is measured.

The proposed algorithm improves the detection performance by a 20% against the original algorithm and gets a decrease of 50% of the false positives detected objects.

V. CONCLUSIONS

The initial CENTINELA's detection algorithm is outperformed by the designed algorithm. The novelty of the algorithm relies on the use of an edge detection operation over the sum of the intensity of the movement vectors estimated by the Lucas-Kanade operation.

Some improvement must be follow to continue this work. A new automatic parameters detector should be the first step. In second place, a better close contours algorithm.

REFERENCES

- [1] S. Blackman, R. Popoli, "Design and Analysis of Modern Tracking Systems" Artech House Publishers, 1999
- [2] Yaakov. Bar-Shalom, Xiao-Rong Li, "Multitarget-Multisensor Tracking: Principles and Techniques", YBS Publishing, 1995.
- [3] Ayala-Ramirez, V.; Parra, C.; Devy, M.; , "Active tracking based on Hausdorff matching," Pattern Recognition, 2000. Proceedings. 15th International Conference on , vol.4, no., pp.706-709 vol.4, 2000.
- [4] C. Goerick, N. Detlev, and M. Werner, "Artificial Neural Networks in Real-Time Car Detection and Tracking Applications," Pattern Recognition Letters, vol. 17, pp. 335-343, 1996.
- [5] H. Mallot, H. Bulthoff, J. Little, and S. Bohrer, "Inverse Perspective Mapping Simplifies Optical Flow Computation and Obstacle Detection," Biological Cybernetics, vol. 64, no. 3, pp. 177-185, 1991.
- [6] J. W. T. H. J. M. G. O. B. R. S. R. D. Koller, «Towards Robust Automatic Traffic Scene Analysis in Real-time,» de Proc ICPR, 1994.
- [7] A. Lipton, H. Fujiyoshi y R. Patil, «Moving Target Classification and tracking from real-time video,» de Proc. IEEE Workshop Applications of Computer Vision, 1998.
- [8] B. Lucas y T. Kanade, «An Iterative Image Registration Technique with an Application to Stereo Vision,» de Proc. Imaging Understanding Workshop, 1981.

Máster de Tecnologías de Telecomunicación

Trabajo Fin de Máster

Actualización y mejora del Algoritmo de Detección de Vehículos incorporados al Sistema Optrónico CENTINELA

Carlos Tarajano Beracoechea

Dr. Gustavo I. Marrero Callicó

Julio 2012

Resumen:

- El objetivo de este Trabajo Fin de Máster es estudiar y analizar las técnicas de visión artificial para la detección de vehículos más modernas y actualizar el algoritmo de detección de vehículos integrado en el sistema de seguimiento del CENTINELA de Tecnobit
- Se ha diseñado un **nuevo algoritmo de detección de vehículos** que combina alguna de las técnicas de detección de forma novedosa.

Diseño del Algoritmo

- Se combinan dos técnicas de detección de vehículos:
 - **Flujo óptico**
 - **Detección de contornos**
- La combinación de ambas permite detectar objetos artificiales sobre el fondo (detección de contornos) y si están en movimiento con respecto al fondo (flujo óptico).



Sistema de vigilancia CENTINELA

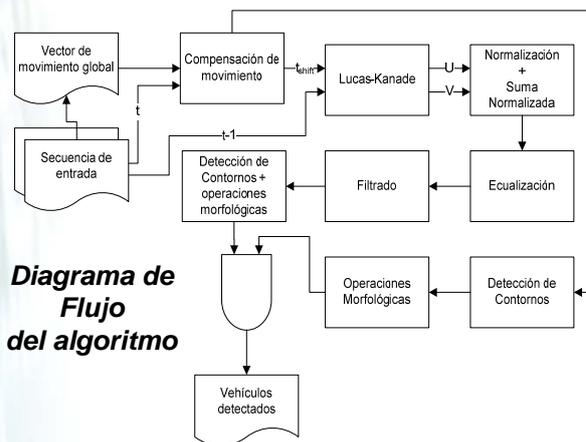


Diagrama de Flujo del algoritmo

- Como aportación **novedosa** se ha introducido el uso de un filtrado de Canny para detectar las zonas con movimiento sobre el flujo óptico de la secuencia.



Secuencia de test

Vectores de movimiento

Detección de movimiento

Optimización y Pruebas

- El diseño del algoritmo ha sido iterativo. Se han probado multitud de configuraciones y parámetros hasta llegar a la versión final
- Se ha evaluado el rendimiento de cada versión del algoritmo con 30 secuencias de prueba. Dichas secuencias son tanto del sensor infrarrojo como del sensor CCD y capturan gran variedad de situaciones reales.

Conclusiones

- Se ha diseñado un algoritmo que cumple con los requisitos solicitados
- Se estima que hay un amplio campo de mejora, sobre todo en:
 - Estimación del flujo óptico.
 - Estimación de los parámetros en función de la secuencia de entrada.
 - Cerrado morfológico de contornos.

