

Universitat Jaume I
Departamento de inform tica



Tesis doctoral

ESTIMACI3N Y SEGMENTACI3N DE MOVIMIENTO BASADO EN REGIONES Y SU APLICACI3N A TAREAS DE MONITORIZACI3N DE TRAFICO

Jorge Badenas Carpio

Dirigida por:
Dr. Filiberto Pla Ba n
Dr. Jos  Miguel Sanchiz Mart 

A MI ESPOSA, XENIA,
Y A MIS HIJAS, SELINA Y ALEXANDRA.

A MIS PADRES, JUAN Y CARMEN.

Resumen

La presente Tesis trata sobre la aplicación de técnicas de visión por ordenador al problema de la monitorización del tráfico. Con este objetivo, se han estudiado y desarrollado diversas técnicas, como la segmentación, estimación de movimiento, seguimiento, etc., que de manera entrelazada permiten obtener un sistema de control de tráfico basado en visión. Como principales tareas de dicho sistema podemos citar: medición del volumen del tráfico, seguimiento de trayectorias de los vehículos, y la detección de diversas situaciones anómalas, como vehículos parados, invasiones de carriles contrarios, etc.

El método de segmentación que se muestra en el Capítulo 2 divide cada imagen en regiones uniformes. Entre otras características, esta técnica está basada en la técnica de las *k-medias*, es *no supervisado*, se adapta automáticamente a imágenes con cualquier número de objetos y ofrece un excelente comportamiento frente a la principal de nuestras exigencias: no mezclar píxeles de diferentes objetos en una misma región.

Las regiones extraídas para cada imagen son tomadas por un proceso de segmentación basada en movimiento que une las que corresponden a la proyección de un mismo objeto. Se asume que dos regiones adyacentes pertenecen a un mismo objeto si sus parámetros de movimiento son coherentes. Dichos parámetros se estiman mediante un método basado en el cálculo de la *Diferencia de la Imagen Desplazada*, y que emplea estimadores robustos y una estrategia multiresolución.

El Capítulo 4 trata de hacer uso de una idea no del todo explotada hasta ahora: ir acumulando (integrar) la información obtenida tras el procesamiento de cada imagen de la secuencia, para hacer uso de ella en el tratamiento de cada nueva imagen. De esta forma las estimaciones son más acertadas (se dispone de mayor información) y la influencia de los errores tiende a ser menor. La integración de los parámetros de movimiento frecuentemente se encuentra en la bibliografía por medio de filtros como el de Kalman. En esta Tesis se plantea un tipo de integración temporal que a penas ha sido estudiado en la literatura: la *integración temporal morfológica*. Básicamente consiste en fundamentar la segmentación de cada nueva imagen en todas las segmentaciones de las imágenes previas y en la información obtenida a partir del proceso de seguimiento. Esto, además, permite introducir una nueva técnica llamada *segmentación virtual* que permite reducir el coste de computación.

Las técnicas propuestas se han verificado en diversas secuencias de tráfico, tanto urbano como de carretera, que se presentan a lo largo del documento. Los resultados muestran que el sistema implementado es capaz de trabajar en tiempo real, sin emplear

un hardware dedicado y de alto coste, y dando una adecuada respuesta a las tareas que se le demandan.

Agradecimientos

El trabajo presentado en esta Tesis es producto tanto del esfuerzo personal como de la colaboración de diversas personas e instituciones. Es por ello que me gustaría agradecer sus aportaciones tanto en el terreno científico como en el personal.

Diversas instituciones han hecho posible este trabajo por medio de proyectos o becas. Entre ellas debo citar especialmente a la Fundació *Caixa Castellò* quien me concedió la beca para realizar el *Master on Multimedia Technologies and Systems* en la Universidad de Surrey del Reino Unido. Fue a partir de la *Msc. Thesis* que allí realicé de donde surgieron las primeras ideas y trabajos que han fructificado en esta Tesis. Además, se ha contado con la ayuda financiera de la Comunidad Económica Europea a través del proyecto ESPRIT EP-21007 IOTA, del Ministerio de Educación y Ciencia con el proyecto CICYT TIC 98-0677-C02-01, y de la Generalitat Valenciana por medio de los proyectos GV97-TI-05-26 y GV97-TI-05-27.

A lo largo los años que ha durado la realización de esta Tesis he tenido el placer de colaborar con diversas personas de quienes siempre he conseguido aprender algo. Debo hacer mención especial de mis directores de Tesis, tanto el Dr. Filiberto Pla, quién me ha dirigido desde el principio, como el Dr. José Miguel Sanchiz que se incorporó en la fase de realización de los métodos de seguimiento e integración temporal. Ambos han aportado innumerables consejos y opiniones que han influido tanto en el resultado final como en mi manera de enfocar la investigación desde un punto de vista más científico.

Especialmente importante fue la ayuda del Dr. Mirosław Bober, mi supervisor durante el master que realicé en la Universidad de Surrey. De él aprendí gran parte de los conceptos básicos sobre análisis de movimiento que me han permitido desarrollar el trabajo posterior. Debo resaltar que una parte importante de las técnicas de estimación y segmentación de movimiento que aquí se presentan (Capítulo 3) fueron desarrolladas para la *Msc. Thesis* que él me dirigió en ese master.

Me siento afortunado de formar parte del *Grupo de Visión por Ordenador* de la Universitat Jaume I de Castellón, donde aparte de disponer de todos los medios técnicos necesarios, he tenido el placer de contar con el apoyo y amistad de todos sus miembros.

Finalmente, me gustaría dedicar esta Tesis a mis padres, Juan y Carmen, ya que de algún modo esta Tesis también es fruto de sus esfuerzos y sacrificios. Igualmente, quisiera dedicar esta Tesis a mi mujer, Xenia, y a mis hijas, Selena y Alexandra. A mi mujer, por su constante cariño, comprensión, paciencia, y por haber llenado mi vida de amor y alegría. A mis hijas, que nacieron casi al final de este trabajo, por

aumentar mi felicidad y esperanza en el mañana.

Castellón, 26 de enero de 2001.

Índice General

Resumen	v
Agradecimientos	vii
Notación y listado de símbolos	xvii
1 Introducción	1
1.1 Análisis de movimiento y control de tráfico	1
1.2 Dominio del problema	5
1.3 Breve revisión bibliográfica	8
1.4 Objetivos	13
1.5 Estructura del trabajo	15
2 Segmentación de la imagen de referencia	19
2.1 Partición de una imagen en regiones	19
2.1.1 Métodos basados en umbralización (<i>thresholding</i>)	21
2.1.2 Métodos basados en bordes	22
2.1.3 Métodos basados en regiones	22
2.1.4 Elección de un método de segmentación	25
2.2 Un algoritmo de segmentación de imágenes estáticas	29
2.2.1 Estructura del algoritmo	29
2.2.2 El algoritmo de clasificación de las <i>k-medias</i>	31
2.2.3 Algoritmo jerárquico de división de regiones	33
2.2.4 Unión de regiones	35
2.3 Resultados	37
2.4 Conclusiones	41
3 Análisis de movimiento	43
3.1 Introducción	43
3.1.1 Métodos basados en sustracción de imágenes	44
3.1.2 Métodos basados en el gradiente	47
3.1.3 Métodos basados en características	50
3.1.4 El método propuesto	54
3.1.5 Estructura del capítulo	56

3.2	Detección de movimiento	57
3.3	Reducción del número de regiones	61
3.4	Estimación de movimiento	64
3.4.1	El algoritmo básico de estimación de movimiento	64
3.4.2	La estrategia multiresolución	67
3.4.3	Estimación de movimiento con precisión subpíxel	67
3.4.4	Aplicación de estadísticos robustos	67
3.5	Segmentación	70
3.5.1	Segmentación de regiones con movimiento coherente	70
3.5.2	Segmentación con reestimación de movimiento	71
3.6	Resultados	72
3.7	Conclusiones	85
4	Seguimiento e integración temporal	87
4.1	Introducción	87
4.1.1	Breve revisión bibliográfica	88
4.1.2	Un método de seguimiento e integración temporal morfológica	93
4.1.3	Estructura del capítulo	95
4.2	El algoritmo	95
4.3	Seguimiento de regiones	97
4.3.1	Correspondencias entre regiones	97
4.3.2	Estimación de los parámetros de movimiento	99
4.4	Memoria temporal	102
4.5	Recuperando regiones perdidas	106
4.5.1	Planteamiento del problema	107
4.5.2	Búsqueda de la región R^k	107
4.5.3	Generación de la región R^l	108
4.6	Generación de una segmentación final	108
4.7	Resumen del algoritmo	108
4.8	Resultados	109
4.9	Conclusiones	114
5	Un sistema de monitorización de tráfico	117
5.1	Introducción. Monitorización de tráfico	118
5.2	Tareas de monitorización de tráfico	118
5.2.1	Medición del volumen del tráfico	119
5.2.2	Seguimiento de trayectorias	121
5.2.3	Detección de vehículos parados	122
5.2.4	Detección de eventos anómalos	124
5.3	Un sistema en tiempo real	124
5.3.1	Arquitectura del sistema	124
5.3.2	Segmentación virtual	128
5.4	Resultados	129
5.5	Conclusiones	142

6 Conclusiones	145
6.1 Conclusiones generales	145
6.2 Aportaciones	147
6.3 Líneas de trabajo futuro	148
6.4 Lista de publicaciones surgidas de esta Tesis	149
A Operadores morfológicos	151
B El filtro de Kalman	155
B.1 Introducción	155
B.2 Algoritmo del filtro de Kalman	156
Bibliografía	158
Índice de materias	178

Índice de Figuras

1.1	Escena de tráfico. Un vehículo acaba de adelantar a otro en una carretera de dos direcciones.	4
1.2	Flujo de información hacia el Centro de Control de Tráfico. Cada sistema de monitorización procesa las imágenes de tráfico capturadas por su sistema de adquisición. El Centro de Gestión de Tráfico recibe el estado del tráfico en cada punto monitorizado de la red vial.	7
1.3	Flujo de datos entre los procesos que componen el sistema de monitorización de tráfico.	16
2.1	Una imagen típica de vehículos circulando por una carretera	20
2.2	Diversas iteraciones del proceso de segmentación aplicando el algoritmo de Kottke y Sun.	27
2.3	Diagrama de flujo del método de segmentación	30
2.4	Árbol de representación del proceso de división de las imágenes	33
2.5	Las cuatro etapas del proceso de unión de regiones	35
2.6	Segmentación de una imagen de tráfico en una carretera.	38
2.7	Segmentación de una imagen de tráfico urbano.	39
2.8	Resultado final de la segmentación una vez unidas las regiones de tamaño mayor a η	40
3.1	Varias etapas del método de Steward et al. para detección de los carriles de una carretera.	47
3.2	Comportamiento del <i>error de la diferencia de la imagen desplazada (DFD)</i> en la vecindad del movimiento óptimo.	56
3.3	Pirámide de imágenes. Cada nivel corresponde a una resolución diferente.	57
3.4	Secuenciación de los procesos que forman parte del análisis de movimiento y que serán vistos en el presente capítulo.	57
3.5	Actualización y uso de la imagen referencia.	59
3.6	Elementos estructurales empleados en las dos operaciones morfológicas aplicadas sobre la imagen diferencia.	61
3.7	Pasos fundamentales del procedimiento de reducción de regiones.	62
3.8	Imagen diferencia obtenida a partir de la imagen de la figura 2.1 y la siguiente de su misma secuencia.	64
3.9	Algoritmo de estimación de movimiento.	66

3.10	Interpolación de los niveles de intensidad en la resolución subpíxel.	68
3.11	Fases de la segmentación de movimiento.	70
3.12	Varias etapas de la segmentación de una imagen de tráfico urbano.	73
3.13	Varias etapas de la segmentación de la imagen siguiente a la de la Figura 3.12.	75
3.14	Varias etapas de la segmentación de la imagen siguiente a la de la Figura 3.13.	76
3.15	Segmentación de una escena grabada en la Durlacher-Tor de Karlsruhe. Estas imágenes fueron usadas en [Koller et al. 1993]. (KOGS/IAKS Universität Karlsruhe)	77
3.16	Segmentación de dos imágenes de la secuencia grabada en la N-340.	80
3.17	Resultados de la segmentación de la imagen de la subfigura 3.16.a utilizando varias combinaciones de parámetros.	82
3.18	Cuatro imágenes de una secuencia de autopista con sus respectivas segmentaciones.	84
4.1	Algoritmo de seguimiento e integración temporal morfológica.	96
4.2	Pasos que componen el seguimiento de regiones.	97
4.3	Se establece una correspondencia entre las regiones de la <i>lista maestra</i> , master list, y las regiones extraídas de la imagen actual.	98
4.4	Algoritmo iterativo de correspondencia de regiones.	100
4.5	Lista maestra de regiones.	103
4.6	Inicialización de la lista maestra de regiones.	104
4.7	Actualización de la lista maestra de regiones.	105
4.8	Seguimiento de vehículos en una escena de tráfico	110
4.9	Secuencia de vehículos en una carretera. Primera columna: imagen original. Segunda columna: segmentación inicial. Tercera columna: segmentación final de la imagen	111
4.10	Varias segmentaciones de una secuencia de tráfico en ciudad. Primera columna: imagen original. Segunda columna: segmentación inicial. Tercera columna: segmentación final de la imagen.	113
4.11	Secuencia de vehículos en un cruce de una avenida Primera columna: imagen original. Segunda columna: segmentación final de la imagen	114
5.1	Flujo de informaciones del sistema. Las cajas representan informaciones que se obtienen en alguna fase del procesamiento, mientras que las flechas indican que una información (la apuntada) se genera a partir de la otra (la que se encuentra en el origen de la flecha).	120
5.2	Dos casos típicos donde no solo interesa saber el volumen total del tráfico, sino también el de cada carril o calle.	121
5.3	Trayectoria de un coche en una curva.	122
5.4	Secuencia en la que un coche se detiene en el arcén de una carretera.	123
5.5	Arquitectura del sistema de monitorización de tráfico. Programas que interactúan para llevar a cabo el análisis de movimiento y las tareas de monitorización de tráfico.	125

5.6	Distribución de los distintos componentes del sistema de monitorización de tráfico entre los programas implementados.	127
5.7	La comunicación entre procesos se realiza mediante secciones críticas.	128
5.8	Diez imágenes de una secuencia urbana con sus correspondientes segmentaciones.	131
5.9	Una secuencia de vehículos en un cruce de una ciudad. Las subfiguras de la derecha corresponden a las segmentaciones de las imágenes de las subfiguras de la izquierda.	132
5.10	Secuencia de tráfico en carretera. Vehículos de distintos tipos circulan sobre una carretera de dos sentidos.	135
5.11	Secuencia de coches deteniéndose ante un semáforo. Primera columna: imagen original. Segunda columna: segmentación inicial. Tercera columna: segmentación final de la imagen.	137
5.12	Secuencia de tráfico urbano un cruce de una avenida.	138
5.13	Vehículos detectados en cada una de las direcciones de salida.	139
5.14	Secuencia de tráfico en una autopista.	141
A.1	Operaciones morfológicas de erosión y dilatación.	152
B.1	Ciclo del algoritmo de Kalman.	157

Notación y listado de símbolos

A la hora de expresar los distintos símbolos y fórmulas matemáticas se han empleado las siguientes notaciones:

- Las matrices se representan mediante letras mayúsculas en negrita: **A**, **B**, etc.
- Los vectores se representan mediante letras minúsculas en negrita: **v**, **w**, etc.
- Las estructuras de datos, tales como conjuntos, regiones, imágenes, etc., se representan con letras mayúsculas: *R*, *I*, *D*, etc.

A continuación se presenta una lista de las variables y símbolos más importantes que aparecen en los distintos capítulos:

Símbolos y variables generales

R^k	Región <i>k</i> de la imagen.
$R_{i,j}^k$	Píxel (<i>i,j</i>) de la región R^j .
t	Instante de tiempo.
I_t	Imagen obtenida en el instante t .
$I_t(i, j)$	Píxel (<i>i,j</i>) de la imagen obtenida en el instante t .

Símbolos y variables del Capítulo 2

\mathbf{W}^j	Matriz de pesos de la región R^j .
\mathbf{p}^i	Vector de características del píxel i .
\mathbf{m}^j	Vector del centro de masas de la región R^j .
m_x^j, m_y^j, m_g^j	Medias de las coordenadas x e y y de la intensidad (g) en la región R^j .
$\sigma_x^j, \sigma_y^j, \sigma_g^j$	Desviaciones típicas de las coordenadas x e y y de la intensidad (g) en la región R^j .
w_x^j, w_y^j, w_g^j	Pesos de las coordenadas x e y y de la intensidad (g) que forman la diagonal principal de la matriz \mathbf{W}^j .
N_{R^j}	Número de píxeles de la región R^j .
$T_{\sigma_{max}}$	Umbral de máxima desviación típica.
η	Máximo tamaño de una región.
μ	Mínimo tamaño de una región.

Símbolos y variables del Capítulo 3

\mathbf{M}_t	Matriz de actualización de la imagen referencia en el instante t .
B_t	Imagen referencia en el instante t .
D_t	Imagen diferencia en el instante t .
ψ_1	Constante de actualización de la imagen de referencia.
γ	Umbral para calcular los píxeles diferencia.
l_{cad}	Longitud mínima de una cadena de píxeles diferencia.
dx	Desplazamiento en la coordenada x .
dy	Desplazamiento en la coordenada y .
$DFD_i(dx, dy)$	Valor absoluto de la diferencia de la imagen desplazada con el movimiento (dx, dy) para la región R^i .
α, λ	Constantes del kernel robusto.
Θ, κ	Umbrales para la segmentación.

Símbolos y variables del Capítulo 4

d^{mn}	Distancia entre los centroides de las regiones R^m y R^n .
\mathbf{m}^m	Vector centroide de la región R^m .
\mathbf{W}	Matriz de pesos para la correspondencia entre regiones.
$w_x, w_y, w_g, w_{v_x}, w_{v_y}$	Pesos de las coordenadas x e y , intensidad y velocidad en las coordenadas x y y .
$\mathbf{e}_k, \mathbf{v}_k, \mathbf{a}_k, \mathbf{c}$	Posición, velocidad, aceleración y variación de la aceleración en el instante k .
$\mathbf{A}_k, \mathbf{H}_k$	Matrices del filtro del Kalman.
M^m	Matriz de memoria temporal de la región R^m .
$M_{ij}^m.v, M_{ij}^m.o$	Valores v y o del píxel (i, j) , de la matriz de memoria temporal de la región R^m .
O_{max}	Máximo valor para un valor $M_{ij}^m.o$.
P_{max}	Constante para declarar regiones como <i>permanentes</i> .
A_s	Acumulador de una región R^s .
$P(R^m (i, j))$	Probabilidad de que un píxel pertenezca a una región R^m .

Capítulo 1

Introducción

Contenido

1.1	Análisis de movimiento y control de tráfico	1
1.2	Dominio del problema	5
1.3	Breve revisión bibliográfica	8
1.4	Objetivos	13
1.5	Estructura del trabajo	15

1.1 Análisis de movimiento y control de tráfico

Tal y como afirman diversos autores [Torr y Murray, 1993, Singh, 1992], la percepción de movimiento es una función importante en el sistema visual humano y animal, que juega un papel primordial en tareas tales como la percepción de profundidad o la detección y segmentación de objetos.

“Motion is a powerful cue for image and scene segmentation in the human visual system as evidenced by the ease with which we see otherwise perfectly camouflaged creatures as soon as they move, and by the strong cohesion perceived when even disparate parts of the image move in a way that could be interpreted in terms of a rigid motion in the scene ...”
[Torr y Murray, 1993]

Durante las últimas décadas, se ha prestado una importante atención al estudio de la percepción del movimiento. Este estudio ha sido acometido por investigadores de diversas disciplinas como psicología, fisiología y visión por ordenador. Dicho estudio tenía una doble meta: entender cómo los humanos perciben el movimiento y formular una teoría computacional para la percepción del movimiento que pudiera ser usada en un autómata.

El análisis de movimiento consiste, básicamente, en inferir conocimiento a partir de una sucesión de imágenes. Para ello se comparan entre sí las imágenes de la secuencia, analizando similitudes y diferencias. A partir de este análisis y desde un punto de vista práctico, existen tres tipos de problemas que pueden ser resueltos [Fleet, 1992, Murray y Buxton, 1989, Sonka *et al.*, 1993, Mitiche y Bouthemy, 1996]:

1. *Detección de movimiento.* Consiste en detectar si se ha producido algún tipo de movimiento.
2. *Detección y localización de objetos en movimiento.* Este tipo de problemas es más complejo que el anterior, ya que no sólo consiste en saber si algo se ha movido, sino en saber la forma que ello tiene en el plano de la imagen. Además, puede requerir estimar la trayectoria de los objetos móviles y predecir su posición futura. Esto conlleva el empleo de técnicas de estimación de movimiento, segmentación y seguimiento. Se entiende por *estimación de movimiento* en el plano de la imagen, el cálculo de los vectores desplazamiento de los pixels de la imagen. Esta estimación puede ser densa, cuando se calcula para todos los puntos de la imagen, o dispersa, cuando sólo se hace para ciertos puntos o entidades de la imagen. El movimiento 2D, en el plano de la imagen, es una proyección del movimiento 3D experimentado por los objetos de la escena. La información de movimiento puede ser usada para determinar el número de objetos móviles y sus velocidades. Para ello los puntos que se mueven con parámetros de movimiento coherentes son agrupados en un proceso que recibe el nombre de *segmentación de movimiento*. Se llama *seguimiento* o *tracking* al proceso que relaciona las apariciones de una misma entidad u objeto en las distintas imágenes de la secuencia, y que por tanto permite seguir su trayectoria.
3. *Derivación de propiedades 3D de los objetos de la escena.* La profundidad relativa y la velocidad en la escena pueden ser relacionadas cuantitativamente con la posición y la velocidad en el plano de la imagen [Ballard y Brown, 1982, Aggarwal y Nandhakumar, 1988, Horn, 1986]. Así, en este tipo de problemas se analizan las proyecciones 2D de los objetos que han sido adquiridas en diferentes instantes de tiempo y a partir de ellas se obtienen características 3D del objeto, tales como su posición, trayectoria o forma en el mundo real.

El término *flujo óptico* fue originalmente acuñado por Gibson [Gibson, 1950] para definir el campo de vectores velocidad que puede ser asociado a la proyección de cada punto visible de la escena. Este flujo óptico puede ser estimado a partir de los patrones de intensidad espacio-temporal registrados por el sistema de visión. “*La estimación se realiza mediante la resolución de un conjunto de restricciones derivadas a partir de la hipótesis hecha sobre la naturaleza del patrón de intensidad observado, y sobre el flujo óptico*” [Mitiche y Bouthemy, 1996].

Las técnicas desarrolladas para análisis de movimiento han sido utilizadas en una amplia variedad de aplicaciones¹, de entre las que podemos destacar:

- *Codificación*. El análisis de movimiento permite en aplicaciones de compresión de vídeo predecir la siguiente imagen de la secuencia. De este modo, se consigue ahorrar espacio.
- *Robótica*. En este área el análisis de movimiento se utiliza para la navegación de los robots dotados de un sistema de visión, permitiéndoles detectar obstáculos, conocer su posición, evitar objetos móviles, etc.
- *Meteorología*. El seguimiento de las nubes permite realizar mapas de vientos.
- *Aplicaciones militares*. Búsqueda de objetivos.
- *Biomedicina*. Análisis del movimiento del corazón, comportamiento de microorganismos y estudio del movimiento humano en el deporte o la rehabilitación.
- *Interfaces y realidad virtual*. En estos campos se está investigando la interacción a través de la interpretación de las expresiones faciales o los gestos de las manos.

Una de las aplicaciones donde el análisis de movimiento parece tener un mayor futuro es la monitorización de tráfico. En los últimos años, se han presentado muchos trabajos en este campo [Giachetti *et al.*, 1995, Beymer *et al.*, 1997, Tan *et al.*, 1998, Koller *et al.*, 1994a, Waterfall y Dickinson, 1984, Frank *et al.*, 1996, Zhu *et al.*, 2000, Haag y Nagel, 2000], los cuales han abordado la solución de diversos problemas: detección de vehículos, su seguimiento, su clasificación, etc..

Los medios que han venido siendo utilizados para la monitorización de tráfico, y que siguen en uso, son:

- Los *detectores basados en microondas*, los cuales permiten medir velocidades individuales de vehículos, pero son incapaces de advertir vehículos parados o con movimiento lento. Además, y dado que deben situarse a un lado de la carretera, su rango de acción se ve limitado por el tamaño y forma de los vehículos.
- Los *detectores de bucle inductivo*. Consisten, básicamente, de un cable enterrado bajo la superficie de la calzada con una corriente que lo recorre. Cuando un vehículo pasa sobre él, éste induce una sobrecarga que al ser registrada permite contabilizar el vehículo. Este tipo de sistemas tan sólo permiten contar vehículos, y deben ser instalados en cada carril para abarcar toda la calzada. Como desventajas principales, además de su utilidad limitada, cabe decir que su instalación es costosa y requiere perturbar la circulación del tráfico.

¹Nagel en [Nagel, 1981] presenta una revisión exhaustiva sobre las aplicaciones del análisis de movimiento. En [Nagel, 1983a] y [Thévenaz, 1990] pueden encontrarse otras amplias revisiones, aunque bastante más escuetas.



Figura 1.1: Escena de tráfico. Un vehículo acaba de adelantar a otro en una carretera de dos direcciones.

Frente a los anteriores sistemas de detección, los sistemas basados en visión por ordenador ofrecen varias ventajas. Primeramente, requieren un menor coste, con una instalación menos compleja [Broggi y Dickmanns, 2000]. Y en segundo lugar, un único dispositivo permite abarcar un mayor espacio de carretera, proporcionando una estimación de más parámetros.

Entre las tareas que permite un sistema basado en visión por ordenador, podemos señalar: recuento de vehículos y velocidades, clasificación de vehículos, estimación de la velocidad media del tráfico para cada carril o sentido, detección de cambios de carril o adelantamientos, detección de atascos, medición de la longitud de las colas, etc..

El problema del análisis de movimiento es difícil por sí mismo, pero en el contexto de su aplicación a la monitorización del tráfico, estas dificultades se hacen más manifiestas. La primera dificultad estriba en el hecho de que la monitorización debe ser realizada en tiempo real, y a ser posible con un hardware no excesivamente costoso, del tipo de un ordenador personal. La restricción del tiempo real viene justificada por el hecho de que la mayoría de los parámetros que pueden extraerse pierden parte de su utilidad cuando se difiere en el tiempo su comunicación, ya que en último término deberían servir para la toma de decisiones por parte de un sistema central de gestión de tráfico. La restricción acerca del coste del hardware se justifica por el hecho del gran número de puntos de la red vial que sería deseable controlar. Teóricamente, y dado un único presupuesto, a mayor coste del sistema, menor número de puntos podrán ser monitorizados.

Hasta la actualidad se han desarrollado un gran número de técnicas para el análisis de movimiento, pero en pocas de ellas se suele tener en cuenta la posibilidad de que funcionen en tiempo real [Liu *et al.*, 1998, Welch y Wood, 1993, Isard y Blake, 1996, Siyal *et al.*, 2000]. No obstante, cuando esto se ha hecho, ha sido usando un hardware dedicado y de alto coste.

La existencia de numerosos objetos móviles, vehículos, en el campo de visión, plantea una dificultad doble. Por un lado, esto implica la posibilidad de que unos vehículos oculten total o parcialmente a otros. Además, estas oclusiones pueden ser progresivas, es decir, que un vehículo que originalmente era totalmente visible, acabe total o parcialmente oculto, y viceversa. Por otro lado, a mayor número de objetos móviles, mayor coste computacional. La Figura 1.1 muestra uno de estos casos. En él, un vehículo oculta parcialmente a un segundo que acaba de adelantar. Sin embargo, en esta misma secuencia, la situación era la contraria antes de producirse el adelantamiento.

El tamaño de los vehículos es otro problema, pues si deseamos que la imagen abarque un gran espacio de carretera, su tamaño será inevitablemente pequeño, unos cientos de píxeles a lo sumo. También se debe tener en cuenta que tanto los vehículos como la calzada carecen de textura. Es bien sabido que la falta de textura es una de las mayores dificultades con que se puede encontrar el análisis de movimiento, y principalmente los métodos basados en flujo óptico [Horn y Schunck, 1981, Nagel, 1983b]. La falta de textura acentúa el conocido *problema de la apertura*, dificultando el uso de métodos locales de estimación de movimiento. Por último, otra importante complicación, consubstancial a todas las escenas de exteriores, es la existencia de cambios en la iluminación. Estos cambios serán, por lo general, graduales, aunque en ciertas ocasiones pueden ser bruscos.

Para la realización de la investigación presentada en este trabajo se han usado diversas secuencias de imágenes, siendo la mayoría de ellas correspondientes a escenas de tráfico, tanto urbanas, como de carretera. En ambos tipos de escenas se ha empleado una única cámara situada sobre una posición elevada y preferentemente enfocando en la dirección paralela a la trayectoria de los vehículos. De esta forma es posible abarcar un mayor tramo de carretera y por lo tanto monitorizar un mayor número de vehículos. En las secuencias de carretera la cámara ha sido situada sobre un puente, mientras que en las urbanas se ha usado la terraza de algún edificio elevado.

1.2 Dominio del problema

El trabajo presentado en esta tesis ha sido principalmente realizado en la *Universitat Jaume I* de Castellón, dentro del grupo de Investigación de *Visión por Ordenador*, al que pertenece el autor. Sin embargo, las primeras investigaciones fueron realizadas como *Msc. Thesis* [Badenas, 1996] del *Master on Multimedia Technology and Systems* que el autor realizó durante el curso 1995-1996 en la Universidad de Surrey del Reino Unido. Aquella *Msc. Thesis* fue supervisada por el Doctor Miroslaw Bober del *Department of Electronic and Electrical Engineering* y miembro del *Vision and Speech Signal Processing Group* de la citada universidad.

Así mismo, parte de las investigaciones incluidas en esta tesis surgieron de la colaboración en diversos proyectos de investigación con financiación pública. Estos proyectos son:

- *Estimación de movimiento e interpretación de secuencias de imágenes. Aplicaciones al control de tráfico.* CICYT, Ministerio de Educación y Ciencia, 1996-1998.
- *IOTA. Intelligent object tracking and analysis.* ESPRIT, European Union, 1996-1998.
- *Análisis de Imagen para la Asistencia a la Conducción en Tráfico no Urbano (GV97-TI-05-26).* Conselleria de Cultura, Educació i Ciència. Generalitat Valenciana, 1998-2000.
- *Seguimiento de Objetos en Movimiento Mediante Visión Activa Foveal. Aplicaciones al Control de Vigilancia (GV97-TI-05-27).* Conselleria de Educació Cultura i Ciència. Generalitat Valenciana, 1998-2000.
- *Visión Activa basada en la Representación Log-Polar para la Monitorización y Seguimiento de Objetivos Móviles (TIC98-0677-C02-01).* CICYT Ministerio de Educación y Ciencia, 1998-2001.

La presente tesis aborda el problema del diseño de un sistema de monitorización de tráfico basado en visión por ordenador. Dicho sistema consiste, físicamente, de una cámara conectada a un ordenador, enfocando a un tramo de carretera o calle. El ordenador debe disponer de una aplicación software para procesamiento de imágenes que permita extraer información acerca del tráfico en la zona enfocada. Esa información debe ser de dos tipos: parámetros y eventos.

- **Parámetros.** Miden la actividad del tráfico. Por ejemplo, el número de vehículos por minuto o la longitud de una cola en una intersección.
- **Eventos.** Son incidencias, como por ejemplo, un atasco, un accidente o un vehículo circulando en sentido contrario.

Idealmente, el sistema de monitorización debería estar instalado en diversos puntos de la red vial, generando de esta forma información detallada, y al instante, de la situación de los puntos conflictivos de la red. Dicha información sería enviada a un Centro de Control de Tráfico, el cual tendría tres cometidos principales. En primer lugar, almacenar la información estadística para en el futuro, gracias a ella, mejorar la red vial. En segundo lugar, prever problemas a corto plazo en el flujo de tráfico, y plantear soluciones. Por ejemplo, ante un incremento continuo del tráfico en ciertas vías, cambiar la prioridad de los semáforos, o generar rutas alternativas que eviten una previsible congestión. Y en tercer lugar, recoger los diversos eventos que puedan enviarle los sistemas de monitorización, actuando en consecuencia ante cada uno de ellos. La Figura 1.2 muestra este flujo de información entre los sistemas de monitorización de tráfico y el Centro de Control de Tráfico.

El trabajo presentado en esta tesis se dirige hacia el diseño de un sistema de monitorización de tráfico que analice las imágenes capturadas por la cámara y sea capaz de generar los eventos y parámetros que reflejen el estado del mismo. Para ello,

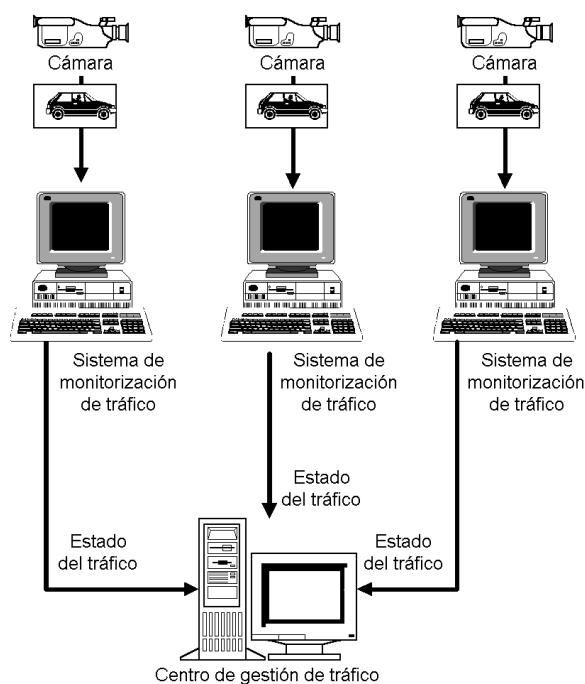


Figura 1.2: Flujo de información hacia el Centro de Control de Tráfico. Cada sistema de monitorización procesa las imágenes de tráfico capturadas por su sistema de adquisición. El Centro de Control de Tráfico recibe el estado del tráfico en cada punto monitorizado de la red vial.

y como no podría ser de otro modo, el protagonista principal será el vehículo. Así, el sistema desarrollado debe ser capaz de detectar su presencia en una escena, y además seguir su comportamiento durante todo el tiempo que permanezca dentro de ella. Para conseguir esto se utilizarán diversas técnicas de procesamiento de imágenes, entre las que cabe destacar: segmentación estática, estimación de movimiento, segmentación de movimiento y seguimiento.

La segmentación estática permite agrupar los píxeles en regiones. La técnica empleada debe garantizar que cada región corresponda, si no a un objeto completo, sí al menos a una parte de la imagen perteneciente a un solo objeto. De esta forma, las regiones irán siendo agrupadas en la segmentación de movimiento para formar objetos.

El método de estimación de movimiento calculará el movimiento que cada región ha experimentado en el plano de la imagen durante el tiempo transcurrido entre dos imágenes consecutivas de una secuencia. Con los parámetros estimados, la técnica de segmentación de movimiento unirá aquellas regiones que se desplazan con un movimiento coherente, formando de esta forma objetos.

Por su parte el método de seguimiento será el encargado de relacionar las distintas apariciones de un mismo vehículo en cada una de las imágenes de la secuencia. Es decir, dado un objeto segmentado en una imagen, y dado un conjunto de objetos

segmentados en la siguiente imagen, este método debe ser capaz de decidir qué objeto del conjunto corresponde al primer objeto.

Este proceso de seguimiento es el que permite saber realmente cuántos vehículos han circulado y cuál ha sido su comportamiento. Por lo tanto, es en él en quien, en última instancia, debe apoyarse el sistema de monitorización de tráfico para generar toda la información que le es requerida. De este modo, conociendo el comportamiento individual de cada vehículo, será sencillo generar parámetros globales o eventos.

1.3 Breve revisión bibliográfica

Desde principios de los años ochenta, el interés por la utilización de las técnicas de procesamiento de imágenes en la monitorización de tráfico ha sido creciente. Esto se ha visto reflejado en la aparición de un importante volumen de trabajos que desarrollan nuevos métodos o adaptan a este tipo de entornos otros ya existentes.

Uno de los primeros trabajos fue el aportado por Waterfall y Dickinson [Waterfall y Dickinson, 1984], el cual describe un método para análisis automático del tráfico en carreteras. El sistema estaba diseñado para extraer medidas estadísticas que ayudasen a arquitectos e ingenieros a mejorar los trazados de las carreteras. El método no estaba pensado para trabajar en tiempo real, sino para grabar las imágenes y posteriormente procesarlas en un laboratorio. El análisis de movimiento estaba basado en una simple substracción de imágenes. Este procedimiento tan simple no es el más adecuado, ya que aporta poca información y además es muy sensible al ruido de la imagen. No obstante, mostró la utilidad de la visión por ordenador en este, por aquel entonces, nuevo ámbito de aplicación.

En la actualidad existen varios grupos de investigación notablemente especializados en el control de tráfico. Uno de estos grupos es el de la Universidad de Reading, Reino Unido. Como denominador común de todos sus trabajos se encuentra el uso de modelos alámbricos tridimensionales [Ferryman *et al.*, 1995, Worrall *et al.*, 1991, Tan *et al.*, 1991, Tan *et al.*, 1994, Tan *et al.*, 1998, Worrall *et al.*, 1994]. Estos modelos alámbricos consisten de líneas tridimensionales, las cuales se hacen corresponder con las líneas bidimensionales que son extraídas de la imagen. De esta forma, se adaptan los modelos a las proyecciones de los vehículos. Sin embargo, esta adaptación constituye la tarea más difícil del proceso, ya que para realizarla es preciso estimar previamente la posición, la orientación y el tipo de vehículo (coche, camión, furgoneta, etc.). Para facilitar su realización, se asume que tanto la escala de los vehículos como la geometría de la escena son conocidas. El conocimiento de la escena consiste en una calibración previa de la cámara y de una estimación de la orientación de la carretera, que permite reducir los grados de libertad del modelo, de seis grados a sólo tres. Esta reducción se basa, obviamente, en que las ruedas del vehículo deben descansar sobre la carretera, y por lo tanto, sólo cabe esperar rotaciones alrededor del eje perpendicular al suelo. Una vez fijado el modelo, el proceso de seguimiento queda facilitado por el conocimiento de la orientación y el propio modelo. Mediante el uso de un filtro

de Kalman ², el cual modela el movimiento del vehículo, se predice la posición en la siguiente imagen de la secuencia. Con esta predicción, la correspondencia entre líneas 2D y 3D se simplifica, pues se conoce su previsible posición.

Un enfoque similar del problema puede encontrarse en [Koller *et al.*, 1993] y [Koller, 1993], donde el modelo de un coche genérico es proyectado sobre la imagen, haciendo corresponder, de nuevo, los segmentos del modelo con los extraídos de la imagen. Sin embargo, en este método, antes de establecer la correspondencia entre el modelo y los segmentos de la imagen, se realiza una segmentación de movimiento, asumiendo que el modelo de movimiento es eminentemente translacional. En esta segmentación, basada en una correspondencia de características entre imágenes consecutivas, se agrupan aquellos segmentos de la imagen que exhiben movimiento coherente. De esta forma se reduce el número de posibles correspondencias entre segmentos del modelo y de la imagen, evitando interferencias de líneas que corresponden al fondo. En [Koller *et al.*, 1993] se supone conocido el tipo de modelo, y por tanto, la clase de modelo que le corresponde a cada vehículo de la imagen, mientras que en [Koller, 1993], aunque se realiza una selección del modelo, se requiere que el vehículo real sea muy aproximado a alguno de los modelos disponibles.

Otros métodos que también utilizan modelos tridimensionales pueden encontrarse en [Frank *et al.*, 1996] y [Gardner y Lawton, 1996]. Gardner y Lawton [Gardner y Lawton, 1996] introducen un *modelo de gravedad* que permite orientar la adaptación del modelo a la imagen y, a la vez, restringir sus movimientos, mientras que en [Frank *et al.*, 1996] se modela toda la escena, incluyendo objetos estáticos, para abordar el problema de la oclusión parcial de vehículos por parte de elementos de la escena.

El análisis de movimiento basado en modelos tridimensionales aporta una gran cantidad de información al sistema, ya que define completamente, si no toda la escena, sí al menos cada uno de los objetos móviles. Estos modelos realizan una clasificación de los vehículos, y permiten predecir con gran certeza el movimiento de los mismos. Además, una vez inicializados, y dada esta calidad de la predicción, el proceso de seguimiento se realiza fácilmente, resolviendo correctamente las agrupaciones u oclusiones de vehículos. Sin embargo, estos métodos plantean varios problemas que reducen significativamente su utilidad. Primeramente, requieren que se disponga de gran número de modelos, y que los objetos móviles que aparezcan en la escena se puedan adaptar, en menor o mayor medida, a ellos. Parece algo irreal el pretender disponer de modelos para cualquier vehículo que exista o pudiera existir en el futuro. Por ejemplo, téngase en cuenta que el modelo usado para un utilitario pequeño sin maletero trasero será diferente del usado por un coche con maletero, que a su vez será distinto de un coche ranchera. Evidentemente, esto se seguiría complicando si pensamos que el coche ranchera pudiera ser una ambulancia con una sirena, o que cualquiera de los anteriores modelos podría llevar encima una baca con equipaje o una bicicleta.

²Este mismo grupo propone un nuevo filtro en [Maybank *et al.*, 1996] que implícitamente modela el comportamiento del vehículo mediante procesos estocásticos.

Probablemente, el mayor inconveniente que plantean este tipo de técnicas es el de su tremendo coste computacional. En realidad, este elevado coste, aunque pueda ser necesario en algún tipo específico de aplicaciones, no puede justificarse por ninguna de las tareas usuales que se le requieren a un sistema de control de tráfico.

Otro problema importante en este tipo de métodos es la elección del modelo. Dado un conjunto de características extraídas de la imagen, hay que determinar qué modelo es el que mejor se adapta. La elección de modelo no es nada trivial, más aún si en la escena, como a menudo sucede en las imágenes de tráfico, tenemos objetos situados a distintas distancias de la cámara. Normalmente, este tipo de métodos parten de la premisa de la existencia de poca perspectiva, con lo que se elude el problema de la escala. Sin embargo, éste no es el único problema que plantea la elección de método. Algo más importante es la agrupación de vehículos. Ya hemos comentado, que una vez inicializado el seguimiento, y cuando ya se ha elegido el modelo para cada vehículo, éste puede separar correctamente los agrupamientos de coches. Sin embargo, la inicialización no es tan fácil cuando se deben asignar modelos a un grupo de vehículos que se ocultan entre sí. Por desgracia, éste es un problema de difícil solución, pues es común que los coches entren agrupados en una escena de tráfico. Además, este problema se repite también para los modelos bidimensionales, como indican autores como Beymer y Malik [Beymer *et al.*, 1997], que tras usar esta clase de modelos en varios de sus trabajos, en sus más recientes investigaciones se han decantado por el uso de características más sencillas.

“However, the inability to segment vehicles that are partially occluded remains. If one could initialize a separate contour for each vehicle, then one could track even in the presence of partial occlusion. However, initialization is the difficult part of the problem!...” [Beymer *et al.*, 1997]

El uso de modelos bidimensionales reduce drásticamente el coste computacional del procedimiento. En [Dubuisson *et al.*, 1996] se modela un vehículo genérico deformable mediante un polígono de N vértices. Este modelo deformable se adapta a la silueta lateral del vehículo, que es el ángulo de visión desde el cual más fácilmente se puede clasificar el tipo de vehículo (coche, furgoneta, etc.). El método emplea dos cámaras que se colocan lateralmente. Una situada en un punto de entrada de la carretera, y otra situada en un punto de salida, de modo que los automóviles que atraviesan la carretera obligatoriamente deben pasar frente a ambas cámaras. Cuando un vehículo pasa por el punto de entrada, éste es clasificado y memorizado hasta que pasa por el punto de salida. De esta forma se calcula el tiempo necesitado por cada vehículo para recorrer un determinado camino. El conjunto de estas mediciones de tiempos permiten estimar la velocidad media del tráfico, y por lo tanto, medir su fluidez.

El uso de imágenes laterales ofrece algunas ventajas de cara al desarrollo del sistema. La principal de todas ellas es que el movimiento de los vehículos sigue un modelo completamente translacional. Este modelo se estima con mayor rapidez, y además con mejor precisión. Por contra, la principal desventaja de este posicionamiento de las cámaras es la reducida visión que se tiene de la carretera. En primer lugar la cámara debe estar muy próxima a la carretera, para que los vehículos ocupen una

parte significativa de la imagen (20%-30%), y así los modelos se ajusten correctamente. En segundo lugar, el campo de acción del sistema se reduce a un único carril, al más próximo a la cámara. El resto de carriles no podrán ser controlados, pues los automóviles del carril más cercano ocultarán a los del resto.

Dado que el punto de vista frontal de la cámara con respecto a la carretera es el más frecuente, la mayoría de trabajos que utilizan modelos deformables emplean aproximaciones convexas, *blobs*, a la forma de los vehículos. Esto se realiza mediante contornos activos, también conocidos como *snakes* [Kass *et al.*, 1988], [Bartels *et al.*, 1987], [Blake y Isard, 1998]. En [Koller *et al.*, 1994b] se presenta un método que emplea B-splines cúbicas. Estos modelos son adaptados a los contornos de las zonas de la imagen donde se detecta movimiento. La detección de movimiento se realiza mediante una comparación de la imagen actual con una imagen estática de la escena, imagen que es actualizada tras procesar cada nueva imagen de la secuencia según el método descrito en [Karmann y von Brandt, 1990], el cual estima la evolución de los valores de intensidad de los píxeles del fondo. Para ello, se tienen en cuenta cómo afectan a las condiciones de iluminación tanto las circunstancias climatológicas como la hora del día.

En [Koller *et al.*, 1994b] se emplean dos filtros de Kalman: uno para adaptar a los *blobs* los doce puntos de control usados en las B-splines; y otro para realizar el seguimiento de los vehículos mediante un modelo de movimiento afín. Este método es completado en [Koller *et al.*, 1994a] con una *red de creencia*, *Belief Network*, la cual permite efectuar razonamientos a mayor nivel sobre el comportamiento de los vehículos.

Métodos similares pueden encontrarse en [Remagnino *et al.*, 1997] y [Ferrier *et al.*, 1994]. En [Ferrier *et al.*, 1994] se simplifica el modelo a una forma rectangular para conseguir que el sistema siga un vehículo en tiempo real, y en [Remagnino *et al.*, 1997] se combinan modelos bidimensionales para los peatones y tridimensionales para los vehículos. Se detectan las zonas de movimiento mediante una comparación con un fondo estático, y dependiendo del tamaño de cada zona, se le asigna un modelo de peatón o de vehículo.

Como ya se ha comentado con anterioridad, el principal problema del uso de modelos bidimensionales es el de la inicialización. De hecho, en los métodos que se han mostrado, ni tan siquiera se prevé la posibilidad del agrupamiento de los vehículos en el instante de la inicialización. Por ello, parece que el uso de otro tipo de características más simples, no tan directamente influenciadas por la forma tan diversa que los vehículos presentan, pueden proporcionar mejores resultados.

Dubuisson y Jain [Dubuisson y Jain, 1995] describen un procedimiento basado en regiones que combina segmentación de movimiento y de color. La segmentación de movimiento se basa en substracción de imágenes, empleando tres imágenes (actual, anterior y posterior). Después de la resta de imágenes, se obtiene una imagen donde los píxeles que se encuentran alrededor de las aristas de los objetos móviles quedan marcados. La segmentación de color está basada en un método de *unión y separación*, *split and merge*, de regiones. Ambas técnicas de segmentación son combinadas en

un último paso, donde los bordes extraídos permiten unir aquellas regiones que aún perteneciendo a un mismo objeto, no fueron unidas por la segmentación de color. Al igual que en [Dubuisson *et al.*, 1996], se emplea una visión lateral de la carretera, pero sin realizar una estimación de movimiento, con lo que no es posible separar vehículos que aparezcan agrupados.

Smith y Brady [Smith y Brady, 1995], al igual que Malik y otros en [McLauchlan y Malik, 1997], [Beymer *et al.*, 1997] describen técnicas basadas en puntos característicos. En [Smith y Brady, 1995]³ se describe un método de estimación y segmentación basado en características que permite seguir vehículos. La cámara se supone posicionada en un coche que se halla en movimiento, y se emplean como características tanto puntos característicos, como bordes. Los puntos son seguidos a lo largo de la secuencia empleando un método de correspondencia, agrupándose en la fase de segmentación aquellas características con movimiento coherente. Este método utiliza *integración temporal*, que permite utilizar el conocimiento de la forma de los objetos adquirido durante toda la secuencia y no en unas pocas imágenes. Con este mayor conocimiento se consigue mejorar la segmentación final de los objetos. Beymer y otros en [Beymer *et al.*, 1997] describen un procedimiento basado en puntos. En este trabajo se establece una correspondencia de puntos a lo largo de la secuencia y se agrupan teniendo en cuenta el movimiento rígido de los automóviles. Aquellos puntos que se mueven rígidamente juntos durante varias imágenes son agrupados como un único vehículo. Por su parte, Klausmann y otros en [Klausmann *et al.*, 1999] emplean los puntos de interés de la imagen para detectar vehículos. Una variación en la distribución de los puntos de interés de una región de la imagen se interpreta como la presencia de un vehículo.

Además de las anteriores tendencias, existen otros métodos que desarrollan aspectos puntuales de la aplicación de la visión por ordenador a la monitorización del tráfico. Gil *et al.* [Gil *et al.*, 1996] ha desarrollado un método para combinar las técnicas basadas en contornos y las basadas en regiones. En este artículo se comparan varias técnicas de combinación, mostrando la idoneidad de un procedimiento basado en el filtro de Kalman. Sin lugar a dudas, la combinación de varias técnicas trabajando en paralelo puede representar en el futuro una forma de aumentar la precisión de la estimación de movimiento. Sin embargo, y de cara a una implementación en tiempo real, significa un incremento del coste de computación, por lo que, de momento, parece más rentable la optimización de una buena técnica. Fathy y Siyal [Fathy y Siyal, 1997] consiguen la implementación en tiempo real concentrando la acción del método sobre un único problema: contar el número de vehículos que atraviesan un cruce de calles, pero sin estimar ni el movimiento ni la forma de los vehículos. En [Siyal *et al.*, 2000] se presenta un enfoque similar para contar vehículos en autopistas. La ejecución en tiempo real se consigue mediante un sistema multiprocesador especialmente diseñado para procesar en paralelo cada uno de los carriles de la autopista.

³En [Smith, 1995] puede encontrarse una versión más amplia del mismo trabajo.

1.4 Objetivos

Observando la literatura existente acerca de la aplicación de las técnicas de procesamiento de imágenes a la monitorización del tráfico, podemos observar que existen dos tendencias, dependiendo de si se realiza o no una explícita extracción y uso de información tridimensional. Después de un cuidadoso estudio del problema que deseamos abordar, nos parece acertada la conclusión de que el análisis de movimiento sobre el plano de la imagen puede proporcionar todo el conocimiento necesario para resolver las tareas de monitorización de tráfico. El uso de estructuras tridimensionales es muy costoso computacionalmente y este costo no puede ser justificado por la realización de ninguna de las tareas fundamentales de este tipo de aplicaciones.

Existen algunas tareas adicionales, tales como *la estimación de la velocidad de vehículos individuales* o *la estimación de la velocidad media del tráfico* donde podría parecer necesario conocer la posición tridimensional de los vehículos sobre la carretera. Sin embargo, esta posición puede ser inferida si poseemos a priori cierto conocimiento acerca de la escena. Dado que la posición de la cámara es fija, se puede calcular previamente la estructura tridimensional de la carretera. De esta forma, el sistema podría hacer corresponder puntos del plano de la imagen con puntos tridimensionales de la carretera. Con esta correspondencia se podría estimar la velocidad individual de cada vehículo, y a partir de ella, la velocidad media del tráfico.

No obstante, y dado que la estimación de velocidades de vehículos individuales no la consideramos como fundamental, se pueden utilizar medidas alternativas para estimar la fluidez del tráfico. Las medidas que proponemos son: el número de vehículos que abandonan la escena por unidad de tiempo, y la relación entre el número de vehículos presentes en la escena por unidad de tiempo y la anterior medida. La primera medida permite saber el volumen del tráfico que atraviesa un determinado punto, mientras que la segunda nos hace saber el volumen de la congestión que pueda existir en ese punto.

En esta Tesis se propone un método de análisis de movimiento basado en regiones. El seguimiento de regiones es menos sensible a oclusiones dada la extensa información que proporcionan. Características tales como el tamaño, forma o intensidad media pueden ser directamente obtenidas a partir de ellas. Además, las regiones resultan muy adecuadas para secuencias donde el fondo es estático, ya que una vez que el movimiento ha sido estimado, resulta sencillo separar las regiones que se mueven de las que permanecen estáticas. En las escenas de tráfico el fondo permanece inmóvil, mientras que los vehículos suelen ocultarse parcialmente.

Para conseguir realizar el sistema completo de monitorización de tráfico, será necesario emplear una combinación de técnicas nuevas o ya existentes. Esta combinación constará de: una segmentación estática, una estimación de movimiento, una segmentación de movimiento y un seguimiento de regiones. La segmentación estática será el paso inicial y extraerá de cada imagen las regiones. Dado que en los subsiguientes pasos las regiones serán unidas para formar vehículos, es importante que esta segmentación genere regiones, que si bien no definan completamente la forma del objeto

correspondiente, sí al menos se garantice que sólo pertenecen a un único objeto.

La estimación de movimiento calculará para cada una de las regiones los vectores de movimiento en el plano de la imagen. Ésta suele ser la parte con mayor carga computacional de los métodos de análisis de movimiento, y por tanto, la que suele impedir las implementaciones en tiempo real. Para evitar esto, esta estimación deberá ser lo más simple posible, aunque aportando la suficiente precisión para poder usar el movimiento como medida discriminadora entre las regiones que pertenecen a diferentes objetos. Precisamente será la segmentación basada en movimiento quien utilice las velocidades estimadas para unir las regiones que pertenecen al mismo objeto, vehículo o fondo.

Será necesario un módulo para el seguimiento de las trayectorias de los vehículos. Para cada par de imágenes consecutivas de la secuencia, donde en cada una de ellas se han extraído las formas de los vehículos, este módulo será el encargado de relacionar qué dos regiones, una en cada imagen, corresponden al mismo vehículo. Esto se hará buscando las regiones que, de algún modo, sean las más similares.

En la mayoría de los métodos de análisis de movimiento, el seguimiento es el paso final, el cual se realiza tras la segmentación. Nosotros pensamos que el proceso de seguimiento también puede ayudar a mejorar la segmentación. El seguimiento relaciona las distintas apariciones de una misma entidad en distintas imágenes. Por lo tanto, este proceso puede ser un medio para realimentar la segmentación, de forma que en ésta se tenga en cuenta el conocimiento acumulado a lo largo de toda la secuencia. Conociendo la evolución de la morfología y la trayectoria de un objeto durante n imágenes, se puede predecir su forma y posición en la imagen $n + 1$. Esto nos permitirá corregir la segmentación de movimiento cuando se aprecien errores. Entendemos por errores, súbitos cambios en la forma de un objeto, o también repentinas apariciones o desapariciones de objetos. Se entiende que en la mayoría de escenas con movimiento, y principalmente en las escenas de tráfico, los objetos aparecen, desaparecen o van cambiando la forma que de ellos apreciamos, pero de una manera gradual y no repentina. Si, por ejemplo, de repente, aparece una nueva región tras la segmentación, se entenderá que probablemente se debe a un error en este proceso, y se deberá corregir la segmentación. No obstante, el proceso deberá tomar nota de esta región, por si su aparición se repite en las siguientes segmentaciones, ya que entonces se debería entender que realmente sí que correspondía a un objeto.

Otro de los objetivos fundamentales es conseguir que el sistema final funcione en tiempo real, es decir, que procese las imágenes a la misma velocidad con que son capturadas por el sistema de adquisición. Para ello, se deberán escoger técnicas lo más simples posibles, pero que proporcionen resultados satisfactorios. Además, se deberán utilizar otras que reduzcan la complejidad del problema, por ejemplo, reduciendo el espacio de búsqueda de objetos móviles en la imagen.

Por tanto, podemos resumir los objetivos de esta tesis de la siguiente manera:

- El estudio de los métodos de segmentación de imágenes, seleccionando y adaptando aquel que resulte más adecuado para la segmentación de vehículos.

- El estudio de las técnicas de estimación y segmentación de movimiento, buscando aquellas que produciendo resultados satisfactorios, no aumenten considerablemente el coste computacional del sistema.
- El estudio y desarrollo de estrategias de seguimiento de regiones que sean aplicables al problema del tráfico.
- La realización de técnicas que permitan utilizar en la segmentación de cada imagen, toda la información generada en el procesamiento de las imágenes previas de la secuencia, *integración temporal*.
- Aplicación de las técnicas estudiadas para el desarrollo de un sistema de monitorización de tráfico en tiempo real que permita llevar a cabo las siguientes tareas:
 - Estimar el número de vehículos por unidad de tiempo que han circulado por un determinado carril.
 - Medir la fluidez del tráfico.
 - Estimar el número de coches que se hallan detenidos en una cola.
 - Detectar atascos.
 - Detectar comportamientos anómalos de ciertos vehículos como por ejemplo: vehículos que se salen de la calzada o adelantamientos en zona prohibida.

1.5 Estructura del trabajo

La Figura 1.3 muestra un diagrama de flujo del sistema a alto nivel. Los procesos allí mostrados se repiten para cada imagen de la secuencia. Así, en cada iteración del sistema se toman dos imágenes consecutivas de la secuencia, n y $n + 1$, y se producen como resultados la segmentación de la imagen n y los parámetros que muestran el estado actual del tráfico.

Al procesar la imagen n con ayuda de la $n + 1$ somos conscientes de que estamos introduciendo un cierto retraso que no se produciría de usar la imagen $n - 1$ en lugar de la $n + 1$. Sin embargo, bajo nuestro punto de vista, esto carece de importancia. En primer lugar porque parte de ese retraso se consume en la segmentación estática de la imagen n , y para llevar a cabo este procedimiento no se requiere la imagen $n + 1$. Y en segundo lugar, porque estamos hablando de una latencia del orden de unas pocas décimas de segundo. Debemos ser conscientes de que para el tipo de aplicación que estamos desarrollando, este retraso no es significativo, pues en ningún caso se va a interactuar con los vehículos.

El primer proceso es la *detección de movimiento*. Este procedimiento detecta en qué zonas de la imagen se han producido cambios debidos al movimiento de los objetos. Este paso es importante, pues sólo sobre estas zonas es necesario realizar el procesamiento, ya que en ellas se encuentran los coches que están circulando.

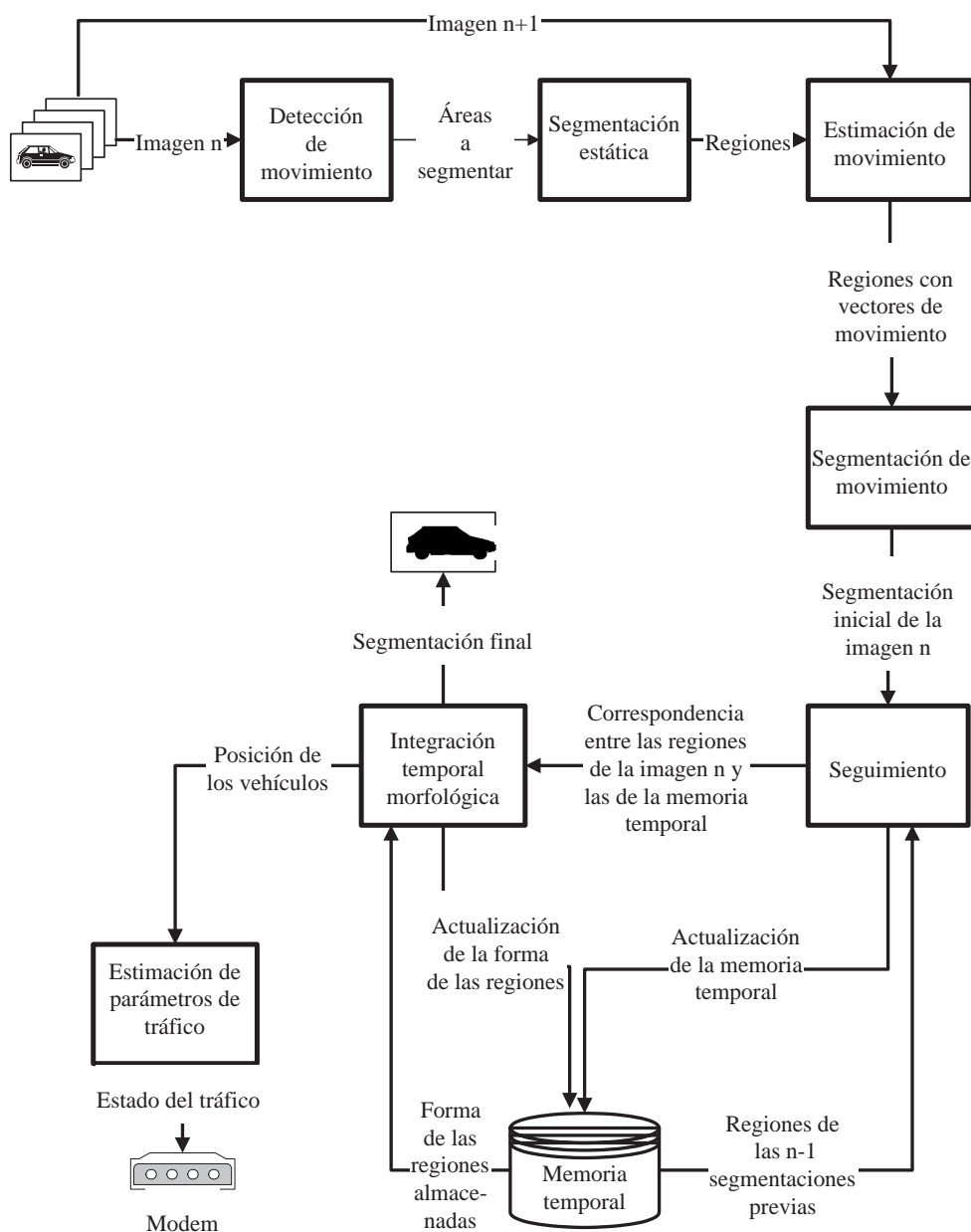


Figura 1.3: Flujo de datos entre los procesos que componen el sistema de monitorización de tráfico.

A continuación se realiza la *segmentación estática* de la imagen actual, n . Para llevar a cabo este proceso hemos desarrollado un método el cual divide la imagen en regiones. Esta división es hecha sin tener en cuenta ninguna consideración acerca del movimiento, aquí tan solo se atiende a las características de la imagen.

La imagen segmentada se pasa al proceso de *estimación de movimiento*, el cual, tomando la siguiente imagen, $n + 1$, estima el vector de movimiento asociado a cada región, utilizando un modelo translacional. Esta estimación se realiza por correlación, es decir, se busca para cada región de la imagen n aquella posición en la imagen $n + 1$ que minimiza la diferencia de nivel de gris. Esta sencilla estrategia, la cual usa el modelo de movimiento más sencillo, el translacional, nos aporta suficiente información para la fase de segmentación, pero a un coste computacional asumible.

Los vectores de movimiento calculados para cada región son utilizados por el siguiente proceso, *segmentación de movimiento*, el cual compara los movimientos de las regiones vecinas y va uniéndolas para formar objetos. En este proceso se asume que si dos regiones se desplazan con movimiento coherente, esto significa que ambas corresponden a un mismo objeto. A esta segmentación la denominaremos *segmentación inicial*, para distinguirla de la que se genera tras el proceso de seguimiento y en la que se hace uso de la *integración temporal morfológica*. A esta última segmentación la llamaremos *segmentación final*.

En esta tesis hemos desarrollado un método para generar la segmentación de cada imagen a partir de no sólo la información obtenida en su procesamiento, sino de toda la acumulada a partir de las imágenes previas. Así, una vez que la imagen ha sido dividida en regiones con movimiento diferente, el proceso de *seguimiento* relaciona estas regiones con las que están almacenadas en la *memoria temporal*, produciendo como resultado una correspondencia entre las mismas. Esta correspondencia relaciona las diferentes apariciones de una misma entidad, un objeto, en diferentes imágenes. La memoria temporal almacena cada nueva región que aparece en la segmentación de una imagen. Se considera que una región es nueva cuando no se corresponde con una región que hubiera aparecido en una segmentación previa. Sobre cada región se guarda información acerca de su forma y trayectoria. La trayectoria permite predecir la posición de las regiones en las futuras imágenes, mientras que el contenido sobre la forma se utiliza en el proceso de *integración temporal morfológica* para generar la segmentación final de la imagen n . Como ya se comentó con anterioridad, este proceso pretende obtener una mejor segmentación de los vehículos al consensuar todo el conocimiento adquirido en las segmentaciones precedentes.

Como puede observarse en el diagrama de la Figura 1.3, los procesos de *seguimiento* e *integración temporal* no sólo toman información de la memoria temporal, sino que también la actualizan. De este modo, los resultados del procesamiento de una imagen no son olvidados, sino que influirán sobre las restantes segmentaciones.

El último proceso, *estimación de los parámetros de tráfico*, utiliza la información generada por los restantes procesos, que son los que realizan el análisis de movimiento, para estimar la información referente al estado del tráfico. Aquí se realizan las tareas relacionadas en la Sección 1.4, por un lado el cómputo de los parámetros estadísticos,

y por el otro la generación de eventos ante situaciones que requieren efectuar un aviso.

De acuerdo con lo anteriormente expuesto, el resto de los capítulos se organizan de la siguiente forma:

- El Capítulo 2 describe la segmentación estática. En él se muestran algunos métodos de segmentación de regiones y se presentan los resultados del método escogido, junto con algunas conclusiones.
- El Capítulo 3 trata acerca de la estimación y segmentación de movimiento. Primeramente, se realiza una revisión bibliográfica de los distintos métodos de estimación y segmentación de movimiento. Posteriormente, se describe el método desarrollado en este trabajo, mostrando resultados y conclusiones.
- El Capítulo 4 describe el método de seguimiento, y el uso que se hace de la integración temporal para generar la segmentación final de las imágenes.
- El Capítulo 5 presenta los detalles del sistema relacionados con la monitorización del tráfico. Además, se explican las técnicas utilizadas para conseguir el funcionamiento en tiempo real. Se presentan diversos experimentos llevados a cabo sobre varias secuencias de tráfico utilizando el sistema de monitorización global. Se realiza un análisis y una discusión sobre los resultados obtenidos con los métodos expuestos en esta Tesis.
- En el Capítulo 6 se muestran las conclusiones, se resumen las principales aportaciones presentadas a lo largo del documento, se referencian las publicaciones surgidas de este trabajo, y se detallan algunas líneas de investigación futura a partir de los trabajos presentados en esta Tesis.

Capítulo 2

Segmentación de la imagen de referencia

Contenido

2.1	Partición de una imagen en regiones	19
2.1.1	Métodos basados en umbralización (<i>thresholding</i>)	21
2.1.2	Métodos basados en bordes	22
2.1.3	Métodos basados en regiones	22
2.1.4	Elección de un método de segmentación	25
2.2	Un algoritmo de segmentación de imágenes estáticas	29
2.2.1	Estructura del algoritmo	29
2.2.2	El algoritmo de clasificación de las <i>k-medias</i>	31
2.2.3	Algoritmo jerárquico de división de regiones	33
2.2.4	Unión de regiones	35
2.3	Resultados	37
2.4	Conclusiones	41

2.1 Partición de una imagen en regiones

Tal y como se comentó en la introducción de esta tesis, el primer paso para realizar el análisis de movimiento, según nuestra aproximación, consiste en dividir la imagen en regiones. Esta división debe ser realizada conforme a un criterio de homogeneidad que garantice la existencia de una relación entre los píxeles que sean integrados en cada región. En realidad, éste es un paso previo para la segmentación definitiva que será realizada en etapas posteriores. No se pretende establecer una correspondencia biyectiva entre las regiones aquí extraídas y los objetos de la escena. Tan solo se aspira a que cada región pertenezca a un único objeto, aunque un objeto pueda estar compuesto por varias regiones.



Figura 2.1: Una imagen típica de vehículos circulando por una carretera

El anterior requisito es necesario puesto que en pasos posteriores se estimará el movimiento de cada región. Dado que se asumirá que objetos distintos se desplazan con movimientos también distintos, la presencia de píxeles de dos objetos en una misma región llevaría a una estimación de movimiento errónea.

En cuanto al criterio de homogeneidad a elegir, analizando las imágenes de tráfico, parece claro que éste debe ser la intensidad de los píxeles de la imagen. Una segmentación por textura no parece adecuada puesto que ni la carretera, ni los vehículos, que son de colores uniformes, presentan texturas que sean apreciables a estas distancias, como se ve en la Figura 2.1, que muestra una típica imagen de vehículos circulando por una carretera. En esta imagen se puede apreciar tanto la uniformidad de la carretera como la de los vehículos. La uniformidad de los vehículos se va haciendo más manifiesta cuanto más lejana es la posición de la cámara.

Formalmente, se define la *segmentación de imágenes* como un proceso que típicamente particiona el dominio espacial de la imagen en subconjuntos mutuamente exclusivos, llamados *regiones*. Cada región es uniforme y homogénea con respecto a alguna propiedad, tal como la intensidad, el color o la textura, y su valor difiere de una forma significativa del de cada una de sus regiones vecinas. Un proceso de segmentación de imagen que use la intensidad como valor de propiedad, produce regiones llamadas *características tonales discretas*.

Existen dos tipos de segmentación [Sonka *et al.*, 1993]: completa y parcial. El resultado de la segmentación completa es un conjunto disjunto de regiones que unívocamente se corresponden con los objetos de la imagen, mientras que en la segmentación parcial las regiones no se corresponden directamente con los objetos. Para conseguir una segmentación completa es necesaria la utilización de procesos de más alto nivel que empleen un conocimiento específico del dominio del problema. Aunque en la mayoría de imágenes, la segmentación con procesamiento de bajo nivel no consigue resolver completamente el problema, sí al menos consigue reducir su complejidad, pues

los píxeles son agrupados en entidades de mayor nivel.

Según las características dominantes que se empleen en los métodos de segmentación, éstos pueden ser divididos en tres grupos [Sonka *et al.*, 1993, Jain, 1989, Horn, 1986]:

- Basados en un conocimiento global acerca de la imagen. Este conocimiento se representa generalmente mediante un histograma, y la definición de una serie de umbrales u holguras. Por ello son llamados de umbralización (*thresholding*).
- Basados en bordes (*Edge-based*). Detectan aquellos píxeles de la imagen que corresponden a cambios bruscos de intensidad. A menudo, estos cambios representan la frontera entre dos objetos distintos.
- Basados en regiones (*Region-based*). Asignan directamente los píxeles de la imagen a las diferentes regiones sin extraer previamente sus contornos. Los píxeles son asignados mediante el análisis de las vecindades de las regiones, empleando criterios de homogeneidad.

Los dos últimos grupos de técnicas son los que se adaptan a un mayor número de situaciones. Sin embargo, los basados en umbralización también tienen una gran utilidad, principalmente en aquellos dominios donde los objetos presentan una intensidad uniforme y conocida a priori.

En teoría, la segmentación basada en regiones y la basada en bordes deberían ser las dos caras de una misma moneda. Dado que los bordes forman los contornos de las regiones, ambos métodos deberían producir idénticos resultados. Los bordes podrían ser obtenidos a partir de las regiones mediante un algoritmo de seguimiento de contornos, y las regiones podrían ser obtenidas a partir de los bordes usando un algoritmo de relleno de contornos. Desafortunadamente, en imágenes reales es raro obtener bordes correctos a partir de regiones, y viceversa. Diversos factores, como el ruido, impiden esto [Jain *et al.*, 1995]. Sin embargo, los resultados de ambos enfoques pueden ser combinados, por ejemplo mediante un grafo relacional.

2.1.1 Métodos basados en umbralización (*thresholding*)

La umbralización por niveles de gris es el proceso de segmentación más sencillo y rápido. Básicamente, este tipo de métodos consiste en definir n umbrales u_k , y $n + 1$ etiquetas E_j , de forma que al conjunto de píxeles con un valor de intensidad mayor que u_{k-1} y menor que u_k se le asigna la etiqueta E_k . De este modo, cada etiqueta corresponde a una región diferente, con lo que la imagen queda segmentada.

Como es lógico, el éxito de este tipo de técnicas dependerá de la correcta elección de los umbrales. Así, se han propuesto diversos métodos para calcularlos:

- Transformación del histograma [Herman y Liu, 1978, Wezka y Rosenfeld, 1979].

- Aproximación del histograma mediante funciones de densidad de probabilidad de distribución normal [Kittler y Illingworth, 1986, Mardia y Hainsworth, 1988, Gupta y Sortrakul, 1998].
- Refinamiento jerárquico [Gross y Rosenfeld, 1987, Song *et al.*, 1990, Biswass y Pal, 2000].

Además, se han investigado diversos métodos para calcular umbrales en tiempo real [Hassan, 1989, Lavagetto, 1990]. En [Sahoo *et al.*, 1988, Gonzalez y Wintz, 1977, Rosenfeld y Kak, 1982] se presentan revisiones más amplias sobre este tipo de métodos.

2.1.2 Métodos basados en bordes

Consisten en la detección de aquellos píxeles de la imagen que presentan un cambio brusco en su valor de nivel de gris. En la literatura podemos encontrar una gran variedad de métodos para detectar estos píxeles [Canny, 1986, Roberts, 1977, Zhou *et al.*, 1989, Kittler *et al.*, 1987, Gokmen y Li, 1990]. Sin embargo, la imagen resultante de la detección de bordes no puede usarse como una imagen segmentada [Sonka *et al.*, 1993]. Esto se debe a que generalmente lo que se obtiene es un conjunto de bordes parciales, los cuales no definen completamente el contorno de las regiones. Así, es preciso utilizar otro tipo de técnicas [Hong *et al.*, 1980, Milgram, 1979, Elder y Zucker, 1996], las cuales permiten combinar bordes parciales para construir regiones. Desafortunadamente, estos métodos no siempre encuentran regiones aceptables. Además, estos procedimientos suelen ser demasiado complejos, pues requieren el estudio de las posibles conexiones entre todos los píxeles finales de los bordes que forman una determinada vecindad.

2.1.3 Métodos basados en regiones

A diferencia de los anteriores, los métodos basados en regiones tratan de agrupar los píxeles de una manera directa, sin detectar previamente sus contornos. Además, sus resultados suelen ser mejores a los de los métodos de umbralización o bordes, sobretodo cuando las imágenes presentan ruido.

Para asignar los píxeles a una u otra región se emplea un criterio de homogeneidad. Muerle y Allen [Muerle y Allen, 1968], en uno de los primeros trabajos de segmentación basada en regiones, definían una región como “*una porción de una imagen compleja, en la cual la distribución estadística de los niveles de gris es razonablemente uniforme*”. Obviamente, esta definición presupone un criterio de homogeneidad basado en la intensidad de la imagen. Sin embargo, existen otras propiedades que también pueden ser usadas como criterio, por ejemplo el color, la textura o incluso un modelo. El método de Muerle y Allen es considerado una técnica de crecimiento de regiones, ya que en sucesivas iteraciones, a una región se le van añadiendo aquellos píxeles vecinos que cumplen el criterio de homogeneidad.

Brice y Fennema [Brice y Fennema, 1970] presentan un método, también de crecimiento, que emplea una estrategia similar para formar las regiones iniciales. Una

vez que los píxeles han sido distribuidos entre las regiones, se emplean una serie de heurísticas para analizar los bordes que separan cada par de regiones vecinas. Cuando un borde es considerado como débil, el procedimiento lo elimina y une las regiones que separaba. En [Feldman y Yakimovsky, 1974], el algoritmo de Brice y Fennema se completa con la incorporación de dos nuevos pasos. En ellos se hace uso de información semántica acerca de los objetos que se sabe que componen la imagen.

En [Revol y Jourlin, 1997] se presenta un algoritmo que realiza el crecimiento basándose en la preservación de la varianza de las regiones. Para ello se aplican operaciones morfológicas que añaden aquellos píxeles que no modifican significativamente el histograma de una región. Una de las características de este método es que en cada iteración, la pertenencia de un píxel a una región puede ser reconsiderada. La inicialización del método se realiza a partir de unos píxeles semilla que son escogidos por el usuario.

En los métodos hasta ahora vistos en este apartado, la estrategia consiste en la unión sucesiva de regiones. Como caso opuesto podemos considerar las técnicas de *división de regiones* [Robertson *et al.*, 1973] (region splitting). Estas técnicas parten de la imagen como una única región, la cual es sucesivamente subdividida. Xu y Uberbacher [Xu y Uberbacher, 1997] han desarrollado recientemente un método de este tipo. En él la imagen es representada por un *árbol de mínima expansión* en el cual los píxeles de la imagen son los nodos, y los enlaces contienen la diferencia de nivel de gris entre dos píxeles. La segmentación de la imagen consiste en dividir el árbol en ramas que minimicen la diferencia de nivel de gris entre los píxeles que componen cada rama. De esta forma la imagen se divide en regiones de intensidad uniforme.

Las técnicas de división no han tenido un gran éxito, sin embargo, han dado lugar a otras que combinan ambos enfoques, *división y unión* (split and merge). Según el principio desarrollado por Horowitz y Pavlidis [Horowitz y Pavlidis, 1974], las regiones similares adyacentes son unidas, mientras que las regiones no homogéneas son divididas. En este método se usa una representación piramidal de la imagen. Cuando una región de un determinado nivel de la pirámide no es homogénea, entonces es subdividida en cuatro regiones cuadradas que son situadas en el siguiente nivel de la pirámide. Cuando cuatro regiones presentan una homogeneidad similar, se agrupan. Este método presenta la desventaja de requerir que las regiones y sus uniones tengan forma cuadrada. Así, estos mismos autores, en [Horowitz y Pavlidis, 1976], proponen una modificación que permite unir pares de regiones adyacentes cuando son suficientemente uniformes. En este nuevo método se introduce el uso de árboles para representar la estructura piramidal. Las operaciones del algoritmo en la pirámide pueden ser expresadas de una forma concisa mediante un *árbol de segmentación*, donde los nodos hoja representan regiones homogéneas.

En [Hong, 1982] y [Pietikainen *et al.*, 1982] se presentan otras dos modificaciones que permiten el solapamiento de regiones en la estructura piramidal. En [Cho y Meer, 1997] se construye una pirámide de imágenes a partir de la imagen original. Cada nivel de la pirámide corresponde a una resolución diferente y lleva asociado un grafo de adyacencia de regiones. Este grafo permite unir en cada nivel

aquellos píxeles con un nivel de gris similar, con lo que se define una segmentación para cada nivel de la pirámide. La segmentación final de la imagen, llamada por los autores *segmentación de consenso*, se obtiene considerando la probabilidad de que dos píxeles formen parte de una misma región. Por su parte, la probabilidad se calcula a partir del número de veces que dos píxeles fueron unidos en los distintos niveles de la pirámide.

En [Gamba *et al.*, 1997] se presenta un método de segmentación que auna las segmentaciones basadas en regiones y bordes. El método está diseñado para la navegación autónoma de robots. Mediante la estimación de los puntos de fuga de las líneas de la imagen se definen las orientaciones de las regiones. Estas regiones son agrupadas mediante una serie de heurísticas que consideran dichas orientaciones. Saber y otros [Saber *et al.*, 1997] han desarrollado otro método para combinar bordes y regiones, aunque para imágenes en color. La técnica consta de varios pasos: primero una estrategia de crecimiento basada en *Campos Aleatorios de Markov* segmenta la imagen; a continuación, con los bordes extraídos, se dividen aquellas regiones atravesadas por uno de estos bordes; y finalmente, se unen las regiones que son divididas por un borde no significativo.

Un enfoque diferente para la segmentación basada en regiones es el de los métodos de *agrupamiento* (clustering). Esta estrategia consiste en la aplicación de técnicas de agrupación no supervisada para clasificar los píxeles de la imagen en diferentes regiones. Se definen un conjunto de características asociadas a cada píxel, en base a las cuales se realiza la clasificación.

“... al no exigir un conocimiento previo de las clases de objetos existentes, los algoritmos de *agrupamiento* dotan de una gran robustez a la segmentación de escenas ...” [Maravall, 1993]

Coleman y Andrews [Coleman y Andrews, 1979] definieron uno de estos métodos. El algoritmo divide inicialmente la imagen en dos grupos (clusters), a los que se asignan los píxeles mediante la técnica de clasificación de las *k-medias*. Tras cada aplicación de esta técnica se evalúa si el número de grupos es el adecuado, y en caso contrario, se añade un nuevo grupo a la imagen, para, a continuación, proceder de nuevo a la clasificación de los píxeles. Esta estrategia permite que el número de grupos se adapte a las características de la imagen, pero presenta la desventaja del alto coste computacional, pues requiere reclasificar todos los píxeles de la imagen en reiteradas ocasiones.

El método de Kottke y Sun [Kottke y Sun, 1994] también emplea el clasificador de las *k-medias*, pero para un número fijo de grupos. Este número debe establecerse antes de la segmentación, lo cual implica que se debe poseer un cierto conocimiento previo de la imagen. El espacio de características es tridimensional: intensidad y coordenadas x e y . Esto permite agrupar los píxeles por su similitud de nivel de gris, pero evitando la dispersión de los píxeles de cada grupo dentro del dominio espacial. Como en cualquier algoritmo basado en el de las *k-medias*, los píxeles se asignan al grupo cuyo centro es más cercano en el espacio de características. Sin embargo, en este trabajo se emplea

una matriz de pesos que se recalcula cada ciertas iteraciones. Dicha matriz permite ponderar la importancia de cada característica y su actualización adapta los clusters a las singularidades de cada imagen.

2.1.4 Elección de un método de segmentación

“La elección de una técnica de segmentación que sea apropiada para un específico tipo de imagen es un problema difícil” [Pal y Pal, 1993].

Lo ideal sería disponer de un método que pudiera segmentar cualquier tipo de imágenes. Sin embargo, la realidad es muy distinta. Disponemos de un conjunto de técnicas que han sido aplicadas a muy diversos tipos de imágenes.

“No hay una única técnica que pueda ser aplicada a todas las imágenes, ni todos los métodos son igualmente buenos para una determinada clase de imagen” [Pal y Pal, 1993].

Además, no existe ningún procedimiento que permita medir cuantitativamente, de manera objetiva, la calidad del resultado de la aplicación de un método de segmentación [Rosenfeld y Kak, 1982, Pal y Pal, 1993, Pratt, 1991, Zhang, 1997]. Desafortunadamente, sólo un observador humano puede comparar los resultados de diferentes segmentaciones, lo que nos lleva a una forma de evaluación que siempre tendrá algún grado de subjetividad.

Dados los tres grupos generales en los que hemos dividido las diferentes técnicas de segmentación, creemos que la segmentación a utilizar debe estar basada en regiones. La segmentación basada en umbrales requiere cierto conocimiento acerca de los objetos que componen la imagen. En nuestro caso, la imagen se compone de una parte fija, la calzada y sus alrededores, y una parte que irá variando de una imagen a otra, los vehículos. Dada la diversidad de las tonalidades que pueden tener estos últimos es prácticamente imposible asentar ningún tipo de conocimiento acerca de su intensidad. Por su parte, la intensidad de la parte fija sólo es uniforme en la calzada, y no en sus alrededores. Además, su intensidad irá variando debido a las condiciones de iluminación.

En cuanto a los métodos basados en bordes, estos presentan los problemas antes mencionados: discontinuidad de los bordes en imágenes ruidosas, como las grabadas en el exterior, y la necesidad de usar un algoritmo de relleno para obtener las regiones finales. Estas cuestiones hacen que su supuesto menor coste computacional, frente al de los algoritmos basados en regiones, no sea tal, por lo que creemos que un método basado en regiones aborda el problema de forma más directa.

Los vehículos y la calzada se caracterizan por presentar una intensidad uniforme. De aquí, es fácil deducir que el criterio de homogeneidad debe ser el nivel de gris, sin embargo, y dado que antes de aplicar el proceso no conocemos ni el número de vehículos ni sus niveles de intensidad, el método de segmentación debe adaptarse a la imagen sin necesidad de poseer un conocimiento a priori.

Los métodos de agrupamiento cumplen estos requisitos, si bien presentan la desventaja de su elevado coste computacional [Maravall, 1993]. El coste computacional es proporcional al número de píxeles de la imagen y al número de grupos. Esto se debe a que son métodos iterativos, donde en cada iteración, se toma cada píxel y se estudia su inclusión en cada uno de los grupos. Por lo tanto, si deseamos utilizar uno de estos métodos, debemos reducir tanto el número de píxeles, como el número de comparaciones necesarias para incluir un píxel en un grupo. El número de píxeles puede reducirse mediante una técnica de detección de movimiento, es decir, detectando las zonas de la imagen donde se ha producido movimiento. Dado que lo que en realidad nos interesa es segmentar la forma de los vehículos, sólo necesitamos procesar los píxeles que se han movido. Este tipo de técnicas se verán en la Sección 3.2.

La reducción del número de comparaciones para averiguar la pertenencia de un píxel a alguno de los grupos disponibles puede conseguirse mediante un método jerárquico. La estrategia consistiría en iniciar el procesamiento con tan solo unos pocos grupos. De esta forma, la imagen sería dividida en aquellos elementos más preponderantes. Una vez conseguida esta división inicial, a cada uno de los grupos se le volvería a aplicar el método de agrupamiento. Para ello, el grupo sería subdividido en grupos más pequeños, y se aplicaría el proceso de forma que los píxeles que pertenecieran al grupo subdividido solo pudieran ser asignados a uno de estos nuevos grupos. Este procedimiento se repetiría en toda la jerarquía para cada grupo que no cumpliera el criterio de homogeneidad. Criterio que tendrá que estar basado en la varianza del nivel de intensidad, de forma que las regiones tengan un nivel de gris uniforme.

El método de Kottke y Sun [Kottke y Sun, 1994], que fue anteriormente referenciado en el apartado 2.1.3, es un método de segmentación basado en el algoritmo de clasificación de las *k-medias*. Como ya comentamos, su característica más interesante es que el algoritmo recalcula los pesos que se asignan a cada característica (intensidad y coordenadas x e y de cada píxel). Esto hace que los grupos se adapten fácilmente a las peculiaridades de cada imagen.

En la Figura 2.2 se muestran diversas iteraciones del algoritmo de Kottke y Sun. La imagen segmentada es la que aparece en la Figura 2.1, y el resultado final de la segmentación se muestra en la subfigura 2.2.d. La subfigura 2.2.a muestra la inicialización del proceso. En ella la imagen se divide en k grupos cuadrados de igual tamaño, y se calcula el vector de medias de las características (coordenadas x e y y la intensidad).

A partir de la inicialización se van repitiendo iteraciones en las cuales cada píxel se asigna al grupo que minimiza la diferencia entre las características del píxel y el vector de medias del grupo. Las medias de cada grupo son recalculadas tras cada iteración, y los pesos, utilizados en el cálculo de las distancias, lo son cada cierto número de iteraciones. De este modo los grupos se van adaptando a la imagen.

En las subfiguras 2.2.b y 2.2.c se muestran dos estados intermedios de la segmentación. Concretamente, estas subfiguras corresponden a la distribución de los píxeles tras 9 y 12 iteraciones respectivamente. Como puede verse, la adaptación es bastante rápida, de hecho la forma de los vehículos puede apreciarse rudamente desde la primera iteración. Las restantes iteraciones redistribuyen los píxeles para refinar su

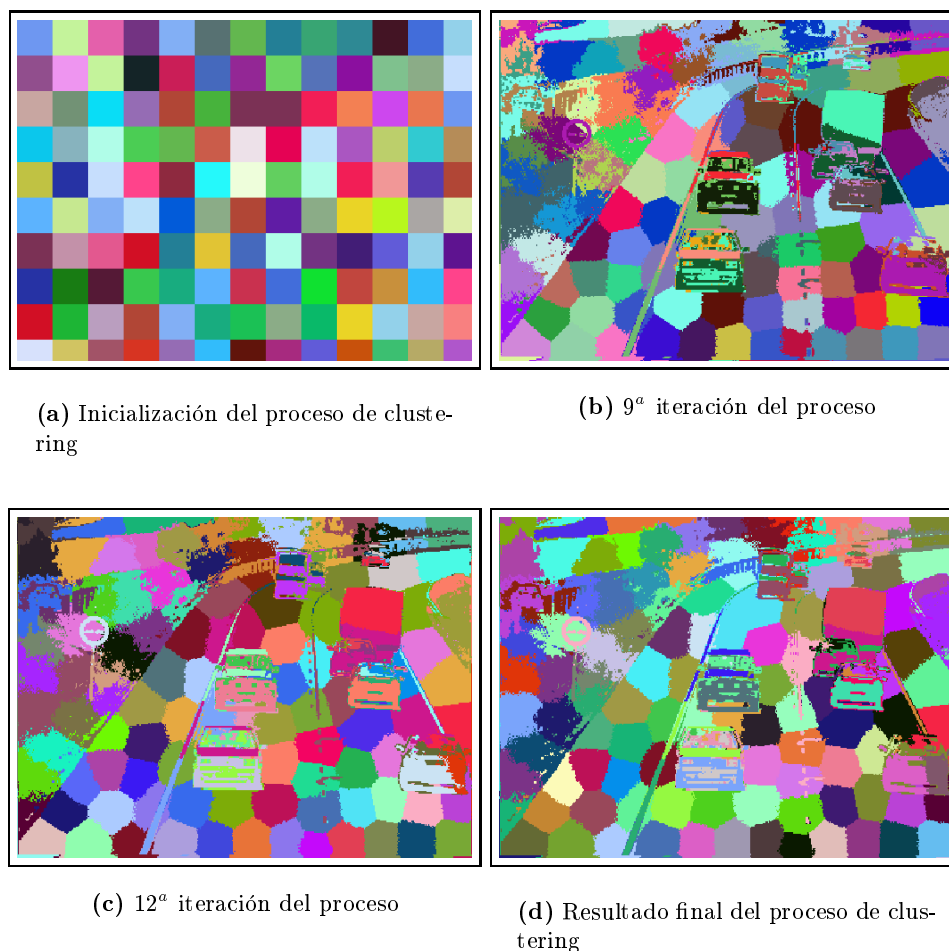


Figura 2.2: Diversas iteraciones del proceso de segmentación aplicando el algoritmo de Kottke y Sun.

forma.

El principal inconveniente de este algoritmo es la necesidad de especificar el número de grupos antes de comenzar a procesar la imagen. Esto implica tener que tomar una decisión en un instante en que aún no se dispone de ninguna información. Además, este inconveniente, combinado con la adaptación del algoritmo a la imagen, produce una desventaja añadida: los grupos tienden a preservar la misma distribución espacial uniforme con la que fueron introducidos en el proceso de inicialización. Esto se debe a que el algoritmo en realidad fue diseñado para imágenes donde el objeto a segmentar ocupase la mayor parte de la imagen. Cuando, como en nuestro caso, esto no es normal que suceda, se necesita utilizar un gran número de grupos, con lo que en realidad la adaptación de los grupos no es global, sino local. Es decir, cada grupo se adapta a los píxeles de su vecindad. Así, algunas zonas de la imagen son excesivamente

segmentadas, mientras que en otras se echa en falta la presencia de algún grupo más para refinar la segmentación.

En la segmentación de la imagen de la Figura 2.1, subfigura 2.2.d, se ven zonas donde pueden apreciarse estos dos casos. El caso más claro de sobresegmentación es el de la carretera. Ésta ha sido dividida en una excesiva cantidad de regiones, todas ellas de similar tamaño y forma. La causa es que en esa zona de la imagen no se producen grandes diferencias de nivel de gris, por lo que el algoritmo prima la posición del píxel frente a su intensidad, con lo que cada píxel es asignado al grupo más cercano en la imagen.

En el caso que nos ocupa, la sobresegmentación no tiene una excesiva importancia, pues un proceso final podría unir fácilmente aquellas regiones vecinas que presentan una intensidad similar. Sin embargo, el caso de la subsegmentación es mucho más grave, e incluso irreversible. Cuando una misma región, grupo, engloba píxeles de dos objetos, es muy difícil dividir de nuevo la imagen, máxime cuando existe una conexión espacial entre los píxeles de ambos objetos.

En la segmentación de la Figura 2.2.d se observa uno de estos casos en el grupo de tres vehículos que circula por el carril de la derecha. Allí, una misma región agrupa píxeles tanto del coche que ocupa la posición central del grupo, como del camión que se encuentra circulando delante de él. En este caso, dado que en la región los píxeles del coche y del camión están conectados, ningún proceso de bajo nivel podrá separarlos. De hecho, sólo un observador humano podría detectar el error.

A pesar de las críticas que hemos vertido sobre el algoritmo de Kottke y Sun [Kottke y Sun, 1994], nos parece que éste es un método de segmentación que se adapta a los requerimientos de nuestro sistema. Se observa claramente que carece de un enfoque global que permita redistribuir los grupos en la imagen de una forma más adecuada, más grupos en las zonas que presentan más detalles y menos en las zonas uniformes. Sin embargo, a pesar de ello, los grupos se adaptan a las formas presentes en cualquier imagen, y en la mayoría de casos consigue que los píxeles de una región sólo pertenezcan a un objeto. Además, se trata de un método no supervisado, lo cual resulta imprescindible.

En el resto del capítulo presentamos el método de segmentación que hemos desarrollado para esta Tesis. Aunque la semilla del método es, sin lugar a dudas, el método de las *k*-medias presentado por Kottke y Sun, nuestro método, además, se basa en las técnicas de *división y unión de regiones*. De estos métodos recogemos el concepto de *segmentación jerárquica*.

La segmentación jerárquica consiste en dividir la imagen aplicando el algoritmo de agrupamiento, inicialmente en sólo unas pocas regiones, para a continuación ir subdividiendo sólo aquellas regiones que no cumplen el criterio de homogeneidad. Esta subdivisión se realiza en cualquier nivel. Es decir, una región surgida de una subdivisión puede volver a ser subdividida si no cumple el criterio. De esta forma, el número final de regiones en que la imagen queda segmentada no se introduce al principio, sino que son las especificidades de cada imagen las que finalmente dictan su valor.

Por lo general, un algoritmo sólo basado en la división de regiones acaba produciendo regiones vecinas similares, por lo que debe completarse con un procedimiento que una este tipo de regiones [Horowitz y Pavlidis, 1974]. En nuestro método, esta unión se realiza al final, una vez que se ha terminado el proceso de división de las regiones. En este último paso se analizan todas las posibles vecindades entre regiones, y se unen aquellas que presentan una intensidad similar.

La siguiente sección, 2.2, detalla el algoritmo de segmentación propuesto en esta Tesis. Primero se describen de una forma general los distintos pasos del algoritmo, Apartado 2.2.1. En el siguiente apartado, 2.2.2, se detalla el algoritmo de clasificación basado en agrupamiento. El Apartado 2.2.3 describe la técnica de división reiterada de regiones que permite al algoritmo adaptar el número de las mismas a cada imagen. Por su parte, el último apartado de la sección, 2.2.4, describe los pasos finales del método. En ellos, primero se dividen aquellas regiones que presentan grupos de píxeles inconexos, y después se unen las regiones vecinas que resultan similares con respecto al criterio de homogeneidad: la intensidad de la imagen. La Sección 2.3 muestra los resultados de la aplicación del algoritmo sobre diversas imágenes, y la Sección 2.4 presenta las conclusiones de este capítulo.

2.2 Un algoritmo de segmentación de imágenes estáticas

2.2.1 Estructura del algoritmo

La Figura 2.3 presenta un diagrama de flujo del método de segmentación propuesto. En él se observan los procedimientos principales, los cuales serán ampliamente detallados en los siguientes apartados. Como puede verse en esta figura, existe un cierto número de procesos que se hallan dentro de un rectángulo de línea discontinua. Estos procesos corresponden al algoritmo de clasificación basado en las *k-medias*. Dicho algoritmo de clasificación se ejecuta varias veces, primero para la totalidad de la imagen, y después para cada una de las regiones, *grupos*, que son subdivididas.

El primer paso del algoritmo inicializa el proceso de clasificación. En él, la imagen se divide en regiones cuadradas de igual tamaño, y se calcula el vector de medias de la región. El siguiente paso asigna cada píxel a la región cuyo centro, vector de medias de las características, es el más cercano. Esta distancia se calcula restando el vector de medias de cada región, del vector de características del píxel que queremos asignar. La diferencia es ponderada por una matriz de pesos que modifica la importancia de cada característica.

Una vez terminada la clasificación, y dado que se ha producido una redistribución de los píxeles, se procede a recalculer los vectores de medias de las regiones, los centros de masas. Y a continuación se toma la decisión de parar o no la clasificación, en base a si se ha obtenido una solución estable. Para ello se calcula el máximo desplazamiento que ha sufrido el centro de una región con respecto a la anterior iteración, y cuando es menor que una constante T_s , se considera que se ha llegado a una clasificación estable.

Si la solución es estable, se pasa al proceso de división de las regiones, y si no lo

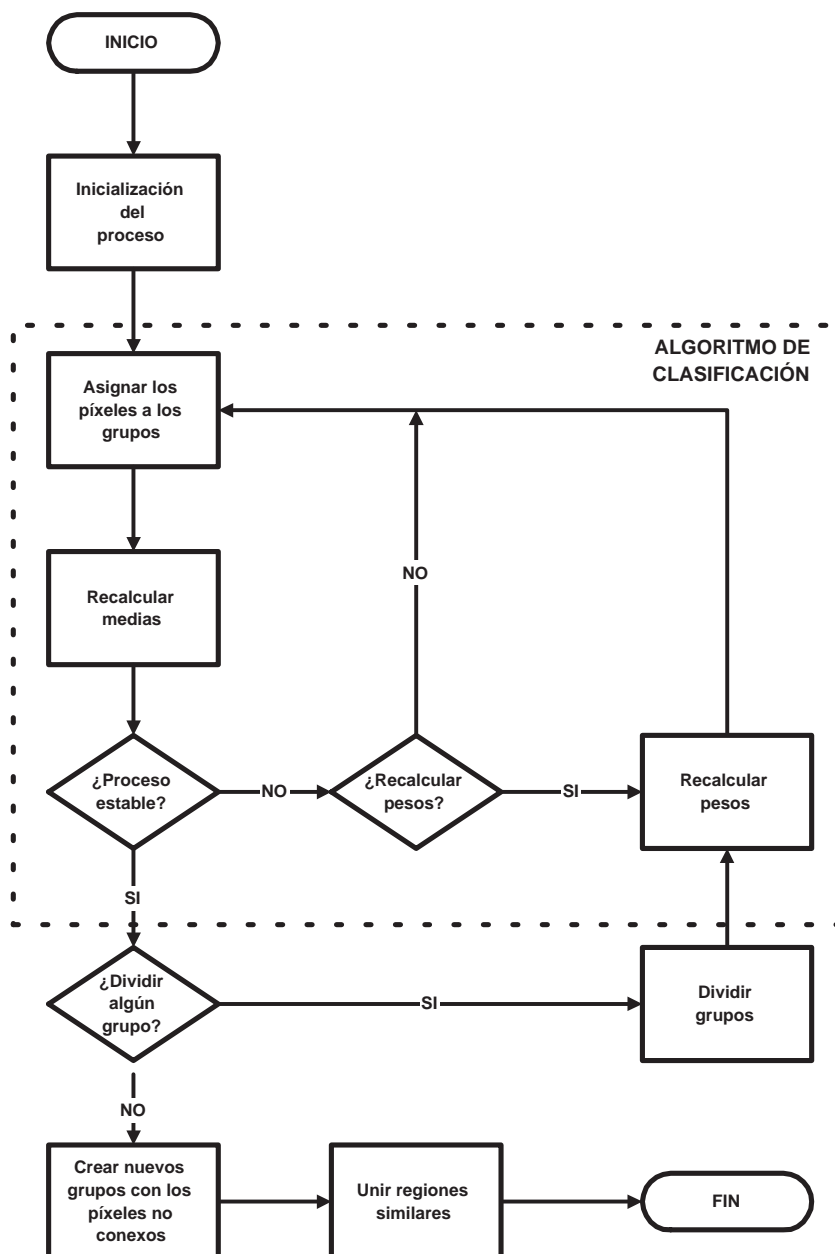


Figura 2.3: Diagrama de flujo del método de segmentación

es se mira si los pesos de las características deben ser recalculados. Estos pesos, que son recogidos en una matriz \mathbf{W}^j para cada región R^j , se recalculan cada t iteraciones. Tanto si se recalculan los pesos, como si no, se vuelve al proceso de asignación de píxeles a regiones, donde las variaciones en los vectores de medias de cada región hacen que la segmentación se vaya adaptando a la imagen.

En el proceso de división de regiones se mira la uniformidad de cada región con respecto a la intensidad. Aquellas regiones que no son uniformes se dividen en varias y a continuación se vuelve a aplicar el algoritmo de clasificación. En esta nueva ejecución de la clasificación se trata cada región dividida como una imagen independiente, de forma que sus píxeles sólo pueden asignarse a las nuevas regiones que la subdividen, y no al resto de regiones de la imagen. Esto permite ir obteniendo, en cada nivel de la subdivisión, regiones de mayor detalle.

Una vez que todas las regiones cumplen los criterios de homogeneidad, y ya no pueden volver a ser subdivididas, se realiza un análisis de la conectividad de los píxeles que pertenecen a cada región. El algoritmo de clasificación garantiza la agrupación de los píxeles de cada grupo en el espacio de características, pero no en el espacio de la imagen. Aunque la división produce regiones relativamente agrupadas en el espacio de la imagen, siempre pueden aparecer conjuntos inconexos de píxeles que pertenecen a un mismo grupo, pero que podrían no pertenecer a un mismo objeto. Por ello, se forma una nueva región para cada uno de estos conjuntos.

El último proceso del algoritmo reduce la subdivisión de la imagen. Toma cada pareja de regiones vecinas y las une cuando presentan una intensidad similar. Este último paso permite unir regiones vecinas surgidas de la misma o distintas ramas del proceso de subdivisión.

En los tres siguientes apartados se describen el algoritmo de clasificación de los píxeles, el método de subdivisión de regiones y la unión de regiones similares. En ellos presentaremos los detalles del método, que hasta ahora tan solo hemos sumariado.

2.2.2 El algoritmo de clasificación de las *k-medias*

El algoritmo de clasificación propuesto es una adaptación del algoritmo de las *k-medias* que emplea un espacio de características tridimensional: las dos coordenadas de la imagen y la intensidad de los píxeles. Mientras que la intensidad actúa como criterio de homogeneidad para agrupar píxeles en regiones, las coordenadas evitan que los píxeles que se asignan a cada región se dispersen por toda la imagen.

El algoritmo de las *k-medias* siempre parte de una división inicial del conjunto de elementos en k grupos. En nuestro caso, la división inicial puede provenir bien del procesamiento de toda la imagen, o bien de la subdivisión de una región. En el primer caso se divide la imagen en k regiones cuadradas uniformemente distribuidas en la imagen y de igual tamaño. En el segundo caso los grupos vienen dados por el propio proceso de subdivisión de cada región.

Una vez que se tiene la división inicial, se procede a calcular los valores medios de las tres características en cada región. De esta forma, cada región es caracterizada

por un centro de masas representado por un vector $\mathbf{m}^j = [m_x^j, m_y^j, m_g^j]$, donde m_x^j es la media de las coordenadas x de los píxeles que pertenecen a la región R^j , m_y^j es la media de las coordenadas y y m_g^j es la media de las intensidades.

Una vez que se tiene la distribución inicial de las regiones con sus respectivos vectores de medias, se comienza un proceso iterativo que examina cada píxel i y lo asigna a aquella región R^j que minimiza el criterio E^{ij} , en el cual \mathbf{p}^i es el vector de características del píxel, \mathbf{m}^j es el vector de características de la región, \mathbf{W}^j es la matriz de pesos y k es el número de regiones.

$$E^{ij} = (\mathbf{p}^i - \mathbf{m}^j) \mathbf{W}^j (\mathbf{p}^i - \mathbf{m}^j) \quad j \in \{1, 2, \dots, k\} \quad (2.1)$$

Con la siguiente forma para la matriz \mathbf{W}^j :

$$\mathbf{W}^j = \begin{bmatrix} w_x^j & 0 & 0 \\ 0 & w_y^j & 0 \\ 0 & 0 & w_g^j \end{bmatrix} \quad (2.2)$$

Esta asignación de todos los píxeles se repite varias veces, recalculándose en cada una de ellas los vectores de medias. El proceso iterativo se detiene cuando el máximo desplazamiento del centro de coordenadas de una región es menor que una cierta constante T_s . El valor de esta constante influye sobre la calidad de la segmentación, pues determina el final de la adaptación de las regiones a la imagen. Cuanto más pequeño sea su valor, mayor número de iteraciones se producirán y, por tanto, mejor será la adaptación. Sin embargo, un valor excesivamente pequeño provocará un elevado coste computacional, con muy poca variación en la asignación de los píxeles durante las últimas iteraciones.

La matriz de pesos \mathbf{W}^j tiene dos funciones: permitir comparar medidas dispares como son la posición y la intensidad, y dar mayor énfasis a aquellas características de la región que presenten una menor varianza σ^2 . Para conseguir la segunda funcionalidad los pesos se fijan de forma que se minimice la distancia entre los píxeles de la región R^j , sujeto a la restricción $w_x^j w_y^j w_g^j = 1$.

$$w_x^j = c^j / \sigma_x^j, \quad w_y^j = c^j / \sigma_y^j, \quad w_g^j = c^j / \sigma_g^j \quad (2.3)$$

$$c^j = (\sigma_x^j \sigma_y^j \sigma_g^j)^{1/3} \quad (2.4)$$

$$(\sigma_x^j)^2 = \left(\frac{1}{N_{R^j} - 1} \right) \sum_{i=1}^{N_{R^j}} (x_i - m_x^j)^2 \quad (2.5)$$

donde N_{R^j} es el número de píxeles en la región R^j , y $(\sigma_x^j)^2$ es la varianza de la coordenada x en la región R^j (las varianzas $(\sigma_y^j)^2$ y $(\sigma_g^j)^2$ se calcularían de manera similar a la Ecuación 2.5).

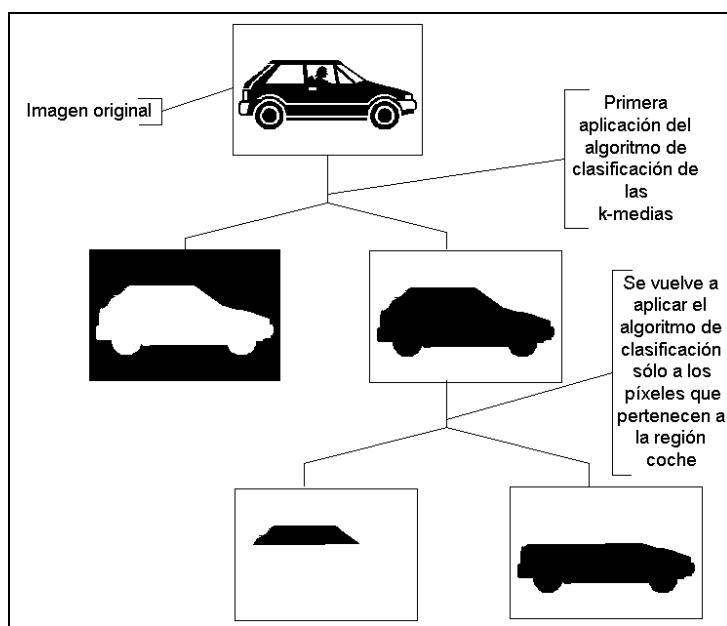


Figura 2.4: Árbol de representación del proceso de división de las imágenes

En ausencia de ruido, y para regiones uniformes, la desviación típica de la intensidad, σ_g^j , puede ser cero. En tal caso se asignarán los siguientes pesos para dar una importancia total a la intensidad de los píxeles: $w_x^j = w_y^j = 0$ y $w_g^j = 1$.

Cada región R^j tiene asociada su propia matriz \mathbf{W}^j . El uso de esta matriz posibilita que el algoritmo se adapte al contenido de la imagen, dando más importancia a aquel conjunto de características que caracterizan mejor las propiedades de una región particular. Estas matrices son recalculadas cada t iteraciones. Gracias a este mecanismo, el proceso requiere un menor número de iteraciones para alcanzar un estado estable.

2.2.3 Algoritmo jerárquico de división de regiones

La mera aplicación del algoritmo de las *k-medias*, aunque adapta las regiones a la imagen, mantiene una distribución uniforme de las mismas. Esto es debido a que el algoritmo no presenta una adaptación global a la imagen, sino que las regiones se adecuan a los píxeles de su vecindad. Sin embargo, las propiedades de la imagen varían en las distintas partes de la misma. Así, algunas partes contienen más detalle y deberían ser divididas en más regiones que otras, donde la distribución de intensidad es más uniforme, y por tanto no requerirían tantas regiones.

Para llevar a cabo una distribución de las regiones más acorde con las características de la imagen, completamos el método de clasificación por *agrupamiento* con una estrategia jerárquica. La Figura 2.4 pretende mostrar de una forma simple esta estrategia. En ella observamos que la imagen original consta básicamente de un coche

sobre un fondo uniforme. El primer paso de la segmentación clasifica los píxeles de la imagen en dos regiones usando el algoritmo descrito en el Apartado 2.2.2. Dado que la región vehículo no es uniforme, el siguiente paso consiste en dividirla en dos nuevas regiones y aplicar de nuevo el algoritmo de clasificación, eliminándose a continuación la región vehículo. Este mismo esquema se iría aplicando hasta alcanzar una división de la imagen donde todas las regiones fuesen uniformes.

El anterior párrafo describe de una forma simplista el algoritmo aplicado. Sin embargo, no difiere en gran medida del algoritmo real. Inicialmente, el algoritmo de *agrupamiento* se aplica a toda la imagen usando un pequeño número de regiones. Lo ideal sería conocer de antemano el número de objetos en la imagen y usar un número similar de regiones, pero, dado que en nuestro caso (monitorización de tráfico) esto no es posible, empleamos de seis a diez regiones iniciales. Éste es un número lo suficientemente pequeño como para no incrementar el coste computacional, y lo bastante grande como para conseguir segmentar inicialmente los detalles principales de la imagen. Una vez obtenida la primera clasificación, se pasa el test de homogeneidad a cada una de las regiones, y aquellas que no lo superan son divididas en cuatro regiones. Este test consiste en calcular la desviación típica σ_g^j de la intensidad de cada región R^j y ver si su valor es mayor que una constante $T_{\sigma_{max}}$.

Un detalle importante a considerar en la segmentación es el tamaño final de las regiones. No es interesante obtener regiones ni demasiado grandes, ni tampoco demasiado pequeñas. Las grandes pueden enmascarar en su interior objetos pequeños, los cuales pueden ser interesantes de segmentar. En cuanto a las regiones demasiado pequeñas, primeramente siempre es necesario establecer un máximo nivel de detalle, pues en caso contrario podríamos obtener ciertas regiones finales de tan sólo unos pocos píxeles. Y en segundo lugar, no debemos olvidar que a posteriori deseamos realizar una estimación de movimiento para las regiones obtenidas. Por tanto, debemos establecer un mínimo tamaño que permita obtener regiones que aporten suficiente información de cara al análisis de movimiento.

Para controlar el máximo y mínimo tamaño de una región se establecen dos constantes η y μ , respectivamente. De esta forma, una región con un número de píxeles superior a η siempre es dividida, mientras que una región de tamaño inferior a μ nunca lo es. Además, en el proceso de división, aquellas regiones que tienen un número de píxeles menor que dos veces la constante μ no se dividen en cuatro regiones, sino en dos. De esta forma tratamos de evitar que al dividir este tipo de regiones, tras el proceso de clasificación se obtengan varias regiones de tamaño inferior a la mitad de μ , hecho que inevitablemente se produciría si dividiéramos la región en cuatro. A continuación se resume la regla de división, en la cual N_{R^i} representa el número de píxeles de una región R^i :

$$\left\{ \begin{array}{ll} \text{No dividir} & \text{Si } N_{R^i} < \mu \\ \text{Dividir en dos} & \text{Si } N_{R^i} < 2\mu \text{ y } N_{R^i} \geq \mu \text{ y } \sigma_g^j > T_{\sigma_{max}} \\ \text{Dividir en cuatro} & \text{Si } N_{R^i} < \eta \text{ y } N_{R^i} \geq 2\mu \text{ y } \sigma_g^j > T_{\sigma_{max}} \\ \text{Dividir en cuatro} & \text{Si } N_{R^i} \geq \eta \end{array} \right.$$

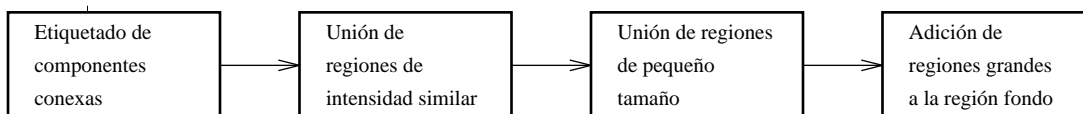


Figura 2.5: Las cuatro etapas del proceso de unión de regiones

De cara al resultado final de la segmentación no es en modo alguno trascendente establecer un valor preciso para las constantes η y μ . En nuestra aplicación se suele usar para η un valor un poco mayor al número de píxeles que esperamos que ocupe el mayor vehículo de la secuencia. Para μ usamos dos tipos de valores dependiendo de las características de la imagen, cuando ésta apenas presenta perspectiva se usa la mitad del número de píxeles que ocupe la proyección de un coche de tamaño normal, mientras que cuando sí existe perspectiva se usa el tamaño de la proyección más pequeña de vehículo que se desplace con un movimiento apreciable. Este último valor se debe a que en los vehículos muy alejados es prácticamente imposible apreciar movimiento sobre el plano de la imagen. Por lo tanto, de nada nos sirve segmentarlos en la imagen original si posteriormente la segmentación basada en movimiento los va a declarar como estáticos, lo cual hará que los una al fondo de la imagen.

2.2.4 Unión de regiones

El cometido de esta etapa final de la segmentación es unir aquellas regiones adyacentes con similares niveles de intensidad media. De esta forma se pretende obtener una división de la imagen con regiones lo más grandes posibles, pero con todos sus píxeles conectados. Es decir, que entre cualquier par de píxeles pertenecientes a una misma región siempre exista un camino interior a la región que permita llegar desde uno hasta el otro. La Figura 2.5 muestra los cuatro pasos del proceso de unión de regiones.

Lamentablemente, el algoritmo de clasificación empleado no garantiza por completo esa interconexión de los píxeles de una región. En cambio, sí que se obtienen grupos de puntos unidos en el espacio tridimensional usado para la clasificación (coordenadas e intensidad). Pero no asegura que en el espacio de la imagen los píxeles de una región sean necesariamente conexos.

Por tanto, el primer paso consiste en dividir cada región en regiones formadas por grupos conexos de píxeles, lo cual nos asegurará que cada región tan sólo pertenece a un único objeto del mundo real. En la práctica, la mayoría de los píxeles que componen una región se hayan conectados (entre un 70 y un 90 por ciento de los píxeles lo están), formando el resto de píxeles pequeños grupos de apenas unos pocos elementos localizados cerca del conjunto principal.

Una vez realizada esta tarea, el siguiente paso es intentar reducir el número de regiones. Para ello se unen aquellas regiones vecinas que presentan una intensidad similar. La similitud puede evaluarse de diversas formas. Una posibilidad es asumir que los valores de intensidad de las regiones siguen distribuciones de probabilidad

normal [Jain *et al.*, 1995], lo cual lleva al cálculo de una razón L

$$L = \frac{\sigma_0^{N_{R^1} + N_{R^2}}}{\sigma_1^{N_{R^1}} \sigma_2^{N_{R^2}}} \quad (2.6)$$

donde σ_0 es la desviación típica conjunta de las regiones R^1 y R^2 , σ_1 y σ_2 son las desviaciones típicas de las regiones R^1 y R^2 , y N_{R^1} y N_{R^2} son el número de píxeles de las regiones R^1 y R^2 . Cuando el ratio L es menor que un determinado umbral, entonces se puede considerar que sólo existe una región.

En la implementación de nuestro método utilizamos esta medida de similitud. Sin embargo, cuando el sistema debe trabajar en tiempo real, el anterior método de similitud es substituido por una simple comparación de las medias de intensidad de las regiones. Aunque se producen ligeras variaciones, esta simplificación del método nos evita el cálculo de todas las varianzas conjuntas entre regiones adyacentes.

Por último, se toman aquellas regiones que son demasiado pequeñas como para ser significativas y se unen a la región adyacente más similar o a la que presenta un mayor contorno de vecindad. En este último paso, las regiones son ordenadas según el número de píxeles que las componen, empezando por las más pequeñas. Tomando como referencia la constante μ definida en la regla de división de regiones, se han definido experimentalmente varias proporciones de este valor, las cuales permiten diferenciar la importancia de las regiones pequeñas. Cuando el tamaño de una región es menor que $\mu/4$, sus píxeles se asignan a la región con mayor vecindad. Cuando el tamaño es mayor que $\mu/4$ y menor que $\mu/2$ entonces la región es asignada a la más similar.

$$\begin{cases} \text{Asignación por vecindad} & \text{Si } N_{R^i} < \mu/4 \\ \text{Asignación por similitud} & \text{Si } \mu/4 \leq N_{R^i} \leq \mu/2 \end{cases}$$

Esta distinción se hace porque se considera que los grupos de píxeles de tamaño menor a $\mu/4$ no representan entidades significativas, pudiendo corresponder a pequeños detalles sin importancia, o quizás a efectos luminosos, ruido o transiciones en los bordes de las regiones. Por ello, su intensidad no puede considerarse como una medida de similitud, y lo que se intenta es incluirlas en la región que las rodea total o parcialmente. Otra posibilidad hubiera sido el desestimar este tipo de regiones, y no realizar un estudio de su movimiento. Sin embargo, esta opción fue desestimada tras observar en algunos experimentos que su no inclusión podía modificar las relaciones de vecindad entre otras regiones más importantes.

En la aplicación del método de segmentación a nuestro problema partimos con una ventaja importante que nos permitirá simplificar aún más la segmentación final. Sabemos que el parámetro η representa el tamaño del mayor vehículo con el que nos podemos encontrar en una imagen. Por tanto, podemos unir todas aquellas regiones con un número de píxeles superior a ese valor, pues sabemos que forman parte del fondo estático.

2.3 Resultados

El algoritmo presentado en este capítulo ha sido probado sobre distintos tipos de imágenes, principalmente imágenes de tráfico tanto urbano como de carretera. Las Figuras 2.6 y 2.7 muestran varias fases del proceso de segmentación sobre dos imágenes de tráfico. Dichas imágenes aparecen en las subfiguras 2.6.a y 2.7.a. La primera forma parte de una secuencia grabada en la carretera N-234 a su paso por las cercanías de la Universitat Jaume I. En esta imagen pueden verse diez vehículos, nueve sobre la carretera y uno más circulando a velocidad lenta por un camino vecinal que discurre paralelo a la misma. La segunda imagen corresponde a una escena de tráfico urbano donde un vehículo está aminorando su velocidad al llegar frente a un semáforo. En esta imagen pueden observarse distintos coches que se encuentran aparcados a ambos lados de la calzada. Esta imagen corresponde a una secuencia tomada en la avenida Blasco Ibañez de Valencia.

La imagen de la Figura 2.6 corresponde a un tamaño de 384x288 y los parámetros usados para su segmentación fueron: número inicial de regiones= 12, $t = 9$, $T_s = 1.0$, $T_{\sigma_{max}} = 12$, $\eta = 3500$ y $\mu = 150$. Por lo que respecta a la imagen de la Figura 2.7, su tamaño es 192x144 y los parámetros usados fueron: número inicial de regiones= 6, $t = 9$, $T_s = 1.0$, $T_{\sigma_{max}} = 12$, $\eta = 2500$ y $\mu = 100$. La variación en los parámetros se debe únicamente al tamaño de las imágenes, lo cual obviamente influye en el tamaño de los vehículos. Se reducen los parámetros μ y η y el número de regiones iniciales, el resto de parámetros permanecen iguales.

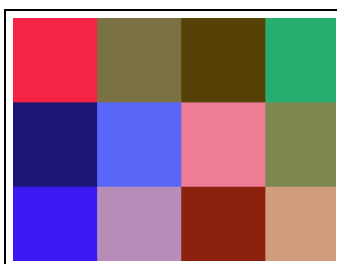
La fijación de parámetros se realiza en el lugar donde se instala el sistema, partiendo de las combinaciones anteriores, para sus respectivos tamaños de imagen. Se realizan una serie de segmentaciones y se adaptan los parámetros según los resultados. Habitualmente, los cambios sólo suelen afectar a los valores de μ y η . Observando el tamaño de los vehículos, se intenta tomar el menor valor posible para η y el mayor posible para μ para reducir el número de regiones resultantes. El parámetro σ puede ser aumentado cuando la iluminación es buena o reducido cuando baja el contraste de la imagen. El parámetro T_s puede ser aumentado cuando los vehículos ocupan un espacio considerable con respecto al tamaño de la imagen. En estos casos una segmentación de menor calidad apenas afectará al resultado final. En cualquier caso, una vez fijados los parámetros ya no es necesario volver a modificarlos, con lo que la segmentación podrá funcionar sin la intervención de un operador humano.

Las subfiguras 2.6.b y 2.7.b corresponden a las fases de inicialización del proceso. Aquí, la imagen es dividida en un reducido número de regiones distribuidas uniformemente en la imagen. Una vez hecho esto, tal y como se comentó en la sección anterior, se procedía a calcular los vectores de medias de cada región y se iniciaba el proceso de clasificación de los píxeles. El resultado de esta clasificación puede verse en las subfiguras 2.6.c y 2.7.c. En ellas puede apreciarse como a pesar del pequeño número de regiones usado, ya empieza a distinguirse la forma de la mayoría de los objetos de la imagen.

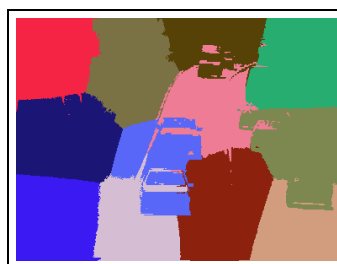
En las dos siguientes fases de la segmentación, subfiguras 2.6.d y 2.7.d, puede



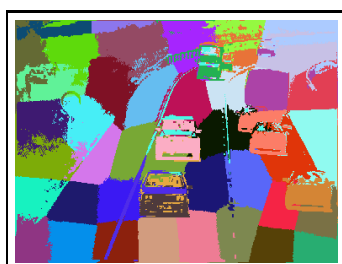
(a) Imagen original.



(b) Inicialización del proceso.



(c) Aplicación del algoritmo de clasificación sobre la división inicial.



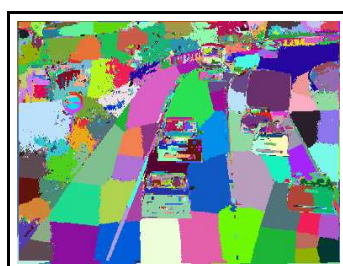
(d) Clasificación de los píxeles tras la primera subdivisión.



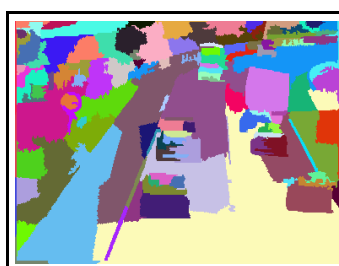
(e) Clasificación de los píxeles tras la segunda subdivisión.



(f) Clasificación de los píxeles tras la tercera subdivisión.



(g) División de la imagen antes del proceso de unión de regiones.



(h) Segmentación de la imagen tras realizar la unión de regiones.

Figura 2.6: Segmentación de una imagen de tráfico en una carretera.

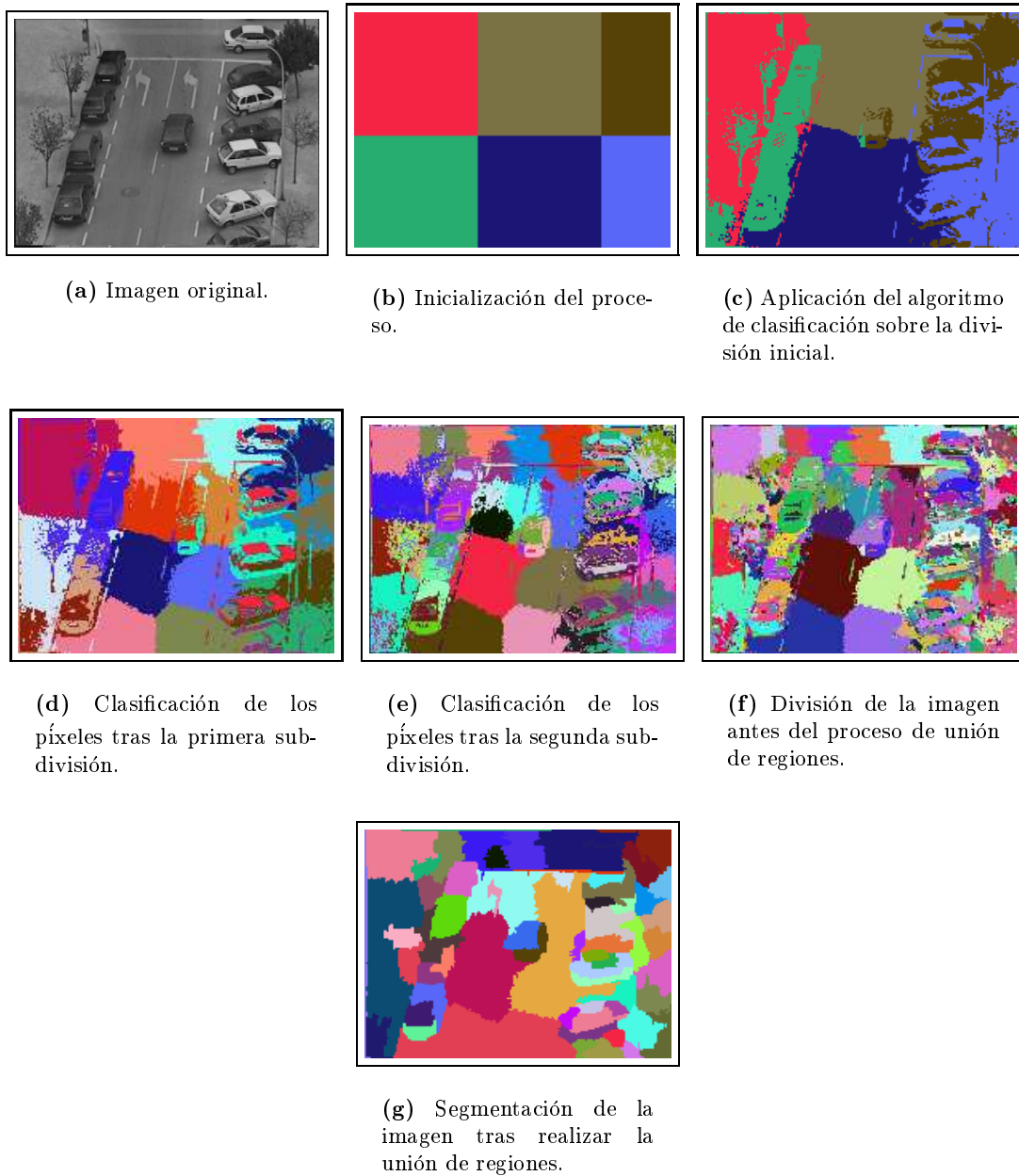
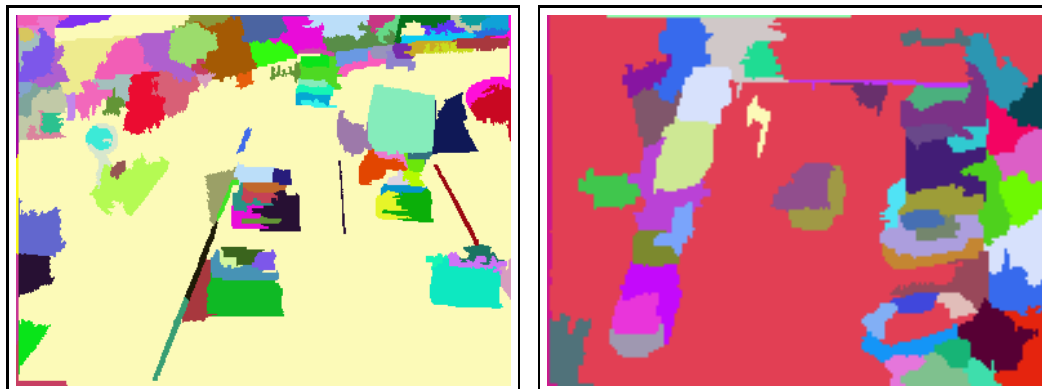


Figura 2.7: Segmentación de una imagen de tráfico urbano.



(a) Segmentación final de la imagen de la carretera.

(b) Segmentación final de la imagen urbana.

Figura 2.8: Resultado final de la segmentación una vez unidas las regiones de tamaño mayor a η .

observarse como todas las regiones han sido subdivididas y los píxeles reclasificados entre las nuevas regiones. En la siguiente subdivisión y clasificación, subfiguras 2.6.e y 2.7.e, puede observarse que algunas de las regiones obtenidas en la anterior clasificación permanecen intactas desde entonces. Esto es debido a que han alcanzado una uniformidad con respecto a la intensidad y un tamaño menor que el parámetro η . Esto se produce principalmente en las regiones situadas sobre la proyección de la carretera, donde apenas existen cambios de intensidad.

Las subfiguras 2.6.f y 2.7.f muestran el resultado tras la última subdivisión y clasificación. Después de esta última ya no se producen nuevas divisiones, pues la totalidad de las regiones cumplen el criterio de homogeneidad. Las siguientes subfiguras, 2.6.h y 2.7.g, corresponden al resultado final de la segmentación de las imágenes. Aquí se ha realizado la unión de aquellas regiones adyacentes que tienen una intensidad media similar.

La Figura 2.8 representa la etapa que hemos añadido para el caso de la segmentación de tráfico. Estos resultados se muestran en una figura aparte, ya que no necesariamente deberían formar parte de un algoritmo de segmentación. La última parte en realidad es sólo aplicable a un caso como el nuestro, donde se sabe que los objetos tienen un tamaño relativamente pequeño comparado con el tamaño de la imagen. La subfigura 2.8.a corresponde al estado final de la segmentación de la imagen de tráfico en carretera, mientras que la subfigura 2.8.b es la respectiva segmentación final de la imagen urbana.

Puede observarse en las dos subfiguras (2.8.a y 2.8.b) que con esta etapa conseguimos formar una región de gran tamaño, la cual aglutina una gran parte del total del fondo estático de la imagen, aunque lo más importante es que prácticamente se

ha conseguido aislar a los objetos móviles. Esto es de gran importancia, pues, como comentaremos en el siguiente capítulo, a menudo una región de tipo carretera que se haya próxima a un vehículo produce una respuesta falsa en la estimación del movimiento. Esta falsa respuesta es inducida por la región vehículo, debido a que al avanzar está ocultando parte de la región carretera. Esto provoca que a veces un falso movimiento que compense la pérdida de esos píxeles produzca un menor error que el movimiento verdadero, el cual es cero. Se profundizará sobre esta problemática en el próximo capítulo, pero en cualquier caso, parece claro que cuanto antes consigamos unir una región al fondo, mejor (por supuesto sólo si realmente pertenece al fondo). Una vez unida al fondo sabemos que su estimación de movimiento será correcta (igual a cero), y que no se la unirá por error a una región vecina con movimiento. Además, esto representa un ahorro fundamental en el tiempo de cálculo, pues para la región *fondo*, no es necesario realizar una estimación de movimiento, ya que al permanecer la cámara estática, éste es igual a cero.

El último paso consigue que prácticamente toda la escena quede dividida en dos grupos principales de píxeles, los que corresponden a la carretera, y los que corresponden a los vehículos. La principal excepción la constituyen las señales horizontales de la calzada, pues ni por su intensidad, ni por su tamaño pueden unirse al resto de la carretera. Su intensidad es claramente diferenciada y su tamaño no es ni tan pequeño como para ser despreciable, ni mayor que el de un vehículo, con lo que no pueden unirse al fondo. En este tipo de regiones, como veremos en el próximo capítulo, sólo las líneas continuas pueden causar más dificultades a la estimación de movimiento. Este es un caso similar al que hemos comentado sobre las regiones carretera que son vecinas a regiones vehículo. En cuanto al resto de señales horizontales, flechas o líneas discontinuas, su estimación siempre es correcta, dado que longitudinalmente están claramente delimitadas.

2.4 Conclusiones

En este capítulo se ha presentado un método no supervisado de segmentación de imágenes. Se ha empleado como criterio de homogeneidad la intensidad de los píxeles, de esta forma, la imagen se divide en regiones de intensidad uniforme. Este enfoque es particularmente útil dentro del ámbito de aplicación del algoritmo, escenas de tráfico. En ellas, tanto los objetos que se mueven, vehículos, como la mayor parte del fondo de la imagen, calzada, se caracterizan por tener texturas casi uniformes, apareciendo como áreas de nivel de gris casi constante.

El método utiliza una estrategia jerárquica en la que la imagen es inicialmente dividida en unas pocas regiones, las cuales son sucesivamente subdivididas hasta que su distribución de intensidad interna es uniforme. Esta estrategia permite, por una parte, adaptar el número de regiones a las características de la imagen, y por otro, reducir el tiempo de cálculo, al simplificar el problema de clasificación de los píxeles, clasificación basada en la técnica de las *k-medias*. De este modo, las regiones se adaptan a la imagen sin requerir un conocimiento previo de la misma.

Una vez acabadas las divisiones de las regiones, una etapa posterior del algoritmo une aquellas regiones que presentan una intensidad similar. Con esto se consigue evitar la sobre-segmentación de la imagen, permitiéndose que regiones surgidas de subdivisiones de distintas regiones puedan finalmente formar parte de una misma región.

Al final del algoritmo se ha incluido una última etapa consistente en unir aquellas regiones de tamaño mayor al que pueda alcanzar un vehículo. El resultado es el surgimiento de una gran región que engloba la mayor parte de las zonas uniformes de la imagen, pero que no corresponden a vehículos. En lo que respecta a la zona de mayor interés en la imagen, la carretera, esto se traduce en que queda prácticamente dividida en dos tipos de píxeles, los de clase “carretera” y los de clase “vehículo”. Sin lugar a dudas, este resultado es un gran salto hacia nuestro objetivo final: la extracción de aquellas regiones que se mueven. Efectivamente, falta tomar la decisión de qué regiones móviles corresponden a cada vehículo, pero esto será realizado con la ayuda del análisis de movimiento, tema que se aborda en el siguiente capítulo.

El método de segmentación presentado en este capítulo no es el único que puede emplear el sistema de monitorización de tráfico. Puede ser aceptable cualquier otro método que cumpla las restricciones impuestas al nuestro:

- Ser no supervisado
- Adaptar el número final de regiones a las características de la imagen
- Que cada una de las regiones resultantes sólo corresponda a un único objeto, aunque un objeto pueda constar de varias regiones.

Capítulo 3

Análisis de movimiento

Contenido

3.1	Introducción	43
3.1.1	Métodos basados en sustracción de imágenes	44
3.1.2	Métodos basados en el gradiente	47
3.1.3	Métodos basados en características	50
3.1.4	El método propuesto	54
3.1.5	Estructura del capítulo	56
3.2	Detección de movimiento	57
3.3	Reducción del número de regiones	61
3.4	Estimación de movimiento	64
3.4.1	El algoritmo básico de estimación de movimiento	64
3.4.2	La estrategia multiresolución	67
3.4.3	Estimación de movimiento con precisión subpíxel	67
3.4.4	Aplicación de estadísticos robustos	67
3.5	Segmentación	70
3.5.1	Segmentación de regiones con movimiento coherente	70
3.5.2	Segmentación con reestimación de movimiento	71
3.6	Resultados	72
3.7	Conclusiones	85

3.1 Introducción

El análisis de movimiento proporciona valiosa información a un sistema basado en visión. Con la ayuda de esta información es más fácil interpretar una escena, permitiendo tanto detectar y segmentar los objetos que se hallan en movimiento, como conocer y predecir su comportamiento actual y futuro.

La idea básica es que si agrupamos aquellos píxeles que se mueven con un mismo movimiento, entonces habremos extraído la forma de los objetos que se están moviendo en la imagen. En el caso de la monitorización de tráfico, cada grupo de píxeles corresponderá a un vehículo diferente. Extrayendo su forma y conociendo su comportamiento en la imagen podemos calcular los parámetros necesarios para llevar a cabo nuestras tareas de control.

Nuestro objetivo en esta fase del algoritmo es tomar dos imágenes consecutivas de una secuencia y analizar los cambios que la primera de ellas ha experimentado para llegar a convertirse en la segunda. Para realizar esto se emplearán diversas técnicas, todas ellas relacionadas con el movimiento. En primer lugar una detección de movimiento. Con esta técnica se pretende averiguar qué zonas de la imagen son las que han experimentado una variación, y por lo tanto son las que pueden corresponder a proyecciones de objetos en movimiento. La detección no aporta información de alta calidad, pero permite reducir el coste computacional del algoritmo al discriminar entre las zonas que han cambiado y las que no. Dado que lo que buscamos son objetos móviles no es necesario aplicar el algoritmo a los píxeles que no se han movido.

El segundo tipo de técnicas a emplear son las de estimación de movimiento. Con ellas se estima cuál ha sido el desplazamiento que ha experimentado cada uno de los píxeles de la imagen. El último tipo de técnicas a emplear son las de segmentación de movimiento. Estas técnicas agrupan los píxeles que se desplazan con el mismo movimiento, lo cual, como ya hemos dicho, se realiza con el objetivo de extraer la forma de los objetos que se están moviendo.

Los métodos de análisis de movimiento son normalmente divididos en tres grandes grupos [Sonka *et al.*, 1993], [Holden, 1993], [Thévenaz, 1990]:

- Métodos basados en substracción de imágenes
- Métodos basados en el gradiente
- Métodos basados en características

3.1.1 Métodos basados en substracción de imágenes

Los métodos basados en substracción de imágenes se usan principalmente para discriminar entre dos grupos de píxeles: los que pertenecen al fondo estático y los que corresponden a los objetos que se están moviendo. Dada la escasa información que proporcionan, no permiten otro tipo de tareas como, por ejemplo, estimar los vectores de movimiento de los píxeles. Sin embargo, su más interesante ventaja es su bajo coste de computación.

En este tipo de métodos, se crea una imagen diferencia al restar dos imágenes. La resta se realiza píxel a píxel y tomando el valor absoluto del resultado. Asumiendo que el valor de los píxeles estáticos no cambia, el resultado de restar un mismo píxel en dos imágenes distintas debería ser cero. Sin embargo, debido a la presencia de ruido, lo que se suele obtener es un valor cercano a cero, por lo que es necesario utilizar un

umbral. De esta forma los píxeles con un valor absoluto de la resta superior a ese umbral son declarados como no estáticos.

Este tipo de técnicas se dividen en dos clases dependiendo de qué dos imágenes de la secuencia se restan. Dichas imágenes pueden ser bien dos consecutivas, o bien, la imagen actual a segmentar y una imagen estática. Esta imagen corresponde a una instantánea de la escena en la cual sólo aparece el fondo estático sin ninguno de los objetos que se moverán. En cualquiera de ambos casos, este tipo de métodos imponen dos requerimientos muy restrictivos: que la cámara permanezca estática y que la iluminación sea constante.

La utilización de este tipo de métodos es especialmente apta para codificación y compresión de vídeo [Haskell *et al.*, 1972], ya que permite detectar las zonas de la imagen que deben ser transmitidas para generar una imagen a partir de la anterior. Sin embargo, han sido usados en diversas aplicaciones, generalmente en conjunción con otros tipos de técnicas: tráfico [Waterfall y Dickinson, 1984], imagen médica [Abdel-Malek *et al.*, 1990], imágenes de satélite [Eghbali, 1979], seguimiento de personas [Segen y Pingali, 1996], etc.

Los primeros trabajos realizaban un simple procesamiento píxel a píxel. Sin embargo, Jain [Jain, 1985] modificó el método para considerar bloques de píxeles, lo cual lo dotaba de una mayor robustez. Hsu et al. [Hsu *et al.*, 1984] propusieron un planteamiento estadístico para el problema de la detección de movimiento entre dos imágenes consecutivas. En él se recorren las dos imágenes con dos ventanas, una para cada imagen, de igual tamaño y situadas sobre la misma posición, y se testean dos hipótesis: que no exista cambio temporal, con lo que las funciones de intensidad de ambas ventanas deben tener la misma parametrización; y que sí exista cambio temporal, con lo que los parámetros de las funciones de ambas ventanas deberán ser distintos. Wiklund y Granlund [Wiklund y Granlund, 1987] emplearon tres imágenes consecutivas, actual, anterior y siguiente, para alcanzar una mayor robustez. De esta forma, la detección sobre la imagen actual es el resultado de integrar los mapas de cambio temporal calculados para las imágenes anterior y actual, y para las imágenes actual y siguiente. En [Bouthemy y Lalade, 1993] se unifican los dos enfoques anteriores: el uso de tres imágenes y el planteamiento estadístico. En él se realiza un etiquetado estadístico de los píxeles de la imagen mediante un enfoque bayesiano y un proceso de relajación. Se emplean campos aleatorios de Markov para obtener máscaras de los objetos en movimiento, usando la detección anterior para inicializar el procesamiento de la imagen actual. Gil et al. [Gil y Pun, 1993, Gil *et al.*, 1994] han desarrollado una estrategia multiresolución, donde se realiza un procesamiento en cada nivel de resolución de la imagen, y los resultados obtenidos en un nivel son incorporados al siguiente nivel de mayor resolución. En este mismo trabajo se incorpora la búsqueda de cambios de signo en las imágenes diferencia para detectar bordes débiles. El resultado es una máscara alrededor de la zona ocupada por cada objeto móvil. Sin embargo, esta máscara suele ser demasiado amplia, por lo que es necesario aplicar un posterior refinamiento.

Un tipo distinto de métodos son los que emplean una imagen referencia correspon-

diente al fondo estático de la imagen. Esto permite simplificar el problema, ya que los mapas de cambio temporal se corresponden con las máscaras de los objetos en movimiento. Sin embargo, este método plantea la dificultad de que la imagen referencia debe ser actualizada constantemente para poder reflejar posibles variaciones tanto en el fondo como en las condiciones de iluminación. La actualización de los píxeles no puede realizarse por una mera sustitución del valor de un píxel por el valor más reciente, por lo que es frecuente el uso de filtros. En [Donohoe *et al.*, 1988] se emplea un filtro lineal diseñado mediante pesos, mientras que en [Karmann y von Brandt, 1990] se usa un filtro de Kalman. Este último método ha sido usado en [Kilger, 1992] y [Koller *et al.*, 1994b]. En [Koller *et al.*, 1994b] el método es mejorado con la aplicación previa de un filtro gaussiano que reduce la influencia del ruido.

No podemos finalizar el comentario acerca de las imágenes diferencia sin reseñar brevemente otro de sus importantes usos: la estimación de las trayectorias más frecuentes. La idea básica es acumular las diferencias obtenidas para cada píxel de la imagen. De esta forma, y tras el procesamiento de un gran número de imágenes de una secuencia, los valores acumulados más altos corresponderán al recorrido que más frecuentemente realizan los objetos móviles de la imagen. Steward *et al.* [Steward *et al.*, 1994] emplean esta idea para detectar los carriles de las autopistas y así reducir el espacio de búsqueda de los vehículos. En la Figura 3.1 pueden verse varias fases de la aplicación de este método. Las dos primeras subfiguras son dos imágenes de la secuencia. Las subfiguras 3.1.c y 3.1.d muestran los valores acumulados para cada píxel en dos instantes del procesamiento. En ellas los valores más claros corresponden a píxeles donde más veces se ha detectado movimiento. Las subfiguras 3.1.e y 3.1.f corresponden a los dos últimos pasos del algoritmo: la umbralización y la aplicación de una operación morfológica que permite eliminar los pequeños grupos de píxeles. Por su parte, en [Ferryhough *et al.*, 1996] se emplea esta técnica para detectar las trayectorias más comunes que siguen las personas dentro de determinados escenarios objeto de análisis.

El método de Steward *et al.* [Steward *et al.*, 1994] representa una automatización del proceso de búsqueda de carriles. En una aplicación como la nuestra, es de gran utilidad el hecho de que el método desarrollado sepa dónde se encuentran, ya que esto puede llegar a reducir el espacio de la imagen donde aplicar el proceso hasta en un 30%. Sin embargo, presentan un problema: los arcones de las carreteras quedan fuera del mapa de actividad, ya que son zonas donde raramente circulan vehículos. Además, como vemos en la subfigura 3.1.f, también alguna zona central de la carretera queda fuera del mapa, con lo que cuando se produzca un adelantamiento por esta zona una parte del vehículo no será segmentada. A pesar de que en alguno de nuestros trabajos anteriores [Badenas, 1996, Badenas *et al.*, 1997] hemos usado este método para reducir el coste computacional, en la actualidad preferimos usar mapas de actividad generados manualmente, ya que evitan los problemas mencionados, y pueden considerarse como una parte del proceso de calibración inicial.

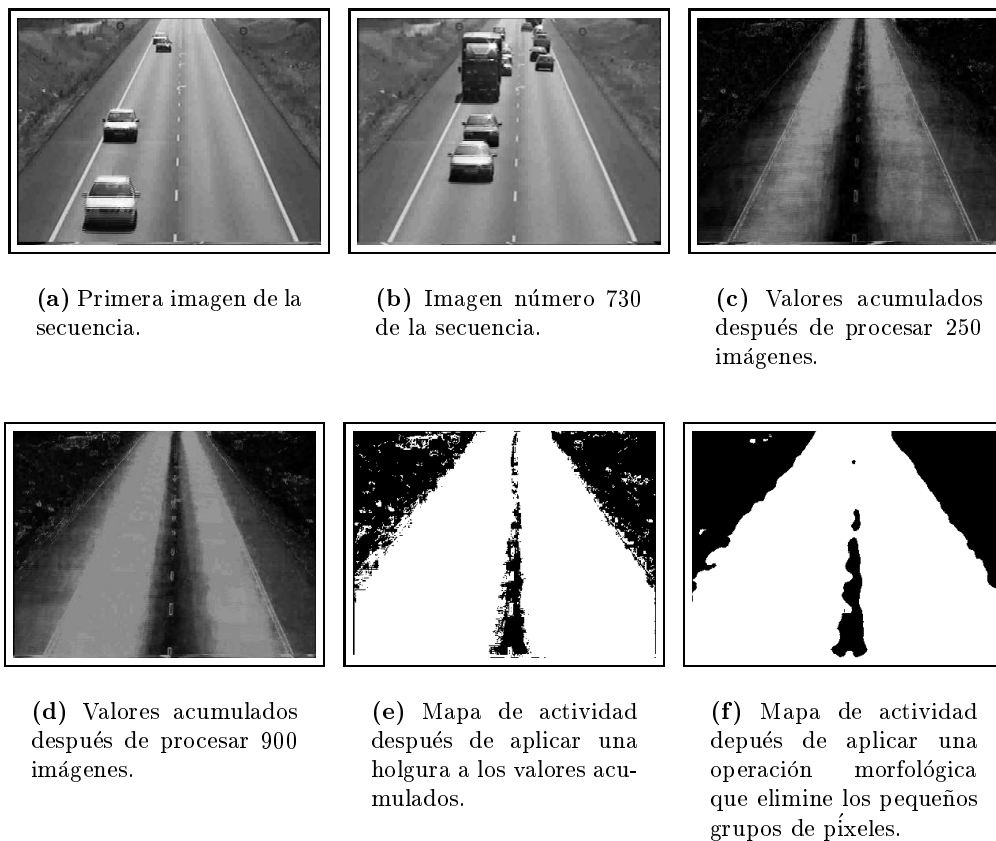


Figura 3.1: Varias etapas del método de Steward et al. para detección de los carriles de una carretera.

3.1.2 Métodos basados en el gradiente

Los métodos basados en el gradiente explotan la relación entre los gradientes temporal y espacial de la imagen. Se basan en la conocida ecuación de la restricción del flujo óptico [Horn y Schunck, 1981, Schunck, 1989, Nagel, 1983b, Nagel y Enkelmann, 1984, Verri y Poggio, 1987], más comúnmente conocida como la ecuación del flujo óptico:

$$I_x(\mathbf{x}, t)u + I_y(\mathbf{x}, t)v + I_t(\mathbf{x}, t) = 0 \quad (3.1)$$

Dicha ecuación relaciona el campo de velocidad ($u = \frac{dx}{dt}$, $v = \frac{dy}{dt}$) con las variaciones temporales y espaciales de la intensidad de la imagen $I(\mathbf{x}, t)$. En la anterior ecuación, las derivadas temporal I_t y espaciales I_x , I_y de la intensidad I se calculan directamente a partir de la imagen. En cada punto de la imagen existen dos incógnitas que deben ser calculadas u y v , y tan sólo una ecuación. Por lo tanto, los componentes de la velocidad en cualquier punto de la imagen no pueden ser determinados de forma

única a partir de la ecuación del flujo óptico, a no ser que se añadan otro tipo de restricciones. Este problema es conocido como el *problema de la apertura*.

“... the so-called 'problem of aperture' which also exists in human vision. This is related to the impossibility of unequivocally recovering the motion direction if the object is observed through an aperture smaller than the object itself. In this context, the features on the object under observation (such as textures - patterns) are not enough to allow the perception of the transversal component of the object motion.” [Nesi, 1993]

La ecuación del flujo óptico no proporciona un vector único de velocidad, por lo que se hace necesaria la utilización de una restricción de regularización. Un ejemplo de dicho tipo de restricción es la propuesta por Horn y Schunck [Horn y Schunck, 1981], la cual asume que la variación del vector de velocidad dentro de una vecindad es pequeña, *restricción de uniformidad*. Esto permite plantear una nueva restricción, con lo que el flujo óptico puede calcularse resolviendo ambas ecuaciones: ecuación del flujo óptico y restricción de regularización. El procedimiento se reduce a minimizar el error $E_1(\mathbf{x})$ sobre toda la imagen,

$$E_1(\mathbf{x}) = \int \int_{\text{Imagen}} (I_x(\mathbf{x}, t)u + I_y(\mathbf{x}, t)v + I_t(\mathbf{x}, t))^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2) dx dy \quad (3.2)$$

donde u_x , u_y , v_x y v_y son las derivadas parciales de las componentes de la velocidad u y v .

El coeficiente λ equilibra los términos del uniformidad y flujo óptico. Dependiendo del valor de λ , se puede obtener una solución que potencie más o menos la uniformidad del campo de flujo (a mayor λ , mayor uniformidad).

El método de Schunck [Schunck, 1989] se basa en la misma ecuación de la restricción del flujo de la imagen, pero, en cambio, expresa la restricción en forma de coordenadas polares. Cuando se aplica la ecuación del flujo óptico a un grupo de píxeles de una región, cada una de las ecuaciones define una línea en el sistema de coordenadas (u, v) . Asumiendo que los vectores de movimiento deberían ser similares dentro de una pequeña vecindad, es decir, que existe un movimiento coherente dentro de una región alrededor de un píxel, entonces las líneas de la restricción generadas por los píxeles de la vecindad deberían intersectar en un mismo punto. De esta forma, los puntos de intersección se hallarán agrupados alrededor de aquel punto del espacio u, v que defina el verdadero vector de flujo del píxel. Ambos métodos [Horn y Schunck, 1981, Schunck, 1989] sólo funcionan satisfactoriamente en regiones donde se cumplan dos restricciones:

- que existan suficientes variaciones de intensidad,
- que el desplazamiento entre imágenes consecutivas de la secuencia sea pequeño.

La ecuación del flujo óptico se deduce a partir de la expansión por medio de las series de Taylor de la superficie de intensidad de la imagen alrededor de un punto (x, y) ,

$$I(x + dx, y + dy, t + dt) = I(\mathbf{x}, t) + I_x(\mathbf{x}, t)dx + I_y(\mathbf{x}, t)dy + I_t(\mathbf{x}, t)dt + O(\delta^2) \quad (3.3)$$

Cuando las variaciones dx , dy y dt son muy pequeñas, los términos de orden más alto $O(\delta^2)$ pueden despreciarse, con lo cual se obtiene la ecuación del flujo óptico (3.1). Algunas veces, especialmente para secuencias donde los objetos se mueven rápidamente, o la cámara no permanece estática, esta aproximación resulta pobre, por lo que los términos de orden más alto, $O(\delta^2)$, deberían ser tenidos en cuenta. Nagel [Nagel, 1983b] propuso considerar los términos de segundo orden.

Varios autores propusieron que la restricción de uniformidad debería imponerse a lo largo de los bordes de la imagen, y no de forma perpendicular a ellos. Hildreth [Hildreth, 1984] minimizó la función criterio de Horn y Schunck (3.2) a lo largo del contorno de los objetos. Nagel y Enkelmann [Nagel y Enkelmann, 1986, Enkelmann, 1988, Nagel, 1987] introdujeron el concepto de uniformidad direccional, el cual permite suprimir la restricción de uniformidad en la dirección del gradiente espacial de la imagen. El flujo óptico sólo es uniformizado en la dirección perpendicular al gradiente de intensidad de la imagen, con lo que los bordes de discontinuidad se preservan mucho mejor. En [Nagel, 1990], Nagel extiende al dominio temporal esta *restricción de uniformidad orientada*, usando no solo derivadas parciales espaciales, sino también derivadas parciales espacio-temporales de la función de intensidad de imagen para determinar la mejor dirección en la cual aplicar la uniformidad. Overington [Overington, 1987] utiliza la ecuación del flujo óptico para calcular la componente normal del flujo en los bordes. A partir de ese cálculo se realiza una segmentación por medio de la búsqueda de discontinuidades en esa componente. En [McLauchlan *et al.*, 1992, Nelson, 1991] se presentan otros dos métodos de segmentación basada en la ecuación del flujo óptico, si bien sólo permiten detectar qué píxeles están en movimiento (detección de movimiento).

Uno de los principales problemas de los métodos anteriores es la imposibilidad que tienen de abordar secuencias donde los objetos experimenten grandes desplazamientos. Para abordar este tipo de secuencias varios autores han empleado estrategias multiresolución [Enkelmann, 1988, Heitz y Bouthemy, 1993, Battiti *et al.*, 1991]. En ellas se construye una pirámide de imágenes a distinta resolución, de forma que se comienza a estimar el movimiento en los niveles de menor resolución. Las estimaciones realizadas en un nivel son usadas como estimación inicial en el siguiente nivel de mayor resolución. De esta forma, por un lado, se pueden abordar mayores desplazamientos, y por otro, se consigue reducir el tiempo de proceso.

Otra importante dificultad es la presencia de oclusiones. El manejo de oclusiones en los métodos basados en el gradiente es doblemente difícil. Por un lado, debido a la presencia de discontinuidades de movimiento, y por otro, debido a que en las áreas donde se presenta este fenómeno, la información de movimiento

proporcionada por la ecuación del flujo óptico no será válida. Dos estrategias diferentes para abordar este tipo de situaciones son: la detección de regiones ocultas y ocluyentes [Heitz y Bouthemy, 1993], y el uso de técnicas de estimación robusta [Ong y Spann, 1995]. En [Heitz y Bouthemy, 1993] se introduce un test que permite comprobar la validez local de la ecuación del gradiente. Ong y Spann en [Ong y Spann, 1995] presentan un método que aplica la ecuación de la restricción del flujo de imagen a regiones de píxeles. En él se asume una transformación afín bidimensional como modelo de movimiento, y se emplea el estimador robusto *least-median square* para resolver el resultante sistema de ecuaciones sobredeterminado. Este estimador robusto permite rechazar aquellos píxeles de la región para los cuales no se puede asignar ningún vector de movimiento debido a que son fruto de la oclusión, o de algún otro movimiento ajeno a la región. Este tipo de píxeles son comúnmente conocidos por su nombre en inglés: *outliers*.

3.1.3 Métodos basados en características

Este tipo de métodos extraen características de la imagen, tales como puntos de interés, líneas, contornos o regiones, mediante las cuales determinan un campo disperso de movimiento. Se analiza el cambio de posición de las características a través de varias imágenes para estimar su desplazamiento.

A menudo este tipo de métodos son planteados mediante un esquema de correspondencia. Se intenta hacer corresponder las características extraídas en la imagen de referencia con las características encontradas en el resto de imágenes de la secuencia. Este tipo de esquemas son habitualmente empleados con características simples, tales como puntos o líneas, debido a que la información que aporta cada característica es muy reducida, reduciéndose al ámbito de una pequeña vecindad.

Barnard et al. [Barnard, 1979, Barnard y Thompson, 1980] presentaron uno de los primeros y más conocidos métodos basados en correspondencia. En él, un procedimiento iterativo recalcula las probabilidades de correspondencia entre los puntos de interés encontrados en dos imágenes consecutivas. El cálculo de probabilidades se basa en la similitud local de los puntos y en el empleo de heurísticas que permiten reducir el espacio de búsqueda: máxima velocidad, coherencia de movimiento entre puntos vecinos y correspondencia biyectiva. Este mismo método fue empleado por Thompson y Pong [Thompson y Pong, 1987] en un trabajo de segmentación de imágenes usando una cámara en movimiento. Aquí se asume que el movimiento de la cámara es conocido y que la superficie de la escena es plana. Haciendo uso de estas restricciones y buscando discontinuidades locales en el flujo de movimiento, se detectan objetos en movimiento. En [Castellow *et al.*, 1987] se utilizan los píxeles de los bordes extraídos mediante el método de Canny, *edgels*, para encontrar el flujo óptico de los bordes. Este método fue originalmente desarrollado por Scott [Scott, 1987]. En los *edgels* de menor curvatura sólo es posible estimar la componente paralela del flujo. Sin embargo, en los *edgels* cercanos a las partes de más alta curvatura sí es posible recuperar ambas componentes. Así, mediante un método iterativo de relación basado en correspondencia, el movimiento es propagado al resto de *edgels* del

borde. Este método ha sido usado por Murray et al. [Murray *et al.*, 1987] en la reconstrucción de objetos tridimensionales (estructura a partir de movimiento). En [Medioni y Nevatia, 1984] podemos encontrar otro método basado en asignación de probabilidades a las correspondencias por medio de un algoritmo de relajación, el cual, al igual que los anteriores, presenta la desventaja de su elevado coste computacional. En [Gong y Brady, 1990] y [Wang y Brady, 1990] se presentan dos métodos que, al igual que en [Castellow *et al.*, 1987], propagan el movimiento estimado para los puntos de los bordes de mayor curvatura hacia los puntos de menor curvatura. En [Park y Han, 1998] se establece la correspondencia de los puntos de los contornos. El método determina el punto correspondiente en la siguiente imagen minimizando el cambio de curvatura de los puntos. En un primer paso, se utilizan *snakes* para localizar los contornos dentro de la imagen. Giachetti en [Giachetti, 2000] realiza un estudio sobre las diferentes medidas de similitud que han venido siendo empleadas a la hora de establecer correspondencias entre puntos característicos. Además, presenta técnicas para mejorar la precisión de la estimación de movimiento y reducir el coste computacional empleando filtros y multiresolución.

El método de Lef y Joshi [Lef y Joshi, 1993] enfoca el problema de la correspondencia desde un punto de vista geométrico. El método se basa en considerar que los parámetros de movimiento obtenidos considerando todos los puntos de interés de dos imágenes, permanecerán casi idénticos si se repite el análisis eliminando un par de puntos correspondientes. En este trabajo se asume que el vector de translación es pequeño cuando se lo compara con la distancia hasta la cámara, asunción que, por otra parte, reduce el ámbito de aplicación del método. En [Sarigianidis y Pycock, 1993] podemos encontrar otro método de correspondencia entre puntos de interés. En él se formula la correspondencia como un problema de optimización de una función de coste global. Se lleva a cabo una minimización por medio de una red neuronal de Hopfield. Esta función de coste se basa en las restricciones de unicidad (cada punto sólo tiene un correspondiente) y uniformidad del campo de movimiento. Esta última restricción es empleada en [Mehrotra, 1998]. Se toman tres imágenes consecutivas y se establece la correspondencia entre aquellas tripletas de puntos que representan las trayectorias más uniformes.

Los métodos basados en características simples, tales como puntos y líneas, no resultan adecuados para segmentación basada en movimiento. Esto es debido a que producen campos dispersos de vectores de movimiento, lo cual dificulta la extrapolación para crear campos densos, sobre los cuales realizar la segmentación. Una posible opción son los métodos basados en bloques, en sus dos principales versiones: métodos de correlación de fase y métodos de correspondencia de bloques.

Los métodos de correlación de fase [Fleet y Jepson, 1990, Fleet, 1992] estiman el desplazamiento entre dos imágenes por medio de una función de correlación cruzada normalizada, la cual es calculada en el dominio espacial bidimensional de Fourier. La idea se basa en el principio por el cual una translación en el dominio espacial produce una variación en el término de la fase del dominio de Fourier. Aunque el modelo habitual para este tipo de técnicas es el translacional, se han introducido variaciones para abordar rotaciones [Castro y Morandi, 1987] y el modelo afín [Pla y Bober, 1997].

También se han planteado esquemas jerárquicos de multiresolución los cuales resuelven el problema de la selección del tamaño de la ventana [Erkam *et al.*, 1993]. Dentro de los métodos de estimación de movimiento en el dominio de frecuencias podemos incluir aquellos que utilizan distribuciones de Wigner o filtros de Gabor. Heeger [Heeger, 1988] usó filtros espacio-temporales de Gabor para encontrar el flujo óptico en cada punto de la imagen. Mediante estos filtros se encuentra la energía de Gabor en doce orientaciones diferentes y en varias frecuencias espaciales. Se selecciona la orientación de la velocidad local más fuerte para dar una estimación del flujo óptico. Wiskott [Wiskott, 1999] integra información obtenida a partir de Gabor-wavelets y Mallat-wavelets. En una primera fase, se emplean Gabor-wavelets para estimar el campo de flujo de la imagen. Esta técnica es sensible a los componentes de la textura en diferentes orientaciones, con lo que no plantea el problema de la apertura. Sin embargo, el campo de flujo que se obtiene es de baja resolución espacial. A partir de este campo, el sistema extrae hipótesis de movimiento que se usan en una segunda fase para restringir correspondencias entre bordes detectados por las Mallat-wavelets.

Los métodos de correspondencia de bloques pueden considerarse en la práctica como los más populares para la estimación de movimiento, debido a su menor complejidad [Jain y Jain, 1981, Gharavi y Mills, 1990, Tekalp, 1995]. Singh [Singh, 1990] desarrolló un método de correspondencia de bloques, que además, proporciona medidas de confianza sobre las estimaciones. Esta confianza se basa en dos factores: cuan buena es la mejor correspondencia de un bloque en términos de errores de predicción, y la consistencia de las estimaciones de movimiento entre un bloque y sus vecinos. En [Bierling, 1988] se propone un esquema multiresolución que emplea un tamaño distinto de ventana en cada nivel de resolución. Lee y Chen [Lee y Chen, 1997] reducen el coste computacional por medio de la multiresolución, combinada con un test de descarte de correspondencias. Oh y Lee [Oh y Lee, 2000] proponen un método para reducir el espacio de búsqueda basándose en correlaciones espacio-temporales. Se han propuesto varios métodos para generalizar la correspondencia de bloques a modelos de movimiento más generales [Flusser, 1992, Nakaya y Harashima, 1994, Seferedis y Ghanbari, 1993]. En [Wang y Adelson, 1994, Borshukov *et al.*, 1997] se presenta un mismo método basado en el error cuadrático mínimo y que sirve para calcular el movimiento afín de los bloques, pero se proponen dos estrategias distintas para realizar la segmentación de movimiento: *agrupamiento* mediante el algoritmo de las *k-medias*, y crecimiento de regiones, respectivamente.

El algoritmo de Bober y Kittler [Bober y Kittler, 1994a] utiliza la transformada de Hough para estimar movimiento. Cada píxel vota por aquel conjunto de parámetros de movimiento que minimiza el *error de la diferencia de la imagen desplazada*¹. Dicho error se calcula empleando un *núcleo robusto* y una estrategia multiresolución. Una de las principales características de este método es que realiza en paralelo la estimación y la segmentación de movimiento. En [Hotter y Thoma, 1988, Diehl, 1991] también se realiza estimación y segmentación de movimiento de manera simultánea, aunque desde

¹Este error se calcula para un grupo de píxeles y un determinado movimiento. Se toman los valores de intensidad de un grupo de píxeles que pertenecen a una imagen y se restan de los valores de sus correspondientes en la siguiente imagen una vez aplicado el movimiento.

un enfoque bien distinto. En lugar de usar un crecimiento de regiones, se utiliza una estrategia de división. Inicialmente se realiza una detección de movimiento, y entonces se considera que cada grupo conectado espacialmente es un objeto diferente. Para cada uno de estos grupos se realiza una estimación de movimiento que a continuación es evaluada calculando el *error de la diferencia de la imagen desplazada*. Aquellos grupos de píxeles a los que no se adecua el movimiento estimado son declarados como objetos independientes, procediéndose a una nueva estimación de movimiento para ellos.

Los métodos basados en regiones realizan una segmentación previa de las imágenes para extraer grupos compactos de píxeles caracterizados por algún criterio de homogeneidad (ver Sección 2.1.3). Una vez hecho esto, se realiza una estimación de movimiento para cada una de las regiones. Este tipo de técnicas producen campos densos de movimiento, ya que al estimar el movimiento de una región, se está calculando para todos los píxeles que la componen. Los métodos presentados en [Hotter y Thoma, 1988, Diehl, 1991] pertenecen a esta clase, ya que primero se extrae la región y luego se estima su movimiento. En ambos casos, el criterio de homogeneidad empleado es la minimización del *error de la diferencia de la imagen desplazada*. Kottke y Sun [Kottke y Sun, 1994] propusieron otro método de esta clase. Dos imágenes consecutivas son segmentadas mediante un método de *agrupamiento* (visto en la sección 2.1.4). Se establece la correspondencia entre las regiones de ambas imágenes mediante un esquema de relajación, en el que se usan como medidas de similitud la posición, la intensidad y la forma de las regiones. Los desplazamientos son estimados a partir de las distancias de los centros de gravedad de las regiones correspondientes. Esta forma de calcular el desplazamiento conlleva problemas cuando las regiones sufren cambios debidos a oclusiones o movimientos no translacionales. En estos casos, el centro de gravedad cambia su posición interna de una imagen a la siguiente, lo cual provoca una incorrecta estimación del movimiento.

El método propuesto por Tabb y Ahuja [Tabb y Ahuja, 1994] también realiza la estimación de movimiento 2-D por medio de la correspondencia de regiones. Aquí, una secuencia de imágenes se considera como un volumen 3-D tomando el tiempo como eje z . Para movimientos relativamente pequeños, una región 2-D se solapará espacialmente consigo misma en imágenes consecutivas. Antes de realizar la estimación de movimiento, se lleva a cabo una segmentación de la imagen por medio de un algoritmo de segmentación multiescala. Una vez obtenidas las regiones, se intenta hacer corresponder las regiones de la imagen referencia con aquellas de la siguiente imagen que las solapan. Cuando una región tiene varias candidatas a la correspondencia, se toma una de ellas arbitrariamente. Las correspondencias incorrectas son detectadas en una etapa posterior, en la cual se verifican las correspondencias a nivel de píxel. Estas correspondencias se establecen mediante la estimación de los parámetros de movimiento que proporcionan la mejor transformación afín para pasar de una región a su correspondiente.

Para llevar a cabo una segmentación de objetos móviles, Dubuisson y Jain [Dubuisson y Jain, 1995] combinan segmentación de regiones con la información proporcionada por el movimiento. Este método no puede tratar los problemas de oclusión, debido a que en ningún momento se llega a realizar una estimación

de movimiento. En [Dubuisson *et al.*, 1996] se completa el método con la incorporación de modelos deformables a los contornos de los objetos extraídos, si bien el método sigue adoleciendo del mismo defecto. Gelgon y Bouthemy en [Gelgon y Bouthemy, 2000] presentan un método de segmentación y seguimiento de regiones. Tras una segmentación espacial basada en color o textura, se construye un grafo donde las regiones son los nodos, y éstos, a su vez, constituyen un Campo Aleatorio de Markov. Para cada región se estima el movimiento 2-D realizándose a continuación un etiquetado en el que las regiones con movimiento similar reciben una misma etiqueta. A partir de aquí, se genera un nuevo grafo donde existe un nodo para cada etiqueta. Este grafo representa los objetos de la escena y sirve como predicción para la segmentación de la siguiente imagen de la secuencia. En [Gil *et al.*, 1996] se presenta la combinación de dos métodos de estimación de movimiento, uno basado en regiones y otro basado en el mínimo rectángulo que engloba la región en movimiento.

3.1.4 El método propuesto

Como hemos visto a lo largo de los apartados anteriores, el trabajo desarrollado en análisis de movimiento es muy amplio, por lo que los enfoques también son muy diversos: gradiente, correlación de fase, utilización de características simples (puntos y líneas), correspondencia de regiones, sustracción de imágenes, etc. La mayoría de estos enfoques presenta algún tipo de prestación superior sobre el resto de métodos, de aquí que nuestra decisión deba fundamentarse en los requerimientos que impone el sistema donde se vaya a aplicar el análisis de movimiento.

En nuestro caso, el objetivo fundamental es conseguir una segmentación de aquellos objetos móviles que pueda contener la imagen, pero a un coste computacional que permita un funcionamiento del sistema en tiempo real. No se requiere una segmentación de gran calidad, como podría ser el caso de las aplicaciones de codificación de imagen, donde las imágenes deben poder reconstruirse a partir del análisis de movimiento. En nuestra aplicación, lo principal es saber cuántos vehículos, durante un periodo de tiempo dado, han circulado a través del tramo de carretera monitorizada, y cuál ha sido su comportamiento en lo referente a su dirección y al carril o carriles que han utilizado.

Guiándonos por estos objetivos, y en lo que se refiere al sistema de análisis de movimiento, lo que pretendemos es que nos proporcione la suficiente información como para poder discriminar qué regiones de la imagen pertenecen a objetos distintos. En cuanto a la dirección y el comportamiento, esto vendrá dado por un análisis posterior. En realidad nos bastará con saber en cada imagen la posición de cada vehículo y el sentido de su circulación.

Atendiendo a la bibliografía existente y a los objetivos expuestos en este apartado, hemos decidido que el mejor enfoque para nuestro método de análisis de movimiento es la utilización de regiones. El análisis de movimiento basado en regiones permite etiquetar por completo la imagen mediante vectores de movimiento, campo de movimiento denso, pero sin tener que recurrir a la estimación individualizada para cada

píxel de la imagen, ya que esto último supone un tremendo coste computacional. Los métodos basados en correspondencia de características simples, puntos y líneas, nos aportarían un relativo bajo coste computacional, sin embargo, a partir de ellos es difícil generar un campo de flujo que describa toda la imagen.

La utilización de regiones supone la combinación de dos enfoques distintos para la segmentación. Por un lado, la segmentación basada en movimiento, y por otro, la segmentación de imágenes individuales, a la que hemos venido llamando segmentación estática. Este segundo tipo de segmentación nos permite incorporar información morfológica al proceso. Como ya hemos visto en el anterior capítulo, basándonos tan solo en la intensidad podemos formar grupos de píxeles con la suficiente garantía como para estar seguros de que sólo pertenecen a un único objeto. Esto nos permite trabajar con entidades de mayor nivel. La fase de segmentación eleva su nivel de abstracción, pues no sólo unirá píxeles, sino que más bien consistirá en la unión de partes de objetos.

El método desarrollado se basa en buscar para cada región aquel movimiento que minimice el *error de diferencia de la imagen desplazada*. Esto expresado de una forma gráfica consiste en mover cada región segmentada sobre la siguiente imagen hasta encontrar el desplazamiento que hace mínima la suma de las restas entre píxeles correspondientes. Es decir, dada una región de una imagen, buscamos en la siguiente imagen la región más parecida.

Dado que una búsqueda exhaustiva del movimiento sería computacionalmente ineficiente, se emplea un *método de búsqueda por descenso*. Es decir, usamos un método que iterativamente va mejorando la estimación, donde en cada iteración se selecciona aquella estimación con error mínimo. Para poder aplicar este tipo de métodos debe cumplirse una premisa: que la función a minimizar sea convexa entre el punto inicial de búsqueda y el mínimo global. Afortunadamente, esta función tiene un buen comportamiento en la vecindad del mínimo global, por lo que este tipo de búsquedas casi siempre converge al mínimo global [Bober y Kittler, 1993]. La Figura 3.2 muestra la forma que suele presentar la función de error en las cercanías del mínimo global.

El método de búsqueda por descenso puede fallar cuando el movimiento de las regiones es grande. En este caso, la presencia de mínimos locales puede impedir alcanzar un mínimo global. Para solventar este problema hemos incorporado una estrategia multiresolución [Bierling, 1988, Bober y Kittler, 1994a, Erkam *et al.*, 1993]. Además, esta estrategia nos aporta una ventaja adicional: dado que un píxel de los niveles de resolución menor representa a varios píxeles del nivel de la imagen original, se acelera el proceso de convergencia.

Además de los niveles de menor resolución, en la pirámide de imágenes incorporamos dos niveles de resolución mayor que la imagen original. Éstos son los niveles que llamamos *subpíxel* y en ellos los valores de intensidad de sus píxeles se calculan por interpolación a partir de la imagen original. De esta forma podemos estimar el movimiento con mayor fidelidad, ya que generalmente el desplazamiento de las regiones no coincide exactamente con la rejilla de píxeles de la imagen.

Las técnicas basadas en diferencias de imágenes ayudan a la detección de aquellas zonas de la imagen que han experimentado cambios. De esta forma, sólo se deben

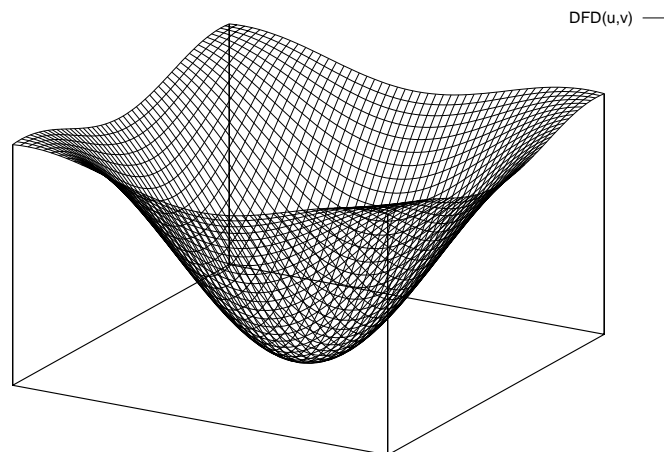


Figura 3.2: Comportamiento del *error de la diferencia de la imagen desplazada (DFD)* en la vecindad del movimiento óptimo.

procesar estas zonas, pues en realidad lo que se intenta analizar es lo que se mueve, y no lo que permanece estático. Con un mínimo coste computacional estas técnicas permiten ahorrar una gran cantidad de tiempo de proceso. Por ello, en nuestro método introducimos una de estas técnicas, la cual se basa en la substracción de la imagen a segmentar y de una imagen de referencia. Esta imagen de referencia corresponde al fondo estático, y se actualiza constantemente para reflejar los cambios en la iluminación. En realidad, la detección de movimiento es el primer paso del algoritmo, incluso previo a la segmentación estática. Obviamente, para qué queremos segmentar lo que sabemos que va a formar parte del fondo.

3.1.5 Estructura del capítulo

En el resto del capítulo veremos detalladamente el método propuesto para estimación y segmentación de movimiento. En la próxima sección se analizará el método de detección de movimiento que empleamos para realizar una discriminación entre los píxeles que pertenecen al fondo estático, y los que pueden corresponder a objetos móviles. La Sección 3.3 expone otra técnica basada en diferencias que permite unir regiones estáticas. A continuación, Sección 3.4, se verá el método de estimación de movimiento. En él explicaremos tanto la estrategia multiresolución, que ya hemos comentado, como el uso de estimadores robustos que permiten el rechazo de los *outliers* (ver definición al final del apartado 3.1.2). En la Sección 3.5 se verá la segmentación de movimiento, y en la Sección 3.6 se mostrarán algunos resultados sobre varias secuencias de imágenes. La última sección expondrá las conclusiones del capítulo.

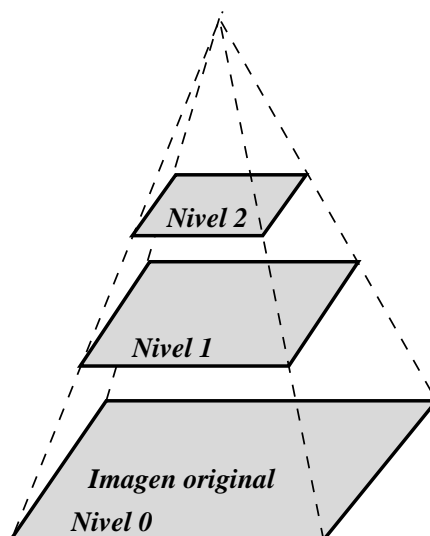


Figura 3.3: Pirámide de imágenes. Cada nivel corresponde a una resolución diferente.

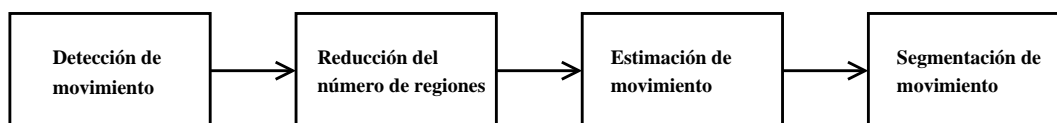


Figura 3.4: Secuenciación de los procesos que forman parte del análisis de movimiento y que serán vistos en el presente capítulo.

3.2 Detección de movimiento

Un frecuente paso previo a la estimación de movimiento es la detección de aquellas partes de la imagen que han experimentado un cambio en su intensidad. Estas partes corresponderán a zonas de la imagen donde se haya proyectado el movimiento de algún objeto. Aunque esto sólo será cierto si se asume que la cámara permanece estática y que la iluminación no presenta cambios apreciables.

Si recordamos lo expuesto en el Apartado 3.1.1, existen dos clases de técnicas basadas en substracción de imágenes: las que utilizan dos imágenes consecutivas y las que usan una imagen de referencia correspondiente al fondo estático de la secuencia. Nosotros nos decantamos por el segundo tipo. La razón es que estas técnicas definen completamente la forma de los objetos que se mueven. Cuando se usa una imagen de referencia la zona de cambio corresponde a todo el objeto móvil, aunque el objeto no presente textura. Cuando se usan dos imágenes consecutivas con este tipo de objetos, las zonas de cambio sólo se encuentran en los píxeles que son cubiertos o descubiertos por el movimiento del objeto. El empleo de dos imágenes supone un caso similar al de intentar obtener regiones a partir de la detección de contornos: se debe estudiar la

unión de los grupos de píxeles para definir completamente el contorno de las regiones. Además, se debe distinguir entre lo que son píxeles cubiertos y descubiertos, pues mientras los primeros forman parte de la región, los segundos no.

Parece claro que el uso de una imagen de referencia es una estrategia más sencilla y rápida. Desafortunadamente, no está exenta de algún inconveniente. Éste es la necesidad de actualizar continuamente la imagen de referencia, para así reflejar los posibles cambios que puedan experimentarse en el fondo estático de la imagen. Koller et al. [Koller *et al.*, 1994b, Koller *et al.*, 1994a] emplean lo que ellos denominan un formalismo de filtro de Kalman. La actualización del fondo se basa en la siguiente ecuación:

$$B_{t+1} = B_t + (\psi_1(1 - \mathbf{M}_t) + \psi_2\mathbf{M}_t)D_t \quad (3.4)$$

En la anterior ecuación, B_t representa la imagen referencia en el instante t , M_t es una matriz del mismo tamaño que la imagen, D_t es una matriz del tamaño de la imagen que contiene los valores de la resta de imágenes, y ψ_1 y ψ_2 son dos pesos. M_t contiene un valor binario para cada píxel, el cual es uno cuando el valor de la diferencia es mayor que un umbral (holgura), y cero en caso contrario. Es decir, cada valor de M_t indica si el píxel correspondiente en la imagen es un píxel de cambio o no. Las constantes ψ_1 y ψ_2 son las que indican cómo afectan a la imagen referencia los valores de intensidad de la imagen actual. El valor de ψ_1 toma un valor del orden de las décimas, mientras que ψ_2 es del orden de las centésimas. En realidad ψ_2 representa la desconfianza que se tiene en que un valor de diferencia mayor que la holgura empleada siempre corresponda a un píxel móvil.

La estrategia empleada por Koller et al. permite adaptar el fondo a los cambios progresivos de iluminación. Lamentablemente, esta estrategia deja indefenso al sistema ante un cambio brusco de la iluminación. Si se emplea un valor pequeño para ψ_2 , la adaptación será muy lenta, si se emplea un valor mayor, se producirá un efecto indeseable: los píxeles móviles afectarán al fondo estático.

Nosotros no utilizamos el peso ψ_2 , ya que la decisión de si un píxel es estático no se fundamenta sólo en la substracción de la imagen referencia. En nuestro sistema tenemos, además, dos fases adicionales: la segmentación estática y la reducción de regiones. La segmentación estática permite englobar en una sola región gran parte del fondo, y la fase de reducción permite añadirle aún más regiones. Esto nos permite que ante la detección de un cambio brusco de iluminación, nuestro algoritmo deseche todo el fondo y lo recalculé tras unas pocas imágenes. En realidad, tras la primera imagen después de desechar el fondo, ya gran parte de él está calculado y por lo tanto puede utilizarse en las siguientes imágenes.

Utilizando un formalismo similar al de Koller et al., nuestra ecuación de adaptación del fondo sería la siguiente:

$$B_{t+1} = B_t + \psi_1(1 - \mathbf{M}_t)D_t \quad (3.5)$$

La Figura 3.5 muestra las fases del método que están involucradas en la actualización o uso de la imagen referencia. El proceso utiliza dos imágenes de la secuencia, donde se quiere segmentar la primera de ellas. La primera fase es la que genera la

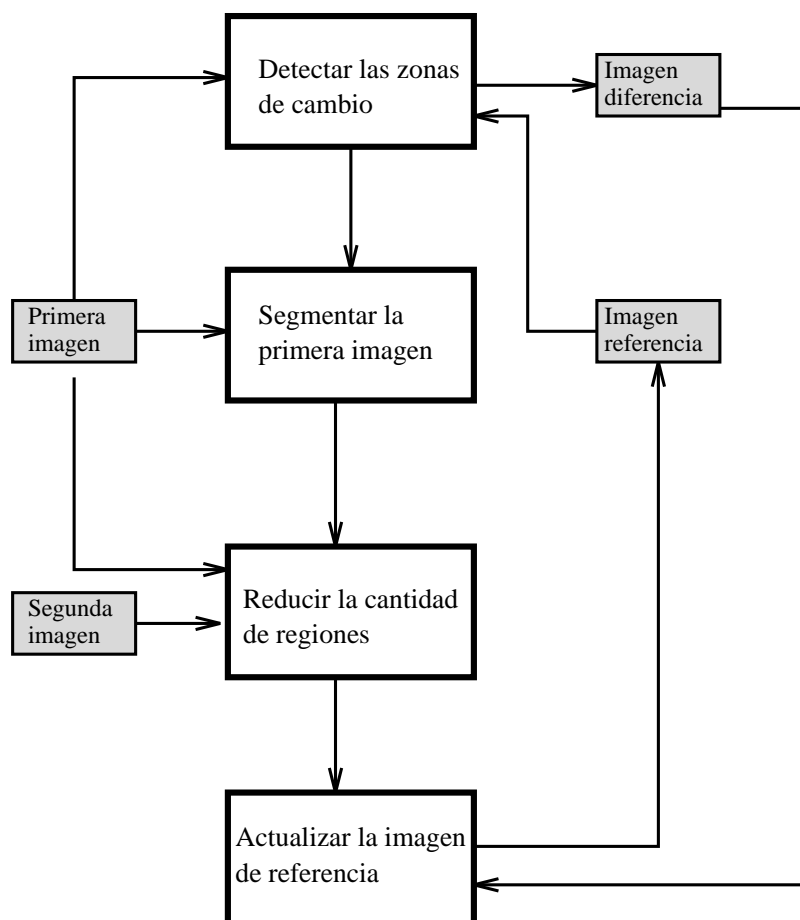


Figura 3.5: Actualización y uso de la imagen referencia.

máscara correspondiente a la detección de las zonas donde se han producido cambios. Para ello se utiliza la imagen a segmentar y la imagen referencia. Para realizar la detección se genera la imagen diferencia que contiene los valores de la resta de las dos imágenes. El siguiente proceso consiste en segmentar la primera de las dos imágenes, segmentación estática. Este proceso recibe la detección de movimiento. De esta forma, sólo se realiza la segmentación estática sobre las zonas donde se ha detectado movimiento. El resto de píxeles forman parte de una región, la cual obviamente formará parte del fondo estático.

Una vez obtenida la segmentación estática, ésta se pasa al proceso de reducción del número de regiones. Este proceso une regiones estáticas, formando una región fondo aún mayor. Para ello no utiliza la imagen referencia, sino las dos imágenes consecutivas. Veremos con mayor detalle el proceso en la siguiente sección.

La región fondo que obtenemos es pasada al proceso de actualización de la imagen referencia. Esto se realiza mediante la ecuación (3.5), donde en la matriz M_t se ponen

a uno los píxeles que pertenecen al fondo, y a cero el resto.

Se asume que se ha producido un cambio brusco en la iluminación cuando en el proceso de detección de cambio el número de píxeles diferencia se incrementa en un porcentaje superior a β . Un píxel diferencia es un píxel que ha sido marcado en la imagen diferencia como un píxel móvil. En este caso, se desecha la imagen referencia y se suspende el proceso de detección. El proceso de actualización asignará directamente los valores de la región fondo a la imagen referencia. Esto implica que una parte de la imagen referencia estará sin determinar, por lo que en la detección de movimiento para la siguiente imagen, esta parte sólo podrá realizarse para esos píxeles, declarándose el resto como pertenecientes a las zonas de cambio. Tras unas pocas imágenes, y conforme los objetos móviles vayan dejando ver los píxeles que ocultan, se podrá obtener la totalidad de la imagen referencia.

La obtención de la imagen diferencia no consiste en una mera substracción píxel a píxel entre las dos imágenes. En primer lugar, la resta no se realiza directamente sobre las imágenes, sino que previamente se les pasa un filtro gaussiano que reduzca el ruido de las imágenes². Seguidamente se realiza la substracción píxel a píxel, que proporciona una primera imagen diferencia conteniendo los valores absolutos de las restas. A esta imagen diferencia volvemos a pasarle un filtro gaussiano³ que uniformice los valores absolutos obtenidos. Una vez pasado el filtro, se aplica un umbral γ para binarizar la imagen diferencia. Los valores absolutos de la diferencia mayores que γ son puestos a uno, y el resto puestos a cero.

La imagen diferencia obtenida puede presentar dos tipos de problemas:

- Pequeños grupos de píxeles aislados que no correspondan a ninguna entidad.
- Grupos inconexos que en la imagen real correspondan a un mismo objeto móvil.

Ambos problemas pueden solventarse mediante operaciones morfológicas: una *apertura* y un *cierre* (ver Apéndice A). La Figura 3.6 muestra los elementos estructurantes utilizados en cada una de las dos operaciones. Es de suponer que los grupos de píxeles diferencia que formen parte de un objeto tendrán un mayor tamaño que los grupos que no correspondan a ninguna entidad real. Por ello, primero aplicamos la *apertura*, la cual eliminará estos pequeños grupos, y después el *cierre*, para unir partes de objetos. Además, la aplicación de las operaciones tiende a dar una forma más convexa a las máscaras obtenidas, lo cual es más acorde con la forma de los objetos.

Como comentábamos en el Apartado 3.1.1, no utilizamos un método de detección automática de carriles, sino que manualmente se genera una imagen donde se marca el área de interés. Los píxeles de este área son todos los de la calzada, incluidos los arcenes, más un ancho adicional que permita englobar totalmente las proyecciones de todos los vehículos. Se debe tener en cuenta que, dependiendo de la perspectiva de

²En la implementación en tiempo real del sistema utilizamos un filtro de media, el cual proporciona resultados algo peores, pero a un coste computacional mucho menor.

³De nuevo, en la implementación en tiempo real, sustituimos el filtro gaussiano por un filtro de media.

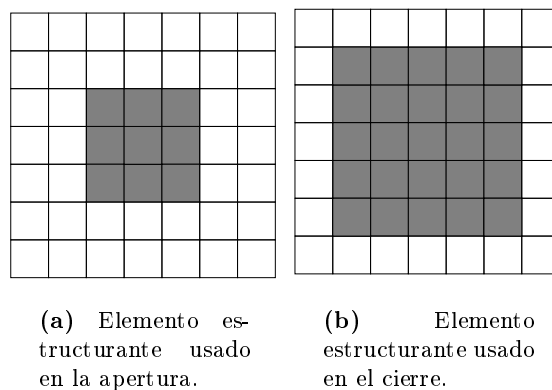


Figura 3.6: Elementos estructurales empleados en las dos operaciones morfológicas aplicadas sobre la imagen diferencia.

la cámara, algunas veces la proyección de las partes más elevadas de los vehículos exceden la proyección de la carretera.

Esta imagen apenas juega ningún papel cuando se utiliza la imagen de referencia, tan sólo ahorra un mínimo coste al señalar aquellos píxeles donde ni siquiera es necesario realizar la detección de movimiento. Sin embargo, cuando se desecha la imagen referencia, su papel es más importante pues permite reducir un mayor coste computacional, ya que sobre los píxeles no marcados no se realizará segmentación.

3.3 Reducción del número de regiones

El propósito de esta fase del algoritmo es intentar reducir aún más el número de regiones. Aunque nos interesa unir cualquier par de regiones adyacentes que se desplacen con el mismo movimiento, en realidad nuestro principal objetivo es incrementar al máximo el tamaño de la región fondo, que es la de un mayor tamaño. En esta fase del algoritmo, y dadas las fases anteriores, el tamaño de esta región ya puede llegar a alcanzar entre un 60% – 80% del total de píxeles de la imagen. Por supuesto, ese porcentaje dependerá del número y tamaño de los objetos móviles que tengamos. Ya hemos comentado que la fase más costosa del algoritmo es la estimación de movimiento, y que para la región fondo no vamos a realizar una estimación, ya que sabemos que su movimiento es cero. Por tanto, cuantas más regiones adhiramos al fondo, menor será el coste de la siguiente fase.

Esta técnica, al igual que la de detección de movimiento, también se basa en la substracción de imágenes, pero a diferencia de ésta, no emplea una imagen del fondo estático. En su lugar, la substracción se realiza entre la imagen a segmentar y su siguiente en la secuencia. El hecho de no usar la imagen fondo es lo que le proporciona la aptitud para colaborar en su regeneración cuando ésta tiene que ser desechada.

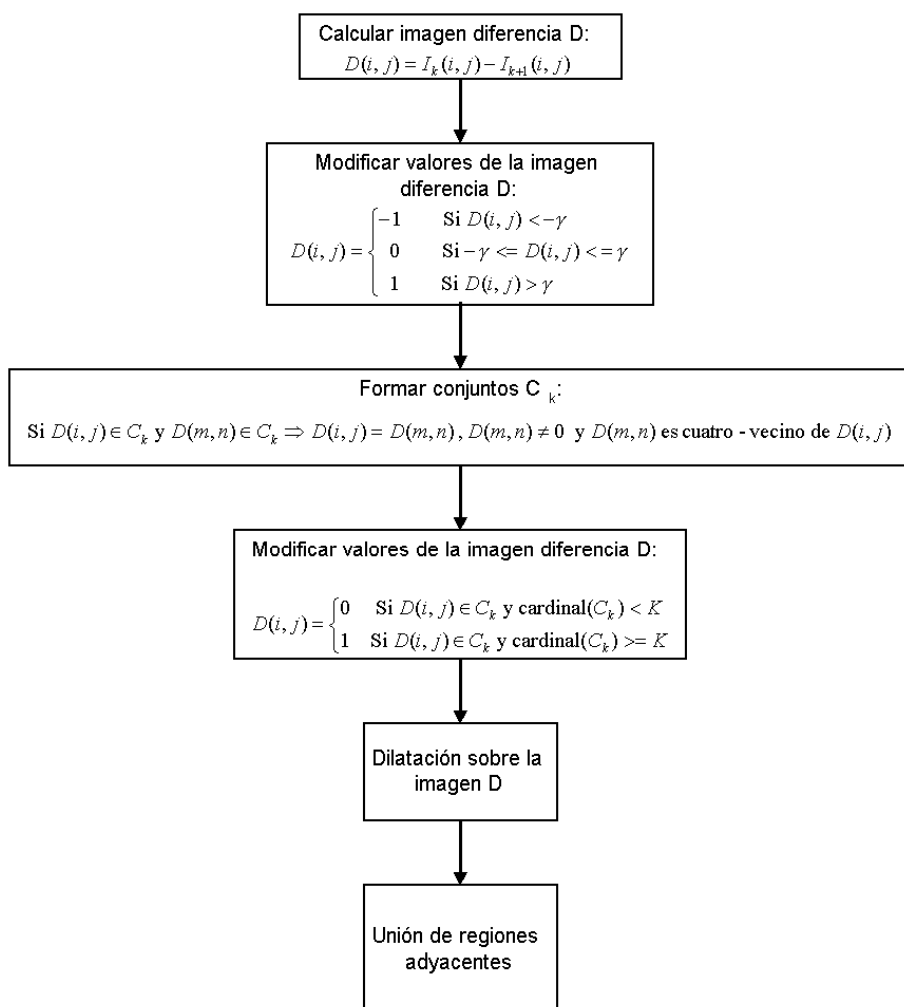


Figura 3.7: Pasos fundamentales del procedimiento de reducción de regiones.

La Figura 3.7 muestra los pasos fundamentales del proceso de reducción de regiones. El procedimiento se inicia con el cálculo de la imagen diferencia. Para ello se realiza la substracción entre las dos imágenes, aunque esta vez sin aplicar ningún tipo de filtro. Otra diferencia es que en realidad no nos interesan todos los píxeles diferencia de la imagen, sino sólo aquellos que se encuentran en el contorno de las regiones. Además, en la imagen diferencia no tomamos los valores absolutos, sino los propios resultados de las restas con su correspondiente signo. Esto es porque un paso posterior elimina los píxeles producidos por el ruido basándose en el análisis de los signos de los valores diferencia.

Como los únicos píxeles que nos interesan son los que se encuentran en los bordes de las regiones, sólo realizaremos la resta entre aquellos píxeles que perteneciendo a una región, tienen un vecino que pertenece a otra. Dado que ahora tendremos valores

positivos y negativos emplearemos dos umbrales, γ y $-\gamma$, los cuales se usarán para obtener una imagen diferencia D_k , a partir de las imágenes referencia I_k y su contigua I_{k+1} :

$$D_k(i, j) = \begin{cases} -1 & \text{Si } I_k(i, j) - I_{k+1}(i, j) < -\gamma \\ 0 & \text{Si } -\gamma \geq I_k(i, j) - I_{k+1}(i, j) \geq \gamma \\ 1 & \text{Si } I_k(i, j) - I_{k+1}(i, j) > \gamma \end{cases}$$

Los valores que ahora contiene la imagen diferencia D_k serán -1 ó 1 cuando el píxel es de cambio, y 0 cuando el píxel parece estático. Un píxel de cambio podrá ser provocado por un movimiento, o por ruido. La diferencia entre un grupo de píxeles producido por el ruido y un grupo de píxeles producidos por un movimiento estribará en que mientras en el segundo caso los signos serán consistentes para todo el grupo, en el segundo esto no sucederá. La idea se basa en que cuando una región de intensidad uniforme avanza sobre otra con intensidad también uniforme, los valores diferencia para los píxeles que son cubiertos tenderán a ser del mismo signo. Evidentemente, lo mismo sucederá para los píxeles que son descubiertos. Basándonos en esta idea lo que hacemos es encadenar píxeles adyacentes en una *cuatro-vecindad* que tengan el mismo signo. De esta forma sólo marcamos los píxeles que están en una cadena de tamaño superior a l_{cad} , rechazando las cadenas de tamaño inferior por considerarlas como ruido. Tras emplear diversos tamaños, hemos comprobado que un valor $l_{cad} = 15$ proporciona buenos resultados. El paso final para obtener la imagen diferencia consiste en aplicar una operación morfológica de *dilatación* (ver apéndice A). Para ello se emplea un elemento estructural de 5 píxeles en forma de cruz. Con ello lo que se pretende es ampliar las zonas de píxeles diferencia, para que haya una mayor coincidencia entre estos y los contornos de las regiones. Esta coincidencia ya es de por sí grande, sin embargo, no en todos los casos las discontinuidades de flujo óptico coinciden exactamente con los contornos de las regiones extraídas en la segmentación estática. En cuanto a los posibles píxeles diferencia pertenecientes a regiones estáticas, la operación de *dilatación* no afecta especialmente a sus contornos. Primeramente, porque los píxeles “ruido” han sido mayoritariamente eliminados con el procedimiento de encadenamiento por signo. Y en segundo lugar, porque la posición de los píxeles diferencia en estas regiones no surgen especialmente en los contornos.

La unión de dos regiones adyacentes se basa en contar el número de píxeles que componen su borde, y de éstos cuántos están marcados con un valor diferente a cero⁴. Dado que el borde común a dos regiones que se desplazan con movimientos diferentes debe estar principalmente compuesto por píxeles diferencia, la heurística para unir dos regiones consiste en contar el porcentaje de píxeles diferencia que hay en el borde común de dos regiones y si este porcentaje no es suficientemente grande, unir las. Éste procedimiento permite principalmente unir regiones estáticas, que son las que más nos interesan por formar parte del fondo. No obstante, también permite unir regiones móviles que no presenten un borde común donde haya un cambio significativo de intensidad.

⁴Para ahorrar tiempo, estos recuentos pueden ser realizados mientras se llevan a cabo los pasos anteriores: la substracción en los bordes y la selección de cadenas.



Figura 3.8: Imagen diferencia obtenida a partir de la imagen de la figura 2.1 y la siguiente de su misma secuencia.

La Figura 3.8 muestra la substracción de dos imágenes consecutivas, donde la primera de ellas es la imagen de la Figura 2.1. En esta figura puede verse que los bordes de los objetos que están en movimiento quedan claramente marcados por los píxeles diferencia. Estos bordes corresponderán con los contornos de las regiones que pertenecen a estos objetos. Por su parte, para las regiones estáticas no es posible apreciar una configuración de píxeles diferencia que denote contornos.

En algunas imágenes los contornos no quedan completamente cerrados, o no existe una exacta coincidencia con los bordes de las regiones. Por ello, no puede exigirse que el cien por cien de los píxeles del contorno de las regiones móviles sean píxeles diferencia. Es mejor usar un porcentaje mucho menor 50–60%, el cual, aunque pueda dejar de unir alguna región, no una al fondo estático ninguna que sea móvil.

3.4 Estimación de movimiento

En este apartado se verá el algoritmo de estimación de movimiento que se ha desarrollado. Básicamente, se trata de un método de búsqueda fundamentado en el cálculo del *error de la diferencia de la imagen desplazada*, **DFD** (Subapartado 3.4.1). Esta búsqueda se realiza por medio de una estrategia multiresolución (Subapartado 3.4.2) y llegando a una estimación con precisión subpíxel (Subapartado 3.4.3). En el Subapartado 3.4.4 se muestra la utilización de estadísticos robustos para mejorar los resultados de la estimación.

3.4.1 El algoritmo básico de estimación de movimiento

El método desarrollado se basa en la búsqueda para cada región de aquel movimiento que minimice el *error de diferencia de la imagen desplazada*. Es decir, dadas dos

imágenes, donde en la primera de ellas hemos realizado una segmentación con el método expuesto en el capítulo anterior, para cada región buscaremos aquel movimiento que haga mínima la suma de las diferencias entre los valores de intensidad de los píxeles de la región en la primera imagen y las intensidades de sus correspondientes píxeles en la siguiente imagen cuando se aplica el movimiento. En adelante, a esta función de error la denominaremos DFD^5 , que es como más comúnmente se referencia en la literatura.

Si bien, un modelo de movimiento puramente translacional no es lo que mejor se adapta al desplazamiento de los vehículos en el plano de la imagen, se ha asumido que no es necesario utilizar un modelo más complicado. Aunque las regiones experimentan un efecto de escalado, debido al acercamiento/alejamiento, éste no es de una magnitud importante: el intervalo entre fotogramas es pequeño, y además, los vehículos son relativamente pequeños si los comparamos con el tamaño de la imagen. Por contra, emplear un modelo más complejo supondría un considerable aumento del coste computacional, que, como muestran los resultados, no es necesario.

Expresándolo de una manera formal, la estimación del movimiento consiste en buscar para cada región R^i el movimiento (dx, dy) que minimice el *valor absoluto de la diferencia de la imagen desplazada*:

$$DFD_i(dx, dy) = \frac{\sum_{j=1}^{N_i} |I_1(x_{1j}, y_{1j}) - I_2(x_{1j} + dx, y_{1j} + dy)|}{N_i} \quad (3.6)$$

donde N_i es el número de píxeles de la región R^i , I_1 es el nivel de gris en la imagen original, I_2 es el nivel de gris en la siguiente imagen, x_{1j} es la coordenada x del j -ésimo punto de la región i en la primera imagen, y_{1j} es la coordenada y del j -ésimo punto de R^i en la primera imagen, y (dx, dy) es el movimiento aplicado a la región. La suma total es dividida por el número de píxeles, N_i , para poder comparar esta medida entre regiones de diferente tamaño.

En la ecuación (3.6) empleamos la *suma del valor absoluto de las diferencias* como medida de correspondencia. En la literatura se han utilizado otras diversas medidas: correlación directa, correlación normalizada de la media, correlación normalizada de la varianza, suma de las diferencias al cuadrado, etc.. Anandan en [Anandan, 1987] realizó una comparación entre diversas medidas, escogiendo finalmente la suma de las diferencias al cuadrado. Para ello aducía dos razones: que su valor es siempre positivo y que su coste computacional es menor. Tomando en cuenta las razones de Anandan, y considerando que el cálculo del valor absoluto tiene menor coste que el cálculo del cuadrado, nos parece acertada nuestra elección. Además, dada su linealidad, el uso del valor absoluto permite una comparación más fácil entre diferentes medidas que el cuadrado.

Ya que una búsqueda exhaustiva del movimiento conllevaría un alto coste computacional, utilizamos una estrategia diferente. Esta estrategia es la de *búsqueda por el descenso más profundo*. La Figura 3.9 muestra un diagrama de flujo de la técnica, la

⁵DFD viene de las iniciales de Displaced Frame Difference

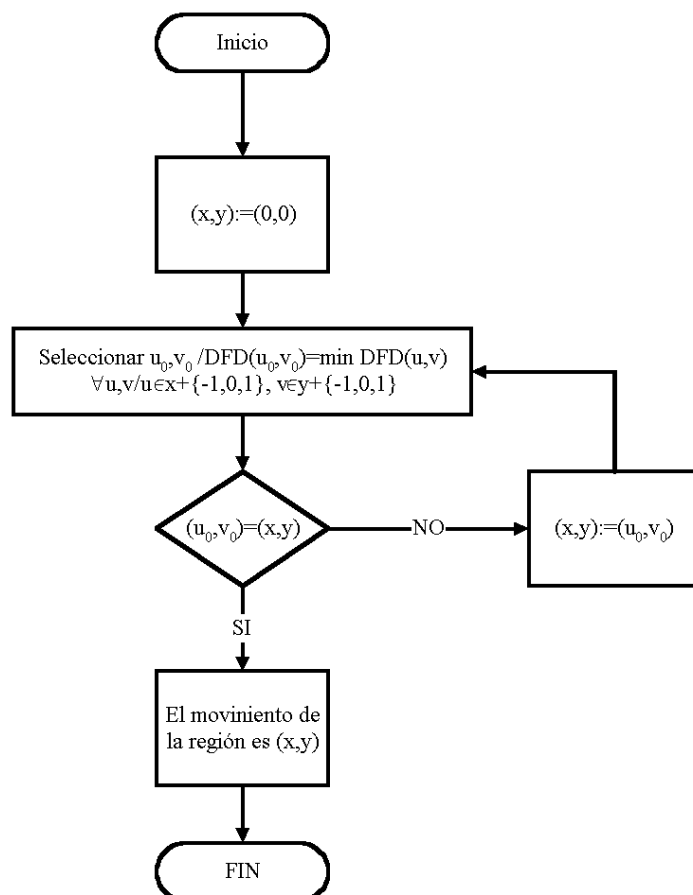


Figura 3.9: Algoritmo de estimación de movimiento.

cual consiste en la repetición, para cada región R^i , de los siguientes pasos:

1. Inicializar el movimiento de la región R^i : $(dx, dy) := (0, 0)$
2. Seleccionar para R^i aquel movimiento $(dx + k, dy + l)$ tal que:
 $DFD_i(dx + k, dy + l) = \min DFD_i(dx + p, dy + q) \forall p, q = \{-1, 0, 1\}$
3. Si $k = 0$ y $l = 0$ saltar al paso 6
4. $dx := dx + k, dy := dy + l$
5. Volver al paso 2
6. (dx, dy) es el movimiento de la región R^i .

El anterior algoritmo se basa en buscar en cada iteración el menor valor de la función DFD entre los vecinos $(p, q = \{-1, 0, 1\})$ del movimiento actual, (dx, dy) . Es

decir, buscamos aquel movimiento que nos hace reducir más rápidamente la función DFD , de aquí que reciba el nombre de búsqueda por el *descenso más profundo*.

El algoritmo reconoce que ha alcanzado el mínimo cuando detecta que el mejor movimiento es el obtenido en la iteración anterior $(dx, dy) = (dx + k, dy + l) / \forall k, l \in \{-1, 0, 1\}$.

3.4.2 La estrategia multiresolución

La estrategia de búsqueda por el *descenso más profundo* puede plantear problemas para algunas regiones. Esto es debido a que el procedimiento de búsqueda puede quedar atrapado en un mínimo local. Para evitar este tipo de mínimos y, además, acelerar el proceso, empleamos una estrategia multiresolución. Se trata de generar una pirámide de imágenes a distinta resolución a partir de cada una de las dos imágenes consecutivas (ver Figura 3.3). Entonces la estimación de movimiento se realiza en primer lugar en los niveles de menor resolución, para posteriormente ir propagando la estimación obtenida a los siguientes niveles de mayor resolución. En otras palabras, si ordenamos los niveles por resolución de 0 a n , donde 0 es la imagen original, la estimación de movimiento siempre se realiza primero en el nivel $k + 1$ antes que en el nivel k , donde $k \in \{0 \dots n\}$. La estimación obtenida en el nivel $k + 1$ es propagada al nivel k , de forma que en el primer paso del algoritmo, ya no se realiza una inicialización a cero de dx y dy , sino al movimiento estimado en el nivel $k + 1$.

Mediante una estrategia multiresolución se pueden tratar mayores desplazamientos en la imagen, pues permite evitar las malas estimaciones provocadas por los mínimos locales.

3.4.3 Estimación de movimiento con precisión subpíxel

En una secuencia real, raramente el movimiento de un objeto coincide con un número exacto de píxeles. Para abordar esta situación añadimos dos niveles más a la pirámide, si bien, en este caso son dos niveles de mayor resolución que la imagen original. Un primer nivel a resolución 0,5 píxeles, y un segundo a resolución 0,1 píxeles.

Los valores de intensidad de los subpíxeles son obtenidos por interpolación lineal. En el caso de la resolución 0,5 píxeles, los valores se obtienen fácilmente mediante una suma y una división. En la resolución de 0,1 píxeles, como muestra la Figura 3.10, para obtener el valor de un subpíxel c , primero se interpolan linealmente los valores de los subpíxeles a y b , y a continuación se calcula el valor para c .

3.4.4 Aplicación de estadísticos robustos

El algoritmo que usamos para estimación de movimiento asume que todos los píxeles de una región se desplazan con el mismo movimiento. Esto se conoce como la *asunción de movimiento simple*. Esta premisa es frecuentemente violada debido a la presencia de transparencias, discontinuidades de profundidad, movimiento independiente de los objetos, sombras y reflexiones especulares. Para conseguir una estimación fiable del

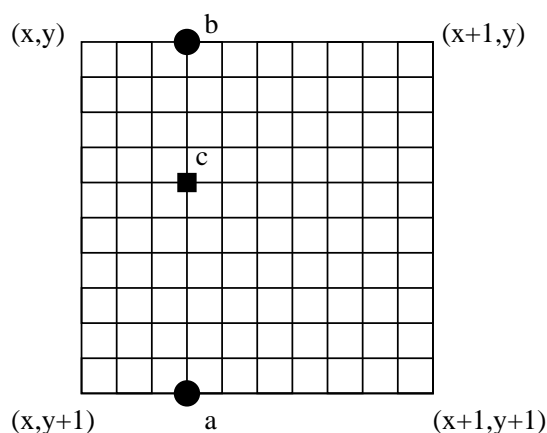


Figura 3.10: Interpolación de los niveles de intensidad en la resolución subpíxel.

movimiento de una región, este debe ser suficientemente grande, sin embargo, cuanto más grande es una región, es más probable que contenga movimientos dispares.

Las regiones de la imagen que corresponden a la carretera pueden provocar complicaciones. Estas regiones no presentan textura, sino más bien, una intensidad bastante uniforme para todos sus píxeles. Esto conlleva que, aunque el verdadero movimiento de estas regiones sea cero, si aplicamos un movimiento diferente éste produzca un valor para la función DFD no mucho peor que el del movimiento correcto.

Dada esta peculiaridad de las regiones carretera, pueden surgir problemas especiales cuando una región coche avanza sobre una región carretera ocultando píxeles de esta región. Los píxeles que han sido ocultados provocan un movimiento aparente de su región, el cual es casi igual al movimiento de la región coche. Esto se debe a que los píxeles ocultos producen grandes errores que el algoritmo trata de compensar mediante el movimiento de la región. Este falso movimiento en realidad trata de recuperar los píxeles perdidos. La falta de textura de la carretera produce que el DFD de un falso movimiento pueda llegar a ser mejor que el del movimiento real. El uso de estadísticos robustos trata de evitar estos problemas, mediante la reducción de la influencia de los píxeles que producen grandes errores.

Los métodos mínimo-cuadráticos o mínimo-modulares son las formulaciones más empleadas, sin embargo, no pueden afrontar el problema de la violación de la asunción del movimiento simple. Tales formulaciones tienden a promediar los movimientos presentes en una región, lo cual produce una incorrecta estimación.

“Robust Statistics is a branch of statistics that investigates the sensitivity of statistical procedures to the violation of the underlying assumptions . . . Two important terms concerning the performance of the estimator should be explained here: efficiency and robustness. Efficiency refers to the ability of a procedure to provide optimal estimates from data that fulfills the underlying assumptions (e.g. assumptions made during the design

of the method). Robustness reflects insensitivity to the assumptions being violated. The Least Square Estimator is efficient but non-robust. The quadratic error term weights heavily the contributions to the “optimal” solution from data points which have large residual errors (e.g. outliers).” [Bober y Kittler, 1993]

El término “robusto” fue inicialmente introducido por Box [Box, 1953]. Posteriormente, en los años sesenta, Huber [Huber, 1964] y Hampel [Hampel, 1968] introdujeron fundamentos matemáticos más fuertes. Dentro de la visión artificial, y más concretamente en el análisis de movimiento, este tipo de estimadores han sido utilizados por diversos autores [Bober y Kittler, 1994b, Bober y Kittler, 1994a, Black y Anandan, 1996, Ong y Spann, 1995] para abordar movimientos múltiples, oclusiones, etc. y tratar de mejorar la seguridad de las estimaciones.

Cuando existen varios movimientos en una misma región, cada conjunto de píxeles será consistente con cada uno de los movimientos. Cuando se examina uno de los movimientos, los píxeles que no son consistentes con ese movimiento provocarán fuertes errores que interferirán en la estimación. Estos píxeles se llaman *outliers*. Una aplicación de los estadísticos robustos consiste en detectar y reducir su influencia, de forma que no afecten a la estimación. De acuerdo con lo expuesto, en nuestro modelo reformulamos el estimador mínimo-modular mediante una formulación robusta.

En la literatura podemos encontrar diversos estimadores robustos, siendo el *cuadrático truncado* el más utilizado

$$\rho(x, \alpha, \lambda) = \begin{cases} \lambda x^2 & \text{Si } |x| < \frac{\sqrt{\alpha}}{\sqrt{\lambda}} \\ \alpha & \text{para el resto de valores} \end{cases}$$

Este estimador es un substituto del *mínimo cuadrático*. De forma similar podemos formular un substituto para nuestro *mínimo-modular*:

$$\rho(x, \alpha, \lambda) = \begin{cases} \lambda |x| & \text{Si } |x| < \frac{\alpha}{\lambda} \\ \alpha & \text{para el resto de valores} \end{cases}$$

donde λ es un factor de escala, el cual nosotros siempre ponemos a 1, y α es el máximo valor que se asigna en nuestro caso a una diferencia entre píxeles. El valor α limita la máxima influencia de una diferencia entre píxeles.

Introduciendo el kernel robusto en la formula de la *diferencia de la imagen desplazada* se obtiene:

$$DFD_i(dx, dy) = \frac{\sum_{j=1}^{N_i} \rho(I_1(x_{1j}, y_{1j}) - I_2(x_{1j} + dx, y_{1j} + dy), \alpha, \lambda)}{N_i} \quad (3.7)$$

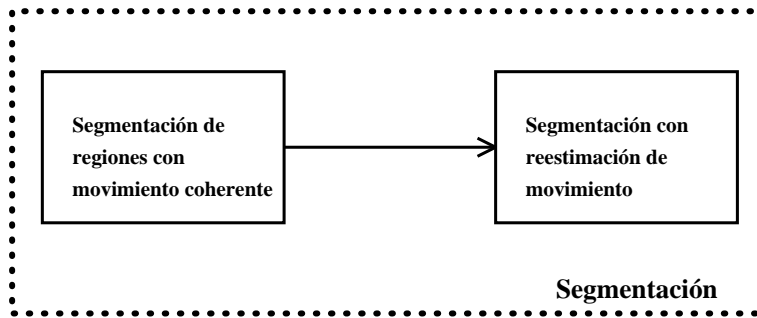


Figura 3.11: Fases de la segmentación de movimiento.

3.5 Segmentación

El proceso de segmentación se basa en la comparación de los parámetros de movimiento estimados para cada par de regiones adyacentes. Cuando este movimiento es coherente, ambas se unen pasando a formar una sola región. Desafortunadamente, aunque el proceso de estimación de movimiento normalmente obtiene estimaciones similares para las regiones que pertenecen a un mismo objeto, a veces esto no ocurre. Este hecho se debe principalmente al efecto de escalado. Nosotros trabajamos con un modelo de movimiento translacional, para lo cual asumimos que el efecto de escalado es despreciable. Sin embargo, este efecto realmente existe en mayor o menor medida, lo cual puede provocar que una misma región obtenga buenos DFDs para varios movimientos cercanos.

Para afrontar este hecho, el procedimiento de segmentación se divide en dos fases consecutivas:

- Segmentación de las regiones con movimiento coherente.
- Segmentación con reestimación del movimiento.

La primera fase une las regiones adyacentes con movimiento similar, mientras la segunda une dos regiones adyacentes si al aplicar el movimiento de una a la otra el resultado de la función DFD es mejor, o casi tan bueno, como el obtenido por el movimiento que se le estimó.

3.5.1 Segmentación de regiones con movimiento coherente

Como ya hemos comentado, esta fase une regiones adyacentes con movimientos coherentes. La cuestión ahora es cómo decidir qué dos regiones tienen movimientos coherentes. Una primera idea podría ser el usar un umbral ζ y basar la decisión en la siguiente regla:

$$\begin{cases} \text{Unir } R^1 \text{ y } R^2 & \text{Si } dif_x < \zeta \text{ y } dif_y < \zeta \\ \text{No unir } R^1 \text{ y } R^2 & \text{Si } dif_x > \zeta \text{ o } dif_y > \zeta \end{cases}$$

donde

$$\begin{aligned} dif_x &= |d_x^1 - d_x^2| \\ dif_y &= |d_y^1 - d_y^2| \end{aligned}$$

y d_x^i y d_y^i son los parámetros de movimiento x e y para la región R^i , respectivamente.

Sin embargo, esta regla no es del todo correcta, pues se olvida el factor de escala. No es lo mismo una diferencia de, por ejemplo, 0.7 píxeles para dos regiones correspondientes a objetos cercanos a la cámara, que esa misma diferencia para dos regiones correspondientes a objetos alejados. En el primer caso podríamos decir que ambas regiones forman parte de un mismo objeto, mientras que en el segundo esa misma afirmación probablemente no será correcta.

La regla que nosotros utilizamos se basa en que la diferencia entre los parámetros de movimiento sea menor que un porcentaje Θ del movimiento de las regiones. Para unir dos regiones debe cumplirse alguna de las siguientes cuatro condiciones:

- $dif_x < d_x^1 \cdot \Theta$ y $dif_y < d_y^1 \cdot \Theta$
- $dif_x < d_x^2 \cdot \Theta$ y $dif_y < d_y^1 \cdot \Theta$
- $dif_x < d_x^1 \cdot \Theta$ y $dif_y < d_y^2 \cdot \Theta$
- $dif_x < d_x^2 \cdot \Theta$ y $dif_y < d_y^2 \cdot \Theta$

Con esta regla, la diferencia de movimientos entre dos regiones se relaciona con la dimensión del movimiento. Esta fase de la segmentación, sobre todo cuando no hay perspectiva, suele ser suficiente para segmentar la imagen. Sin embargo, y casi siempre dependiendo de la cantidad de escalado que sufran los objetos al acercarse a la cámara, la siguiente fase puede ser necesaria para acabar de unir las regiones móviles que pertenecen a un mismo objeto. Dado que esta segunda fase realiza una reestimación del movimiento, preferimos usar un porcentaje Θ no demasiado alto, de forma que se puedan corregir posibles errores en la estimación del mismo.

3.5.2 Segmentación con reestimación de movimiento

Es muy probable que una región se mueva con el mismo movimiento que su vecina, debido a que ambas formen parte del mismo objeto. En esta fase del algoritmo hacemos uso de este conocimiento a priori para, al mismo tiempo, corregir estimaciones erróneas de movimiento y mejorar la segmentación.

El proceso que llevamos a cabo en esta fase consiste en:

1. Ordenar las regiones por orden ascendente en cuanto al número de píxeles.
2. Tomar una región R^A y calcular los distintos DFDs aplicándole los movimientos de sus regiones adyacentes,
3. Seccionar la región R^B cuyo movimiento ha producido un menor DFD.

4. Unir ambas regiones si se cumple la siguiente condición:

$$DFD_{AB} < DFD_{AA} + \kappa$$

donde DFD_{XY} es el DFD resultante de aplicar sobre la región R^X el movimiento estimado para la región R^Y , y κ es una holgura.

5. Si aún quedan regiones, volver al paso 2.

El anterior algoritmo toma las regiones en orden ascendente al número de píxeles que contienen. Esto se hace porque a mayor tamaño de una región, mayor es la confianza que se tiene en su estimación de movimiento. De este modo se comprueba si para cada región existe algún movimiento correspondiente a alguna de las regiones vecinas que produzca un resultado para la función DFD casi tan bueno, o mejor, que el correspondiente a su movimiento.

Cuando se decide unir dos regiones, siempre se une la más pequeña a la más grande. Esto es importante ya que, cuando tras varias adhesiones el tamaño de la región se incrementa en más de un 50%, se realiza una reestimación de su movimiento, reestimación que se calcula partiendo del movimiento que tiene, no de cero. Su objetivo es evitar la propagación de errores. Evidentemente, lo ideal sería reestimar el movimiento de una región cada vez que se produce una unión, sin embargo, el anterior resultado representa un equilibrio entre calidad de las estimaciones y bajo coste computacional.

Aunque habitualmente basta con recorrer sólo una vez la lista completa de regiones, como las uniones producen cambios en las relaciones de vecindad, pueden ser necesarias dos o más iteraciones. Para evitar innecesarias reestimaciones de la función DFD, se marcan aquellas regiones que han experimentado alguna unión. De esta forma, cuando se vuelve a recorrer la lista sólo es necesario visitar aquellas regiones que han cambiado, o son vecinas de alguna de éstas.

Dado que las relaciones de vecindad son básicas para ambas fases de la segmentación, se hace necesario emplear un grafo de adyacencia. En ella se marcan las casillas que corresponden a una vecindad. Estas marcas se corrigen a medida que se van produciendo uniones de regiones. La actualización es sencilla ya que cuando se unen dos regiones basta con pasar los vecinos de la que desaparece a aquella que perdura. El uso de este grafo se hace necesario, pues gracias a ella es inmediato saber si dos regiones son vecinas. En nuestro caso el grafo se construye durante la reducción de regiones, al mismo tiempo que se estudian los bordes de las regiones. De esta forma su construcción no supone ningún coste adicional.

3.6 Resultados

En esta sección mostramos los resultados de los procedimientos de detección, estimación y segmentación de movimiento. Para ello utilizamos cuatro secuencias: dos urbanas, una de carretera y una de autopista.

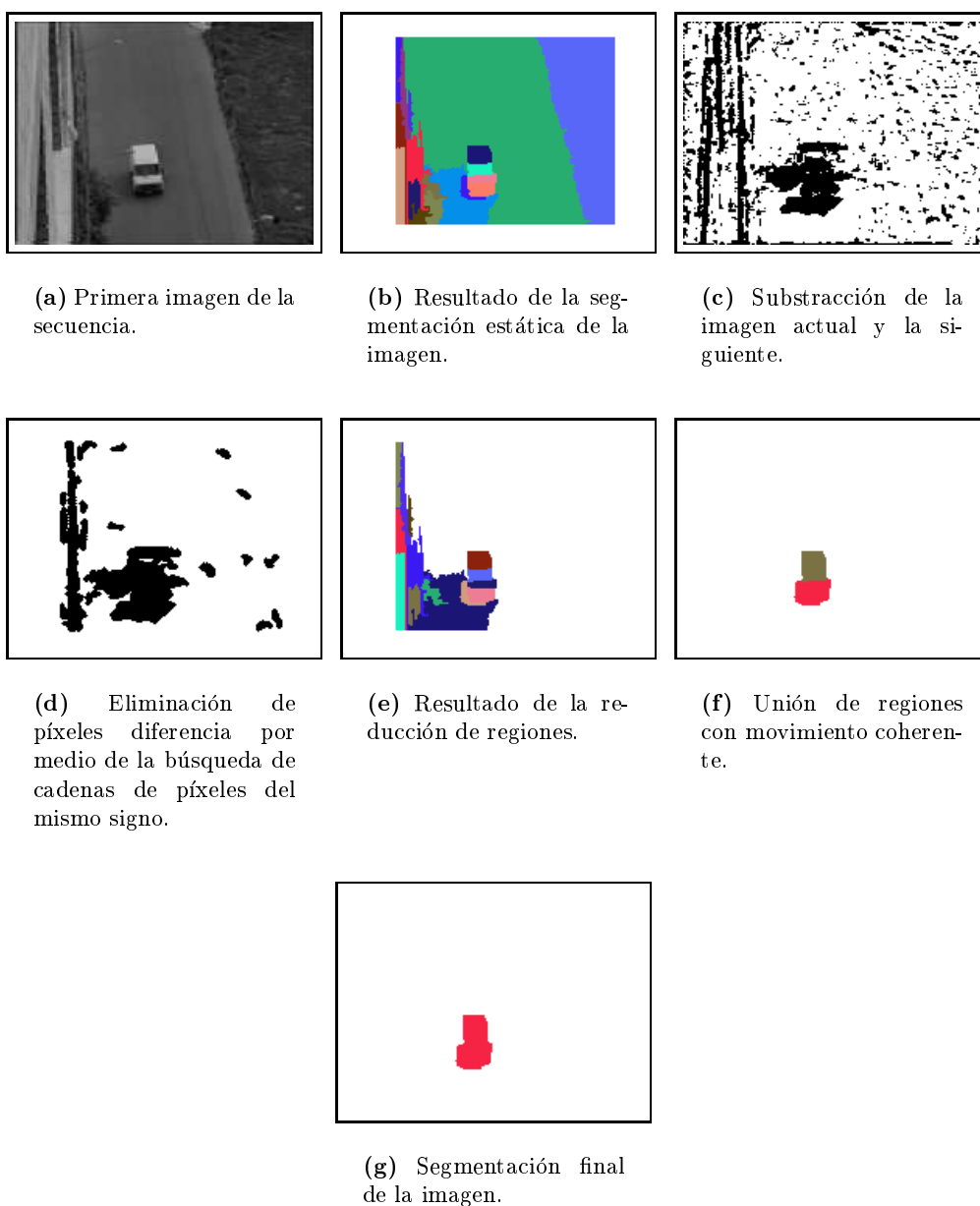


Figura 3.12: Varias etapas de la segmentación de una imagen de tráfico urbano.

La primera secuencia corresponde a tráfico urbano y fue grabada desde la terraza de un edificio. De esta secuencia hemos extraído tres imágenes consecutivas, que son las que aparecen, junto con algunos resultados intermedios del proceso, en las Figuras 3.12, 3.13, 3.14. En ellas, una furgoneta circula por una calle, en dirección a la cámara.

En esta secuencia no hemos marcado como área de interés sólo la calzada, sino que hemos usado un área de interés rectangular. Así, se puede apreciar el comportamiento de otro tipo de regiones estáticas, además de las correspondientes a la calzada. Por supuesto, en una aplicación real se hubiera marcado sólo la calzada, lo cual supone una reducción del tiempo de computación.

La Figura 3.12 contiene la primera imagen de la secuencia, subfigura 3.12.a. Dado que no ha habido imágenes anteriores, no existe una imagen referencia del fondo estático. Por lo tanto, para esta imagen no se realiza una detección de movimiento, y la segmentación estática se aplica a la totalidad del área de interés. El resultado se muestra en la subfigura 3.12.b.

El siguiente paso es unir las regiones que por su gran tamaño no pueden corresponder a vehículos. Esto nos proporciona una región fondo que ocupa aproximadamente las tres cuartas partes de la imagen. La subfigura 3.12.c es el resultado de la substracción de la imagen de la subfigura 3.12.a y la siguiente, subfigura 3.13.a. Tras esta substracción se realiza la selección de cadenas de píxeles diferencia con el mismo signo para reducir la influencia del ruido. El resultado se observa en la subfigura 3.12.d. Puede observarse que la mayor presencia de píxeles diferencia se concentra en la furgoneta y en su lado izquierdo, donde predominan las formas rectas. En la subfigura 3.12.e se muestra el estado de la segmentación después de este paso. A excepción de las dos grandes regiones que ya fueron unidas al fondo debido a su tamaño, las subfiguras 3.12.b y 3.12.e son prácticamente iguales. Debido a la presencia de píxeles diferencia, precisamente sobre la zona donde hay mayor número de regiones, tan sólo ha sido posible unir al fondo un par de regiones pequeñas. A continuación se lleva a cabo la actualización de los valores de la imagen referencia del fondo estático. En este caso, dado que se trata de la primera actualización, se asignarán directamente los valores de los píxeles de la región fondo de la presente segmentación. El resto de píxeles, los que en este momento pertenecen a otras regiones, continuarán con un valor indeterminado, por lo que en la segmentación de la siguiente imagen necesariamente deberán ser procesados por la segmentación estática.

Una vez que se ha actualizado la imagen de referencia, se pasa a estimar el movimiento de las regiones que no pertenecen al fondo. Con las estimaciones realizadas, se procede a unir aquellas que presentan desplazamientos similares. La subfigura 3.12.f muestra esta unión. Como puede observarse, la imagen se ha descompuesto en tres regiones: el fondo y dos más que pertenecen a la furgoneta. El siguiente y último paso es la segmentación con reestimación de movimiento. Aquí se toma la región más pequeña de las dos que componen la furgoneta, y se le aplica el movimiento estimado para la otra. Como el DFD es similar, se produce la unión de ambas, consiguiéndose así la segmentación final (subfigura 3.12.g).

La Figura 3.13 muestra la segmentación de la segunda imagen de la secuencia,

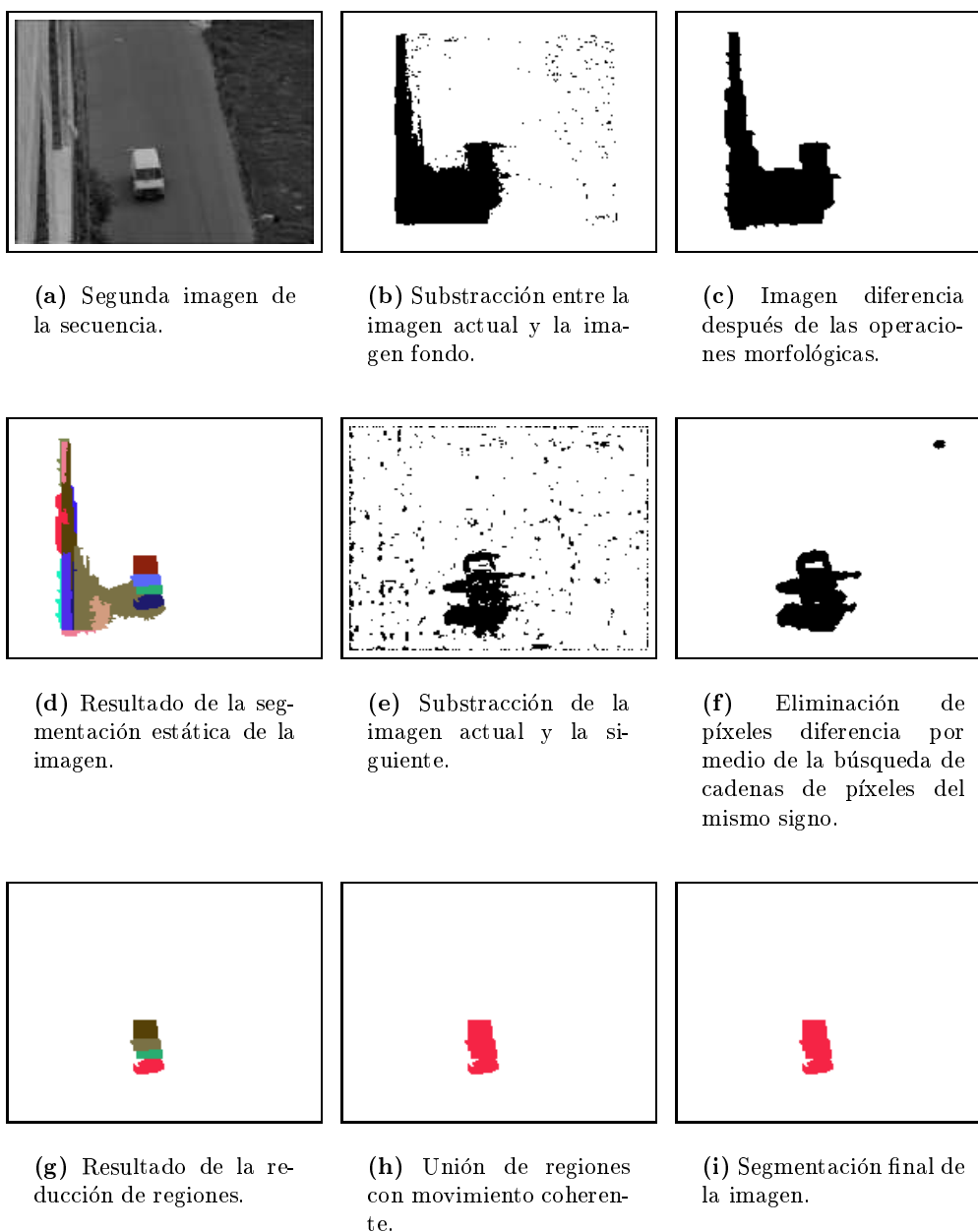
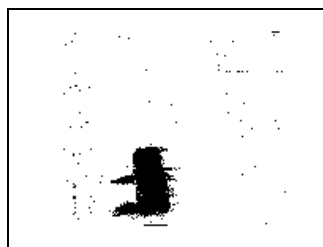


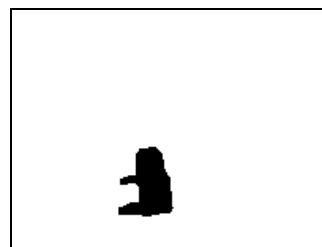
Figura 3.13: Varias etapas de la segmentación de la imagen siguiente a la de la Figura 3.12.



(a) Tercera imagen de la secuencia.



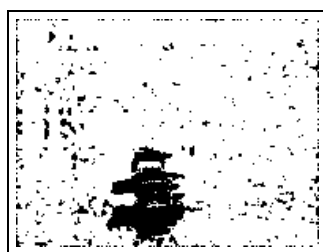
(b) Substracción entre la imagen actual y la imagen fondo.



(c) Imagen diferencia después de las operaciones morfológicas.



(d) Resultado de la segmentación estática de la imagen.



(e) Substracción de la imagen actual y la siguiente.



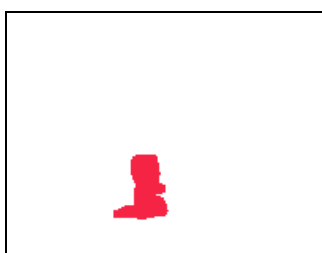
(f) Eliminación de píxeles diferencia por medio de la búsqueda de cadenas de píxeles del mismo signo.



(g) Resultado de la reducción de regiones.



(h) Unión de regiones con movimiento coherente.



(i) Segmentación final de la imagen.

Figura 3.14: Varias etapas de la segmentación de la imagen siguiente a la de la Figura 3.13.

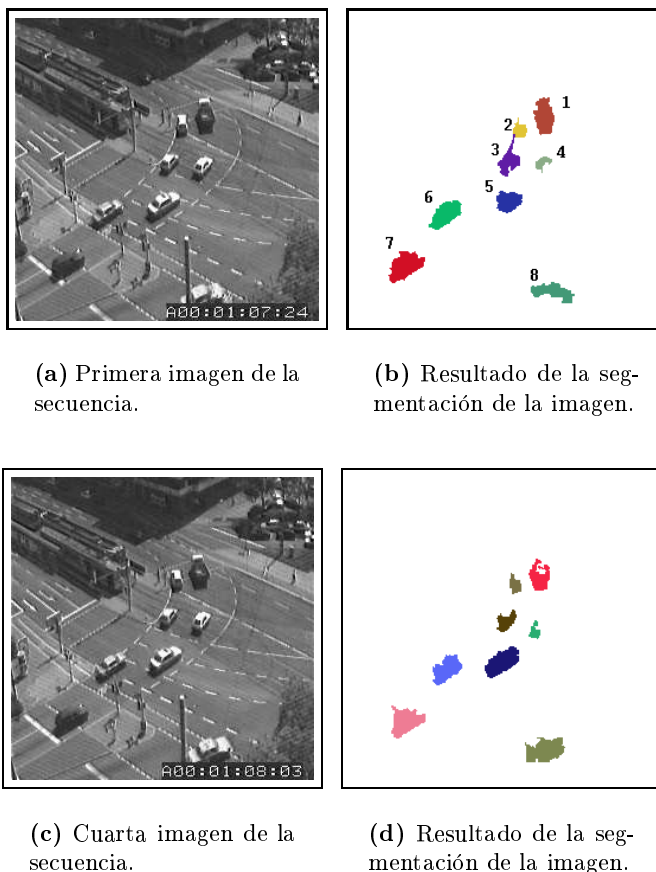


Figura 3.15: Segmentación de una escena grabada en la Durlacher-Tor de Karlsruhe. Estas imágenes fueron usadas en [Koller et al. 1993]. (KOGS/IAKS Universität Karlsruhe)

subfigura 3.13.a. A diferencia de la primera imagen, para la presente sí que disponemos de una imagen de referencia del fondo estático. Al menos una imagen de referencia parcial. La subfigura 3.13.b muestra la imagen diferencia obtenida tras la resta de la imagen actual y el fondo. La subfigura 3.13.c es la misma imagen pero tras aplicar las operaciones morfológicas. Como comentábamos en el Apartado 3.2, estas operaciones tienen como objetivo eliminar pequeños grupos que no corresponden a entidades de interés, y unir grupos que puedan corresponder a un mismo objeto. Como puede observarse, se consiguen ambos objetivos. Por un lado, han desaparecido los pequeños grupos de píxeles, que con casi total seguridad debían corresponder a ruido. Y por otra parte, se ha obtenido una máscara compacta que engloba toda la superficie de la imagen ocupada por la furgoneta. A continuación se realiza la segmentación estática sobre la parte de la imagen ocupada por la máscara de píxeles diferencia. El resultado puede verse en la subfigura 3.13.d.

Una vez segmentada la zona donde se detectó algún cambio, se lleva a cabo la reducción de regiones basada en la substracción de imágenes contiguas. La subfigura 3.13.e muestra la imagen diferencia, y la subfigura 3.13.f muestra el mismo resultado tras la eliminación del ruido. En este caso podemos ver que los píxeles diferencia se concentran únicamente en la zona donde verdaderamente se ha producido el movimiento. Como consecuencia de ello, la reducción de regiones puede aislar correctamente las regiones que pertenecen a la furgoneta. A continuación se procede a la estimación y segmentación de movimiento. A diferencia de la imagen anterior, en esta ocasión se obtienen movimientos similares para todas las regiones que componen la furgoneta. De este modo no es necesario recurrir a la segmentación con reestimación de movimiento. La primera segmentación ya une todos los píxeles de la furgoneta, subfigura 3.13.h, con lo que la siguiente segmentación, aunque se lleve a cabo, no tiene efecto alguno (subfigura 3.13.i).

La Figura 3.14 corresponde a la segmentación de la tercera imagen de la secuencia. El procesamiento de esta imagen es similar al de las anteriores, sin embargo, en esta ocasión ya tenemos una imagen referencia del fondo estático que es casi completa. En la fase de detección de movimiento sólo se encuentran píxeles diferencia en la zona de la furgoneta, de modo que sólo se segmenta esta zona. Dado que toda la zona está en movimiento, la reducción basada en substracción de imágenes contiguas no tiene efecto, por lo que las subfiguras 3.14.d y 3.14.g presentan exactamente las mismas regiones. La subfigura 3.14.h es la segmentación basada en similitud de movimientos. Aquí se consigue unir el cuerpo principal de la furgoneta. El resto de regiones son unidas por la segmentación basada en reestimación de movimiento, subfigura 3.14.i.

Como hemos visto, tras procesar tan sólo dos imágenes se dispone ya de una imagen de referencia del fondo estático. De este modo es posible descartar esta imagen cuando se detecta que se ha producido un cambio importante, pero sin afectar al procesamiento del resto de imágenes.

Las Figuras 3.15 y 3.16 muestran las segmentaciones de otras dos secuencias. En ambas la segmentación es mucho más complicada. Primeramente, porque en ambas el número de vehículos es muy grande: ocho y diez, respectivamente. Además, cada una de ellas presenta inconvenientes adicionales. La primera, Figura 3.15, plantea la dificultad del pequeño tamaño de los vehículos. Este tamaño varía desde 10x14 el más pequeño, hasta 18x30, aproximadamente, el más grande. La segunda secuencia, Figura 3.16, plantea los problemas de la agrupación de vehículos y la perspectiva. En el centro de la imagen, carril derecho, puede observarse un grupo de tres vehículos: dos coches y un camión. En cuanto a la perspectiva, dada la situación poco elevada de la cámara, el tamaño de los vehículos varía considerablemente. Mientras los más pequeños apenas ocupan trescientos píxeles, los más grandes tienen un tamaño en el plano de la imagen de varios miles de píxeles. Además, y lo que es más importante, esto hace que el movimiento no sea puramente translacional. A medida que los vehículos se van acercando a la cámara, su tamaño va aumentando, y, obviamente, cuando se van alejando sucede lo contrario.

La secuencia de la Figura 3.15 muestra un cruce de avenidas en una plaza. Esta

	<i>Imagen 1</i>		<i>Imagen 2</i>		<i>Imagen 3</i>		<i>Imagen 4</i>	
	<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>
<i>Camión 1</i>	-0.2	-0.9	-0.2	-0.7	-0.1	-0.7	-0.3	-0.7
<i>Coche 2</i>	0.0	-0.9	-0.1	-0.8	0.0	-0.9	-0.2	-0.7
<i>Coche 3</i>	1.0	-1.0	0.8	-0.7	0.7	-0.8	0.9	-0.8
<i>Coche 4</i>	0.7	-1.0	1.0	-0.9	0.9	-0.9	0.9	-0.9
<i>Coche 5</i>	1.3	-1.0	1.6	-1.0	1.4	-1.0	1.5	-1.0
<i>Coche 6</i>	1.6	-1.3	1.9	-1.3	1.6	-1.4	1.7	-1.2
<i>Coche 7</i>	1.3	-0.9	1.6	-0.9	1.3	-0.8	2.1	-0.8
<i>Coche 8</i>	1.9	-0.8	2.1	-1.0	2.2	-1.0	2.3	-1.0

Tabla 3.1: Movimientos estimados para los vehículos de la Figura 3.15.

imagen fue obtenida de la página *web* del *Institut für Algorithmen und Kognitive Systeme, Universität Karlsruhe*, y corresponde a una secuencia utilizada por Koller et al. [Koller et al., 1993] en un conocido artículo sobre seguimiento de vehículos. Ya en este artículo, Koller et al. señalaban la dificultad que entraña esta secuencia debido al reducido tamaño de los vehículos. De hecho, su método requería una inicialización interactiva para poder seguir los tres vehículos de menor intensidad. La secuencia original es de un tamaño de 512x512 píxeles. Dado que nuestro método está orientado hacia la implementación en tiempo real, nosotros siempre usamos imágenes de un tamaño inferior, por lo que el usarlo en esta ocasión nos haría partir con una ventaja de la que no se dispondría en una implantación real del sistema. Por ello, y aún a pesar de agravar el problema del pequeño tamaño de los objetos, hemos reducido el tamaño de las imágenes a la mitad: 256x256.

Las subfiguras 3.15.a y 3.15.c corresponden a la primera y cuarta imágenes de la secuencia. Se puede observar que nuestro método consigue segmentar todos los vehículos presentes. Sin embargo, podemos apreciar que el reducido tamaño de los vehículos afecta a la calidad de la segmentación. Esto se debe principalmente a la segmentación estática. Dado el diminuto tamaño de algunos elementos de los coches, tales como ruedas, ventanillas, parachoques, etc, la segmentación estática no les confiere importancia, con lo que a veces se unen a la calzada. También se produce el hecho de que cuando se observan pequeños elementos en la lejanía, compuestos por intensidades claras y oscuras, la visión que se tiene de ellos es la de una única zona de intensidad media, la cual se parece más a la intensidad de la carretera que a la de el propio vehículo.

En el ámbito de aplicación de nuestro método, la calidad de la segmentación carece de relativa importancia, o al menos la importancia que tiene es secundaria. Lo que realmente interesa es saber cuántos vehículos hay, dónde se encuentran, y cómo se están moviendo. Afortunadamente, estos objetivos se alcanzan. La Tabla 3.1, muestra las estimaciones de movimiento para todos los vehículos de la escena a lo largo de cuatro imágenes, las dos presentadas en la Figura 3.15, y las dos que se encuentran entre ambas. En esta tabla se han numerado los vehículos tomando su posición en

la primera imagen. Los números más bajos corresponden a vehículos situados en posiciones más altas de la imagen. Es decir, se ha ordenado de acuerdo a la coordenada y de su centro de masas. Así mismo, el centro de coordenadas está situado en la esquina superior izquierda de la imagen. Observando los movimientos estimados para los vehículos, se puede apreciar que todos ellos se desplazan hacia el centro del cruce, y los que lo alcanzan realizan un giro a la izquierda.



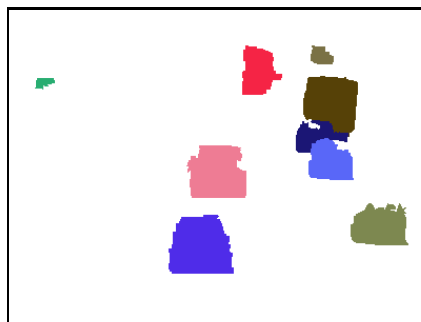
(a) Primera imagen de la secuencia.



(b) Resultado de la segmentación de la imagen.



(c) Cuarta imagen de la secuencia.



(d) Resultado de la segmentación de la imagen.

Figura 3.16: Segmentación de dos imágenes de la secuencia grabada en la N-340.

La secuencia de la Figura 3.16 presenta un caso más complicado debido a la situación de la cámara. De hecho, corresponde a un caso extremo de perspectiva. En una implementación real, lo lógico sería situar la cámara en una posición más elevada, la cual, por un lado, redujera la disparidad de tamaño entre vehículos, y, lo que es más importante, permitiera abarcar un mayor espacio de carretera, y con menores oclusiones parciales. Sin embargo, esta secuencia nos permite observar que el método trabaja correctamente, aún sin que se cumpla la asunción de movimiento puramente translacional.

Las subfiguras 3.16.a y 3.16.c son las imágenes primera y cuarta de la secuencia,

	<i>Imagen 1</i>		<i>Imagen 2</i>		<i>Imagen 3</i>		<i>Imagen 4</i>	
	<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>	<i>x</i>	<i>y</i>
<i>Coche 1</i>	-4.0	0.2	-3.8	0.4	-3.8	0.6	-3.7	0.9
<i>Camión 2</i>	-1.0	0.9	-1.1	0.8	-1.0	1.2	-1.5	1.7
<i>Camión 3</i>	-0.3	-1.2	-0.3	-1.2	-0.6	-1.2	-1.1	-1.1
<i>Coche 4</i>	-0.3	-2.0	-0.2	-1.6	-0.4	-1.5	-0.6	-1.5
<i>Coche 5</i>	-0.5	2.0	-0.6	2.5	-0.6	-2.5	-0.8	3.2
<i>Coche 6</i>	-1.0	-2.5	-1.1	-2.1	-1.2	-2.1	-1.4	-2.0
<i>Coche 7</i>	-0.7	4.0	-0.8	4.0	-0.8	4.3	-0.9	4.6
<i>Coche 8</i>	-2.3	-5.3	-2.6	-4.8	-2.4	-4.4	-2.0	-4.5

Tabla 3.2: Movimientos estimados para los vehículos de la Figura 3.16.

y las subfiguras 3.16.b y 3.16.d son sus respectivas segmentaciones. La Tabla 3.2 muestra los movimientos estimados en cada una de las cuatro imágenes para cada uno de los vehículos que aparecen en la segmentación de la primera imagen. Aquí de nuevo hemos usado la misma numeración, a menor coordenada y para el centro de masas del vehículo, menor número de orden.

La escena contiene 10 vehículos, de los cuales sólo ocho son correctamente segmentados en las cuatro imágenes. Existen dos coches que, dadas las dificultades que presentan, no pueden ser correctamente segmentados en todas las imágenes. El primero de estos vehículos es el que se encuentra casi totalmente oculto tras el camión que se acerca por el carril de la izquierda. Dado que está mayoritariamente oculto, y que por su lejanía, su velocidad en el plano de la imagen es muy similar a la del camión, tan sólo pudo ser segmentado en la tercera imagen. En realidad su situación se va agravando conforme avanza la secuencia, pues cada vez se halla más oculto tras el camión. El otro vehículo es el que se aleja de la cámara por un camino de tierra situado a la izquierda de la carretera. Este coche se desplaza a una velocidad muy baja, y además su visión es interferida por la baja vegetación que se encuentra alrededor del camino. Su movimiento estimado es siempre muy cercano a cero, con lo que el algoritmo tiende a unirlo al fondo. Este vehículo fue segmentado en las imágenes segunda y cuarta.

Pensamos que, a pesar de los dos vehículos que presentan problemas de segmentación, los resultados obtenidos en esta secuencia son muy satisfactorios. No se puede requerir al método que distinga el movimiento de dos objetos adyacentes, en un caso en que, probablemente, ni siquiera un observador humano podría hacerlo⁶. Por lo que respecta a los vehículos principales, son segmentados con suficiente precisión. Cabe destacar la capacidad del método para separar distintos vehículos, aunque estos aparezcan juntos en el plano de la imagen.

La Figura 3.17 muestra los resultados de la segmentación de la imagen de la subfi-

⁶De hecho, y sirva como anécdota, en el caso del vehículo que circula por el camino, fue el algoritmo de segmentación quien descubrió el vehículo antes que nosotros.

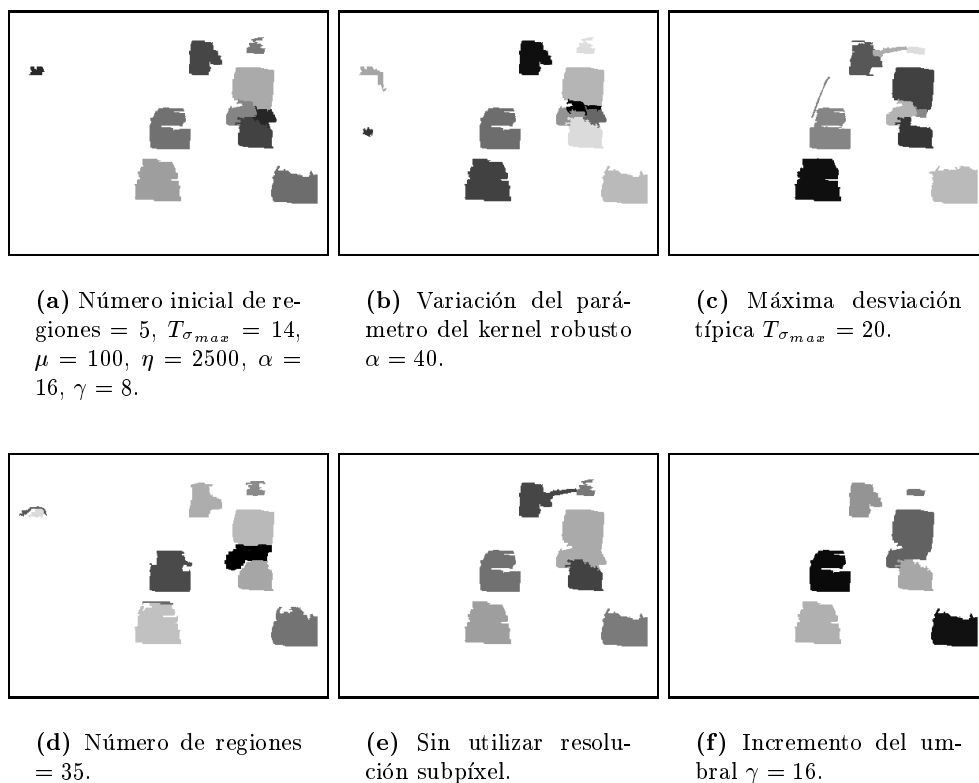


Figura 3.17: Resultados de la segmentación de la imagen de la subfigura 3.16.a utilizando varias combinaciones de parámetros.

gura 3.16.a, pero empleando otras combinaciones de parámetros. En general, se puede observar que el método tiene un buen comportamiento cuando se introducen ligeras modificaciones sobre los parámetros. Esto significa que no es necesario buscar los parámetros óptimos para cada secuencia, sino que más bien, existe un amplio rango de valores que proporcionan resultados satisfactorios.

La subfigura 3.17.a es una segmentación en la que se han utilizado los siguientes parámetros: número de regiones = 5, $T_{\sigma_{max}} = 14$, $\mu = 100$, $\eta = 2500$, $\alpha = 16$, $\gamma = 8$. Los cuatro primeros parámetros corresponden a la segmentación estática (número inicial de regiones, desviación típica, y los tamaños mínimo y máximo de región), mientras que los dos últimos, respectivamente, son el parámetro del kernel robusto, y el umbral empleado en la reducción del número de regiones para calcular las imágenes diferencia. Por su parte, en la Figura 3.16 los parámetros que se utilizaron fueron: número de regiones = 6, $T_{\sigma_{max}} = 12$, $\mu = 150$, $\eta = 2000$, $\alpha = 19$, $\gamma = 8$. En ambas segmentaciones (subfiguras 3.16.b y 3.17.a) los resultados son cualitativamente similares. Las pequeñas diferencias que se observan con respecto a la forma de los vehículos se deben a los cambios introducidos en los parámetros de la segmentación

estática.

Para poder apreciar el efecto de cada parámetro, el resto de subfiguras de la Figura 3.17 muestran modificaciones de un único parámetro con respecto a los que se han usado en la subfigura 3.17.a. Hemos observado que una variación del parámetro α en el rango 15 - 25 no produce cambios significativos en la segmentación final. Así, en la subfigura 3.17.b se ha incrementado este parámetro hasta un valor desmesurado ($\alpha = 40$). Esto hace aumentar la influencia de los outliers en la estimación de movimiento de cada región. Como consecuencia, el algoritmo introduce algunos errores en el cálculo de los parámetros de movimiento de algunas regiones, que, finalmente, no son correctamente segmentadas.

En la subfigura 3.17.c se ha modificado el valor de la máxima desviación típica $T_{\sigma_{max}}$. Este parámetro ha pasado de valer 14 en la subfigura 3.17.a a valer 20 ahora. Su valor afecta a la segmentación basada en nivel de gris (segmentación estática): cuanto mayor es su valor, más grandes son las regiones resultantes. Hemos observado que la variación de este parámetro en el rango 10 - 18 no influye significativamente en las segmentaciones. En cualquier caso, a pesar de haber usado un valor $T_{\sigma_{max}} = 20$ se han segmentado todos los vehículos, si bien, al no haberse producido todas las divisiones necesarias de las regiones, esto ha afectado a las siluetas finales de los vehículos.

La subfigura 3.17.d corresponde a una variación del número inicial de regiones. Aunque las formas extraídas no han variado de manera importante, si que se ha producido un efecto perjudicial: el coste computacional se ha incrementado considerablemente. Al pasar de 6 a 35 regiones iniciales, el coste ha aumentado un 61.5%.

La subfigura 3.17.e trata de mostrar la importancia de la estimación subpíxel. Aquí ha sido suprimida, con lo que algunas regiones adyacentes que pertenecen a diferentes objetos han sido finalmente unidos. La pérdida de la información subpíxel ha sido la que ha impedido distinguir los diferentes movimientos de estos vehículos.

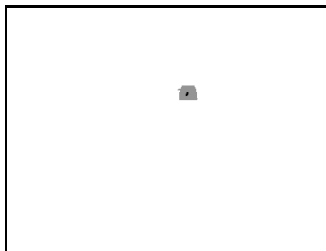
Finalmente, en la subfigura 3.17.f, se ha incrementado el umbral γ del cálculo de imágenes diferencia. Consecuentemente, la cantidad de píxeles diferencia se ha reducido, con respecto a la subfigura 3.17.a, y algunas regiones adyacentes se han unido.

La Figura 3.18 presenta la segmentación de varias imágenes de una secuencia de coches circulando por una autopista. En esta secuencia, los vehículos se mueven a velocidades que rondan los 100km/h. La subfigura 3.18.b corresponde a la segmentación de un coche. Ésta es la primera imagen en la que el coche aparece en la segmentación, ya que en la anteriores su movimiento en la imagen era demasiado próximo a cero. De manera similar, la subfigura 3.18.d es la primera segmentación en la cual se detectan dos nuevos coches cuya velocidad ha pasado a ser suficientemente distinta de cero. La subfigura 3.18.e es la imagen 55 y la subfigura 3.18.f su correspondiente segmentación. Finalmente, la subfigura 3.18.h es la segmentación de la imagen número 100, subfigura 3.18.g.

Los costes computacionales de las fases de estimación y segmentación de movimiento son lineales, siendo éstas, respectivamente, $O(Np^2)$ y $O(Nq)$, donde N es el número de píxeles de la imagen y, p y q son dos constantes de valor pequeño si se las



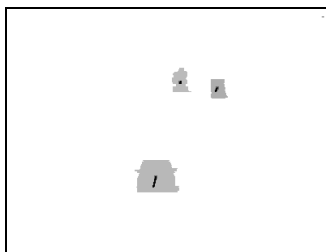
(a) Sexta imagen.



(b) Segmentación de la sexta imagen.



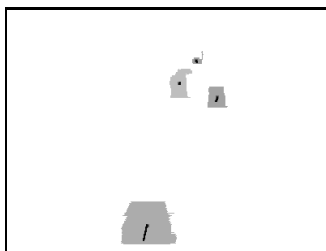
(c) Imagen número 46.



(d) Segmentación de la imagen número 46.



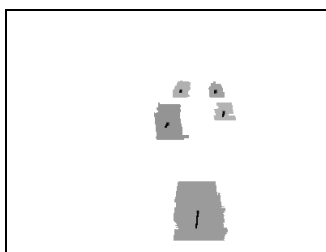
(e) Imagen número 55.



(f) Segmentación de la imagen número 55.



(g) Imagen número 100.



(h) Segmentación de la imagen número 100.

Figura 3.18: Cuatro imágenes de una secuencia de autopista con sus respectivas segmentaciones.

compara con N . Tras una larga experimentación, se ha estimado que los valores de esas dos contantes varían entre los rangos: $1 \leq p \leq 8$ y $0 < q \leq 49$.

3.7 Conclusiones

En este capítulo hemos presentado un método de estimación y segmentación de movimiento especialmente diseñado para escenas de tráfico, aunque también utilizable en otro tipo de secuencias donde la cámara permanezca estática, y el movimiento siga aproximadamente un modelo translacional. El método es una combinación de diversas técnicas, las cuales hemos desarrollado o adaptado para alcanzar el objetivo propuesto.

Podemos descomponer el método en distintas fases: detección, estimación y segmentación. En la primera se realiza una detección de movimiento, la cual reduce el proceso a las zonas de verdadero interés. Esto proporciona dos grandes beneficios: reducción del coste computacional y simplificación del problema. Esta técnica, aunque estrechamente relacionada con otras existentes, proporciona el beneficio de generar una imagen referencia al mismo tiempo que se procesan las imágenes. No es necesario detener el proceso hasta que se genere la imagen de referencia, sino que el algoritmo puede seguir trabajando aunque no haya sido estimada completamente, o incluso si ni siquiera existe.

La fase de reducción del número de regiones basada en substracción de imágenes consecutivas permite, en combinación con la anteriormente comentada detección, aislar del fondo estático las regiones que pertenecen a los vehículos. Esta fase ofrece un diferente enfoque para el mismo problema de la detección de movimiento, pero el hecho de estar basado en diferentes imágenes le permite actual donde la anterior fase pudiera fallar, pero a un mismo reducido coste computacional. Además, esta fase colabora en la regeneración de la imagen referencia del fondo estático cuando ésta debe descartarse.

El procedimiento de estimación de movimiento ha sido diseñado empleando multiresolución, precisión subpíxel y estadísticos robustos. El resultado son estimaciones con el suficiente potencial como para discriminar entre regiones adyacentes con diferentes movimientos, y además afrontar problemas como la existencia de mínimos locales en la función DFD y la violación de la premisa de movimiento simple dentro de cada región.

La segmentación de movimiento ha sido diseñada en dos pasos para conseguir una mayor tolerancia ante posibles errores de estimación. La primera sólo une aquellas regiones adyacentes que por la similitud de sus estimaciones nos proporcionan la suficiente certeza como para considerarlas partes de un mismo objeto. La segunda fase permite reconsiderar las estimaciones para así unir regiones vecinas. Esta reconsideración se basa en la potencialidad de que dos regiones vecinas sean parte de un mismo objeto. Esta reestimación ofrece una mayor robustez al método, permitiéndole afrontar violaciones de la premisa de movimiento translacional.

Como se ha visto en los resultados, el empleo de un modelo de movimiento translacional en modo alguno dificulta la segmentación. Es más, una estimación de movi-

miento basada en este modelo aporta suficiente información como para separar en la mayoría de casos regiones adyacentes que pertenecen a diferentes vehículos.

El método se basa en la realización de una segmentación estática previa. En el capítulo anterior se propuso un método de segmentación para tales fines, que como allí también se dijo, puede ser substituido por cualquier otro que cumpla las mismas condiciones. En este capítulo se ha mostrado que la introducción de dicho paso previo, y el empleo de la información de intensidad como criterio de homogeneidad resultan muy adecuados para las escenas de tráfico. De una manera rápida y segura se consigue realizar la mayor parte de la segmentación, principalmente agrupar los píxeles que corresponden al fondo, sin todavía haber iniciado la estimación de movimiento.

Los costes computacionales de los procesos de estimación y segmentación de movimiento son lineales. Este hecho hace que pueda formar parte del sistema total, al cual se le exigía que trabajase en tiempo real con un hardware no dedicado. Como se verá en los siguientes capítulos, se consigue este requerimiento.

Capítulo 4

Seguimiento e integración temporal morfológica

Contenido

4.1	Introducción	87
4.1.1	Breve revisión bibliográfica	88
4.1.2	Un método de seguimiento e integración temporal morfológica	93
4.1.3	Estructura del capítulo	95
4.2	El algoritmo	95
4.3	Seguimiento de regiones	97
4.3.1	Correspondencias entre regiones	97
4.3.2	Estimación de los parámetros de movimiento	99
4.4	Memoria temporal	102
4.5	Recuperando regiones perdidas	106
4.5.1	Planteamiento del problema	107
4.5.2	Búsqueda de la región R^k	107
4.5.3	Generación de la región R^l	108
4.6	Generación de una segmentación final	108
4.7	Resumen del algoritmo	108
4.8	Resultados	109
4.9	Conclusiones	114

4.1 Introducción

Este capítulo desarrolla dos importantes conceptos del análisis de imágenes dinámicas: el *seguimiento* y la *integración temporal*. Hasta el momento tan sólo hemos presentado

el análisis de pares de imágenes consecutivas. Lo que mostramos a lo largo de este capítulo es la extensión de ese análisis a toda la secuencia.

A continuación y dentro de esta misma sección mostramos una introducción a ambas técnicas, junto con una breve revisión bibliográfica de los trabajos existentes relacionados con las mismas. En el Apartado 4.1.2 describimos someramente el método desarrollado, y en el Apartado 4.1.3 describimos el contenido del resto del capítulo.

4.1.1 Breve revisión bibliográfica

Partimos de la premisa de que los objetos de la escena se desplazan sin variaciones bruscas. Dicha premisa en modo alguno limita la generalidad de nuestro método, puesto que es aplicable a la inmensa mayoría de secuencias. Además, utilizaremos esta restricción para relacionar las segmentaciones obtenidas a partir de los pares de imágenes y obtener así un conocimiento del comportamiento de los objetos mucho más fundamentado. Es obvio que cuanto más información reciba un sistema, mayor será la calidad de sus resultados. En nuestro caso, esto se traduce en que cuantas más imágenes utilicemos para realizar nuestras estimaciones, más fiables serán las mismas, y menor será la influencia de los errores puntuales que puedan producirse en el análisis de pares de imágenes.

Como se ha visto en el capítulo anterior, el análisis de imágenes consecutivas proporciona una medición del movimiento para cada grupo conexo de píxeles que muestra un desplazamiento diferenciado del de sus vecinos. Cada uno de estos grupos, regiones, muy probablemente corresponderá a un objeto diferente, el cual irá variando de una manera continua su posición y forma en la imagen. Desgraciadamente no podemos presuponer que todas las mediciones de movimiento y forma sean precisas, más bien al contrario, hemos de tener en cuenta que la presencia de ruido provocará que las mismas no sigan una evolución completamente uniforme.

A lo largo de este capítulo proponemos un método que permite integrar las diferentes observaciones obtenidas a partir del análisis de pares de imágenes. Esta integración lleva a cabo dos importantes funciones: el seguimiento de la evolución de los objetos, y el aumento de la robustez de las estimaciones de movimiento y forma. Para ello, en primer lugar deberá realizarse una correspondencia entre las regiones que forman parte de la segmentación de cada una de las imágenes, y en segundo lugar, deberá utilizarse algún tipo de memoria temporal que permita integrar las observaciones obtenidas durante la secuencia, de forma que en cada nueva imagen se puedan generar estimaciones basadas en el conocimiento acumulado.

A este acopio de mediciones, el cual permite seguir, e incluso predecir, la evolución de las entidades extraídas de la imagen, es a lo que llamamos *integración temporal*. Este concepto ha sido frecuentemente utilizado para aumentar la robustez de las estimaciones de movimiento [Zheng y Chellappa, 1995, Isard y Blake, 1996, Zhang y Faugeras, 1992, Koller *et al.*, 1994b], lo que se conoce como *seguimiento*. Sin embargo, apenas se ha explotado su utilidad de cara a la mejora de las segmentaciones [Irani *et al.*, 1992, Smith y Brady, 1995], lo que aquí llamamos *integración temporal*

morfológica, y que algunos autores denominan, simplemente, *integración temporal*. Preferimos añadirle el calificativo “morfológica” para distinguirla claramente de otros tipos de integración temporal, como pueden ser el propio seguimiento, o la integración de información de profundidad [Matthies y Kanade, 1989].

El seguimiento se basa en la utilización de correspondencias entre elementos de la imagen, o en estimaciones de flujo óptico. En ambos casos, éstas se obtienen a lo largo de varias imágenes, sirviendo como observaciones para el algoritmo de seguimiento. Este algoritmo, en esencia, determina los parámetros de movimiento que mejor se adecuan al conjunto de observaciones.

En la mayoría de métodos, la segmentación proporciona información para llevar a cabo el seguimiento. Nuestra idea es que a su vez el seguimiento también puede ayudar a mejorar la segmentación. El seguimiento, por medio de las correspondencias, relaciona las distintas apariciones de una misma entidad a lo largo de una secuencia, y además determina su movimiento. Por tanto, nos permite seguir la evolución de las formas. Dado que las variaciones son progresivas y que se conocen los parámetros de movimiento, se puede saber en cada imagen qué píxeles tienen una mayor probabilidad de pertenecer a cada entidad. Ello significa que es posible predecir y corregir las segmentaciones si se integra la información morfológica obtenida a partir de cada imagen, *integración temporal morfológica*.

Con la excepción de unos pocos trabajos, se han ignorado los beneficios que el seguimiento de objetos móviles puede aportar a la segmentación. Irani et al. [Irani et al., 1992] utilizó la integración temporal de los resultados de la segmentación de movimiento para reducir su tiempo de cálculo. Sin embargo, este método no realizaba un seguimiento de la forma, con lo que la información acerca de los eventos de la escena no era directamente disponible. El método de Meyer y Bouthemy [Meyer y Bouthemy, 1994] realiza un seguimiento de los contornos de los objetos móviles. La integración temporal se consigue por medio del almacenamiento de representaciones poligonales de cada uno de los objetos. Ello permite la detección de oclusiones por medio de la comparación entre las regiones segmentadas y las predicciones de sus contornos. Smith y Brady [Smith y Brady, 1995] utilizan un *mapa radial* para mantener la forma de los objetos. Este mapa se combina con cada segmentación para producir la segmentación final de las imágenes. Esto le permite al método detectar oclusiones, permitiendo en estas situaciones obtener una aproximación al contorno total del objeto. En [Mae et al., 1996], se hacen corresponder las discontinuidades en el campo de flujo con las de intensidad, bordes. Estas correspondencias representan bordes móviles, los cuales son acumulados y usados de dos formas: como predicción de la posición de los bordes en la imagen actual, y como substitutos de los bordes perdidos, con el fin de conseguir contornos continuos.

“A tracking process can be defined by (1) a representational model of tokens (e.g., image coordinates of distinguishable points, mid-point coordinates, length and orientation of contour segments), (2) a kinematic model of evolution of tokens (e.g., constant velocity, constant acceleration, affine motion), (3) a set of relations between model parameters and image mea-

surements, and (4) a temporal filter to estimate model parameters from image data (a classic filter that has been commonly used is the Kalman filter).” [Mitiche y Bouthemey, 1996]

Dependiendo de la aplicación, se pueden utilizar diferentes modelos de movimiento, siendo los más sencillos aquellos que asumen velocidad constante. Estos modelos pueden clasificarse en dos grupos [Tekalp, 1995]: modelos de movimiento 2-D, los cuales representan trayectorias en el plano de la imagen, y modelos de movimiento 3-D, que representan cinemáticas de movimientos físicos.

En los *modelos de trayectorias 2-D*, éstas pueden basarse en diferentes transformaciones espaciales, translación [Snyder y Rajala, 1983], afín [Shapiro, 1995], perspectiva o poligonal [Jang *et al.*, 1997], lo cual se traducirá en un diferente número de parámetros para expresar las ecuaciones de movimiento. Con estos modelos se puede realizar el seguimiento tanto de facetas individuales, como de regiones. La evolución temporal de los parámetros de las transformaciones pueden modelarse por medio de rotaciones 2-D, translaciones, y/o dilataciones, o bien mediante la expansión de orden bajo de las series de Taylor [Meyer y Bouthemey, 1994].

Para los *modelos de movimiento rígido 3-D* se suelen emplear dos alternativas [Tekalp, 1995]:

- *La rotación se define con respecto a un sistema fijo de coordenadas del mundo real* [Weng *et al.*, 1993], donde se asume que el centro de rotación coincide con el origen del sistema de coordenadas del mundo real. “*Este enfoque es, por lo general, inadecuado para problemas de seguimiento*” [Tekalp, 1995]. Esto se debe a que la presencia de translaciones puede hacer variar la posición del centro de rotación, con lo que los parámetros de rotación serían diferentes en cada imagen de la secuencia.
- *La rotación se define con respecto a un eje de rotación* [Young y Chellappa, 1990, Haag y Nagel, 1998, Haag y Nagel, 1999]. En este caso, la ecuación del eje de rotación, el cual es inicialmente desconocido y se traslada en el tiempo, sólo puede ser determinada tras resolver las ecuaciones de movimiento.

Otra posible clasificación de los modelos de movimiento empleados en el seguimiento es la de rígidos y deformables. Se ha propuesto el empleo de modelos de contornos activos [Leymarie y Levine, 1993, Stark y Ihle, 1997, Blake y Isard, 1998], tales como snakes [Ip y Shen, 1998] y modelos deformables [Kervrann y Heitz, 94] para seguimiento de movimiento deformable 2-D, y supercuadráticas [Metaxas y Terzopoulos, 1993] para seguimiento de movimiento deformable 3-D.

La correspondencia de entidades y la estimación de flujo óptico plantean dificultades debido a las ambigüedades de las correspondencias o la no existencia de las mismas debido a oclusiones. Para resolver estos casos se han propuesto diversas técnicas de asociación de datos [Bar-Shalom y Fortman, 1988, Rao, 1992, Cox, 1993]. Dichas

técnicas conllevan una búsqueda dentro de una ventana finita, la cual se centra alrededor de la posición que el algoritmo de seguimiento predice. Las siguientes tres asunciones ayudan a la formulación del problema de la correspondencia:

- La posición de un punto será relativamente similar de una imagen a la siguiente.
- La velocidad escalar de un punto será relativamente similar de una imagen a la siguiente.
- La dirección del movimiento de un punto será relativamente similar de una imagen a la siguiente.

Una vez que el modelo dinámico y un número de correspondencias sobre múltiples imágenes han sido determinados, se trata de buscar aquellos parámetros de movimiento que son consistentes con el modelo y las observaciones. Para ello se pueden usar dos tipos de técnicas: por lotes (*batch*) y recursivas.

Los *estimadores por lotes*, tales como el mínimo-cuadrático, procesan todo el conjunto de datos de una vez, después de que estos hayan sido recogidos. Por contra, los *estimadores recursivos*, tales como los filtros de Kalman, van actualizando los parámetros conforme se van introduciendo nuevas observaciones. Los *estimadores por lotes* tienden a ser numéricamente más robustos que los recursivos, principalmente cuando la ecuación de transición de estado, o la de observación, no son lineales. Sin embargo, los métodos recursivos son ventajosos cuando se requieren estimaciones en tiempo real, ya que, a diferencia de los métodos por lotes, no necesitan procesar el conjunto total de observaciones cada vez que se introduce una nueva. En [Weng *et al.*, 1993] se propuso un método híbrido, llamado *recursivo-por-lotes*, el cual combina eficiencia computacional con robustez en los resultados.

Los métodos que se encuentran en la literatura pueden seguir diferentes entidades: puntos, líneas, contornos, regiones, modelos, etc.. Atendiendo al tipo de entidades sobre el que cada método efectúa el seguimiento, podemos realizar una clasificación en cuatro grupos [Beymer *et al.*, 1997]:

- Seguimiento basado en modelos tridimensionales.
- Seguimiento basado en características.
- Seguimiento basado en regiones.
- Seguimiento basado en modelos bidimensionales (modelos de contornos activos).

Los *modelos tridimensionales* consisten en representaciones geométricas precisas de objetos conocidos, los cuales pueden estar colocados en posiciones y orientaciones arbitrarias. Usando el conocimiento de la geometría de la cámara y de la escena, se proyecta el modelo tridimensional sobre el plano de la imagen, tratando de hacer coincidir las facetas del modelo y de la imagen. Para conseguir esta coincidencia se realiza una búsqueda de la posición y orientación que haga máxima la coincidencia.

Una vez obtenida la posición y orientación del objeto en la siguiente imagen, para que esta predicción sea segura, se deben estimar los parámetros de movimiento. Las estimaciones de la posición, orientación y movimiento se actualizan tras cada nueva imagen, conforme se va llevando a cabo el seguimiento.

Los métodos tridimensionales tienen un coste computacional muy elevado. Se puede obtener una reducción significativa si se cumple la premisa de que el objeto descansa sobre el suelo de la escena. Dado que se conoce la geometría de la escena, el seguimiento del objeto se reduce a búsquedas de movimientos de translación sobre el suelo y rotación alrededor del eje vertical al plano del suelo [Worrall *et al.*, 1991, Haag y Nagel, 1998, Haag y Nagel, 1999, Koller *et al.*, 1993, Tan *et al.*, 1998, Frank *et al.*, 1996, Gardner y Lawton, 1996, Gerard y Gagalowicz, 2000].

Los métodos basados en características, tales como puntos [Zheng y Chellappa, 1995, Sudhir *et al.*, 1997, Sethi y Jain, 1987, Beymer *et al.*, 1997], líneas [Deriche y Faugeras, 1990, Castellow *et al.*, 1987, Liu y Huang, 1991, Clarke *et al.*, 1996] o curvas [Zhang, 1992, Isard y Blake, 1996], realizan el seguimiento normalmente usando técnicas de correspondencia. Dada la escasa información que aportan las características, para cada una puede haber varios candidatos, por lo que frecuentemente no basta con asociar cada una de ellas con la más similar de la siguiente imagen, *regla del vecino más próximo*. Se han empleado diversos métodos: filtro de seguimiento-partición (*track-splitting filter*) [Smith y Buechler, 1975, Zhang y Faugeras, 1992], filtro de asociación de datos de probabilidad conjunta (JPDAF) [Fortmann *et al.*, 1983, Bar-Shalom *et al.*, 1990, Chang y Aggarwal, 1991], programación entera de probabilidad conjunta [Morefield, 1977], hipótesis múltiple [Reid, 1979, Cox y Leonard, 1991], programación dinámica [Shapiro *et al.*, 1995]. El uso de facetas presenta dos desventajas importantes: no proporcionan una explícita agrupación de las facetas que se mueven coherentemente [Mitiche y Bouthemy, 1996], y son sensibles a las oclusiones [Mitiche y Bouthemy, 1996, Tekalp, 1995].

Los modelos deformables [Kass *et al.*, 1988, Jain *et al.*, 1996, Cootes *et al.*, 1995] fueron desarrollados para detectar y localizar formas que experimentan modificaciones. Han sido aplicados a una gran variedad de problemas tales como la detección de bordes o contornos [Kass *et al.*, 1988], seguimiento de movimiento [Leymarie y Levine, 1993, Remagnino *et al.*, 1997] y la segmentación de imágenes de resonancias magnéticas [Cohen y Cohen, 1993].

“Snakes are a mechanism for bringing a certain degree of prior knowledge to bear on low-level image interpretation. Rather than expecting desirable properties such as continuity and smoothness to emerge from image data, those properties are imposed from the start.” [Blake y Isard, 1998]

Una *snake* se define como un conjunto ordenado de puntos $\{V_i | i = 1, 2, \dots, N\}$. La energía total de una snake es la suma ponderada de varios términos de energía, los cuales se dividen en energías internas y externas [Park y Han, 1997, Blake y Isard, 1998]. La energía interna trata de preservar la continuidad de la curva, mientras la externa intenta adaptar la curva a la imagen. Cuando se dispone de un conocimiento previo

acerca de la forma de los objetos, éste puede incorporarse en forma de energía [Radeva y Marti, 1995, Radeva *et al.*, 1995, Ip y Shen, 1998]. Dado que la adaptación de la snake a la imagen es un proceso interactivo, es necesario llevar a cabo una inicialización. Esto representa un problema, ya que la snake inicial debe estar cerca de la solución deseada, pues de otro modo puede quedar atrapada en otro mínimo local debido a la energía interna. Dependiendo del tipo de aplicación, la inicialización puede tener que ser realizada por el usuario, o bien podrá ser automática [Jolly *et al.*, 1996]. Una vez inicializada la snake, el proceso consiste en minimizar iterativamente su energía. En la literatura se encuentran diferentes enfoques para realizar este proceso: elementos finitos [Kass *et al.*, 1988], programación dinámica [Amini *et al.*, 1990], algoritmos voraces (greedy) [Williams y Shah, 1992, Jang y Choi, 2000], redes neuronales [Tsai *et al.*, 1993]. Cuando la primera imagen de la secuencia ha sido procesada, se puede usar como inicialización una predicción de la posición del modelo deformable, así, es frecuente el uso de filtros de Kalman [Blake *et al.*, 1993, Koller *et al.*, 1994b, Terzopoulos y Szeliski, 1992, Arsénio y Santos-Victor, 1997, Peterfreund, 1999, Jang y Choi, 2000].

Los *métodos basados en regiones* definen grupos de píxeles conexos que pertenecen a un único objeto. El seguimiento en este caso consiste en relacionar las regiones de las distintas segmentaciones para estimar los parámetros de movimiento. En [Kottke y Sun, 1994, Brock-Gunn *et al.*, 1994, Teal y Ellis, 1996] se realizan seguimientos de regiones basados en buscar las de máxima similitud. Sin embargo, los anteriores trabajos se limitan a establecer correspondencias sin realizar una estimación de movimiento que integre las mediciones de toda la secuencia. Wixson [Wixson, 1996] utiliza un filtro predictivo para facilitar la búsqueda de regiones correspondientes. En [Gil *et al.*, 1994] se incorporan dos filtros de Kalman, uno para estimar la posición de las regiones y otro para estimar su velocidad.

4.1.2 Un método de seguimiento e integración temporal morfológica

“Tracking regions rather than discrete tokens may yield more robust results and provide new means to detect occlusion” [Tekalp, 1995]

Como hemos venido comentando a lo largo de esta Tesis, nuestro método se basa en regiones. Tal y como apuntan Mitiche y Tekalp [Mitiche y Bouthemy, 1996, Tekalp, 1995], el uso de regiones aporta robustez al proceso de seguimiento, puesto que lo hace mucho menos sensible a las oclusiones. Otra ventaja es que facilitan el establecimiento de correspondencias. Dado el mayor tamaño de las regiones frente a características más simples como puntos o curvas, un método de asociación de datos basado en el vecino más próximo produce resultados correctos. No es necesario utilizar técnicas más complejas (JPDAF, hipótesis múltiple, etc.), pues la información que aporta cada región es suficiente para evitar ambigüedades.

Los métodos basados en regiones presentan varias ventajas con respecto a los otros tres grupos de técnicas:

- *Modelos deformables.* Frente a los métodos basados en modelos deformables,

tal y como apuntan varios autores [McLauchlan y Malik, 1997, Beymer *et al.*, 1997, Mitiche y Bouthemy, 1996], las regiones no presentan el problema de la inicialización, dado que no se parte de ningún modelo preestablecido. Al contrario que con los primeros, no importa que los objetos estén parcialmente ocultos, se puede iniciar el proceso con la parte visible.

- *Métodos basados en características.* Gil et al. [Gil *et al.*, 1994] indican que el uso de regiones es más apropiado que el de contornos cuando el tamaño de los objetos es pequeño y se producen cambios en las condiciones de iluminación, como a menudo es el caso en las escenas de tráfico.
- *Modelos tridimensionales.* La principal ventaja de la utilización de regiones frente a los modelos tridimensionales es su menor coste computacional. Es cierto que dichos modelos aportan un mayor conocimiento sobre el comportamiento y tipo de los objetos. No obstante, como ya se comentó en el Capítulo 1, ninguna de las tareas que debe realizar un sistema de monitorización de tráfico justifica el empleo de técnicas tan complejas. Para ser capaces de utilizar dichas técnicas, deberíamos disponer de un sistema dedicado de alto coste, lo cual incumpliría una de nuestras premisas.

En este capítulo proponemos un método de seguimiento e integración temporal morfológica. El seguimiento se basa en una simple correspondencia entre las regiones obtenidas en las segmentaciones de cada imagen. Para ello se busca utilizar una base de datos donde se incluye cada nueva región que aparece en una segmentación. El proceso de correspondencia toma cada nueva segmentación y asocia a cada región de la base de datos aquella otra de la segmentación actual que resulta más similar.

El seguimiento se completa con la utilización de un filtro de Kalman que permite refinar las estimaciones de movimiento obtenidas para cada imagen. Debido al efecto de la perspectiva que aparece en las secuencias de tráfico, en este filtro consideramos que los objetos se desplazan con un modelo de movimiento donde la aceleración se incrementa de manera uniforme. De esta forma se tiene en cuenta el incremento (decremento) de velocidad que experimentan las regiones cuando los vehículos se aproximan a (alejan de) la cámara.

El seguimiento permite conocer la evolución de las regiones. Esta evolución es recogida mediante una estrategia de máscaras que nos permite corregir la segmentación de cada imagen. Se asume que los cambios en la forma de los objetos son graduales, lo mismo que su aparición o desaparición en la escena. Así, la segmentación de una imagen y de la siguiente deben ser muy parecidas. La corrección evita súbitas apariciones o desapariciones de regiones o partes de ellas, asumiendo que este tipo de hechos se deben a errores puntuales de las segmentaciones. Con esta estrategia, no solo la segmentación proporciona información al seguimiento, sino que, a su vez, también el seguimiento colabora en la mejora de la segmentación. La idea básica surgió tras la observación de numerosas segmentaciones de imágenes. En ellas, los errores son inevitables. Así, para la segmentación de un mismo objeto en diferentes imágenes se observan variaciones de forma, que en algunos casos llegan a ser graves: cuando el ob-

jeto no es segmentado, o se une a la segmentación de otro objeto. A partir del estudio de las variaciones, llegamos a una conclusión que puede resultar en cierto modo obvia: incluso en la secuencia peor segmentada, *los píxeles más frecuentemente asignados a un objeto, son aquellos que verdaderamente pertenecen a él*. Partiendo de esta idea, decidimos que era posible llevar a cabo una integración temporal morfológica si se consideraba la pertenencia de cada píxel a un determinado objeto a lo largo de las segmentaciones anteriores. De este modo, la segmentación de cada imagen que se obtiene a partir del análisis de movimiento se considera como una nueva observación que debe tenerse en cuenta para producir la segmentación final. En dicha segmentación final, los píxeles se asignan al objeto a aquel que más frecuentemente han pertenecido.

4.1.3 Estructura del capítulo

A continuación detallamos la estructura del resto del capítulo. La siguiente sección muestra una visión general del método describiendo los principales pasos del algoritmo desarrollado. La Sección 4.3 describe el seguimiento de regiones. Éste consta de dos partes: establecimiento de correspondencias entre regiones, y estimación de los parámetros de movimiento por medio de un filtro de Kalman. La Sección 4.4 muestra lo que denominamos *memoria temporal*. Dicha memoria es una estructura de datos que integra la información proporcionada por cada segmentación acerca de la forma de las regiones. Así mismo, en esta sección, se detalla el citado proceso de integración de información. La Sección 4.5 muestra cómo el algoritmo recupera regiones desaparecidas en la segmentación a partir de la información recogida en la memoria temporal. La Sección 4.6 describe la generación de la segmentación definitiva de cada imagen por medio de la integración temporal morfológica. Dado que el algoritmo presentado en ese capítulo consta de un gran número de pasos, la Sección 4.7 recapitula todo lo expuesto en las secciones precedentes. Aquí se enumeran los diferentes pasos que se han seguido, de manera que el lector tenga una visión global de lo expuesto. Por último, la Sección 4.8 expone algunos resultados sobre varias secuencias de tráfico y la Sección 4.9 comenta las conclusiones del presente capítulo.

4.2 El algoritmo de seguimiento e integración temporal morfológica

La Figura 4.1 muestra el algoritmo general del método. Consiste en un proceso iterativo que va tomando los resultados obtenidos por la estimación y segmentación de movimiento. Dado que la integración temporal genera la segmentación definitiva, tendremos dos segmentaciones para cada imagen. Para distinguirlas, a la obtenida por el análisis de movimiento la llamaremos *segmentación inicial*, mientras que a la segmentación definitiva la denominaremos *segmentación final*.

El primer paso consiste en tomar la segmentación inicial de la primera imagen de la secuencia. Para cada una de las regiones detectadas se crea una entrada en una lista llamada *lista maestra*. Su finalidad es la de servir como referencia para buscar

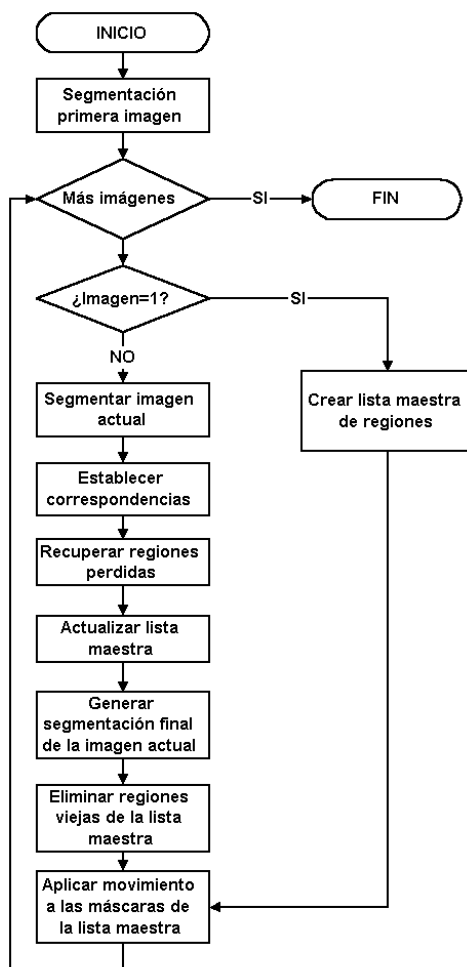


Figura 4.1: Algoritmo de seguimiento e integración temporal morfológica.

las regiones correspondientes a un mismo objeto a lo largo de la secuencia. Para cada una de estas regiones se crea una matriz que contiene información acerca de los píxeles que a ella van siendo asignados en cada segmentación inicial. Para simplificar la explicación del método asumiremos que el tamaño de la matriz es el mismo que el de la imagen, sin embargo, esto puede optimizarse fácilmente usando un tamaño similar al que tiene la región.

En cada iteración se toma una segmentación inicial y se establece una correspondencia entre las regiones de la lista maestra y la segmentación actual. En este mismo paso se utiliza un filtro de Kalman para estimar los parámetros de movimiento de las regiones de la lista maestra que han obtenido correspondencia. Con este filtro se aumenta la robustez de las estimaciones de movimiento, tomando como observaciones los desplazamientos estimados de imagen a imagen.

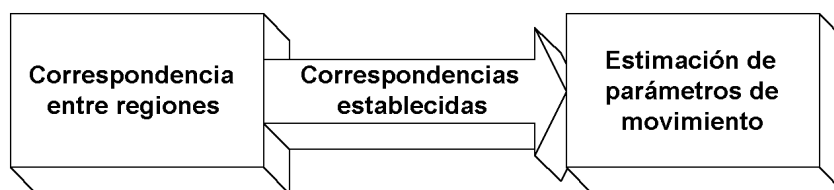


Figura 4.2: Pasos que componen el seguimiento de regiones.

El siguiente paso recupera lo que llamamos *regiones perdidas*. Dado que asumimos que las regiones desaparecen de manera gradual en la imagen, se considera que la desaparición súbita de una región se debe a un error de la segmentación. Lo que se intenta en este paso es recuperar las regiones que estuvieron presentes durante varias imágenes previas y que no lo están en la actual, *regiones perdidas*. Gracias a la información almacenada en las matrices de la lista maestra podemos buscar a dónde han sido asignados sus píxeles y regenerar la región a partir de ellos.

Con la correspondencia entre regiones se actualiza la información de las matrices de la lista maestra y a continuación se genera la segmentación final de la imagen actual. Esta segmentación utiliza los valores de las matrices para asignar cada píxel a la región más probable.

Los dos últimos pasos del algoritmo preparan la lista maestra para la siguiente iteración. Primero se simplifica la lista eliminando las regiones que no han obtenido correspondencia en las últimas iteraciones. Dichas regiones corresponden a objetos que han abandonado la escena o a errores puntuales de las segmentaciones iniciales (regiones que no están relacionadas con un objeto). Por último, se aplica el movimiento de cada región a su matriz. De esta forma se desplaza la información de cada píxel a la posición donde estará en la siguiente imagen.

4.3 Seguimiento de regiones

El seguimiento permite averiguar las trayectorias de las regiones segmentadas. Básicamente, esta fase consta de dos partes: correspondencias entre regiones y estimación de los parámetros de movimiento de cada región.

4.3.1 Correspondencias entre regiones

Las correspondencias se establecen entre las regiones extraídas de cada imagen y las que contiene la lista maestra. De este modo se relacionan las entidades que corresponden a un mismo objeto y se sigue su comportamiento a lo largo de la secuencia.

En un esquema de correspondencia, las regiones presentan dos importantes ventajas frente a otro tipo de características más simples como puntos, líneas o curvas. Ambas ventajas se derivan de su mayor tamaño y son: contienen una mayor información y su número en la imagen es menor. Dichas características permiten que una

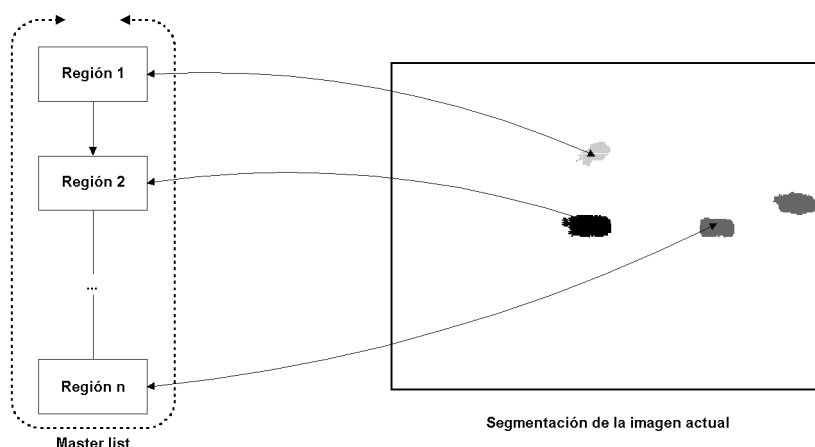


Figura 4.3: Se establece una correspondencia entre las regiones de la *lista maestra*, master list, y las regiones extraídas de la imagen actual.

estrategia basada en el vecino más próximo (correspondencia de una región con aquella otra que resulta más similar) sea suficiente sin la necesidad de recurrir a otro tipo de estrategias de mayor complejidad.

Dado que las características que se usan para establecer la correspondencia son diferentes, empleamos pesos para equilibrar su importancia. El proceso consta de dos pasos: correspondencia y actualización de pesos.

1. *Correspondencia.* Se establece una correspondencia entre cada región de la lista maestra y la más similar de las extraídas de la imagen actual. Para buscar la similitud se consideran cinco características de las regiones: las coordenadas x e y del centro de masas, la intensidad media y los dos parámetros de movimiento translacional. Estas características definen un vector $\mathbf{m} = (m_x, m_y, m_g, v_x, v_y)^t$ para cada región, donde (m_x, m_y) es el centroide de la región, m_g es la intensidad media, y (v_x, v_y) es la velocidad estimada. La correspondencia se establece entre aquellas dos regiones R^n y R^m que minimizan la siguiente distancia euclídea ponderada al cuadrado:

$$d^{mn} = (\mathbf{m}^m - \mathbf{m}^n)^t \mathbf{W} (\mathbf{m}^m - \mathbf{m}^n) \quad (4.1)$$

donde \mathbf{m}^m es el vector de características de la región de la lista maestra R^m , \mathbf{m}^n es el vector de la nueva región R^n y \mathbf{W} es una matriz de pesos con dimensiones 5×5 . \mathbf{W} contiene en su diagonal principal los pesos de las cinco características $(w_x, w_y, w_g, w_{v_x}, w_{v_y})$, y el resto son ceros.

2. *Actualización de los pesos.* En cada iteración se recalculan los pesos de forma que se minimice la suma total de las distancias d^{mn} , siendo R^m y R^n dos regiones

correspondientes. Si k^m es el número de correspondencias que deseamos establecer, se tratará de elegir aquellos pesos $(w_x, w_y, w_g, w_{v_x}, w_{v_y})$ que minimicen la suma total de distancias d_T :

$$\min_{w_x, w_y, w_g, w_{v_x}, w_{v_y}} d_T = \sum_{m=1}^{k^m} d^{mn}, \quad (4.2)$$

Esta minimización estará, además, sujeta a la restricción

$$w_x w_y w_g w_{v_x} w_{v_y} = 1, \quad (4.3)$$

la cual evita el aumento incontrolado de los valores de los pesos.

Aplicando el método de los *Multiplicadores de Lagrange* se pueden calcular los pesos en la anterior minimización.

$$d_T = \sum_{m=1}^{k^m} d^{mn} + \lambda(w_x w_y w_g w_{v_x} w_{v_y} - 1) \quad (4.4)$$

Calculando el valor de λ en (4.4) obtenemos los valores de los pesos:

$$w_x = \frac{\sqrt[5]{(\sum_{m=1}^{k^m} (m_y^m - m_y^n)^2)^2} \sqrt[5]{(\sum_{m=1}^{k^m} (m_g^m - m_g^n)^2)} \sqrt[5]{(\sum_{m=1}^{k^m} (m_{v_x}^m - m_{v_x}^n)^2)}}{\sqrt[5]{(\sum_{m=1}^{k^m} (m_{v_y}^m - m_{v_y}^n)^2)} ((\sum_{m=1}^{k^m} (m_x^m - m_x^n)^2)^{-4/5}} \quad (4.5)$$

$$w_k = \frac{w_x \sum_{m=1}^{k^m} (m_k^m - m_k^n)^2}{\sum_{m=1}^{k^m} (m_x^m - m_x^n)^2}, \quad k = \{y, g, v_x, v_y\} \quad (4.6)$$

Los dos pasos del procedimiento se repetirán hasta que la suma de distancias d_T alcance un mínimo. Para acelerar la convergencia del proceso, y considerando las tres asunciones referidas en la página 91 (posición, velocidad y dirección similares), antes de calcular la distancia d^{mn} se descartan aquellas regiones con centroides separados por una distancia mayor a un umbral u_d .

4.3.2 Estimación de los parámetros de movimiento

La misión de esta parte del método es determinar para cada región los parámetros de movimiento que mejor se adecuan al conjunto de observaciones. El conjunto de observaciones son las estimaciones de movimiento que se realizan para cada imagen, y el modelo de movimiento usado es el translacional. Más concretamente asumiremos un movimiento uniformemente acelerado, lo cual nos permitirá modelar el efecto de aceleración o deceleración que experimentan las regiones cuando se acercan o alejan de la cámara.

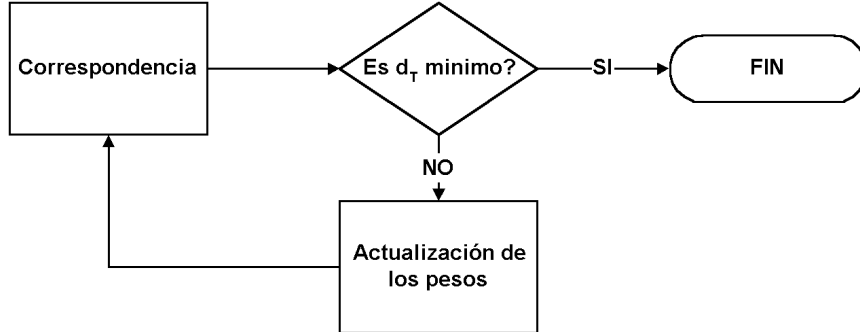


Figura 4.4: Algoritmo iterativo de correspondencia de regiones.

Para realizar la estimación, nosotros, al igual que la mayoría de trabajos, emplearemos un estimador recursivo. Como comentamos en el Apartado 4.1.1, esta clase de estimadores es más apropiada para aplicaciones en tiempo real, debido a que no requieren procesar todo el conjunto de observaciones cada vez que se introduce una nueva.

Como estimador utilizamos un *filtro de Kalman* (ver Apéndice B), el cual permite estimar el mejor valor, en el sentido mínimo-cuadrático, de un vector de estados a partir de un conjunto de mediciones con ruido *gaussiano* en un sistema dinámico lineal. De hecho, modelaremos el movimiento de una región en la imagen como un sistema dinámico y lineal.

La velocidad de una región en el instante k se define por dos componentes $\mathbf{v}_k = \{v_{xk}, v_{yk}\}$. El modelo de movimiento deberá considerar el incremento de la velocidad que experimentan las proyecciones de los vehículos cuando éstos se acercan a la cámara. De este modo, asumiremos que la variación de la aceleración ($\mathbf{a}_k = \{a_{xk}, a_{yk}\}$) es constante y representada por el vector $\mathbf{c} = \{c_x, c_y\}$. Por tanto, el modelo cinemático para las evolución de las regiones quedará definido por las ecuaciones (4.7-4.11), en las que Δt es el incremento de tiempo.

$$\mathbf{e}_k = 1/6\mathbf{c}_{k-1}\Delta t^3 + 1/2\mathbf{a}_{k-1}\Delta t^2 + \mathbf{v}_{k-1}\Delta t + \mathbf{e}_{k-1} \quad (4.7)$$

$$\mathbf{v}_k = 1/2\mathbf{c}_{k-1}\Delta t^2 + \mathbf{a}_{k-1}\Delta t + \mathbf{v}_{k-1} \quad (4.8)$$

$$\mathbf{a}_k = \mathbf{a}_{k-1} + \mathbf{c}_{k-1}\Delta t \quad (4.9)$$

$$\mathbf{c}_k = \mathbf{c}_{k-1} = \mathbf{c} \quad (4.10)$$

En las Ecuaciones (4.7), (4.8) y (4.9), \mathbf{e}_k , \mathbf{v}_k y \mathbf{a}_k denotan la posición, la velocidad y aceleración de una región en el instante k , y Δt es la variación del tiempo, que podemos suponer como la unidad sin pérdida de generalidad.

$$\Delta t = 1 \quad (4.11)$$

La Ecuación (4.10) define un movimiento en el plano de la imagen en el que la aceleración se incrementa uniformemente. Conviene señalar que para obtener un modelo de movimiento más preciso se debería haber tenido en cuenta la velocidad y la aceleración angulares. Por ejemplo, en el caso de una curva es clara la relación entre la variación de las coordenadas x e y . No obstante, para el problema planteado no se requiere tal precisión, ya que con el modelo planteado es suficiente para establecer las correspondencias entre diferentes imágenes. De esta forma se optó por desestimarlas para tener un modelo más simple. Además, en la mayoría de situaciones, los vehículos siguen trayectorias rectilíneas, con lo que ambas componentes, velocidad y aceleración angulares, serían cero, y por lo tanto, los sistemas para x e y serían independientes..

De acuerdo con las ecuaciones (4.7-4.10), podemos separar este sistema en dos subsistemas independientes, cada uno para una coordenada diferente (x , y). Así, construimos dos filtros de Kalman con diferentes vectores de estado (\mathbf{x} e \mathbf{y}). Estos vectores se definen como:

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ v_{xk} \\ a_{xk} \\ c_{xk} \end{bmatrix} \quad \mathbf{y}_k = \begin{bmatrix} y_k \\ v_{yk} \\ a_{yk} \\ c_{yk} \end{bmatrix} \quad (4.12)$$

Dado que el resto de ecuaciones son idénticas para ambos filtros, sólo describiremos el relacionado con la coordenada x . El proceso de estimación se controla mediante las ecuaciones de estado y medición. La ecuación de estado define la transición desde el estado \mathbf{x}_k al \mathbf{x}_{k+1} :

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{r}^p_k \quad (4.13)$$

La ecuación de medición relaciona las observaciones con el estado del proceso:

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{r}^m_k \quad (4.14)$$

En las Ecuaciones (4.13) y (4.14), las variables \mathbf{r}^p_k y \mathbf{r}^m_k representan los ruidos del proceso y medición. Se asume que ambos son independientes con distribuciones de probabilidad normal de media 0 y matrices de covarianza \mathbf{Q} y \mathbf{R} .

$$P(\mathbf{r}^p) \longrightarrow N(0, \mathbf{Q})$$

$$P(\mathbf{r}^m) \longrightarrow N(0, \mathbf{R})$$

La matriz \mathbf{A}_k ($n \times n$) relaciona el estado en el instante k con el estado en el instante $k + 1$. La matriz \mathbf{H}_k ($m \times n$) en la Ecuación (4.14) relaciona el estado en el instante k con la medición en el mismo instante. De acuerdo con las Ecuaciones (4.7-4.11), \mathbf{A}_k y \mathbf{H}_k se definen como:

$$\mathbf{A}_k = \begin{bmatrix} 1 & 1 & 0.5 & 1/6 \\ 0 & 1 & 1 & 0.5 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.15)$$

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (4.16)$$

La introducción de errores graves en una medición puede perturbar las estimaciones producidas por el filtro de Kalman en las subsiguientes iteraciones. Por lo tanto, es importante evitar dichos errores o, al menos, reducir su influencia. En cada iteración, el filtro de Kalman proporciona predicciones de los vectores de estado que son generadas antes de la introducción de las mediciones. Nosotros utilizamos estas predicciones para detectar graves errores en las mediciones de la posición y la velocidad. Así, las mediciones x_k , y_k , v_{xk} y v_{yk} son substituidas por \tilde{x}_k , \tilde{y}_k , \tilde{v}_{xk} y \tilde{v}_{yk} mediante la función ρ :

$$\tilde{u} = \rho(u, \hat{u}, \sigma_u) = \begin{cases} \hat{u} - 2\sigma_u & \text{Si } u < \hat{u} - 2\sigma_u \\ u & \text{Si } \hat{u} - 2\sigma_u \leq u \leq \hat{u} + 2\sigma_u \\ \hat{u} + 2\sigma_u & \text{Si } u > \hat{u} + 2\sigma_u \end{cases}$$

$$\tilde{\mathbf{u}} = \{\tilde{x}_k, \tilde{y}_k, \tilde{v}_{kx}, \tilde{v}_{ky}\}$$

$$\tilde{x}_k = \rho(x_k, \hat{x}_k, \sigma_{x_k}), \quad \tilde{y}_k = \rho(y_k, \hat{y}_k, \sigma_{y_k}), \quad \tilde{v}_{kx} = \rho(v_{kx}, \hat{v}_{kx}, \sigma_{v_{kx}}), \quad \tilde{v}_{ky} = \rho(v_{ky}, \hat{v}_{ky}, \sigma_{v_{ky}})$$

donde \hat{x}_k , \hat{v}_{xk} , \hat{y}_k y \hat{v}_{yk} son las predicciones de cada medición, y σ_{x_k} , $\sigma_{v_{xk}}$, σ_{y_k} y $\sigma_{v_{yk}}$ son las respectivas desviaciones típicas.

4.4 Memoria temporal

La memoria temporal se implementa en el sistema como una lista a la que denominamos *lista maestra*. Su utilidad es la de almacenar información acerca de las proyecciones de los objetos de la escena en la imagen. Principalmente, se guardan datos relativos a la forma y desplazamiento del objeto. Con ellos se lleva a cabo la integración temporal morfológica que proporciona la segmentación final de cada imagen.

En la lista maestra se insertan aquellas regiones de las segmentaciones iniciales que no han obtenido correspondencia con ninguna región de la lista. Se considera que la aparición de una nueva región puede deberse a dos motivos: un nuevo objeto que ha entrado en la escena, o un error de segmentación (básicamente, un objeto¹ que ha quedado dividido en más de una región). El proceso de correspondencia permite diferenciar ambos casos. Una vez insertadas en la lista, sólo las regiones que son

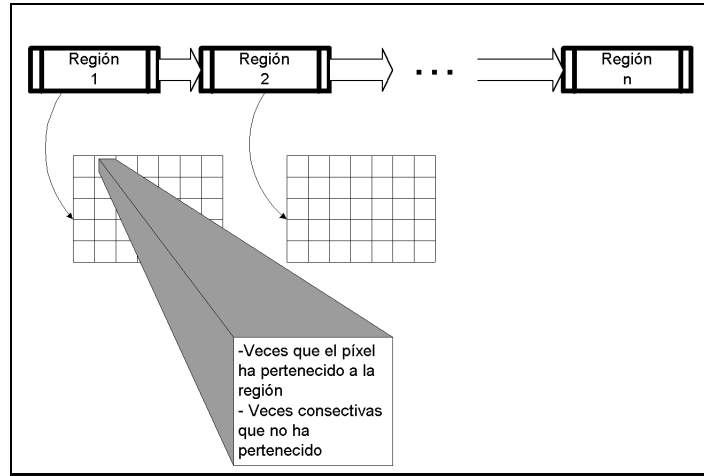


Figura 4.5: Lista maestra de regiones.

objetos completos obtendrán correspondencias en la mayoría de imágenes, mientras que las erróneas tenderán a lo contrario.

Cada vez que se inserta una región R^m en la lista maestra se crea para ella una matriz M^m . Como ya se comentó, por motivos de simplicidad asumiremos que su tamaño es similar al de la imagen. Sin embargo, en el sistema real, y para optimizar la ocupación de memoria, usamos el mínimo rectángulo que engloba la región, más una pequeña franja de seguridad, por si el número de píxeles de la región aumenta. En caso de que en algún momento la región exceda la matriz, se vuelve a crear una nueva siguiendo las mismas directrices y transfiriendo la información entre ellas.

La Figura 4.5 muestra la lista maestra con las matrices que se crean para cada región insertada en ella. Cada celda M_{ij}^m de la matriz M^m contiene dos campos ($M_{ij}^m.v$ y $M_{ij}^m.o$). El campo $M_{ij}^m.v$ sirve para contar el número de segmentaciones en las que el píxel (i, j) ha pertenecido a la región, mientras que el campo $M_{ij}^m.o$ guarda el número *consecutivo* de segmentaciones en las que el píxel no ha pertenecido a la región. La utilidad del primer campo es estimar la probabilidad de pertenencia de un píxel a las distintas regiones. Cuando se genera la segmentación final de la imagen, cada píxel será asignado a la región en la que las segmentaciones iniciales más veces lo hayan colocado. Por medio del valor $M_{ij}^m.v$ podemos expresar la probabilidad de que un píxel (i, j) pertenezca a una región R^m de la siguiente forma:

$$P(R^m|(i, j)) = \frac{M_{ij}^m.v}{\sum_{l \in LM} M_{ij}^l.v} \quad (4.17)$$

donde LM es la lista maestra.

Por otro lado, el campo $M_{ij}^m.o$ permite detectar píxeles que han pertenecido a la región durante varias segmentaciones iniciales, pero que han dejado de pertenecer a

¹En esta definición también se considera al fondo estático como un objeto

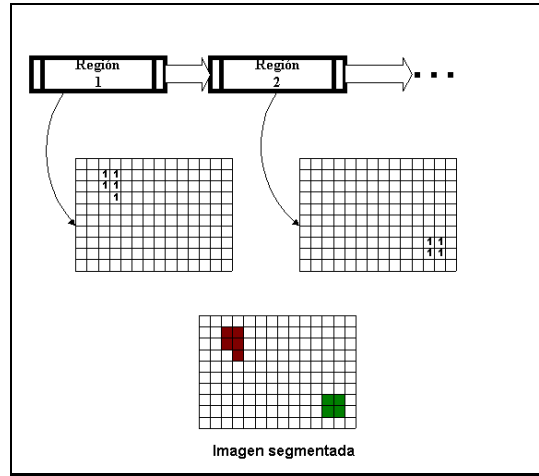


Figura 4.6: Inicialización de la lista maestra de regiones.

ella. Un caso típico de estos píxeles son los del contorno de una región correspondiente a la proyección de un objeto que se va alejando de la cámara. En ese caso, la región va reduciendo su área, con lo que dichos píxeles deben eliminarse de la región aunque hayan pertenecido a ella durante varias imágenes. Otro caso típico es el de dos objetos alejados que se aproximan juntos. Dada la lejanía, el sistema los empieza a segmentar como una única región, sin embargo, conforme se van acercando, empiezan a ser detectados como dos regiones independientes. Esto se traduce en la aparición de una nueva región, cuyos píxeles deben ser sustraídos de la más antigua. En ese caso, cada región, la nueva y la antigua, serán asignadas a cada uno de los dos objetos, probablemente la más vieja será asignada al objeto con una proyección mayor, debido a que mantendrá una mayor similitud.

Cuando se inserta en la lista maestra una nueva región R^m , todos los valores $M_{ij}^m.o$ toman el valor cero, mientras que los valores $M_{ij}^m.v$ son puestos a uno o cero según su pertenencia a la región.

$$M_{ij}^m.v = \begin{cases} 1 & \text{Si } (i, j) \in R^m \\ 0 & \text{Para el resto de píxeles} \end{cases}$$

donde (i, j) es un píxel de la imagen que ha sido asignado en la segmentación inicial a alguna región R^x .

La Figura 4.6 muestra un ejemplo de inicialización de dos regiones que han sido insertadas en la lista maestra. La proyección del objeto de la izquierda se inserta como región R^1 y la de la derecha como R^2 . En las matrices de ambas regiones, los píxeles que pertenecen a la región son inicializados con valor $M_{ij}^x.v = 1$ y $M_{ij}^x.o = 0$, y aquellos que no pertenecen se inicializan con valores $M_{ij}^x.v$ y $M_{ij}^x.o$ iguales a cero.

La actualización de los valores de esta matriz se realiza tras establecer las correspondencias entre las regiones de la segmentación inicial actual y las de la lista

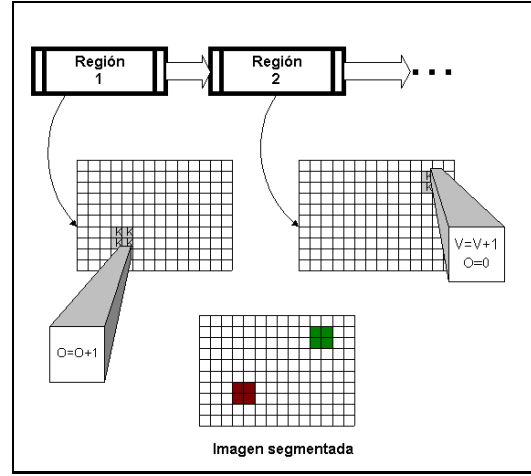


Figura 4.7: Actualización de la lista maestra de regiones.

maestra. Como hemos dicho, las regiones que no obtienen correspondencia son insertadas, mientras que para las que sí que la obtienen se realiza una modificación de los valores $R_{ij}^x.v$ y $R_{ij}^x.o$. Dada una correspondencia entre una región R^m de la lista maestra y una región R^k de la segmentación inicial actual, la actualización de valores sigue las siguientes reglas:

- Si $(i, j) \in R^k \implies M_{ij}^m.v = M_{ij}^m.v + 1$ y $M_{ij}^m.o = 0$
- Si $(i, j) \notin R^k$ y $M_{ij}^m.v > 0 \implies M_{ij}^m.o = M_{ij}^m.o + 1$
- Si $M_{ij}^m.o = O_{max} \implies M_{ij}^m.v = 0$
- Si $M_{ij}^m.v = 0 \implies M_{ij}^m.o = 0$

Según la primera regla, cuando un píxel (i, j) pertenece a la región, el valor $M_{ij}^m.v$ se incrementa y el valor $M_{ij}^m.o$ se pone a cero. Según la segunda, cuando un píxel (i, j) que ha pertenecido a la región se pierde en la segmentación inicial actual, se incrementa el valor $M_{ij}^m.o$. La tercera regla indica que cuando para un píxel de una región R^m se alcanza un valor máximo O_{max} , se elimina el píxel de la región. Esto se hace asignando un cero al campo $M_{ij}^m.v$ de ese píxel. Por último, la cuarta regla dice que el valor $M_{ij}^m.o$ de cualquier píxel que no pertenezca a la región debe permanecer siempre a cero.

Hemos fijado de manera experimental el valor de O_{max} como 3. Se considera que la desaparición de un píxel durante tres segmentaciones consecutivas viene a indicar que ese píxel ha dejado de pertenecer al objeto donde estaba asignado. Por otra parte, valores mucho más grandes impedirían el necesario dinamismo en la variación de la forma. Por ejemplo, evitaría que la segmentación final reflejase los efectos de escalado que se producen cuando los vehículos se acercan o alejan de la cámara.

La Figura 4.7 muestra un ejemplo de actualización de las regiones que se insertaban en la Figura 4.6. La matriz de la parte inferior de la imagen representa la segmentación

inicial de la imagen actual. En ella hay dos regiones, la de la derecha ha sido hecha corresponder con la región R^2 y la de la izquierda con R^1 . En el caso de la región R^2 , la forma segmentada se mantiene idéntica, por lo que para todos los píxeles que tenían un valor $M_{ij}^m.v$ distinto de cero (que habían pertenecido a la región en alguna segmentación anterior) se incrementa $M_{ij}^m.v$ y se mantiene $M_{ij}^m.o$ igual a cero. En el caso de R^1 , en la segmentación actual se ha perdido el píxel inferior. En este píxel se incrementa el campo $M_{ij}^m.o$, mientras que para el resto de píxeles que pertenecen a la región roja se incrementa $M_{ij}^m.v$ y se pone a cero $M_{ij}^m.o$.

Los valores de la lista maestra se utilizarán para la generación de la segmentación final de la imagen. Tras obtener esta segmentación, y como preparación para el procesamiento de la siguiente imagen, la información almacenada en cada matriz se desplaza con el movimiento estimado para cada región. De esta forma se mueven los valores correspondientes de cada píxel a la posición donde se espera encontrarlos en la siguiente segmentación.

4.5 Recuperando regiones perdidas

Algunas veces, una región de la lista maestra que ha aparecido regularmente durante varias segmentaciones iniciales desaparece en una imagen, sin que ello se deba a que ha abandonado la escena. Nuestro método cuenta el número de veces que cada región de la lista ha encontrado correspondencia en una segmentación inicial. Cuando el contador de una región alcanza un valor P_{max} , ésta es declarada *región permanente*. Ello significa que no puede desaparecer excepto si sobrepasa los límites de la imagen (abandona la escena). La selección de P_{max} no es crítica, puesto que un vehículo suele aparecer en al menos 25 o 30 imágenes. Por ello basta con elegir un valor que razonablemente nos de a entender que la persistencia de una región corresponde a la presencia de un vehículo. En todas las secuencias que hemos procesado hemos fijado $P_{max} = 10$, obteniendo resultados satisfactorios.

Una *región permanente* es aquella que por el número de veces que ha aparecido, nos ofrece la total seguridad de que corresponde a un objeto del mundo real. Como contrapartida existen otro tipo de regiones que aparecen de manera circunstancial, debido a errores de segmentación.

La desaparición de una región *permanente* puede deberse a errores de segmentación o a que el desplazamiento del objeto no permite diferenciar el movimiento de su proyección del de las regiones circundantes. Por ejemplo, en escenas de tráfico, esto último puede suceder cuando un grupo de vehículos se aleja de la cámara, o también cuando un coche se detiene. En el primer caso, las segmentaciones iniciales comenzarán gradualmente a agrupar varios de estos vehículos en una misma región, para finalmente unirlos todos en una sola. En el segundo caso, cuando la velocidad del coche sea cero o casi cero, su proyección será irremisiblemente fusionada a la región fondo.

En todos los anteriores casos, la información guardada para cada región en la lista maestra puede ayudar a recuperar su forma. La idea básica es, cuando se detecta la desaparición de una *región permanente*, buscar qué región es la que le ha arrebatado

sus píxeles. Una vez hallada ésta, se trata de ver qué píxeles deben corresponder a cada una. Para ello se analizan los valores $M_{ij}^m.v$ de las matrices de ambas regiones y se asigna cada píxel a la región con mayor valor. Esto se basa en que, como hemos dicho, estos valores aportan el historial de cada píxel. Así, se trata de asignar cada píxel a la región donde más veces ha pertenecido.

La recuperación de regiones perdidas es un paso previo a la actualización de la lista maestra. Por ello, una vez recuperados los píxeles no se actualiza directamente la región sino que se corrige la segmentación inicial generando una región que supuestamente debería haber aparecido en ella.

4.5.1 Planteamiento del problema

Con todo lo anteriormente expuesto, el planteamiento del problema sería: dada una *región permanente* R^p que no ha encontrado una región correspondiente, generar en la segmentación inicial una región R^l , que se hará corresponder con R^p , tomando los píxeles de una región R^k de la segmentación inicial con región correspondiente R^m en la lista maestra. El algoritmo propuesto consta de dos pasos:

- Encontrar la región R^k como aquella con mayor probabilidad de contener píxeles de la región R^p .
- Generar la región R^l tomando píxeles de la región R^k . Para ello habrá que estudiar para cada píxel si tiene mayor probabilidad de pertenecer a la región R^p o a R^m , la región correspondiente a R^k .

4.5.2 Búsqueda de la región R^k

Para encontrar la región R^k analizamos los valores $M_{ij}^p.v$ de la región R^p . Se crea un acumulador A_s por cada región R^s de la segmentación inicial al que se le van sumando los valores $M_{ij}^p.v$ de los píxeles (i, j) que pertenecen a la región R^s .

$$\forall (i, j) \in \text{Imagen} / (i, j) \in R^s, A_s = A_s + M_{ij}^p.v$$

Una vez procesados todos los píxeles de la imagen, R^k corresponderá a la región con un máximo A_s .

$$R^k/A_k = \max_{\forall R^s \in SI} \{A_s\}, \quad (4.18)$$

donde SI es la segmentación inicial de la imagen actual.

La región R^k será aquella que tiene la mayor cantidad de píxeles con alta probabilidad de pertenecer a la región perdida R^p .

4.5.3 Generación de la región R^l

La región R^k tiene como región correspondiente en la lista maestra a R^m . Ahora tratamos de ver qué píxeles de R^k corresponden a R^m y cuales a R^p . Para ello analizaremos los valores $M_{ij}^p.v$ y $M_{ij}^m.v$ y asignaremos cada píxel a la región con un mayor valor para ese píxel. Los píxeles que tengan mayor probabilidad de corresponder a R^p se los asignaremos a la nueva región R^l , estableciendo una correspondencia entre ellas.

Inicialmente, $R_l = \emptyset$. Los píxeles son asignados a R_l siguiendo la siguiente regla:

$$\forall (i, j) \in R_n, \text{ Si } M_{ij}^k.v < M_{ij}^p.v \text{ entonces } R_l \Leftarrow R_l + \{(i, j)\} \text{ y } R_n \Leftarrow R_n - \{(i, j)\}$$

4.6 Generación de una segmentación final por medio de la integración temporal morfológica

La generación de la segmentación final de cada imagen se hace en base a la información almacenada en la lista maestra. El procedimiento es bastante sencillo: se asigna cada píxel de la imagen a la región donde el píxel tiene una mayor probabilidad de pertenecer de acuerdo con la probabilidad expresada en la ecuación (4.17).

$$\forall (i, j) \in \text{Imagen}, (i, j) \in R^x \longleftrightarrow P(R^x|(i, j)) = \max_{\forall R^y \in LM} \{P(R^y|(i, j))\},$$

donde LM es la lista maestra y la probabilidad se expresa de acuerdo con la expresión que aparece en la ecuación (4.17) y que a continuación reproducimos:

$$P(R^y|(i, j)) = \max_{\forall R^y \in LM} \left\{ \frac{M_{ij}^y.v}{\sum_{l \in LM} M_{ij}^l.v} \right\}$$

4.7 Resumen del algoritmo

Para mejor comprensión del algoritmo propuesto, a continuación resumiremos los diferentes pasos del mismo. Para ello seguiremos el algoritmo de la Figura 4.1 (ver página 96).

1. *Inicialización.* Se toma la segmentación inicial de la primera imagen de la secuencia y se construye la *lista maestra* de regiones. Este paso sólo se realiza una vez, para la primera imagen, a diferencia del resto de pasos que se repiten para cada nueva imagen.
2. Para cada imagen se toma su segmentación inicial que es el resultado del proceso explicado en el Capítulo 3.

3. *Correspondencia.* Se establece la correspondencia entre las regiones de la lista maestra y de la nueva segmentación inicial.
4. *Recuperación de regiones perdidas.* Se buscan regiones declaradas como *permanentes* y que no aparecen en la segmentación inicial. Su recuperación consiste en reintegrarlas en la segmentación inicial tomando píxeles de otras regiones.
5. *Actualización de la lista maestra.* Se integra en la memoria temporal la información aportada por la segmentación actual.
6. *Segmentación final.* A partir de la lista maestra se genera una nueva segmentación (segmentación final de la imagen actual).
7. *Eliminación de regiones.* Las regiones que no se han usado durante varias iteraciones se borran de la lista maestra. Éstas corresponden a vehículos que han abandonado la escena o a regiones fruto de errores de la segmentación inicial.
8. *Mover regiones.* Se aplican los parámetros de movimiento estimado para cada región de la lista maestra a su matriz. De esta forma la información de la región se traslada a la posición donde se espera encontrarla en la siguiente imagen.

4.8 Resultados

El método presentado en este capítulo ha sido probado sobre varias secuencias de tráfico, en esta sección mostramos los resultados de las secuencias. La Figura 4.8 muestra una secuencia de 15 imágenes donde varios vehículos se desplazan por una carretera. Las subfiguras 4.8.a y 4.8.b contienen, respectivamente, la primera y última imágenes de la secuencia. La subfigura 4.8.c muestra la segmentación de la primera imagen y la subfigura 4.8.d presenta el seguimiento de las diferentes regiones.

En la subfigura 4.8.d se observan las trayectorias de cada uno de los vehículos a lo largo de la secuencia. Éstas vienen representadas por la evolución de los centroides de las regiones. Hemos presentado las posiciones estimadas por el filtro de Kalman en cinco imágenes de la secuencia: imágenes 1, 4, 7, 10 y 13. Las líneas conectan los centros que corresponden a proyecciones del mismo vehículo en diferentes instantes, manteniendo el orden temporal. Dado que la secuencia representa menos de dos segundos cabría esperar que las trayectorias de los vehículos tuvieran una forma muy similar a una línea recta. Esto se obtiene para casi todas las regiones, sin embargo, se pueden apreciar pequeños errores en algunas de las trayectorias. Cuando ello sucede, los errores suelen presentarse en alguno de los primeros dos puntos (imágenes 1 y 4). Esto se debe a que el filtro de Kalman necesita algunas iteraciones iniciales para empezar a converger. Durante estas iteraciones, los errores producidos en el proceso de medición (estimación y segmentación de movimiento) influyen seriamente en las estimaciones generadas por el filtro. Una vez superadas, la influencia de los errores comienza a decrecer, y las estimaciones se acercan cada vez más a la trayectoria correcta.

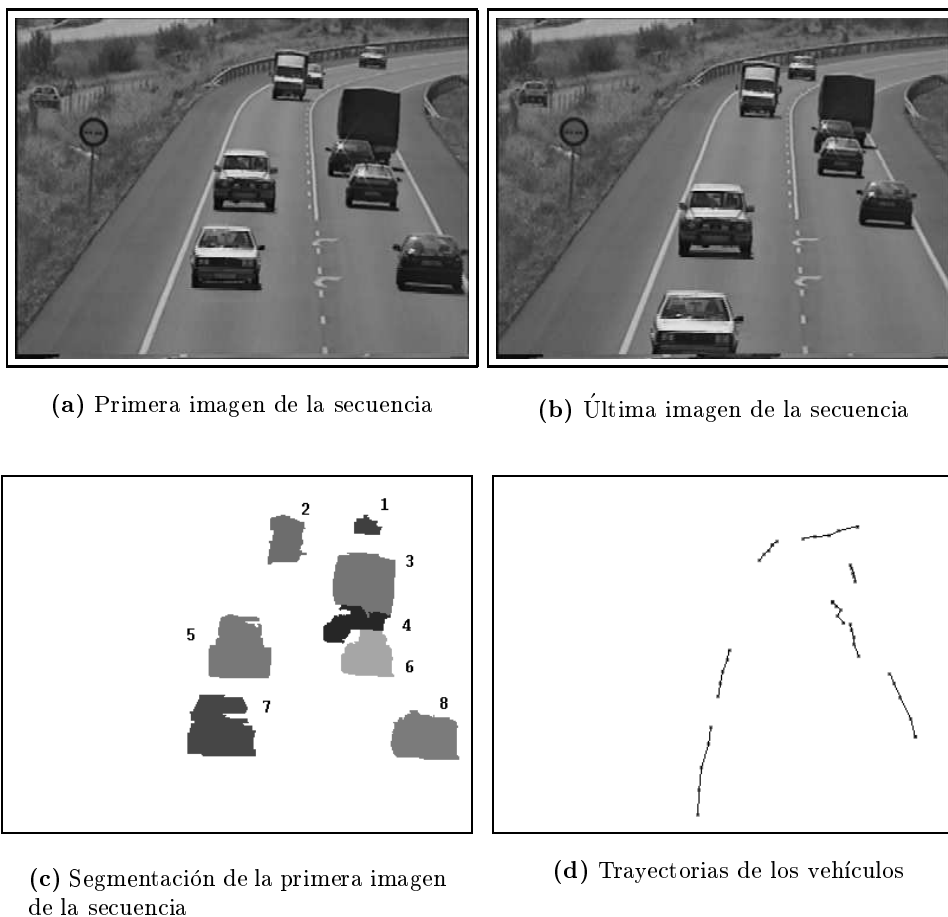


Figura 4.8: Seguimiento de vehículos en una escena de tráfico

Las Figuras 4.9 y 4.10 muestran las segmentaciones de dos secuencias de imágenes, correspondiendo la primera a la misma secuencia que la Figura 4.8. En ambas figuras, la primera columna presenta varias imágenes de la secuencia, la segunda columna muestra la segmentación inicial de cada imagen, y la tercera columna es la segmentación final conseguida por medio de la integración temporal.

Las imágenes de la Figura 4.9 han sido escogidas por ser las que producen las peores segmentaciones iniciales de la secuencia. La subfigura 4.9.a corresponde a la quinta imagen de la secuencia, siendo la subfigura 4.9.b su segmentación inicial. En ella se observa que el camión más alejado ha sido dividido en dos regiones. Sin embargo, la integración de la información de las segmentaciones previas, donde el camión estaba formado por una única región, consigue que la segmentación final una ambas regiones, subfigura 4.9.c.

La subfigura 4.9.d muestra la décima imagen de la secuencia. La segmentación

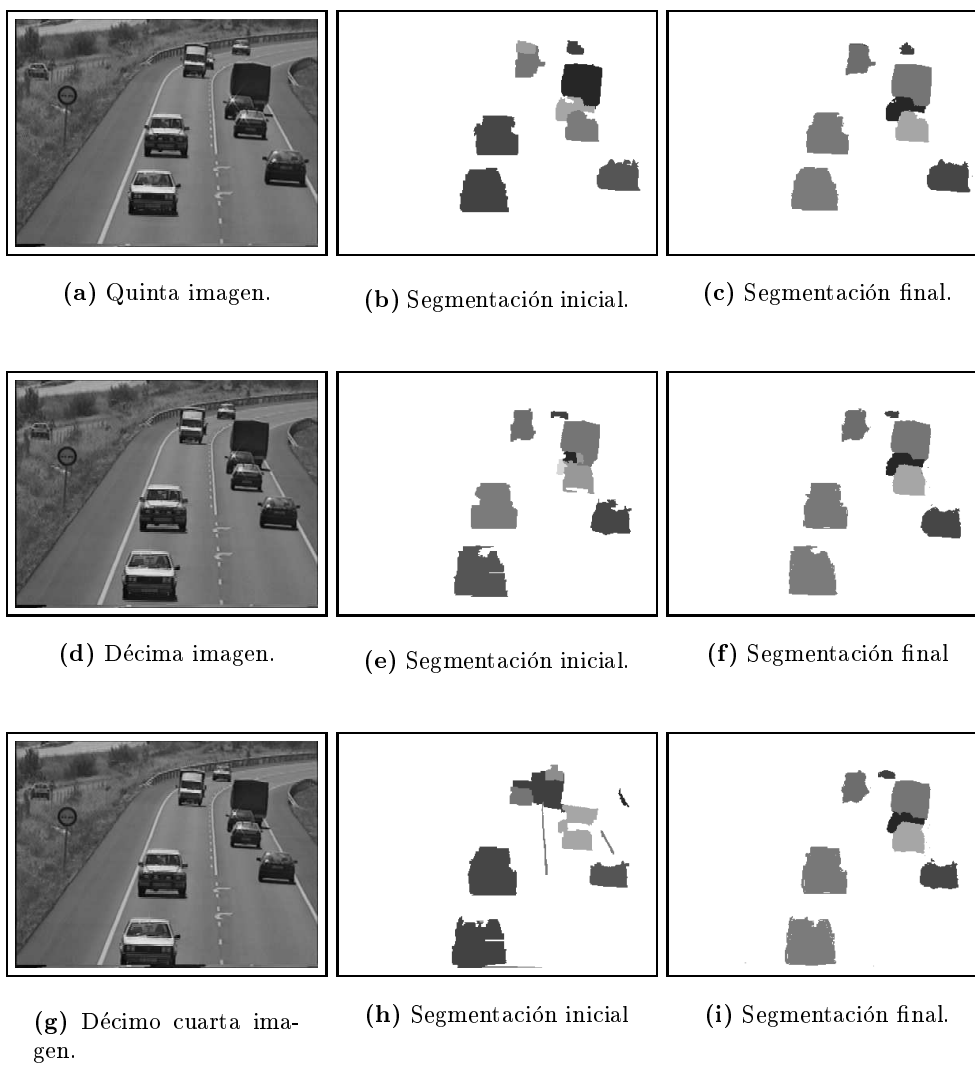


Figura 4.9: Secuencia de vehículos en una carretera. Primera columna: imagen original. Segunda columna: segmentación inicial. Tercera columna: segmentación final de la imagen

inicial, subfigura 4.9.e, presenta un problema similar al de la quinta imagen, aunque en este caso el vehículo subdividido es el que se encuentra en medio del grupo de tres que circula por el carril de la derecha. De nuevo se observa en la subfigura 4.9.f que la segmentación final es correcta, obteniendo una proyección del coche formada por una única región.

La segmentación inicial de la imagen número catorce, subfigura 4.9.h, es, con diferencia, la peor de la secuencia. Algunas regiones estáticas se han detectado como móviles, parte del camión más alejado se ha perdido, se han unido el primer y segundo coche del grupo de tres, y el camión de este grupo ha desaparecido. A pesar de estos errores, el método corrige la segmentación: desaparecen las regiones estáticas y los vehículos recobran su forma. El último camión del grupo de tres es recuperado gracias a que su región fue declarada como *permanente*. Así, sus píxeles se extraen de los de la región fondo.

En la Figura 4.10 se muestran cuatro imágenes de una misma secuencia. La secuencia registra el tráfico de una calle y consta de 858 imágenes, de las que mostramos la segmentación de cuatro de ellas. El tamaño de cada imagen es de 192×144 píxeles. La subfigura 4.10.a muestra la imagen número 488. En ella dos vehículos circulan en diferente dirección. Las subfiguras 4.10.d y 4.10.g corresponden a las imágenes número 555 y 644. En ellas circulan dos vehículos con intensidades de gris muy parecidas a la de la calzada. Ello provoca que en determinadas segmentaciones iniciales los vehículos sólo sean parcialmente segmentados, o incluso divididos en varias regiones. Sin embargo, la segmentación final corrige estas situaciones gracias a la acumulación de información obtenida en las imágenes previas. La subfigura 4.10.j presenta otra imagen difícil. En ella una moto cuya proyección ocupa alrededor de 50 píxeles circula alejándose de la cámara. Además, y aunque la intensidad de la moto sí que ofrece suficiente contraste con la calzada, presenta el inconveniente de su reducida velocidad. Ambos problemas, tamaño y velocidad, impiden su correcta segmentación inicial en algunas imágenes, aunque de nuevo la segmentación final es correcta.

La Figura 4.11 muestra el cruce de una calle con una avenida. Los vehículos entran en la imagen desde los bordes izquierdo e inferior de la imagen, y la abandonan por los bordes superior, inferior y derecho. En la subfigura 4.11.b aparece la segmentación y trayectoria de un coche que toma una curva. Se puede apreciar que el sistema es capaz de seguir su trayectoria, a la vez que la segmentación va actualizando la forma del vehículo. A medida que el coche gira, la parte que se observa del mismo va cambiando. La segmentación resultante va adaptándose gradualmente, eliminando los píxeles que han dejado de pertenecer a la región y añadiendo los que son nuevos. La subfigura 4.11.c corresponde a otra escena de la misma secuencia, donde cinco coches se hallaban detenidos en un semáforo y ahora cuatro de ellos circulan a lo largo de la avenida y uno gira por una calle a la izquierda. Debido a que inicialmente se hallaban todos juntos y a que casualmente varios de ellos presentan colores muy similares (en este caso nos referimos al color real de los coches, y no a la intensidad de gris reflejada en las imágenes de la secuencia), las primeras segmentaciones de estos coches presentan algunos errores iniciales que, como puede observarse en la subfigura 4.11.d, afectan al tramo inicial de las trayectorias estimadas para algunos de ellos. No obstante, tras

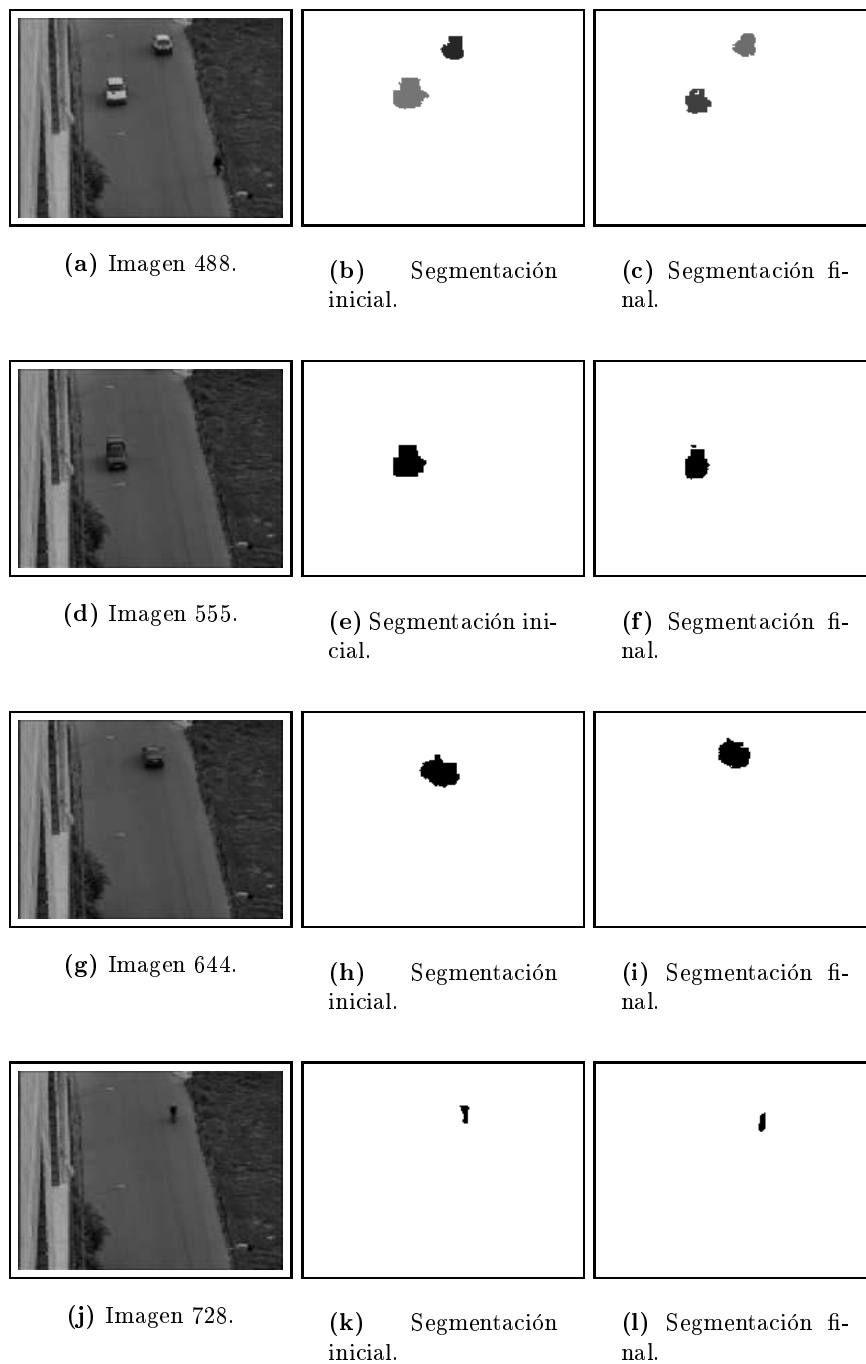


Figura 4.10: Varias segmentaciones de una secuencia de tráfico en ciudad. Primera columna: imagen original. Segunda columna: segmentación inicial. Tercera columna: segmentación final de la imagen.

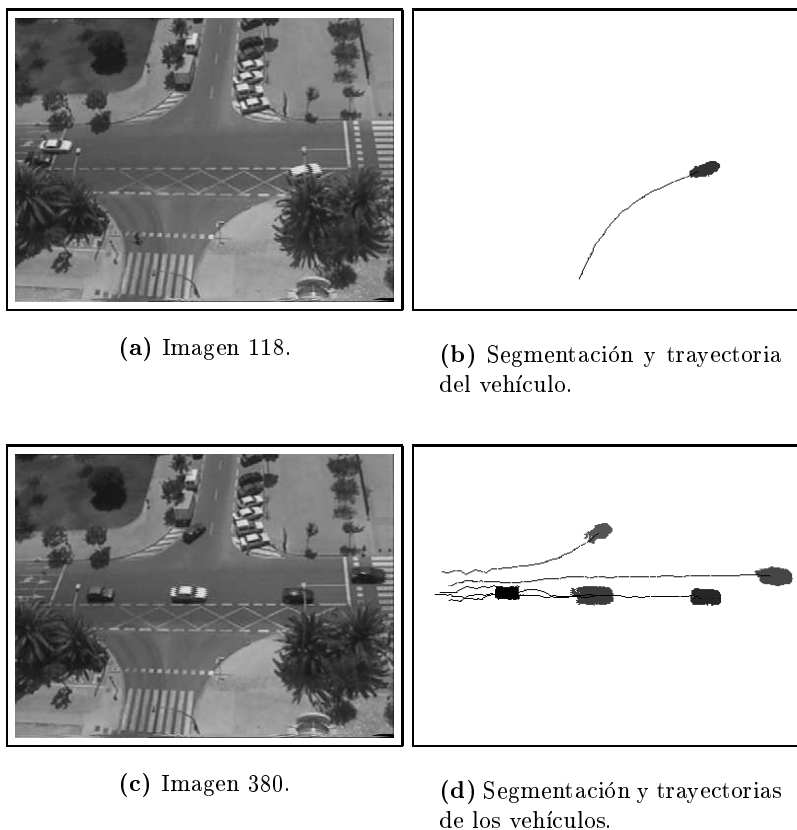


Figura 4.11: Secuencia de vehículos en un cruce de una avenida Primera columna: imagen original. Segunda columna: segmentación final de la imagen

unas pocas imágenes las trayectorias comienzan a ser más correctas gracias, tanto a que las segmentaciones comienzan a ser mejores debido a que el movimiento se hace más diferenciado, como a que se hace notar el efecto estabilizador del filtro de Kalman.

4.9 Conclusiones

En este capítulo se ha presentado un método novedoso de segmentación por medio de la integración temporal. El método propuesto integra las segmentaciones que proporciona el módulo de análisis de movimiento para generar segmentaciones mejoradas. La mayoría de técnicas existentes emplean la segmentación para aportar información al módulo de seguimiento. En nuestro caso, además, el seguimiento ayuda a la segmentación relacionando la información morfológica obtenida en cada imagen.

Como se ha visto, la integración temporal morfológica aporta grandes beneficios pues permite fundamentar la segmentación no en unas pocas imágenes, sino en todas

las precedentes. En esencia, lo que aporta es una estabilización de la forma de los objetos, la cual minimiza la influencia de los errores puntuales que puedan surgir en el procesamiento de algunas imágenes: por ejemplo la aparición o desaparición total o parcial de regiones.

Capítulo 5

Un sistema de monitorización de tráfico en tiempo real

Contenido

5.1	Introducción. Monitorización de tráfico	118
5.2	Tareas de monitorización de tráfico	118
5.2.1	Medición del volumen del tráfico	119
5.2.2	Seguimiento de trayectorias	121
5.2.3	Detección de vehículos parados	122
5.2.4	Detección de eventos anómalos	124
5.3	Un sistema en tiempo real	124
5.3.1	Arquitectura del sistema	124
	Equipamiento	124
	Software del sistema	126
5.3.2	Segmentación virtual	128
5.4	Resultados	129
	Secuencia de una calle de doble dirección	130
	Secuencia de la rotonda	130
	Secuencia de la carretera	134
	Secuencia del semáforo	136
	Secuencia del cruce de calles con la avenida	136
	Secuencia de la autopista	140
	Velocidad de procesamiento	142
5.5	Conclusiones	142

5.1 Introducción. Monitorización de tráfico

La *monitorización de tráfico* es una labor de indudable interés en nuestros días. El gran caudal de tráfico de las calles y carreteras hacen necesaria la automatización de su control. Frente a los sistemas usados hasta ahora, como *detectores basados en microondas* y *detectores de bucle inductivo*(ver página 3), los sistemas basados en visión presentan varias ventajas:

- Menor coste económico.
- Instalación más fácil.
- Permiten la estimación de más parámetros.

Un sistema de monitorización de tráfico debe obtener mediciones acerca del tráfico que permitan dos tipos de actuaciones:

- Solventar situaciones anómalas, como pudieran ser atascos o accidentes.
- Ayudar en el diseño de la red vial aportando información sobre el volumen y tipo de tráfico de las vías monitorizadas.

El presente capítulo trata de mostrar cómo se llevan a cabo las tareas de monitorización en el sistema propuesto en esta Tesis, al mismo tiempo que se muestra la arquitectura de dicho sistema. La Sección 5.2 describe las mencionadas tareas y la Sección 5.3 explica cómo se han implementado para conseguir el funcionamiento del sistema en tiempo real. De ese modo, el Apartado 5.3.1 detalla el equipamiento mínimo requerido y los programas que se han escrito, así como los procesos a que han dado lugar los mismos, prestando especial interés a la información que fluye entre ellos. El Apartado 5.3.2 trata una técnica propuesta para el presente sistema la cual evita tener que procesar todas las imágenes de una secuencia para obtener su segmentación. Esta reducción del número de imágenes resulta de especial interés, puesto que, obviamente, contribuye a disminuir el coste computacional, pero tratando de no afectar a los resultados.

En este capítulo se prestará una importante atención a la sección dedicada a los resultados, Sección 5.4. En ella se utilizarán diversas secuencias para mostrar cómo funciona el sistema propuesto y cuáles son los resultados que aporta. Como es habitual, se concluirá con una sección donde se esbozarán las conclusiones del capítulo.

5.2 Tareas de monitorización de tráfico

El sistema de monitorización de tráfico propuesto en esta Tesis proporciona las siguientes tareas:

- Medición del volumen de tráfico.

- Seguimiento de trayectorias individuales de vehículos.
- Detección de vehículos parados.
- Detección de eventos anómalos.

Para el buen funcionamiento de las técnicas propuestas en este capítulo resulta necesario delimitar en la imagen los bordes de la carretera y de los carriles que hayan en ella. Esto puede realizarse de manera manual o automática:

- *Manualmente.* Se posiciona y enfoca la cámara lista para empezar a monitorizar el tráfico, si bien antes de comenzar el procesamiento se captura una imagen. Sobre esta imagen y mediante cualquier programa de dibujo, se marcan en distintas intensidades de gris los diferentes carriles, así como el resto de imagen que no es carretera. Esto último se marca con intensidad cero. La imagen resultante se pasa al programa de monitorización.
- *Automáticamente.* Se emplea un algoritmo como el propuesto en [Steward *et al.*, 1994] (la Figura 3.1 muestra un ejemplo del funcionamiento de este algoritmo). El algoritmo proporciona de manera automática la imagen requerida por el programa de monitorización.

Cada uno de estos métodos tiene sus ventajas e inconvenientes. El método manual tiene la ventaja de poder realizar un ajuste más exacto de los límites. Sin embargo, presenta el serio inconveniente de requerir un operador que lleve a cabo esta función.

El método automático presenta una seria desventaja que nos hizo desecharlo: las zonas de la carretera con escaso tráfico son declaradas como zonas no-carretera. Esto, por ejemplo, afecta a los arcones, que son zonas por las que raramente se circula, pero que es preciso controlar.

A continuación, en los diferentes apartados, se irán viendo las tareas de monitorización de tráfico que el sistema lleva a cabo. En cada apartado veremos en qué consiste la tarea correspondiente y cuál es el modo en que se efectúa. Para que el lector tenga una visión más clara de cómo se obtienen las informaciones resultantes de cada tarea, la Figura 5.1 muestra el flujo de informaciones que el sistema produce, y cómo éstas son combinadas para producir nuevos resultados.

5.2.1 Medición del volumen del tráfico

Tal y como se ha visto en el capítulo anterior, el sistema desarrollado permite conocer el número de vehículos que se encuentran en cada imagen. Además, el sistema establece correspondencias entre las diferentes apariciones de un mismo vehículo en diferentes imágenes, con lo que se puede obtener, también de forma directa, otra medición de interés para el control de tráfico: el número de vehículos que han circulado por la carretera entre las imágenes n y $n + k$ ¹.

¹Dado que se conoce el instante de adquisición de cada imagen, esto es equivalente a saber el número de vehículos entre dos instantes de tiempo.

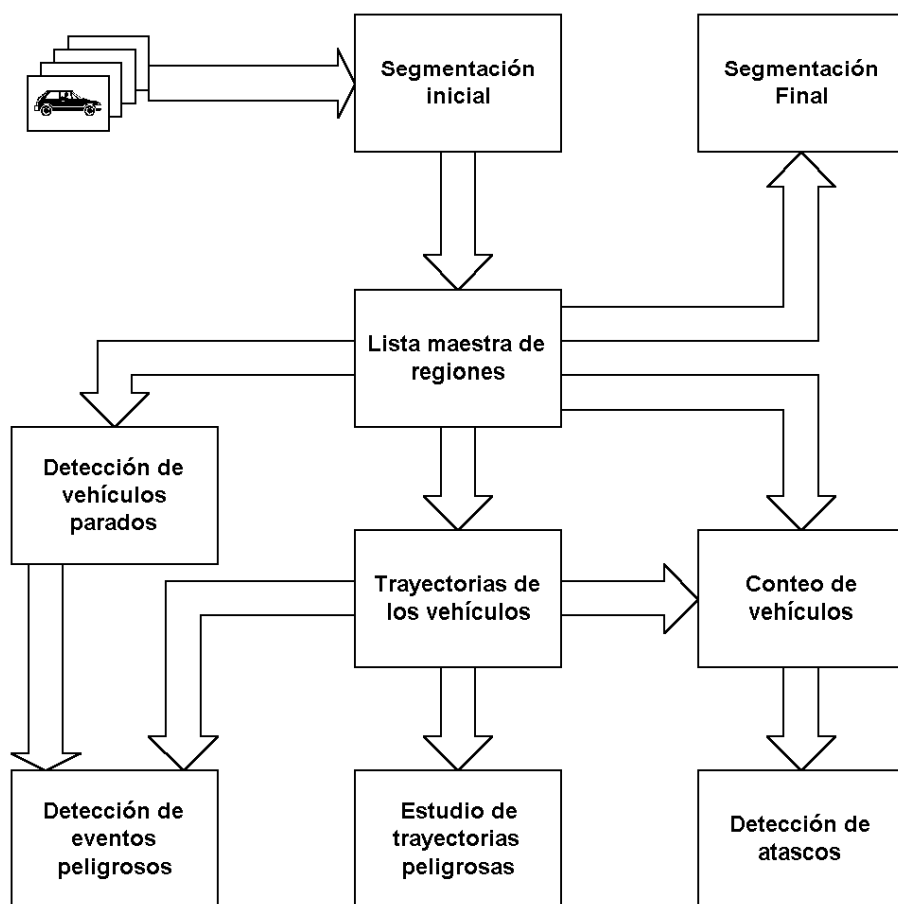


Figura 5.1: Flujo de informaciones del sistema. Las cajas representan informaciones que se obtienen en alguna fase del procesamiento, mientras que las flechas indican que una información (la apuntada) se genera a partir de la otra (la que se encuentra en el origen de la flecha).

Sabiendo los límites de los diferentes carriles y la posición de los vehículos en la imagen², podemos medir el número de vehículos que ha circulado por cada carril (por ejemplo, en una autopista). Y puesto que se conoce el sentido del tráfico en cada carril, se sabrá el número de vehículos que ha circulado en cada dirección.

La Figura 5.2 muestra dos casos donde resulta de interés poder desglosar el volumen de tráfico. En el caso de la autopista es interesante saber el volumen de cada carril, por ejemplo, en una bifurcación o en una salida de autopista pueden existir diferencias sensibles entre carriles. En el caso de la avenida además de conocer la intensidad de tráfico de cada carril, conviene saber qué direcciones toman los vehículos que abandonan la avenida y desde qué calles se suelen incorporar a la misma.

²Cada vehículo se asigna a aquel carril donde se encuentra su centroide.

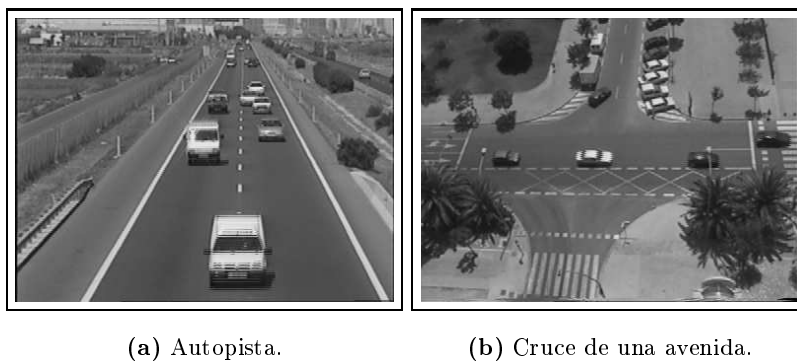


Figura 5.2: Dos casos típicos donde no solo interesa saber el volumen total del tráfico, sino también el de cada carril o calle.

Los atascos pueden detectarse calculando la proporción entre el número de vehículos presentes en una imagen y el número medio de vehículos que abandonan la escena (vehículos cuyo centroide sale de los bordes de la imagen). Un gran número de vehículos en la escena y una media muy baja indicarán una situación de atasco.

$$r = \frac{\text{Número de vehículos en la escena}}{\text{Número medio de salidas}} \quad (5.1)$$

5.2.2 Seguimiento de trayectorias

El seguimiento de trayectorias de vehículos en el plano de la imagen se consigue de manera directa a partir del proceso de seguimiento que se mostró en el Capítulo 4. Dicho proceso nos permite conocer la posición de los distintos vehículos en cada una de las imágenes. Esto se conseguía mediante un proceso de establecimiento de correspondencias entre las regiones, las que se obtienen en cada segmentación inicial y las que pertenecen a la *lista maestra*. Esto, además, se completaba con la incorporación de un filtro de Kalman que permitía estabilizar las trayectorias haciéndolas menos sensibles a posibles errores en la segmentación inicial.

El trazado de trayectorias de vehículos es de gran utilidad en el estudio de curvas de carretera. El análisis de las trayectorias de varios vehículos permite saber por qué parte de la curva (hacia el interior o exterior) suelen pasar los coches, así como dónde suelen frenar. Esto permite detectar anomalías en la construcción y diseño como, por ejemplo, un peralte erróneo.

En la Figura 5.3 tenemos un ejemplo del seguimiento de un vehículo a lo largo de una curva. Analizando varias de estas trayectorias se puede entender el comportamiento general de los coches en una curva como ésta.

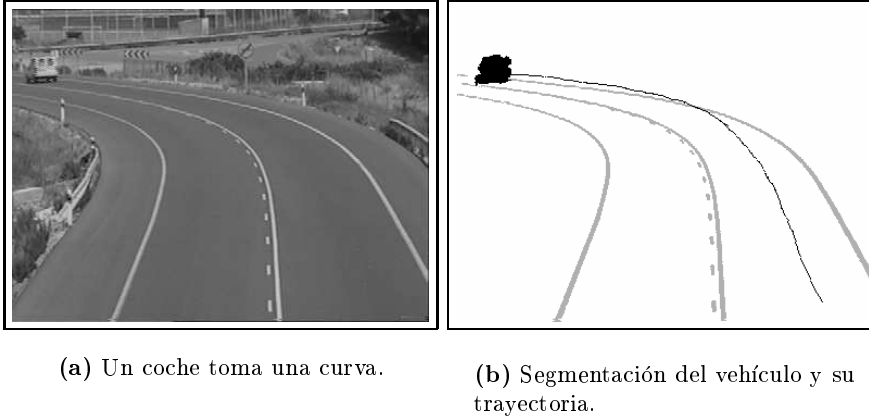


Figura 5.3: Trayectoria de un coche en una curva.

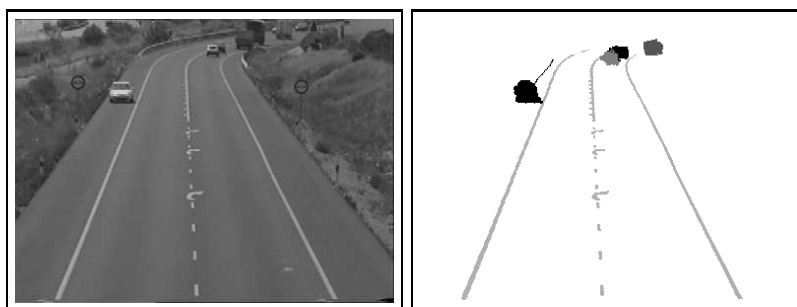
5.2.3 Detección de vehículos parados

Cuando una región vehículo tiene una velocidad muy cercana a cero, el módulo de estimación y segmentación de movimiento la une al fondo debido a la similitud de movimientos. No obstante, tal y como se vio en la Sección 4.5, a una región de la lista maestra que haya aparecido repetidamente en diversas segmentaciones se la declara *región permanente*. Ello significa que esa región sólo puede desaparecer al abandonar la imagen por alguna de sus zonas de salida.

También se vio en la misma sección que cuando en el módulo de integración temporal no se encuentra ninguna *región permanente* en la segmentación inicial, entonces es el sistema quien la añade a la segmentación final (recuperación de regiones perdidas, Sección 4.5). De esta forma, aunque un vehículo se detenga, su forma en la segmentación nunca se pierde.

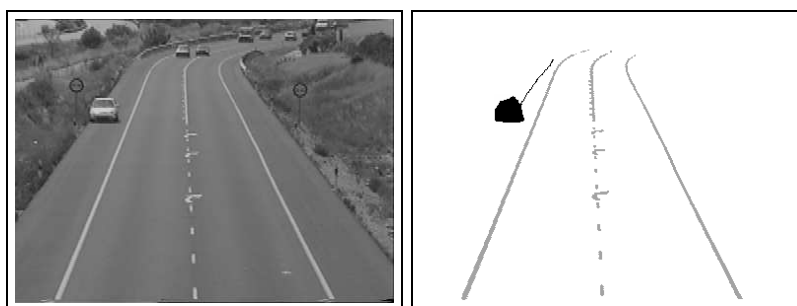
Cuando una región reduce su velocidad a valores muy próximos a cero, se genera un aviso que indica que un vehículo se ha detenido. Esta información es de gran interés pues puede significar una situación potencial de peligro para el resto de vehículos, o quizás la avería de un coche al que se debería atender. Del mismo modo, cuando se detecte que el vehículo reemprende la marcha se generará otro aviso que informe de ello.

La Figura 5.4 muestra un vehículo que se detiene en el arcén de una carretera. En la subfigura 5.4.a el coche se aparta al arcén mientras reduce su velocidad. En la subfigura 5.4.c el vehículo ya se ha detenido y es cuando el sistema avisa de esta situación de posible riesgo. Puede verse en la subfigura 5.4.d que a pesar de estar detenido, el sistema sigue segmentando el vehículo. En la subfigura 5.4.e el vehículo ha reemprendido la marcha y el sistema reemprende su seguimiento. Puede verse, por la trayectoria del automóvil en la subfigura 5.4.f, que en este caso no se produce una reincorporación gradual a la calzada, sino que rápidamente gira las ruedas y sigue su



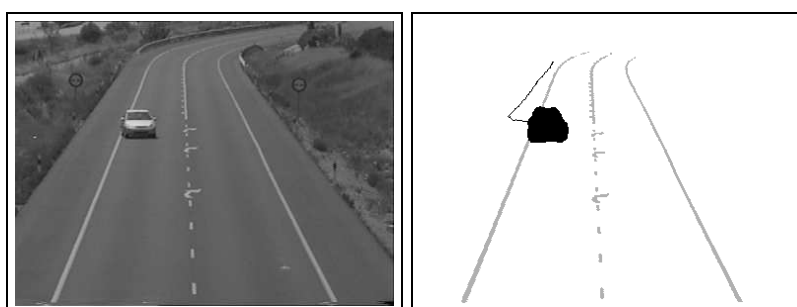
(a) Un coche comienza a detenerse en el arcén de una carretera.

(b) Segmentación y trayectoria del vehículo.



(c) El coche se ha detenido completamente.

(d) Segmentación y trayectoria del vehículo.



(e) El vehículo reinicia la marcha.

(f) Segmentación y trayectoria del vehículo.

Figura 5.4: Secuencia en la que un coche se detiene en el arcén de una carretera.

recorrido.

5.2.4 Detección de eventos anómalos

Las situaciones peligrosas, tales como accidentes, deben ser detectadas por un sistema de monitorización de tráfico de manera que se provoque un aviso instantáneo. Nuestro sistema puede detectar tres tipos de situaciones peligrosas:

- Vehículos que cruzan líneas marcadas como de peligro. Estas líneas virtuales se definen en el sistema y delimitan la parte de la imagen en la cual los vehículos se mueven. De este modo, marcarán los bordes de la carretera, así como separarán carriles con sentido de circulación contraria. Cuando un vehículo cruce una de estas líneas se lanzará un aviso que alerte de una situación de potencial peligro.
- Vehículos que circulan en un mismo carril pero en sentido contrario. Esto engloba también a los vehículos que se encuentran detenidos en un carril por el que circula un vehículo. De igual forma, aquí también se lanzará una alarma.
- Posibles colisiones. El hecho de que un vehículo reduzca su velocidad de manera brusca puede significar una posible colisión contra otro vehículo u objeto. Por lo tanto, cuando se detecte esta situación se lanzará un aviso, tanto si se produce dentro como fuera de la calzada, por ejemplo el arcén.

5.3 Un sistema en tiempo real

Uno de los principales objetivos de esta Tesis ha sido conseguir un sistema que funcione en tiempo real, sin emplear hardware dedicado de coste excesivo. En esta sección se expondrá cómo se ha llevado a cabo la implementación del sistema. Primero se verá cuál es el equipamiento mínimo requerido y a continuación cómo se han distribuido las tareas vistas en los capítulos precedentes en procesos que se ejecutan de manera concurrente. El último apartado de esta sección describe una técnica llamada *segmentación virtual*. con ella se consigue reducir significativamente el coste de computación al permitir no tener que procesar todas las imágenes de la secuencia.

5.3.1 Arquitectura del sistema

En este apartado mostraremos la manera en que ha sido implementado el sistema de monitorización de tráfico. Primeramente veremos el equipamiento requerido y a continuación cuál es la distribución de procesos que se han desarrollado y qué tareas lleva a cabo cada uno de ellos.

Equipamiento

El sistema requiere el siguiente equipamiento para llevar a cabo sus funciones:

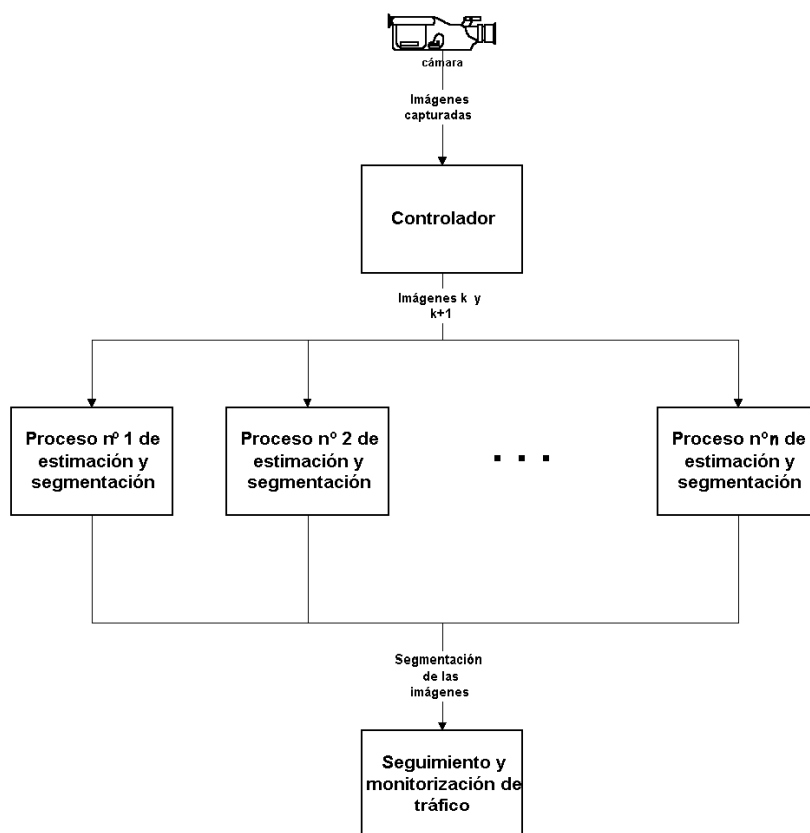


Figura 5.5: Arquitectura del sistema de monitorización de tráfico. Programas que interactúan para llevar a cabo el análisis de movimiento y las tareas de monitorización de tráfico.

- Una cámara en blanco y negro.
- Un ordenador que cumpla los siguientes mínimos:
 - Cuatro procesadores Pentium II a 400 Mhz.
 - Tarjeta de adquisición de imágenes.
 - Sistema operativo Linux.

Conviene aclarar que para poder realizar un mejor estudio de las secuencias, todos los experimentos que se presentan en esta Tesis se han realizado *off-line*. De esta forma se pueden repetir las mismas secuencias para analizar circunstancias de interés. Para ello las secuencias se han grabado mediante una cámara de vídeo VHS y luego se han digitalizado. Este modo de trabajo presenta el inconveniente de que las imágenes tienen una peor calidad, si bien, al hacer que el sistema funcione satisfactoriamente con ellas, garantizamos que éste es más robusto.

Software del sistema

Aquí se verá qué programas se han desarrollado, qué tareas lleva a cabo cada uno de ellos y cuál es la interconexión entre los mismos. Para ello nos serviremos de varias figuras, entre ellas la Figura 5.5. En dicha figura se observan los tres tipos de programas que se han realizado y la relación que existe entre ellos.

- El programa *controlador* es el encargado de comunicarse con la cámara y transferir las imágenes a los procesos de estimación y segmentación.
- Los *procesos de estimación y segmentación* son los que, como su nombre indica, efectúan la estimación y segmentación de movimiento. Se trata de una serie de programas similares que se arrancan al mismo tiempo y que tomando como entrada un par de imágenes consecutivas, proporcionan como resultado las segmentaciones iniciales de cada imagen. Junto a cada segmentación, además, proporcionan la estimación de movimiento correspondiente a las regiones detectadas.
- El programa de *seguimiento y monitorización de tráfico* es el encargado del resto de tareas del sistema. Por un lado, realiza las labores de seguimiento e integración temporal, al mismo tiempo que lleva a cabo las tareas propiamente dichas de monitorización de tráfico: medición del volumen de tráfico, seguimiento de trayectorias, detección de vehículos parados y detección de eventos anómalos.

La Figura 5.6³ muestra las tareas que son propias de cada programa. El módulo de *estimación y segmentación de movimiento* engloba las tareas de *detección de movimiento*, *segmentación estática*, *estimación de movimiento* y *segmentación de movimiento*. Por su parte, el módulo de *seguimiento y monitorización de tráfico* realiza las tareas de *seguimiento*, *integración temporal morfológica* y *estimación de parámetros de tráfico*.

El programa *controlador* no aparece en la Figura 5.6, puesto que en principio sólo se encarga de enviar órdenes a los procesos de estimación y segmentación. Cada vez que se captura una nueva imagen, el programa busca el proceso que se encuentra libre, y a éste se le envía la orden indicativa de que debe realizar la estimación y segmentación de la imagen previa, haciendo uso de la imagen recibida.

La comunicación entre procesos se realiza por medio de *secciones críticas*, las cuales han sido implementadas mediante *semáforos* (como muestra la Figura 5.7). Dichos mecanismos se encuentran disponibles en la mayoría de sistemas operativos multitarea, y su cometido es el de permitir la sincronización entre procesos. Esta

³Con respecto a esta figura resulta necesario hacer un par de aclaraciones. Según el diagrama de flujo que allí aparece, una vez que se han estimado los parámetros de tráfico, éstos son transmitidos. Conviene aclarar que dado que no existe ningún sistema de nivel superior que recopile estos datos, la parte de comunicación no ha sido desarrollada. En cuanto a la *memoria temporal*, el símbolo que la representa significa almacenamiento, sin que se trate específicamente de una unidad de disco. En realidad, y para conseguir una mayor velocidad de procesamiento, la memoria temporal, la *lista maestra de regiones*, reside en memoria principal.

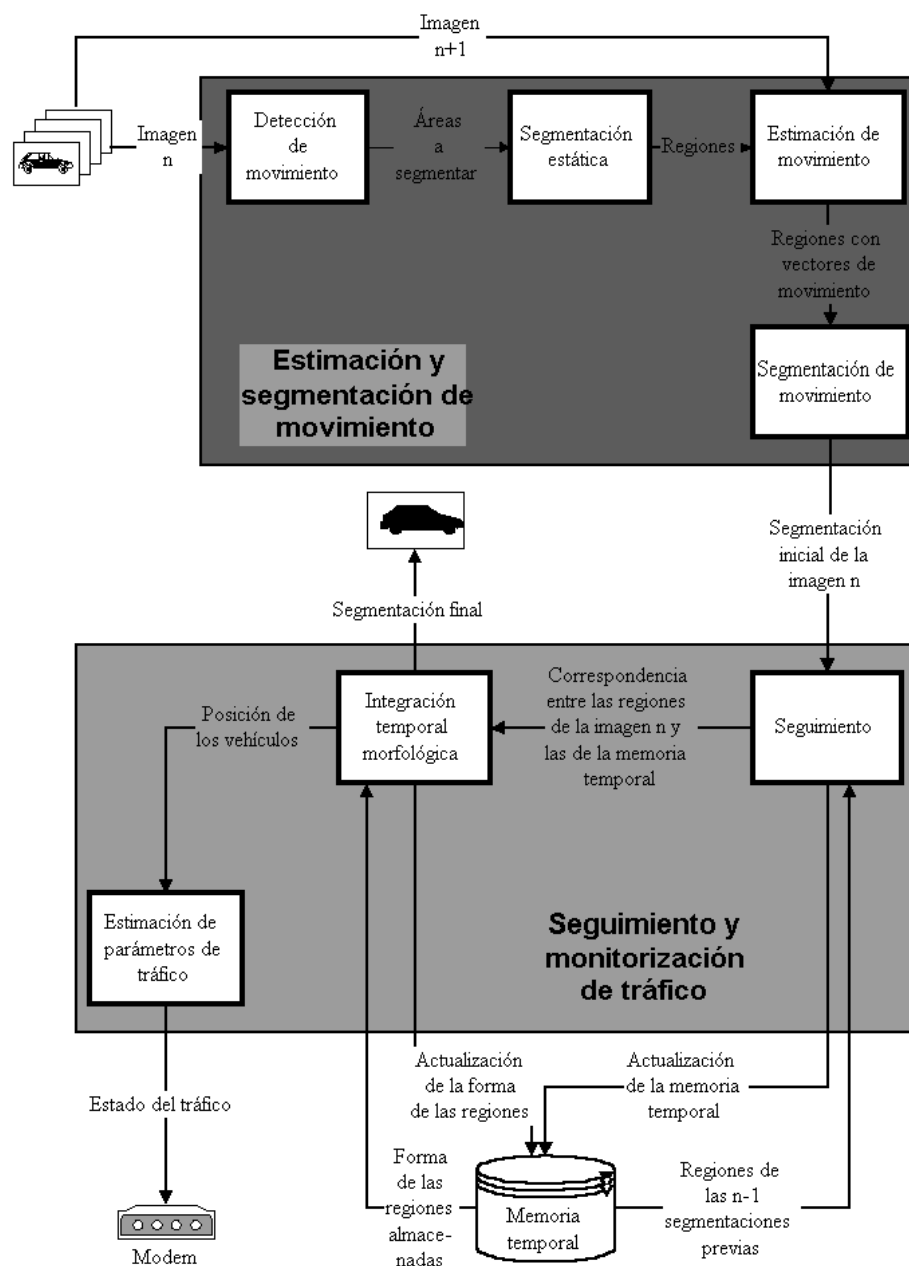


Figura 5.6: Distribución de los distintos componentes del sistema de monitorización de tráfico entre los programas implementados.

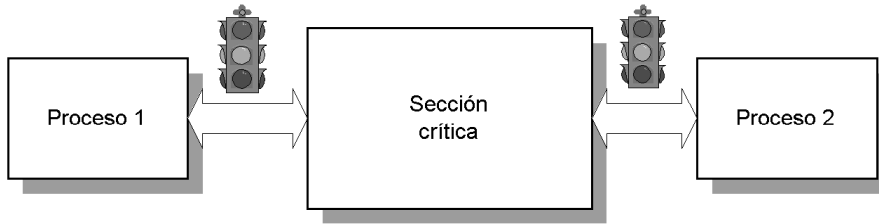


Figura 5.7: La comunicación entre procesos se realiza mediante secciones críticas.

sincronización es la que hace posible establecer restricciones de acceso a determinadas zonas de memoria (secciones críticas).

En nuestro caso las restricciones de acceso son de dos tipos:

- *Escritura.* En una sección crítica no puede haber dos procesos escribiendo al mismo tiempo.
- *Lectura.* En una sección crítica puede haber varios procesos leyendo al mismo tiempo, si bien no es posible leer mientras algún proceso está escribiendo.

5.3.2 Segmentación virtual

Una forma de acelerar el procesamiento de una secuencia es, sin duda, minimizar el número de imágenes por segundo que el sistema debe tratar. Sin embargo, reducir excesivamente esta proporción no es tampoco deseable, pues lleva aparejados dos efectos:

- Cuantas menos imágenes se tomen, mayor será el desplazamiento de los objetos de una imagen a la siguiente. Este hecho supone tanto a nuestro método de estimación de movimiento (ver Sección 3.4), como a la mayoría del resto de métodos basados en correspondencias, un aumento del tiempo de cálculo del movimiento que es debido al inevitable incremento del espacio de búsqueda.
- Cuanto más se amplíe el espacio de búsqueda, mayor es la probabilidad de obtener correspondencias erróneas. Concretamente en nuestro método esto aumentará la posibilidad de que la estimación se detenga en un mínimo local de la función de error **DFD**.

La *integración temporal morfológica* que se presentó en el capítulo anterior proporciona una forma de reducir el número de imágenes, pero sin aumentar el tiempo entre las dos imágenes que se utilizan para calcular la estimación de movimiento.

Recordemos que la integración temporal morfológica almacenaba la forma de las regiones junto con sus parámetros de movimiento y una probabilidad de pertenencia de los píxeles de la imagen a cada región. De este modo, es posible a partir de la

segmentación final de la imagen n predecir la segmentación de la imagen $n + 1$: se desplazan las regiones por medio de las predicciones de movimiento que aporta el filtro de Kalman y se asignan los píxeles de la imagen de acuerdo a las probabilidades de pertenencia.

A estas predicciones de segmentación es a lo que denominamos *segmentaciones virtuales*. Intercalándolas se consigue reducir el coste computacional, pero sin incrementar el intervalo temporal entre pares de imágenes. Obviamente, esto representa una disminución del número de observaciones que el sistema integra, y por tanto los resultados estarán menos fundamentados. En un entorno ideal, donde no existiera ruido en las observaciones, la intercalación de segmentaciones virtuales no afectaría a los resultados. Sin embargo, esto no es posible, y lo que realmente sucede es que cuanto más ruidosas son las observaciones más afecta la introducción de este tipo de segmentaciones, y por tanto, más debería limitarse su uso.

Además de reducir tiempo de computación, en un sistema en tiempo real, las segmentaciones virtuales sirven de salvaguarda ante saturaciones del sistema. En el Apartado 5.3.1 se explicó que el programa *controlador* captura imágenes a intervalos regulares de tiempo y se las va pasando a los procesos de *estimación y segmentación de movimiento*. Estos procesos son ejecuciones simultáneas de un mismo programa, y sus resultados son integrados por el programa de *seguimiento y monitorización de tráfico*. Resulta fundamental que dichos resultados sean entregados justo a tiempo, puesto que en caso contrario se comenzarían a acumular retrasos que acabarían desfasando de manera excesiva la escena real con las segmentaciones y resultados finales del sistema de monitorización.

Cuando el programa de *seguimiento y monitorización de tráfico* no recibe los resultados una vez transcurrido el intervalo previsto, envía una orden de *cancelar operación* al proceso de *estimación y segmentación de movimiento* en cuestión y a continuación intercala una segmentación virtual. De esta forma se evitan retardos sin afectar significativamente a los resultados.

5.4 Resultados

En este apartado se describirán varios experimentos. En ellos se trabajará sobre diversas secuencias, tratando de mostrar las tareas que el sistema presentado en esta Tesis desempeña de cara al control de tráfico.

Quizás la función más importante que se le puede requerir al sistema es la medición de la densidad del tráfico. Esto, a su vez, se basa en la capacidad que tenga el sistema de contar de manera segura el número de vehículos que han circulado por el tramo de carretera bajo control entre dos instantes de tiempo. Para comprobar esta capacidad, se procesarán seis secuencias correspondientes a distintas situaciones de tráfico:

- Una calle de doble dirección (Figura 5.8).
- Una vista parcial de una rotonda (Figura 5.9).

- Una carretera de doble dirección (Figura 5.10).
- Una calle con un semáforo donde se van deteniendo los vehículos (Figura 5.11).
- Un cruce de calles con una avenida (Figura 5.12).
- Una autopista (Figura 5.14).

Secuencia de una calle de doble dirección

La secuencia que aparece en la Figura 5.8 muestra una calle con dos carriles, cada uno con un sentido diferente de circulación. En realidad se trata de la misma secuencia que hemos usado en los Capítulos 3 y 4 (Figuras 3.12, 3.13, 3.14 y 4.10).

La secuencia consta de 1230 imágenes grabadas a razón de 10 imágenes por segundo. La escena presenta el inconveniente de tener un bajo contraste debido a que fue grabada al atardecer. No obstante, los resultados de la detección de vehículos fueron óptimos. Durante los 2 minutos que dura la secuencia, circularon 22 vehículos por la calle, 9 en dirección norte-sur y 13 en sentido contrario. El sistema detectó correctamente los 22 vehículos junto con sus direcciones de circulación.

En la Figura 5.8 se muestran ocho instantes de la secuencia junto con sus correspondientes segmentaciones (colocadas a la derecha de su correspondiente imagen). En las subfiguras 5.8.a, 5.8.i y 5.8.k aparecen dos de los tres⁴ peatones que quedaron registrados en la secuencia. Por su parte, en la subfigura 5.8.m lo que está circulando es una motocicleta. A pesar de la similitud en la forma de los peatones y la motocicleta, el sistema contó a ésta como un vehículo, pero no a las personas. Este hecho se debe a la escasa velocidad de los peatones, lo cual provoca que en la gran mayoría de casos sean unidos al fondo.

El resto de subfiguras muestran imágenes con varios de los vehículos que circulan durante el transcurso de la secuencia. A su derecha aparecen sus correspondientes segmentaciones. Los vehículos de las subfiguras 5.8.c y 5.8.o son un ejemplo claro del poco contraste que presenta la secuencia. Los valores de intensidad de los píxeles de estas dos imágenes se encuentran todos ellos en el rango 0-166. Se puede observar que las intensidades de la calzada y los vehículos de estas dos subfiguras son bastante parecidas.

Secuencia de la rotonda

La Figura 5.9 presenta una nueva secuencia de tráfico urbano. Los vehículos parten de la zona inferior izquierda de la imagen, donde se encontraban parados tras un semáforo, y cuando llegan al centro deben optar por girar a la derecha o a la izquierda. Los coches que van hacia la derecha giran alrededor de una gran rotonda, de la que puede observarse una pequeña parte en el centro derecha de las imágenes, mientras los de la izquierda continúan por una avenida.

⁴El peatón de las subfiguras 5.8.i y 5.8.k es el mismo. Debido a su velocidad lenta, transcurridas 48 imágenes de la secuencia, a penas se ha desplazado unos pocos metros.

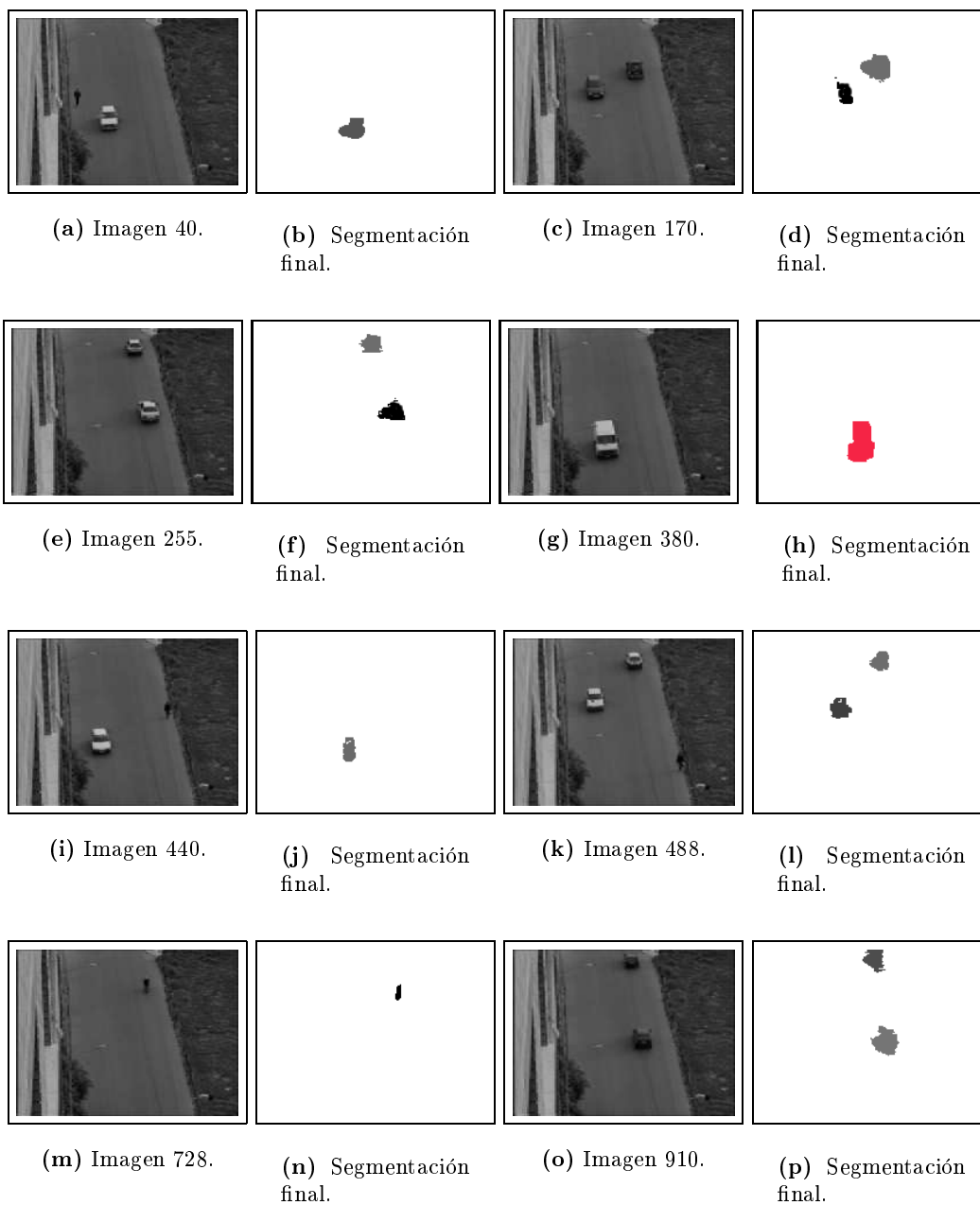


Figura 5.8: Diez imágenes de una secuencia urbana con sus correspondientes segmentaciones.



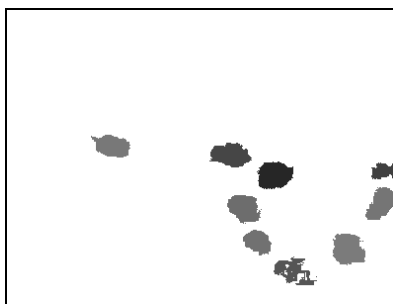
(a) Imagen 4.



(b) Segmentación de la cuarta imagen.



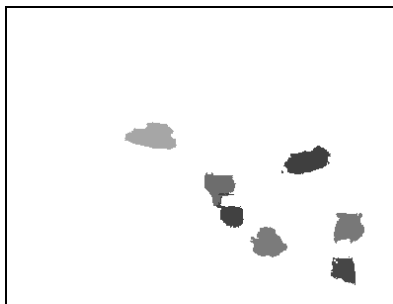
(c) Imagen 27.



(d) Segmentación de la imagen 27.



(e) Imagen 37.



(f) Segmentación de la imagen 37.

Figura 5.9: Una secuencia de vehículos en un cruce de una ciudad. Las subfiguras de la derecha corresponden a las segmentaciones de las imágenes de las subfiguras de la izquierda.

Las subfiguras de la izquierda muestran tres imágenes de la secuencia, mientras que las de la derecha son sus respectivas segmentaciones finales. En esta secuencia se intentó comprobar cómo afectan las siguientes situaciones al funcionamiento del sistema:

- Que las trayectorias no sean rectilíneas, sino curvas, como ocurre en la secuencia. Los vehículos que giran a la izquierda, inicialmente sólo muestran su parte trasera, mientras que finalmente aparecen de costado.
- La presencia de elementos estáticos que oculten parcialmente la visión de los vehículos. En esta secuencia aparecen varias farolas en la parte izquierda, mientras que en la parte inferior están los números de la fecha de grabación y un semáforo. Todos estos elementos deterioran las formas de los vehículos que se extraen en las segmentaciones iniciales, incluso produciendo la división de un vehículo en varias regiones, cada una de las cuales podría ser interpretada como un vehículo diferente.

La primera dificultad, la de las trayectorias curvilíneas, no plantea ningún problema al sistema. Dado que el cambio en la forma de los vehículos se produce gradualmente, el sistema también se adapta de manera progresiva. Las correspondencias se establecen correctamente, al tiempo que las segmentaciones van cambiando de forma gradual para adecuarse a las diferentes vistas que ofrece cada coche.

En cuanto a los objetos estáticos, éstos deterioran las segmentaciones iniciales, pero gracias a la integración temporal (ver Capítulo 4) las segmentaciones finales tienden a mitigar sus efectos. Tanto la fecha de grabación, *27 7 1997*, como un semáforo que hay junto al año, dificultan las segmentaciones de los coches. Esto puede apreciarse claramente en el vehículo que se encuentra más abajo en la subfigura 5.9.d. Su segmentación es bastante deficiente en la imagen 27, pero en la imagen 37 vuelve a ser compacta.

En el tramo de calzada que se encuentra a la izquierda de la imagen, aparecen dos farolas que tienden a dividir las segmentaciones de los vehículos. En la subfigura 5.9.b el coche que está tras una de estas farolas es segmentado parcialmente, es decir, sólo la parte trasera es extraída en la segmentación final. En las siguientes imágenes el vehículo será totalmente perdido, y de hecho ni siquiera será contabilizado como un coche. Ello se debe a que no han pasado suficientes imágenes como para que la integración temporal pueda acumular la información temporal necesaria para corregir los errores. En la segmentación inicial del vehículo, cuando éste llega a la farola, es dividido en dos regiones, con estimaciones de movimiento erróneas provocadas por los píxeles estáticos de la farola. Por todo ello, la integración temporal no puede seguir al vehículo, con lo que nunca llega a declararlo como una *región permanente*, lo cual implica que no sea contabilizado como un vehículo.

Afortunadamente, cuando se cuenta con suficientes observaciones de un vehículo, la anteposición de las farolas no afecta al resultado final. Esto podemos verlo en la subfigura 5.9.d, donde un vehículo diferente sí es segmentado correctamente. En este caso, tanto la integración temporal morfológica, como el filtro de Kalman han

contado con suficientes observaciones como para que unos cuantos errores no afecten a la estimación final.

Con respecto al conteo de vehículos, la secuencia es bastante corta, algo más de un minuto, tiempo durante el cual circularon 19 vehículos. De estos 19 coches el sistema sólo falló en la detección del vehículo mencionado anteriormente, es decir detectó 18 de los 19 vehículos que circularon.

También se contabilizó cuántos seguían girando alrededor de la rotonda y cuántos torcían a la derecha. De los 18 coches detectados, 10 continuaron en la rotonda, y 8 tomaron la avenida de la derecha.

Secuencia de la carretera

En la Figura 5.10 aparecen varias imágenes de una secuencia de carretera, junto a sus correspondientes segmentaciones. Se trata de una vía de dos sentidos, con un carril para cada uno de ellos. En el tramo que muestra la escena está permitido el adelantamiento, por lo que en ambos carriles pueden circular vehículos en cualquiera de los dos sentidos.

Es una secuencia relativamente larga, unos 4 minutos, en la que aparece un gran número de vehículos que además son de diversas clases: coches, camiones, coches con caravana, etc. Otra característica adicional de esta secuencia es la gran profundidad de campo que presenta. Así, no es posible detectar el movimiento de los vehículos que al encontrarse muy alejados de la cámara presentan una velocidad muy cercana a cero en el plano de la imagen. Esto se aprecia en la segunda imagen que se presenta, subfigura 5.10.c. donde el vehículo más alejado no puede detectarse. No obstante, cuando su velocidad en la imagen permite diferenciarlo del fondo, sí que es segmentado y contabilizado.

En esta misma subfigura 5.10.c un coche está realizando un adelantamiento. Aunque en este tramo de carretera están permitidos los adelantamientos, se definió una *línea de peligro* sobre la raya discontinua. De esta forma, se simuló una situación de potencial peligro cuando un vehículo rebasa a otro en zona prohibida. Cuando el centro de masas de la segmentación del coche se situó a la derecha de la línea, el sistema generó un aviso. Como ya se comentó, este aviso debería ser recibido por un programa de más alto nivel⁵, donde posiblemente un operador estudiaría la potencialidad del peligro que plantea la situación.

El resto de subfiguras de la Figura 5.10 corresponden a distintas imágenes de la secuencia. En la subfigura 5.10.i se observan dos camiones de gran tamaño, mientras que en las restantes imágenes los vehículos son coches. Durante el transcurso de la secuencia se detectaron 22 de los 23 vehículos que por ella circularon. De estos 22 vehículos, 14 circulaban por el carril de la izquierda, y 8 por el de la derecha.

⁵Se comentó con anterioridad que estos programas de más alto nivel no han sido desarrollados por corresponder a un tipo de problemas que escapan del ámbito de esta Tesis. El sistema desarrollado se limita a realizar mediciones y detectar eventos

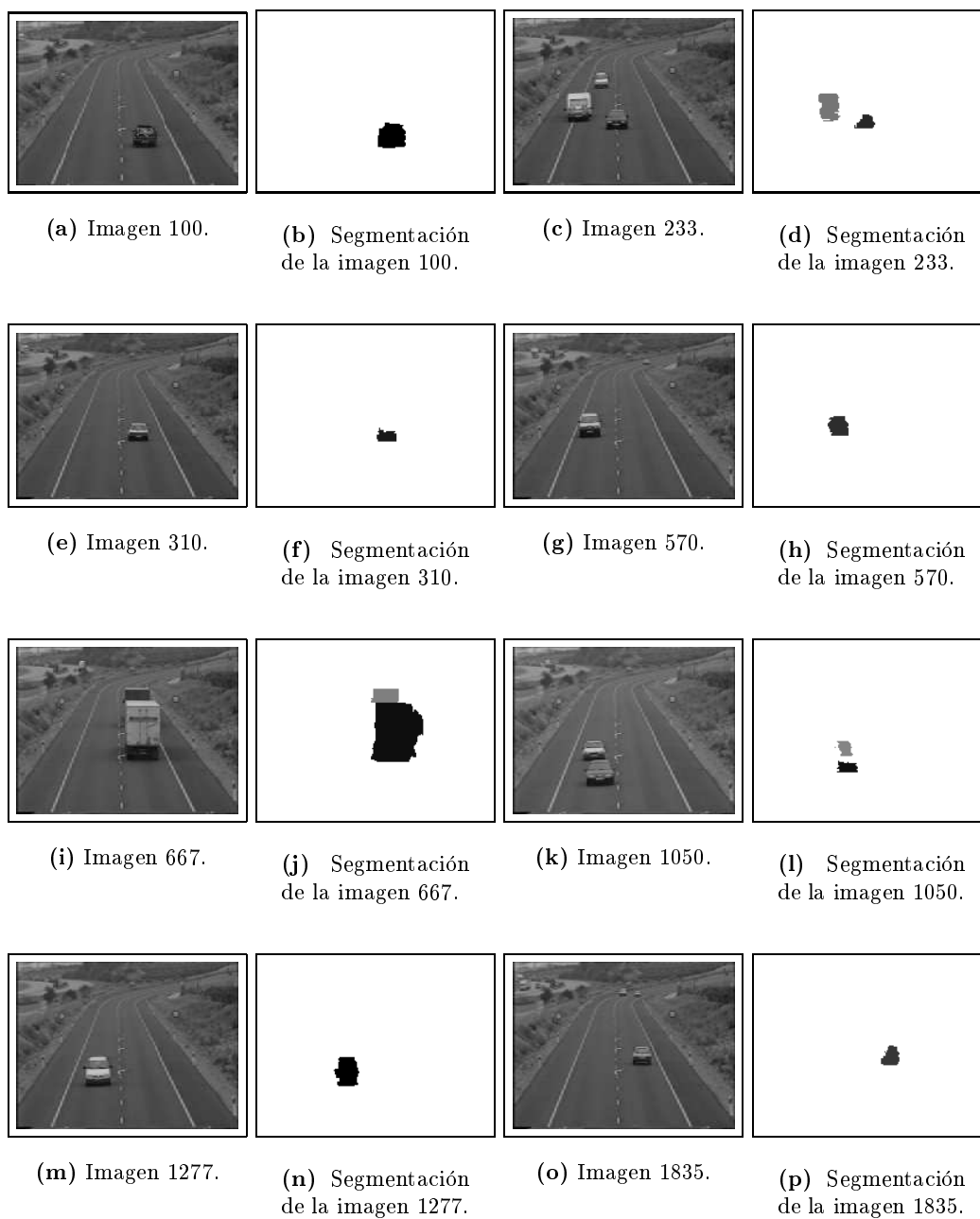


Figura 5.10: Secuencia de tráfico en carretera. Vehículos de distintos tipos circulan sobre una carretera de dos sentidos.

Secuencia del semáforo

En la Figura 5.11 se muestran cuatro imágenes de una misma secuencia. En ella dos coches se aproximan a un semáforo, en el cual se detienen. En la subfigura 5.11.a se muestra la imagen número 15 de la secuencia. En ella un coche comienza a detenerse ante el semáforo que se encuentra en rojo. De este modo, la imagen número 37 (subfigura 5.11.d) es en la última donde se puede apreciar el movimiento del vehículo en la imagen.

Las subfiguras 5.11.b y 5.11.c representan las segmentaciones inicial y final de la imagen 15, mientras que, por su parte, las subfiguras 5.11.e y 5.11.f corresponden a la imagen 37. En todas ellas podemos apreciar que la forma del vehículo ha sido extraída con bastante exactitud.

En la imagen 79 (subfigura 5.11.g) un nuevo vehículo está reduciendo su velocidad ante la proximidad del semáforo, que continúa en rojo. Obviamente, al no cambiar el semáforo, el primer coche sigue detenido, por lo que la segmentación inicial no puede extraer su forma, puesto que sigue sin detectarse movimiento. No obstante, la segmentación final sigue conteniendo su silueta. Esto se consigue por medio de la integración temporal (ver Capítulo 4). Dado que la región del primer vehículo fue declarada como permanente, sólo se le permite desaparecer cuando abandona la escena por alguno de los bordes de la imagen.

Por último, la Subfigura 5.11.j corresponde a la imagen número 94, donde ambos vehículos se hayan totalmente parados. Por ello, la segmentación inicial no detecta objetos móviles, pero de nuevo la segmentación final contiene la segmentación de ambos vehículos.

Secuencia del cruce de calles con la avenida

Las imágenes de la Figura 5.12 corresponden a un cruce urbano entre una avenida y dos calles. La avenida contiene tres carriles, uno de ellos para autobuses, y el sentido de circulación es de izquierda a derecha. La calle inferior es de doble dirección con un carril para cada sentido. Por lo tanto, por esta calle los vehículos pueden tanto abandonar la avenida, como incorporarse a ella. Por contra, la calle que completa el cruce, y que se halla en la parte superior, es de una única dirección, pero tiene dos carriles por los que los coches salen de la avenida.

La duración de la secuencia es de algo más de 15 minutos. Tiempo durante el cual circularon 188 vehículos de cuatro tipos: motos, coches, furgonetas y autobuses. El sistema contabilizó 191 con lo que el error producido fue de un 1,59%. Asimismo, también se controló el tráfico de cada una de las calles. Para ello se contó cuantos vehículos abandonaban la escena por cada una de las tres calles, dando los siguientes resultados:

- *Avenida*. De los 191 vehículos contabilizados, 135 salieron de la imagen por la vía principal.
- *Calle superior*. 42 vehículos.

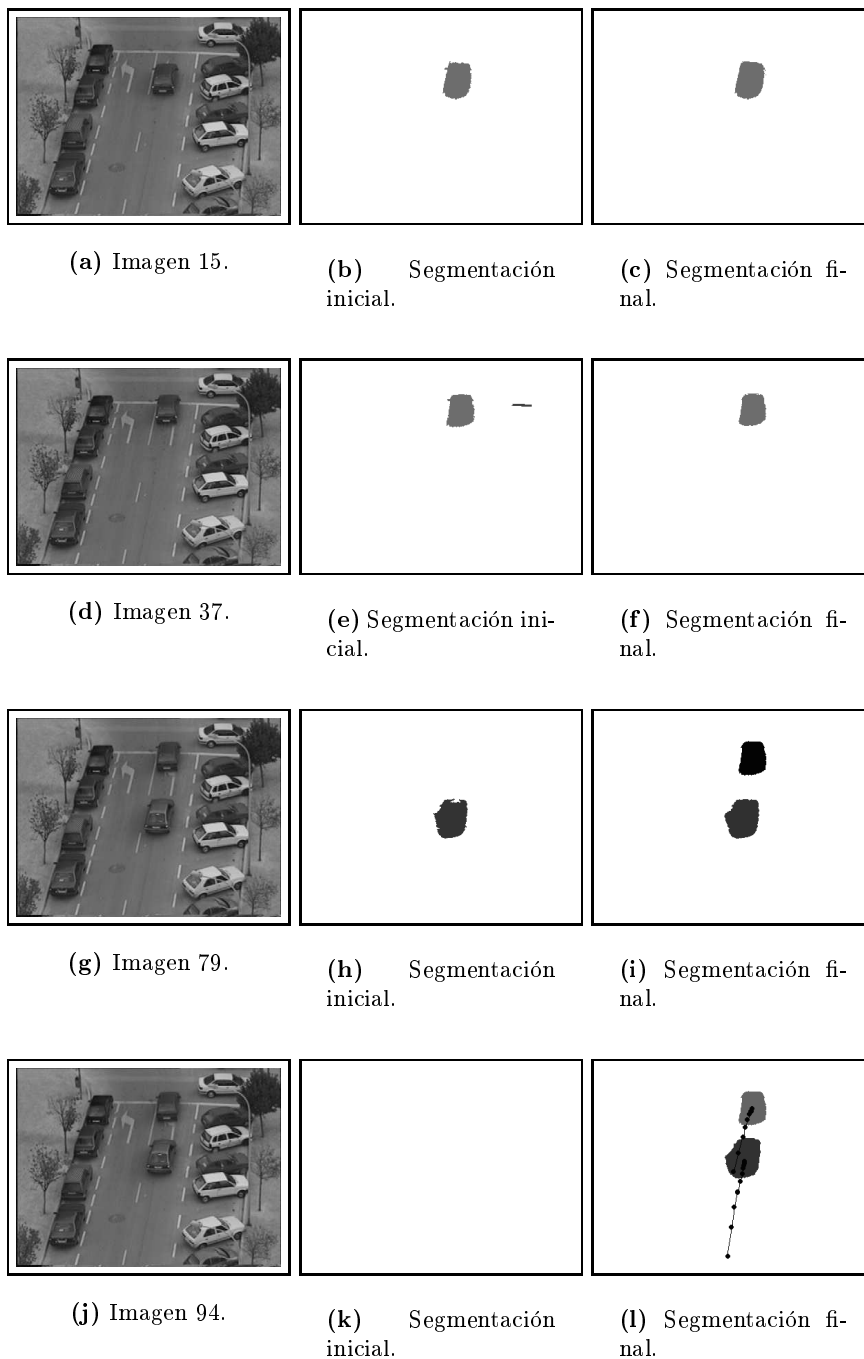


Figura 5.11: Secuencia de coches deteniéndose ante un semáforo. Primera columna: imagen original. Segunda columna: segmentación inicial. Tercera columna: segmentación final de la imagen.

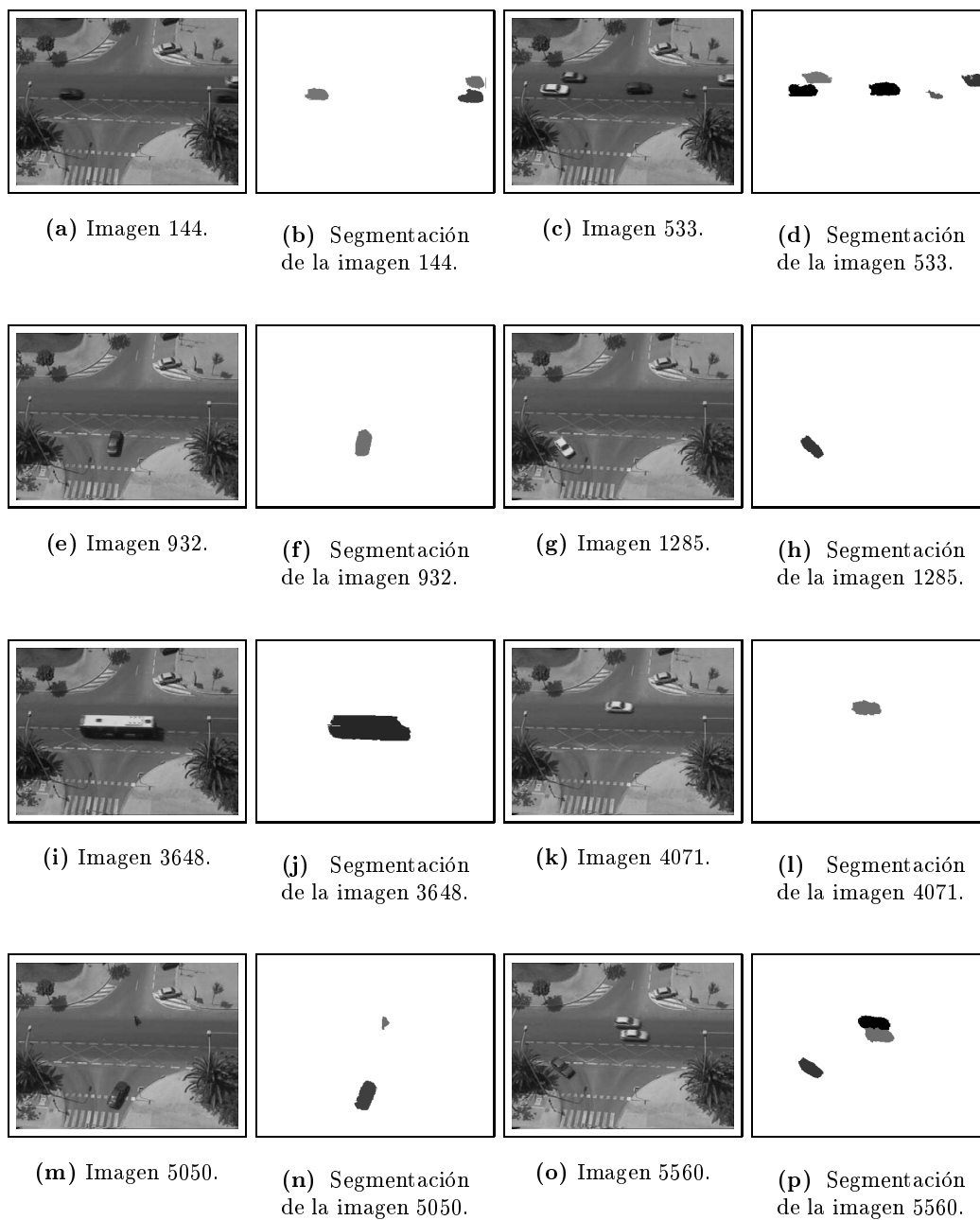


Figura 5.12: Secuencia de tráfico urbano un cruce de una avenida.

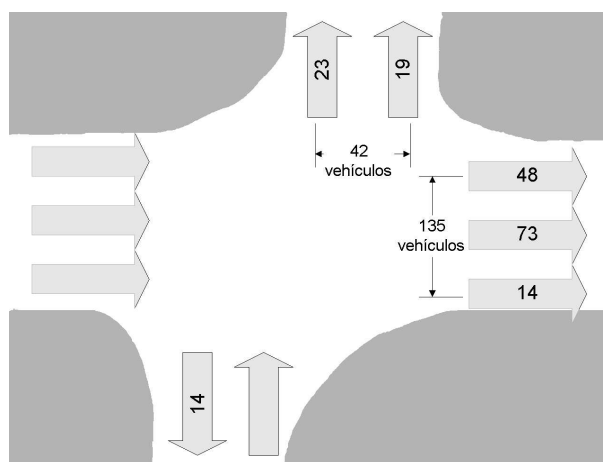


Figura 5.13: Vehículos detectados en cada una de las direcciones de salida.

- *Calle inferior.* 14 vehículos giraron por esta calle desde la avenida.

La Figura 5.13 muestra gráficamente el volumen de tráfico. Las flechas que contienen un número representan carriles por los que se abandona el campo visual de la cámara. El resto de flechas indican que por el correspondiente carril entran vehículos. Se observa que la mayor parte de vehículos tras superar el cruce siguen circulando por la avenida, 135 vehículos. El número contenido en cada flecha indica cuántos vehículos circularon por el carril sobre el que está situada. Así, se puede ver que en la avenida el carril más utilizado es el central, y el menos usado el inferior, al tratarse del carril “Bus”. La calle superior es por la que en segundo lugar más vehículos abandonan el cruce, 42 vehículos. En este caso el uso de los dos carriles es bastante similar. Por último citar que el carril de salida menos usado es el de la calle inferior, por el que circularon tan solo 14 vehículos.

Los tres errores que se produjeron en esta secuencia se debieron a tres vehículos que desaparecieron de la segmentación inicial durante un excesivo número de imágenes consecutivas. Esto se debió al fuerte viento que había en el instante de la grabación, el cual producía constantes vibraciones en la cámara. Se debe tener en cuenta que la cámara estaba montada sobre un trípode, sin que hubiera ningún tipo de protección contra el aire. En una instalación real del sistema se contaría con más medios, de manera que la fijación de la cámara impidiera vibraciones tan grandes como las que se registraron.

Durante toda la secuencia el sistema mostró una gran robustez frente a las oscilaciones provocadas por el viento. Si embargo, en los instantes en que circulaban estos tres vehículos las vibraciones fueron especialmente ostensibles, con lo que se impidió que las segmentaciones fueran correctas durante varias imágenes consecutivas. Debido a esto, se perdió el rastro de los vehículos, con lo que al volver a ser segmentados correctamente, el sistema los contabilizó como nuevos.

Como consecuencia de estas situaciones, el sistema lanzó tres avisos, al entender que los tres vehículos habían reducido bruscamente su velocidad. Estos eventos son de especial interés pues pueden indicar un choque y una posible situación posterior de peligro.

Las subfiguras de la Figura 5.12 muestran ocho escenas distintas de esta secuencia. En la subfigura 5.12.a tres coches circulan a lo largo de la avenida, de los cuales dos están a punto de salir del campo visual de la cámara. La subfigura 5.12.b corresponde a la segmentación final de estos vehículos.

En la subfigura 5.12.c aparecen 4 coches y una moto. Puede verse en la segmentación, subfigura 5.12.d, que a pesar de su tamaño (unos 100 píxeles) y su elevada velocidad (80 ó 90 km/h), la moto es correctamente segmentada.

A continuación, subfigura 5.12.e, un coche de color oscuro cruza transversalmente la avenida desde la calle inferior hacia la superior. La subfigura 5.12.g corresponde al giro de un vehículo hacia la calle inferior, y la subfiguras 5.12.i y 5.12.k muestran un autobús y un coche, respectivamente.

En la subfigura 5.12.m, una moto y un coche han salido de la calle inferior. Mientras el coche gira hacia la avenida, la moto la cruza hacia la calle superior. Por último, la subfigura 5.12.o contiene tres coches, dos de ellos muy pegados, pero que son correctamente segmentados.

En esta secuencia se puede apreciar que el sistema puede tratar vehículos de diversos tamaños: desde motocicletas, subfiguras 5.12.c y 5.12.m, hasta autobuses, subfigura 5.12.i. Esto nos permite pensar en una sencilla extensión de las tareas del sistema. En ella se realizaría una clasificación de los vehículos de una secuencia en tres categorías: motos, coches, y vehículos grandes (autobuses y camiones). La clasificación se basaría en el número de píxeles que componen una región.

Secuencia de la autopista

La Figura 5.14 muestra una secuencia correspondiente a una autopista. Este tipo de vías presentan como característica particular la elevada velocidad de los vehículos que por ella circulan. Aunque el límite de velocidad en España esté fijado en 120km/hora, es frecuente que algunos vehículos superen este límite, a veces ampliamente. Este hecho obliga a que el sistema deba procesar más imágenes por segundo de las que sería necesario en el resto de vías. En el caso de la secuencia que presentamos se han utilizado 12 imágenes por segundo, si bien todos los vehículos parecían circular a velocidades relativamente moderadas, menos de 140km/hora. Es por ello, que para una implantación real del sistema en una autopista quizás debiera ampliarse el ritmo de captura hasta las 15 imágenes por segundo.

En el caso concreto de la autopista de la Figura 5.14 la densidad normal de tráfico es bastante baja. Así, durante los nueve minutos que dura la secuencia, circularon solamente 32 vehículos. De estos, la mayoría eran camiones, y el resto coches o furgonetas.

El sistema contabilizó 33 vehículos en lugar de los 32 que circularon. El error se

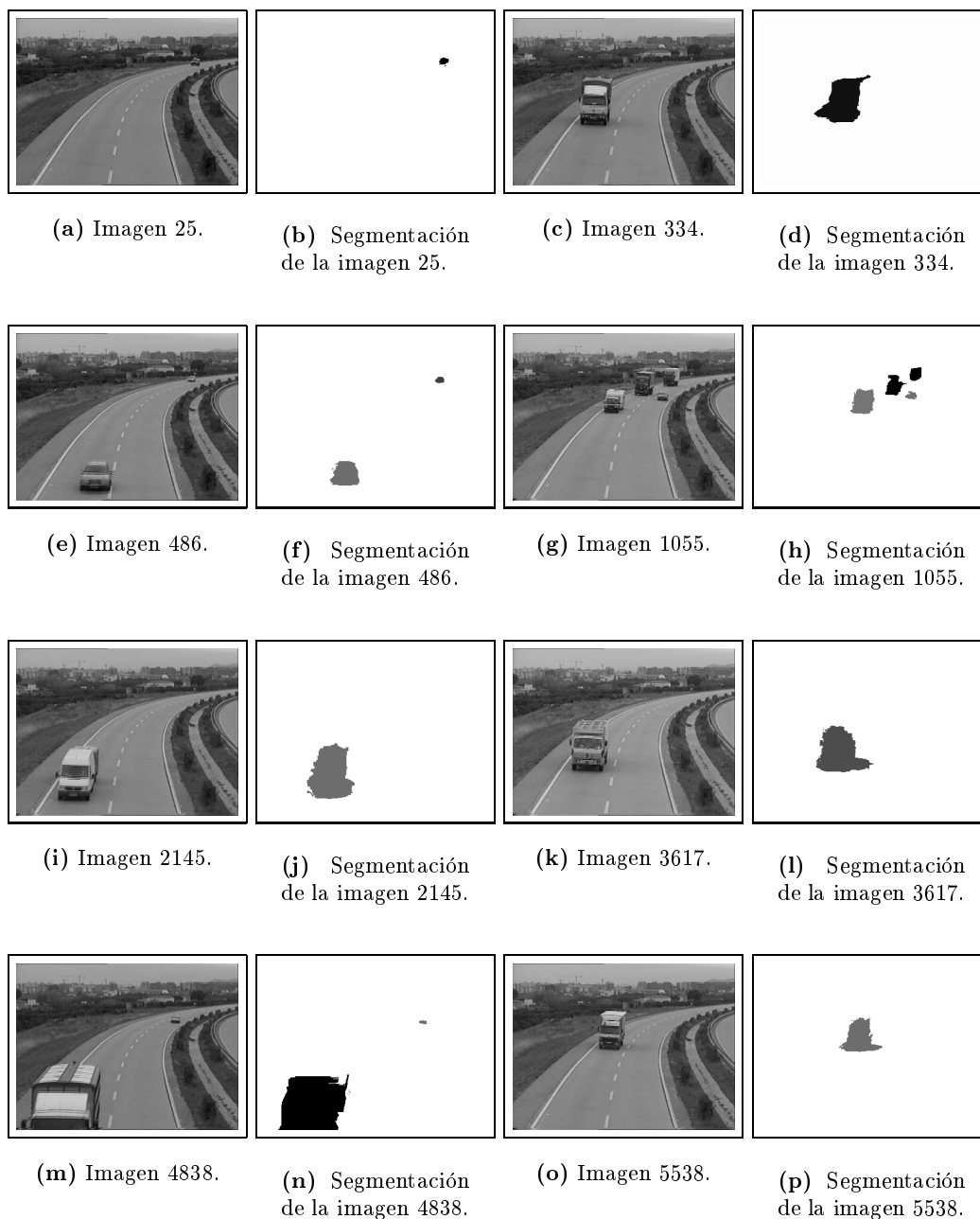


Figura 5.14: Secuencia de tráfico en una autopista.

produjo al dividir en dos un camión con un gran tráiler durante varias segmentaciones iniciales. El número de vehículos en cada carril fue de 27 (teniendo en cuenta el vehículo contabilizado de más) por el carril de la izquierda de la imagen y 6 por el derecho.

En la Figura 5.14 se muestran varias escenas de la secuencia junto a sus correspondientes segmentaciones. La subfigura 5.14.a presenta un coche que se acerca a la cámara. A pesar de su gran lejanía, el vehículo puede ser segmentado. En otros tipos de vías, los vehículos no pueden ser detectados cuando están tan lejos, debido a que su velocidad en la imagen es muy cercana a cero. En este caso, al circular más rápidamente los vehículos, también las velocidades en la imagen son más altas y por tanto se pueden detectar a una mayor distancia de la cámara.

Las subfiguras 5.14.c, 5.14.k, 5.14.m y 5.14.o corresponden a varios camiones, que es el tipo de vehículo que más aparece en la secuencia. Se puede ver que los camiones se encuentran a diferentes distancias relativas a la cámara, donde el más cercano es el de la subfigura 5.14.m que ya casi ha salido del campo visual de la cámara.

En la subfigura 5.14.e aparecen dos coches situados a diferentes distancias y en la subfigura 5.14.g hay dos camiones y dos coches. Uno de los coches lleva detrás una caravana, sin embargo el sistema lo contabilizó correctamente como un único vehículo. En la subfigura 5.14.i hay una furgoneta con un remolque, que también fue segmentada correctamente como una misma unidad.

Velocidad de procesamiento

El sistema descrito en este capítulo ha sido probado sobre un ordenador de cuatro procesadores Pentium II a 400 Mhz. Sobre este equipo se ha conseguido el procesamiento de las anteriores secuencias en tiempo real, a excepción de la secuencia de la autopista, donde se tuvo que incrementar el número de imágenes por segundo hasta 12. En todas las secuencias procesadas en tiempo real, el tamaño de imagen empleado fue 192×144 , y el ritmo de adquisición de imágenes 10 por segundo, utilizando una segmentación virtual por cada segmentación real.

La secuencia de la autopista no pudo procesarse en tiempo real, al tener que usar 12 imágenes por segundo. No obstante, en el momento en que se está acabando de escribir esta Tesis, ya es posible encontrar fácilmente ordenadores PC con procesadores Pentium III que trabajan a 900Mhz o más. Así pues, sería razonable pensar que con uno de estos equipos, también esta secuencia se podría procesar en tiempo real.

5.5 Conclusiones

En el presente capítulo se han mostrado las distintas tareas que el sistema de monitorización propuesto en esta Tesis puede llevar a cabo. Dichas tareas son básicamente:

- Medición del volumen de tráfico.
- Seguimiento de trayectorias.

- Detección de vehículos parados.
- Detección de eventos anómalos.

Como consecuencia de las anteriores tareas, se puede no sólo mejorar la situación del tráfico en un instante dado, sino también obtener estadísticas a más largo plazo que ayuden a planificar la red vial.

Como se ha visto en la Sección 5.3, el sistema desarrollado puede trabajar en tiempo real. Para ello se han implementado tres tipos de procesos que cooperan en el procesamiento de las imágenes:

- *Controlador*, que se encarga de ordenar el procesamiento de cada imagen.
- *Estimación y segmentación de movimiento*, que se encarga de la segmentación inicial de la imagen.
- *Seguimiento y monitorización de tráfico*, que realiza la segmentación final de cada imagen y lleva a cabo las tareas de monitorización de tráfico.

Además, se ha mostrado una técnica llamada *segmentación virtual* que permite evitar el procesamiento de ciertas imágenes, lo cual permite un ahorro computacional. Dicha técnica consiste básicamente en usar la información recogida por el sistema a lo largo de la secuencia para predecir los resultados de esas ciertas imágenes.

En este capítulo se ha prestado una especial atención al apartado dedicado a los resultados de la aplicación. Se han usado escenas tanto de tráfico urbano como de carretera, probando el funcionamiento del sistema bajo diversas circunstancias que se presentan usualmente: semáforos, cruces, adelantamientos, etc. El sistema se ha mostrado apto para ambos tipos de vías, donde las velocidades medias de los vehículos son diferentes. También se ha visto que el funcionamiento del sistema no es dependiente de la dirección de circulación de los vehículos: tanto rectilíneas, vertical y horizontal a la imagen; como circulares, en curvas y rotondas.

En el caso de las trayectorias curvas, la integración temporal no se ve afectada por la variación del punto de vista del objeto que se aprecia en cada imagen de la secuencia. La memoria temporal va siendo actualizada a medida que el vehículo gira, con lo que las segmentaciones se van adaptando a la visión que la cámara percibe de los objetos.

Se ha comprobado el funcionamiento frente a diferentes tipos de vehículos: coches, motos, furgonetas y camiones. A pesar de los diferentes tamaños, sobre todo en el caso de las motos que pueden ocupar alrededor de 100 píxeles, el sistema es capaz de realizar un seguimiento de todos ellos.

Se debe resaltar que el sistema ha mostrado una gran fiabilidad, fundamentada en gran medida por la integración temporal. Esta técnica elimina el efecto de la mayoría de errores que se producen en el tratamiento de imágenes puntuales de la secuencia. Además, permite que la existencia de determinados elementos estáticos que se interponen en la visión de los coches, como semáforos o farolas, no afecten a los resultados.

Se ha utilizado un equipamiento basado en ordenadores PC, sin tener que recurrir a sistemas dedicados que dispararían el coste económico. Con el ordenador empleado, el sistema es capaz de procesar hasta diez imágenes por segundo, lo cual es suficiente para la gran mayoría de situaciones.

Capítulo 6

Conclusiones

Contenido

6.1 Conclusiones generales	145
6.2 Aportaciones	147
6.3 Líneas de trabajo futuro	148
6.4 Lista de publicaciones surgidas de esta Tesis	149

Este capítulo resume el trabajo presentado en esta Tesis. La Sección 6.1 presentará las conclusiones generales que pueden extraerse a partir de la investigación realizada. La Sección 6.2 resumirá las principales aportaciones y la Sección 6.3 tratará las posibles líneas de investigación que pueden surgir a partir del trabajo desarrollado en esta Tesis. Por último, en la Sección 6.4 se muestra una lista de las publicaciones que el presente trabajo ha ido generando durante su desarrollo.

6.1 Conclusiones generales

Como primera conclusión, creemos que puede afirmarse que los objetivos que en un principio se plantearon han sido completados satisfactoriamente. Se han desarrollado un conjunto de métodos y técnicas que se interconectan para alcanzar la meta principal: un sistema basado en visión por ordenador capaz de realizar diversas tareas de monitorización de tráfico. Para ello se ha realizado un profundo estudio de la literatura existente sobre cada una de las partes que integran el trabajo: monitorización de tráfico, segmentación de imágenes estáticas, estimación y segmentación de movimiento, integración temporal, y seguimiento.

A lo largo de los diferentes capítulos de esta Tesis se ha mostrado que el análisis de movimiento sobre el plano de la imagen puede proporcionar todo el conocimiento necesario para resolver las tareas de monitorización de tráfico. Por lo que el uso de estructuras tridimensionales incrementaría substancialmente el coste computacional sin apenas aportar ventajas a la consecución de las tareas más importantes.

Dentro del análisis bidimensional, las regiones, como característica básica del proceso, han mostrado dotar de una gran robustez a los métodos. Permiten asignar vectores de movimiento a todos los píxeles de la imagen sin disparar el coste computacional de su estimación. Como se vio en el Capítulo 3, resultan especialmente interesantes en las secuencias de tráfico, donde las oclusiones son frecuentes y el fondo es estático.

En el Capítulo 2 se realizó un estudio de las técnicas de segmentación de imágenes estáticas, y más concretamente de su aplicación al problema del tráfico. Como se muestra en esta Tesis, no parece necesario el uso de color, ya que que la intensidad de gris aporta suficiente información. La técnica a utilizar debe ser no supervisada, capaz de adaptarse automáticamente a los cambios en el número y tamaño de los objetos y sobre todo obtener regiones compuestas por píxeles de no más de un objeto. Los resultados de este capítulo muestran que la intensidad de gris resulta ser un buen criterio de homogeneidad.

El método de segmentación estática desarrollado (Capítulo 2) cumple las anteriores restricciones. Se adapta dinámicamente al número de vehículos que aparezcan en la escena. Además, ofrece un excelente comportamiento con respecto a la restricción más importante: no formar regiones donde haya píxeles que pertenezcan a más de un vehículo.

El Capítulo 3 mostró que un sencillo modelo de movimiento translacional es suficiente para realizar una correcta segmentación de movimiento en las escenas de tráfico. Frente a modelos de movimiento más complejos, reporta un considerable ahorro en el coste computacional.

En este mismo capítulo se vio que las técnicas basadas en substracción de imágenes, a pesar de su sencillez, son de enorme utilidad en el análisis de movimiento. Con un reducido coste computacional, permiten agrupar la mayor parte de las regiones que pertenecen a un mismo objeto.

Al emplear técnicas basadas en substracción se suele elegir entre una de las dos posibilidades más comunes: substracción de la imagen actual y una de referencia, o de dos imágenes consecutivas. Hemos mostrado que su combinación no es redundante. La primera ofrece mejores resultados cuando las condiciones son estables, mientras que la segunda es menos sensible a las rápidas variaciones de las condiciones de la escena y a pequeñas vibraciones de la cámara.

Seguimiento y segmentación son dos procesos que pueden colaborar mutuamente. De manera contraria a lo que en otros trabajos se ha venido haciendo, en el Capítulo 4 se presenta un método que no sólo usa la información proveniente de la segmentación para realizar el seguimiento, sino que, además, este segundo proceso permite mejorar las segmentaciones. La integración temporal morfológica permite acumular conocimiento sobre la forma de las regiones de la imagen, para a partir de ahí obtener segmentaciones más fiables.

La utilización de un filtro de Kalman en el seguimiento reduce la influencia de los errores de medición que inevitablemente aparecen en cualquier técnica de estimación de movimiento. Además, aporta predicciones que pueden ser empleadas por la

integración temporal morfológica.

El Capítulo 5 muestra que es posible implementar el sistema en un hardware común (por ejemplo, un ordenador del tipo PC). Esto significa un menor coste frente a sistemas dedicados, como por ejemplo: sistemas basados en DSP's.

6.2 Aportaciones

Las principales aportaciones del trabajo desarrollado en esta Tesis se pueden resumir en las siguientes:

- Un método de segmentación de imágenes estáticas que utiliza la intensidad de los píxeles como criterio de homogeneidad. Este criterio ha resultado especialmente útil dentro del ámbito de aplicación de este trabajo, las escenas de tráfico, puesto que tanto los objetos, vehículos, como el fondo, la carretera, se caracterizan por la ausencia de texturas. Éste método cumple la premisa de ser *no supervisado* y, además, utilizar una estrategia jerárquica que le permite adaptarse a secuencias con un número cambiante de objetos.

Ya se comentó en capítulos precedentes que esta técnica puede ser substituida por alguna otra que se encuentre en la literatura o que esté disponible en un futuro. Para ello se deberán satisfacer las tres mismas condiciones que nuestro método cumple: ser no supervisado, adaptarse automáticamente a la variación del número de objetos de la escena, y conseguir segmentaciones donde, aunque un objeto pueda ser subdividido en diversas regiones, nunca varios objetos formen parte de la misma región final.

- Un método de detección de movimiento (Capítulo 3) basado en una combinación de dos técnicas de substracción de imágenes: una de substracción de una imagen referencia que representa el fondo estático, y otra de substracción de imágenes consecutivas. Este método realiza un rápido agrupamiento de la mayor parte de las regiones que pertenecen a un mismo objeto. La imagen referencia se recalcula tras cada imagen, permitiendo así su adaptación a las variaciones de las condiciones de la escena.
- El método de análisis de movimiento que se describe a lo largo del Capítulo 3 constituye en sí una combinación novedosa de técnicas, unas nuevas y otras ya existentes, que permiten la estimación y segmentación de movimiento en tiempo real en una escena de tráfico. Esta estimación se basa en un modelo de movimiento translacional, el cual permite un ahorro en el coste computacional frente a otros modelos más complejos.
- Con respecto al seguimiento, se plantea un nuevo enfoque de su relación con la segmentación: no sólo la segmentación aporta información al seguimiento, sino que también el seguimiento puede ayudar a mejorar las segmentaciones.

- La integración temporal morfológica que en esta Tesis proponemos es una técnica totalmente nueva que permite obtener segmentaciones basadas en todas las imágenes precedentes. De este modo, las soluciones son más estables y, por tanto, menos influenciadas por los errores.
- Se ha introducido el concepto de *segmentación virtual*, el cual permite, basándose en la integración temporal morfológica, proyectar las segmentaciones anteriores evitando el procesamiento de ciertas imágenes. Esto permite reducir el coste computacional, a la vez que se salvaguarda al sistema frente a aumentos desmesurados de carga computacional (cuando el sistema no es capaz de procesar a tiempo una imagen, se introduce una segmentación virtual).

6.3 Líneas de trabajo futuro

En general, la mayoría de técnicas aquí tratadas pueden ser tomadas como punto de partida para desarrollar trabajos futuros. A continuación citaremos algunas de las posibles líneas de trabajo a seguir:

- El estudio de nuevos métodos de estimación de movimiento que empleen modelos algo más complejos que el puramente translacional que se ha usado en esta Tesis. Ya se comentó con anterioridad que la utilización de este modelo de movimiento respondía al deseo de minimizar al máximo el coste computacional de este proceso. Aunque este modelo simple ha respondido de manera bastante satisfactoria, no cabe duda que se podrían reducir los errores de estimación si se usara un modelo que al menos contemplara el efecto de escalado que se produce en los vehículos cuando se acercan o alejan de la cámara.
- Dado que el modelo de movimiento era el translacional, la integración temporal morfológica se desarrolló sin considerar modelos más complejos. Sería altamente interesante extender dicha técnica para ampliar las posibilidades de su utilización.
- Podría incrementarse la precisión en la estimación de trayectorias al combinar las técnicas propuestas en esta tesis con otras basadas en correspondencias de puntos o bordes. Dadas las segmentaciones de dos imágenes consecutivas se establecerían correspondencias entre las características de ambas imágenes. Esto permitiría estimar los parámetros de movimiento de cada región con mayor precisión, al mismo tiempo que permitiría establecer correspondencias entre regiones, por ejemplo, dada una región, su correspondiente podría establecerse como aquella que contiene un mayor número de características que son correspondientes de las de la primera región.
- Otra extensión del presente trabajo consistiría en la estimación de las velocidades reales de los vehículos. Para ello sería necesario incorporar algún tipo de conocimiento acerca de la geometría de la escena, calibración del sistema, etc.

6.4 Lista de publicaciones surgidas de esta Tesis

El embrión de la presente Tesis fue la tesina de master que el autor realizó durante el curso 1995-96 en la Universidad de Surrey [Badenas, 1996] y que fue dirigida por el Doctor Mirosław Bober. A partir de este primer trabajo, se han atravesado diversas etapas que han ido aportando conocimientos y resultados que han sido presentados a diversos congresos o revistas. La presente lista ordena cronológicamente las publicaciones relacionadas con esta Tesis que hasta la fecha ha generado el presente trabajo:

- [Badenas, 1996] J. Badenas. Application of Motion Estimation and Segmentation Techniques to Traffic Monitoring. Msc. Thesis del *Master on Multimedia Technology and Systems, University of Surrey, U.K.*, 1996.
- [Badenas *et al.*, 1997a] J. Badenas, M. Bober y F. Pla. Combining Intensity and Motion-based Segmentation for Robust Traffic Monitoring. En *Proceedings of the VII Symposium Nacional de Reconocimiento de Formas y Análisis de Imágenes (SNRFAI'97), Barcelona, España*, páginas 281-286, abril 1997.
- [Badenas *et al.*, 1997b] J. Badenas, M. Bober y F. Pla. Motion and Intensity-Based Segmentation and Its Application to Traffic Monitoring. En *Proceedings of the International Conference on Image Analysis and Processing (ICIAP'97), Florencia, Italia*, páginas 502-509, septiembre 1997.
- [Badenas y Pla, 1998a] J. Badenas y F. Pla. Applying Computer Vision Techniques to Traffic Monitoring. En *Proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA-98-AIE), Benicassim, España*, páginas 776-785, junio 1998.
- [Badenas y Pla, 1998b] J. Badenas y F. Pla. Segmentation Based on Region-Tracking in Image Sequences for Traffic Monitoring. En *Proceedings of the 14th International Conference on Pattern Recognition (ICPR'98), Brisbane, Australia*, páginas 999-1001, agosto 1998.
- [Badenas *et al.*, 1999] J. Badenas, J.M. Sanchiz y F. Pla. Tracking Regions using Temporal Integration for Traffic Monitoring. En *Proceedings of the VIII Symposium Nacional de Reconocimiento de Formas y Análisis de Imágenes (SNRFAI'99), Bilbao, España*, páginas 321-328, 1999.
- [Badenas *et al.*, 2000a] J. Badenas, J.M. Sanchiz y F. Pla. Seguimiento de Regiones Utilizando Integración Temporal y su Aplicación al Control de Tráfico, *Revista Electrónica de Visión por Computador (REVC)*, No.1, <http://www.cvc.uab.es/revc/revista/01/0104-abs.htm>, enero 2000.
- [Badenas *et al.*, 2000b] J. Badenas, J.M. Sanchiz y F. Pla. Using Temporal Integration for Tracking Regions in traffic Monitoring Sequences. En *Proceedings of the 15th International Conference on Pattern Recognition (ICPR'2000), Barcelona, España*, volumen 3, páginas 1137-1140, septiembre, 2000.

- [Badenas *et al.*, 2001a] J. Badenas, J.M. Sanchiz y F. Pla. Motion-based segmentation and region-tracking in image sequences. *Pattern Recognition*, 34(3):661-670, 2001.
- [Badenas *et al.*, 2001b] J. Badenas, M. Bober y F. Pla. Segmenting Traffic Scenes from Gray Level and Motion Information. *Pattern Analysis and Applications* (Aceptado en julio de 2000).
- [Badenas *et al.*, 2001c] J. Badenas, F. Pla y J.M. Sanchiz. A Computer Vision System for Real-Time Traffic Monitoring. Enviado al *IX Simposium Nacional de Reconocimiento de Formas y Análisis de Imágen*, Castellón, España, 2001.

Apéndice A

Operadores morfológicos

Las operaciones morfológicas realizan modificaciones sobre la forma de las regiones. Su acción cambia la pertenencia de los píxeles a una región atendiendo sólo a propiedades geométricas, por lo que su aplicación natural es sobre imágenes binarias, donde “1” indica pertenencia a la región, y “0” indica no pertenencia.

Existen diversos operadores morfológicos, sin embargo en este apéndice tan sólo veremos los más básicos, aquellos que afectan directamente a esta Tesis. Puede encontrarse una explicación más completa en la mayoría de libros generales sobre visión artificial [Low, 1991, Sonka *et al.*, 1993, Jain *et al.*, 1995].

A continuación veremos los siguientes cuatro operadores:

- Dilatación
- Erosión
- Apertura
- Cierre

Para poder expresar estas operaciones de una manera formal, debemos considerar la imagen como un espacio bidimensional E^2 . Cualquier punto de la imagen puede ser tratado como un vector con respecto al origen $(0, 0)$. Una región \mathbf{X} será un conjunto de puntos pertenecientes al espacio E^2 .

Una transformación morfológica viene dada por la relación entre el conjunto de puntos \mathbf{X} , y otro pequeño conjunto de puntos \mathbf{B} llamado el *elemento estructurante*. \mathbf{B} se expresa con respecto a un origen local \mathcal{O} , llamado *punto representativo*.

Aplicar un operador morfológico sobre una región \mathbf{X} significa mover el elemento \mathbf{B} sistemáticamente a través de todos los puntos de \mathbf{X} .

Dilatación La operación morfológica de *dilatación* \oplus combina dos conjuntos de puntos usando la adición de vectores. La dilatación $\mathbf{X} \oplus \mathbf{B}$ es el conjunto de puntos

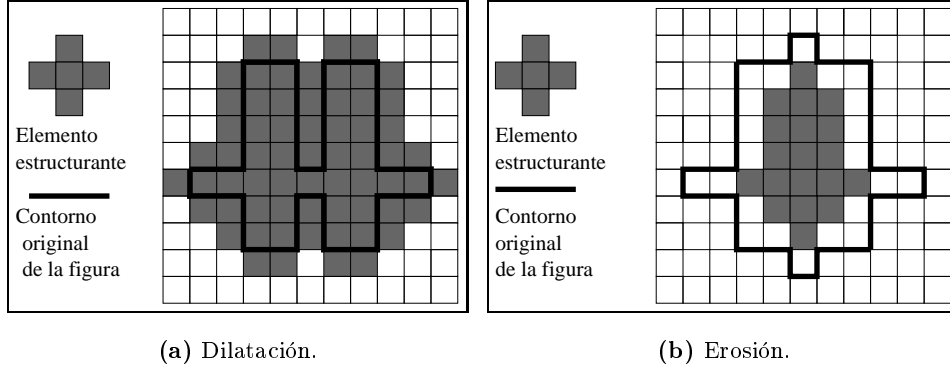


Figura A.1: Operaciones morfológicas de erosión y dilatación.

resultante de todas las posibles sumas de pares de elementos, uno de cada conjunto \mathbf{X} y \mathbf{B} .

$$\mathbf{X} \oplus \mathbf{B} = \{d \in E^2 / d = x + b, \forall x \in \mathbf{X} \text{ y } \forall b \in \mathbf{B}\} \quad (\text{A.1})$$

La subfigura A.1.a muestra el resultado de aplicar un elemento estructurante de cinco píxeles a una región. Dada la forma del elemento estructurante, el resultado es la adición a la región de aquellos píxeles vecinos situados al norte, sur, este y oeste de algún píxel de la frontera.

Erosión La operación morfológica de *erosión* \ominus es la operación recíproca de la dilatación. El resultado de la erosión son todos los puntos d para los cuales todas las posibles adiciones $d + b$ pertenecen a la región \mathbf{X} .

$$\mathbf{X} \ominus \mathbf{B} = \{d \in E^2 / d + b \in \mathbf{X}, \forall b \in \mathbf{B}\} \quad (\text{A.2})$$

La subfigura A.1.b muestra un ejemplo de la aplicación del mismo elemento estructurante sobre una región diferente. El resultado es la supresión de todos aquellos píxeles de la región que se hayan en una frontera, y que no tienen cuatro puntos (norte, sur, este y oeste) alrededor que también pertenezcan a la región.

Apertura La operación morfológica de *apertura* \circ consiste en una *erosión* seguida de una *dilatación*.

$$\mathbf{X} \circ \mathbf{B} = (\mathbf{X} \ominus \mathbf{B}) \oplus \mathbf{B} \quad (\text{A.3})$$

El efecto de una operación de *apertura* es el de eliminar pequeños grupos de píxeles y uniformizar el contorno de las regiones al suprimir salientes pequeños.

Cierre La operación morfológica de *cierre* \bullet es la operación inversa a la *apertura*. Consiste de una *dilatación* seguida por una *erosión*.

$$\mathbf{X} \bullet \mathbf{B} = (\mathbf{X} \oplus \mathbf{B}) \ominus \mathbf{B} \quad (\text{A.4})$$

Como consecuencia de esta operación se conectan regiones cercanas, se rellenan pequeños ahujeros y se uniformizan los contornos al rellenar los pequeños entrantes de los bordes. Por supuesto, tanto en el *cierre* como en la *apertura* los conceptos de pequeño y cercano están relacionados con el tamaño que pueda tener el elemento estructurante.

Apéndice B

El filtro de Kalman

B.1 Introducción

El filtro de Kalman fue propuesto por R.E. Kalman en 1960 [Kalman, 1960] como una solución recursiva al problema del filtrado lineal de datos discretos. En los últimos años, este filtro ha sido ampliamente investigado y aplicado a una gran diversidad de áreas científicas, entre ellas la visión por ordenador.

El filtro consiste en un conjunto de ecuaciones que proporcionan una solución computacionalmente eficiente del método mínimo-cuadrático. Su importancia radica en que permite realizar estimaciones de estados pasados, presentes y futuros, aún cuando la naturaleza del sistema modelado no se conoce con precisión.

Lo aquí expuesto tan solo pretende ser una sencilla introducción al filtro discreto de Kalman, la cual está principalmente basada en [Welch y Bishop, 1997]. En [Maybeck, 1979, Bar-Shalom y Fortman, 1988, Brown y Hwang, 1992, Jacobs, 1993] pueden encontrarse descripciones mucho más extensas y profundas acerca de este filtro.

El objetivo del filtro de Kalman es estimar el estado $\mathbf{x} \in R^n$ de un proceso controlado en tiempo discreto. Este proceso es gobernado por la ecuación lineal:

$$\mathbf{x}_{k+1} = \mathbf{A}_k x_k + \mathbf{w}_k, \quad (\text{B.1})$$

con una medición $\mathbf{z} \in R^m$, tal que

$$\mathbf{z}_k = \mathbf{H}_k x_k + \mathbf{v}_k \quad (\text{B.2})$$

En las anteriores dos ecuaciones, las variables \mathbf{w}_k y \mathbf{v}_k representan los ruidos del proceso y medición, respectivamente. Para ambos, se asume que son independientes entre sí, que corresponden a ruido blanco, es decir, que siguen distribuciones de probabilidad normal de media cero.

$$P(\mathbf{w}) \longrightarrow N(0, \mathbf{Q})$$

$$P(\mathbf{v}) \longrightarrow N(0, \mathbf{R})$$

La matriz \mathbf{A} $n \times n$ de la ecuación (B.1) relaciona el estado en el instante k con el estado en el instante $k + 1$. Por su parte, la matriz \mathbf{H} $m \times n$ de la ecuación (B.2) relaciona el estado en el instante k con la medición en el mismo instante.

Se define $\hat{\mathbf{x}}_{k|k-1}$ como la estimación a priori en el instante k dado el conocimiento previo a dicho instante, y $\hat{\mathbf{x}}$ como la estimación a posteriori del estado en el instante k dada la estimación \mathbf{z}_k . Podemos definir los errores de estimación a priori y a posteriori como

$$\mathbf{e}_{k|k-1} \equiv \mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}$$

$$\mathbf{e}_k \equiv \mathbf{x}_k - \hat{\mathbf{x}}_k$$

siendo las respectivas covarianzas de error:

$$\mathbf{P}_{k|k-1} = E[\mathbf{e}_{k|k-1} \mathbf{e}_{k|k-1}^T]$$

$$\mathbf{P}_k = E[\mathbf{e}_k \mathbf{e}_k^T]$$

La estimación del estado a posteriori es una combinación lineal de la estimación a priori $\hat{\mathbf{x}}_{k|k-1}$ y la diferencia entre la medición real \mathbf{z}_k y la predicción de la medición $\mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$.

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}) \quad (\text{B.3})$$

La diferencia $(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1})$ recibe el nombre de *residuo* y refleja la discrepancia entre la medición predicha y la medición real. La matriz \mathbf{K} $n \times n$ se llama la *ganancia de Kalman* y se calcula de forma que minimice la covarianza de error a posteriori \mathbf{P}_k .

$$\mathbf{K}_k = \frac{\mathbf{P}_{k|k-1} \mathbf{H}_k^T}{\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k} \quad (\text{B.4})$$

B.2 Algoritmo del filtro de Kalman

El filtro de Kalman lleva a cabo la estimación empleando una forma de control realimentado: se estima el estado en un instante dado, y se realimenta el proceso mediante las mediciones. Las ecuaciones del algoritmo pueden ser agrupadas en dos categorías: predicción y actualización. Las ecuaciones de predicción estiman el estado y la covarianza del error actuales a partir del estado y covarianza del instante previo.

Ecuaciones de predicción:

1. Proyectar el estado hacia adelante

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_{k-1} \hat{\mathbf{x}}_{k-1} \quad (\text{B.5})$$

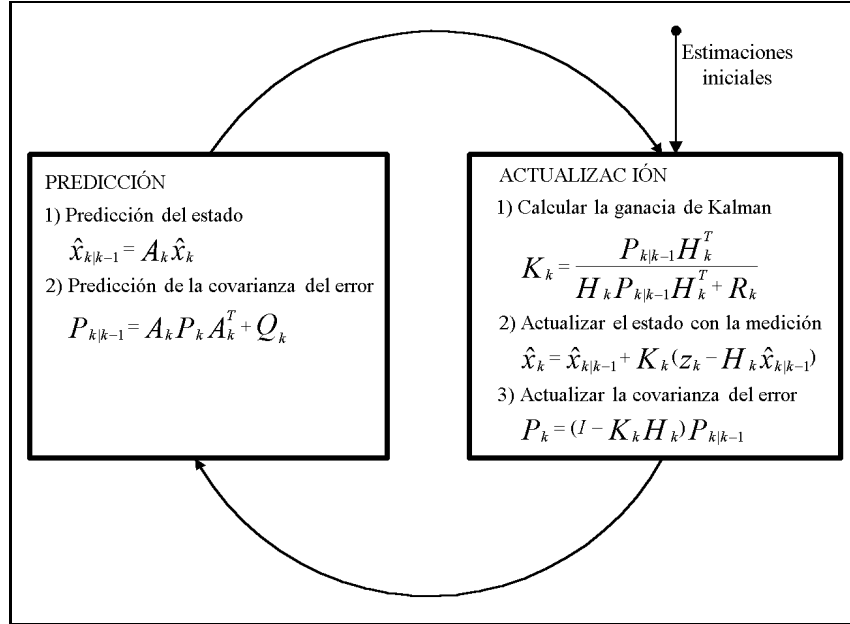


Figura B.1: Ciclo del algoritmo de Kalman.

2. Proyectar la covarianza del error hacia adelante

$$\mathbf{P}_{k|k-1} = \mathbf{A}_{k-1} \mathbf{P}_{k-1} \mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1} \quad (\text{B.6})$$

Por su parte, las ecuaciones de actualización son las que realizan la realimentación al incorporar las nuevas mediciones a las estimaciones a priori para mejorarlas obteniendo unas estimaciones a posteriori. Las estimaciones de actualización llevan a cabo una corrección en base a las mediciones que se van obteniendo.

Ecuaciones de actualización:

1. Cálculo de la ganancia de Kalman

$$\mathbf{K}_k = \frac{\mathbf{P}_{k|k-1} \mathbf{H}_k^T}{\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k} \quad (\text{B.7})$$

2. Actualización de la estimación mediante la medición z_k

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (z_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}) \quad (\text{B.8})$$

3. Actualización de la covarianza del error

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \quad (\text{B.9})$$

El algoritmo se inicializa con la introducción de las estimaciones iniciales de $\hat{\mathbf{x}}_{k|k-1}$ y $\mathbf{P}_{k|k-1}$, para así pasar a la fase de actualización. Una vez ésta ha sido completada, se inicia el proceso iterativo consistente en la predicción seguida de la actualización.

Bibliografía

- [Abdel-Malek *et al.*, 1990] A. Abdel-Malek, O. Hasekioglu, y J. Bloomer. Image segmentation via motion vector estimates. En *Medical imaging IV: Image Processing*, páginas 366–371. SPIE, 1990.
- [Aggarwal y Nandhakumar, 1988] J.K. Aggarwal y N. Nandhakumar. On the computation of motion from sequence of images - a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3:917–935, 1988.
- [Amini *et al.*, 1990] A.A. Amini, T.E. Weymouth, y R.C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):855–867, 1990.
- [Anandan, 1987] P. Anandan. *Measuring visual motion from image sequences*. PhD thesis, COINS Dept., Univ. of Massachusetts, Amherst, 1987.
- [Arsénio y Santos-Victor, 1997] A. Arsénio y J. Santos-Victor. Active monocular tracking with temporal integration of visual cues. En *9th Portuguese Conference on Pattern Recognition*, páginas 191–198, 1997.
- [Badenas *et al.*, 1997] J. Badenas, M. Bober, y F. Pla. Combining intensity and motion-based segmentation for robust traffic monitoring. En *Proceedings of the Conference on Pattern Recognition and Image Analysis, Barcelona, Spain*, páginas 281–286, Abril 1997.
- [Badenas, 1996] Jorge Badenas. Application of motion estimation and segmentation techniques to traffic monitoring. Master's thesis, Master on Multimedia Technology and Systems, University of Surrey, 1996.
- [Ballard y Brown, 1982] D.H. Ballard y C.M. Brown. *Computer Vision*. Prentice Hall INC, New Jersey, 1982.
- [Bar-Shalom *et al.*, 1990] Y. Bar-Shalom, K.C. Chang, y H.A. Blom. Automatic track formation in clutter with a recursive algorithm. En *Multitarget-Multisensor Tracking: Advanced Applications*, páginas 25–42, 1990.
- [Bar-Shalom y Fortman, 1988] Y. Bar-Shalom y T.E. Fortman. *Tracking and Data Association*. Academic Press, Inc., 1988.

- [Barnard y Thompson, 1980] S.T. Barnard y W.B. Thompson. Disparity analysis of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(4):333–340, 1980.
- [Barnard, 1979] S.T. Barnard. *The image correspondence problem*. PhD thesis, University of Minnesota, 1979.
- [Bartels *et al.*, 1987] R. Bartels, J. Beatty, y B. Barsky. *An introduction to splines for use in computer vision*. Morgan Kaufmann, 1987.
- [Battiti *et al.*, 1991] R. Battiti, E. Amaldi, y C. Koch. Computing optical flow across multiple scales. *International Journal of Computer Vision*, 6(2):133–145, 1991.
- [Beymer *et al.*, 1997] D. Beymer, P. McLauchlan, B. Coifman, y J. Malik. A real-time computer vision system for measuring traffic parameters. En *Computer Vision and Pattern Recognition*, páginas 495–501, 1997.
- [Bierling, 1988] M. Bierling. Displacement estimation by hierarchical block-matching. En *Visual Communication and Image Processing*, páginas 942–951, 1988.
- [Biswass y Pal, 2000] S. Biswass y N.R. Pal. On hierarchical segmentation for image compression. *Pattern Recognition Letters*, 21:131–144, 2000.
- [Black y Anandan, 1996] M.J. Black y P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, páginas 75–104, 1996.
- [Blake *et al.*, 1993] A. Blake, R. Curwen, y A. Zisserman. Affine-invariant contour tracking with automatic control of spatiotemporal scale. En *Proceedings of the Fourth International Conference on Computer Vision, Berlin, May 1993*, páginas 66–75, 1993.
- [Blake y Isard, 1998] A. Blake y M. Isard. *Active contours*. Springer-Verlag, 1998.
- [Bober y Kittler, 1993] M. Bober y J. Kittler. Estimation of complex multi-modal motion: An approach based on robust statistics and Hough transform. En *Proceedings of the British Machine Vision Conference*, páginas 239–248. BMVA Press, 1993.
- [Bober y Kittler, 1994a] M. Bober y J. Kittler. Estimation of general multimodal motion: an approach based on robust statistics and Hough transform. *Image and Vision Computing*, 12(12):661–668, 1994.
- [Bober y Kittler, 1994b] M. Bober y J. Kittler. Robust motion analysis. En *Proceedings, CVPR '94 (IEEE Computer Society Conference on Computer Vision and Pattern Recognition), Seattle, June 20-24, 1994*, páginas 947–952. IEEE Computer Society Press, 1994.

- [Borshukov *et al.*, 1997] G.D. Borshukov, G. Bozdagi, Y. Altunbasak, y A.M. Tekalp. Motion segmentation by multistage affine classification. *IEEE Transactions on Image Processing*, 6(11):1591–1594, 1997.
- [Bouthemy y Lalade, 1993] P. Bouthemy y P. Lalade. Recovery of moving objects masks in an image sequence using local spatiotemporal contextual information. *Optical Engineering*, 32(6):1205–1212, 1993.
- [Box, 1953] G. E. Box. Non-normality and tests on variance. *Biometrika*, 40:318–335, 1953.
- [Brice y Fennema, 1970] C.R. Brice y C.L. Fennema. Scene analysis using regions. *Artificial Intelligence*, 1:205–226, 1970.
- [Brock-Gunn *et al.*, 1994] S.A. Brock-Gunn, G.R. Dowling, y T.J. Ellis. Tracking using colour information. Technical report, Department of Computer Science, City University, London, 1994.
- [Broggi y Dickmanns, 2000] A. Broggi y E.D. Dickmanns. Applications of computer vision to intelligent vehicles. *Image and Vision Computing*, 18:365–366, 2000.
- [Brown y Hwang, 1992] R.G. Brown y P.Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. Wiley and Sons, Inc, second edition, 1992.
- [Canny, 1986] J.F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.
- [Castellow *et al.*, 1987] D.A. Castellow, D.W. Murray, G.L. Scott, y B.F. Buxton. Matching canny edgels to compute the principal components of optic flow. En *3rd Alvey Vision Conference*, páginas 193–200, Septiembre 1987.
- [Castro y Morandi, 1987] E. De Castro y C. Morandi. Registration of translated and rotated images using finite fourier transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):700–703, Septiembre 1987.
- [Chang y Aggarwal, 1991] Y.L. Chang y J.K. Aggarwal. 3d structure reconstruction from an ego motion sequence using statistical estimation and detection theory. En *IEEE Workshop in Visual Motion*, páginas 268–273, 1991.
- [Cho y Meer, 1997] K. Cho y P. Meer. Image segmentation from consensus information. *Computer Vision and Image Understanding*, 68(1):72–89, Octubre 1997.
- [Clarke *et al.*, 1996] J.C. Clarke, S. Carlsson, y A. Zisserman. Detecting and tracking linear features efficiently. En *7th British Machine Vision Conference*, 1996.
- [Cohen y Cohen, 1993] L.D. Cohen y I. Cohen. Finite-element methods for active contour models and balloons for 2-d and 3-d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, 1993.

- [Coleman y Andrews, 1979] G.B. Coleman y H.C. Andrews. Image segmentation by clustering. *IEEE*, 67(5):773–785, 1979.
- [Cootes *et al.*, 1995] T.F. Cootes, C.J. Taylor, D.H. Cooper, y J. Graham. Active shape models – their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.
- [Cox y Leonard, 1991] I.J. Cox y J.J. Leonard. Probabilistic data association for dynamic world modeling: a multiple hypothesis approach. En *International Conference on Advanced Robotics*, Pisa, Italy, 1991.
- [Cox, 1993] I.J. Cox. A review of statistical data association techniques for motion. *International Journal of Computer Vision*, 10(1):53–66, 1993.
- [Deriche y Faugeras, 1990] R. Deriche y O. Faugeras. Tracking line segments. *Image and Vision Computing*, 8(4):261–270, 1990.
- [Diehl, 1991] N. Diehl. Object-oriented motion estimation and segmentation in image sequences. *Signal Processing: Image Communication*, 3:23–56, 1991.
- [Donohoe *et al.*, 1988] G.W. Donohoe, D.R. Hush, y N. Ahmed. Change detection for target detection and classification in video sequences. En *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, páginas 1084–1087, New York, 1988.
- [Dubuisson *et al.*, 1996] M.P. Dubuisson, S. Lakshmanan, y A.K. Jain. Vehicle segmentation and classification using deformable templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(3):293–308, 1996.
- [Dubuisson y Jain, 1995] M.P. Dubuisson y A.K. Jain. Contour extraction of moving objects in complex outdoor scenes. *International Journal of Computer Vision*, 14:83–105, 1995.
- [Eghbali, 1979] H.J. Eghbali. K-s test for detecting changes from landsat imagery data. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):17–23, 1979.
- [Elder y Zucker, 1996] J.H. Elder y S.W. Zucker. Computing contour closure. En *4th European Conference on Computer Vision*, páginas 399–412, Cambridge, U.K., Abril 1996.
- [Enkelmann, 1988] W. Enkelmann. Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences. *Computer Vision, Graphics and Image Processing*, 43:150–177, 1988.
- [Erkam *et al.*, 1993] Y.M. Erkam, M.I. Sezan, y A.T. Erdem. A hierarchical phase-correlation method for motion estimation. En *Conference on Information Science and Systems*, páginas 419–424, Baltimore, Mar. 1993.

- [Fathy y Siyal, 1997] M. Fathy y M.Y. Siyal. Measuring traffic movements at junctions using image procesing techniques. *Pattern Recognition Letters*, 8:493–500, 1997.
- [Faugeras, 1993] O. Faugeras. *Three-Dimensional Computer Vision. A Geometric Viewpoint*. The MIT Press, 1993.
- [Feldman y Yakimovsky, 1974] J.A. Feldman y Y. Yakimovsky. Decision theory and artificial intelligence: I.a. semantics-based region analyser”. *Artificial Intelligence*, 5(4):349–371, 1974.
- [Fernyhough *et al.*, 1996] J.H. Fernyhough, A.G. Cohn, y D.C. Hogg. Generation of semantic regions from image sequences. En Jan-Olof Eklundh, editor, *7th European Conference on Computer Vision*, páginas 475–484, Cambridge, Abril 1996. Springer-Verlag.
- [Ferrier *et al.*, 1994] N.J. Ferrier, S.M. Rowe, y A. Blake. Real-time traffic monitoring. En *Proceedings, 2nd IEEE Workshop on Applications of Computer Vision*, páginas 81–87, 1994.
- [Ferryman *et al.*, 1995] J.M. Ferryman, A. D. Worrall, G.D. Sullivan, y K.D. Baker. A generic deformable model for vehicle recognition. En *Proceedings of the British Machine Vision Conference*, páginas 128–136. BMVA Press, 1995.
- [Fleet y Jepson, 1990] D.J. Fleet y A.D. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5:77–104, 1990.
- [Fleet, 1992] D.J. Fleet. *Measurement of Image Velocity*. Kluwer Academic Publishers, 1992.
- [Flusser, 1992] J. Flusser. An adaptative method for image registration. *Pattern Recognition*, 38:45–54, 1992.
- [Fortmann *et al.*, 1983] T.E. Fortmann, Y. Bar-Shalom, y M. Scheffe. Multi-target tracking using joint probabilistic data association. *IEEE Journal on Oceanic Engineering*, 8:173.184, 1983.
- [Frank *et al.*, 1996] T. Frank, M. Haag, H. Kollnig, y H-H. Nagel. Tracking of occluded vehicles in traffic scenes. En Jan-Olof Eklundh, editor, *7th European Conference on Computer Vision*, páginas 485–494, Cambridge, Abril 1996. Springer-Verlag.
- [Gamba *et al.*, 1997] P. Gamba, R. Lodola, y A. Mecoci. Scene interpretation by fusion of segment and region information. *Image and Vision Computing*, 15:499–509, 1997.
- [Gardner y Lawton, 1996] W.F. Gardner y D.T. Lawton. Interactive model-based vehicle tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:1115–1121, 1996.

- [Gelgon y Bouthemy, 2000] M. Gelgon y P. Bouthemy. A region-level motion-based graph representation and labeling for tracking a spatial image partition. *Pattern Recognition*, 33:725–740, 2000.
- [Gerard y Gagalowicz, 2000] P. Gerard y A. Gagalowicz. Three dimensional model-based tracking using texture learning and matching. *Pattern Recognition Letters*, 21:1095–1103, 2000.
- [Gharavi y Mills, 1990] H. Gharavi y M. Mills. Block-matching motion estimation algorithms: New results. *IEEE Transactions on Circuits and Systems*, 37:649–651, 1990.
- [Giachetti *et al.*, 1995] A. Giachetti, M. Cappello, y V. Torre. Dynamic segmentation of traffic scenes. En *IEEE International Vehicles Symposium*, páginas 258–263, 1995.
- [Giachetti, 2000] A. Giachetti. Matching techniques to compute image motion. *Image and Vision Computing*, 18(3):247–260, 2000.
- [Gibson, 1950] J.J. Gibson. *The perception of the visual world*. Houghton Mifflin, Boston, MA, 1950.
- [Gil *et al.*, 1994] S. Gil, R. Milanese, y T. Pun. Feature selection for object tracking in traffic scenes. En *International Symposium on Smart Highways*, Boston, Massachusetts, 1994. SPIE.
- [Gil *et al.*, 1996] S. Gil, R. Milanese, y T. Pun. Combining multiple estimates for vehicle tracking. En Jan-Olof Eklundh, editor, *7th European Conference on Computer Vision*, páginas 307–320, Cambridge, Abril 1996. Springer-Verlag.
- [Gil y Pun, 1993] S. Gil y T. Pun. Non-linear multiresolution relaxation for alerting. En *Proceedings of the European Conference on Circuit Theory and Design*, páginas 1639–1644, 1993.
- [Gokmen y Li, 1990] M. Gokmen y C.C. Li. Edge detection with iteratively refined regularization. En *10th International Conference on Pattern Recognition*, páginas 690–693, 1990.
- [Gong y Brady, 1990] S. Gong y M. Brady. Parallel computation of optic flow. En O. Faugeras, editor, *1st European Conference on Computer Vision*, páginas 124–133, 1990.
- [Gonzalez y Wintz, 1977] R.C. Gonzalez y P. Wintz. *Digital Image Processing*. Addison-Wesley, 1977.
- [Gross y Rosenfeld, 1987] A.D. Gross y A. Rosenfeld. Multiresolution object detection and delineation. *Computer Vision, Graphics and Image Processing*, 39:102–115, 1987.

- [Gupta y Sortrakul, 1998] L. Gupta y T. Sortrakul. A gaussian-mixture-based image segmentation algorithm. *Pattern Recognition*, 31(3):315–325, 1998.
- [Haag y Nagel, 1998] M. Haag y H-H. Nagel. Tracking of complex driving manoeuvres in traffic image sequences. *Image and Vision Computing*, 16:517–527, 1998.
- [Haag y Nagel, 1999] M. Haag y H-H. Nagel. Combination of edge element and optical flow estimates for 3d-model-based vehicle tracking in traffic image sequences. *International Journal of Computer Vision*, 35(3):295–319, 1999.
- [Haag y Nagel, 2000] M. Haag y H-H. Nagel. Incremental recognition of traffic situations from video image sequences. *Image and Vision Computing*, 18(2):137–153, 2000.
- [Hampel, 1968] F. R. Hampel. *Contributions to the theory of robust estimation*. PhD thesis, Univ. of California, Berkeley, 1968.
- [Haskell *et al.*, 1972] B. Haskell, F. Mounts, y J. Candy. Interframe coding of videotelephone pictures. En *Proceedings of the IEEE*, volumen 60, páginas 792–302, 1972.
- [Hassan, 1989] M.H. Hassan. A class of iterative thresholding algorithms for real-time image segmentation. En *Seventh Meeting of Intelligence Robots and Computer Vision*, páginas 182–193, 1989.
- [Haubecker y Spies, 1999] H. Haubecker y H. Spies. Motion. En Jahne *et al.* [1999], capítulo 13, páginas 309–396.
- [Heeger, 1988] D.J. Heeger. Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, 1:279–302, 1988.
- [Heitz y Bouthemy, 1993] F. Heitz y P. Bouthemy. Multimodal estimation of discontinuous optical flow using markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:1217–1232, 1993.
- [Herman y Liu, 1978] G.T. Herman y H.K. Liu. Dynamic boundary surface detection. *Computer Graphics and Image Processing*, 7:130–138, 1978.
- [Hildreth, 1984] E.C. Hildreth. Computations underlying the measurement of visual motion. *Artificial Intelligence*, 23:309–354, 1984.
- [Holden, 1993] M. Holden. *Parallel Algorithms for Digital Image Processing*. John Wiley and Sons, 1993.
- [Hong *et al.*, 1980] T.H. Hong, C.R. Dyer, y A. Rosenfeld. Texture primitive extraction using an edge-based approach. *IEEE Transactions on Systems, Man and Cybernetics*, 10(10):659–675, 1980.

- [Hong, 1982] T.H. Hong. Image smoothing and segmentation by multiresolution pixel linking further experiments. *IEEE Transactions on Systems, Man and Cybernetics*, 12(5):611–622, 1982.
- [Horn y Schunck, 1981] B.K.P. Horn y B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
- [Horn, 1986] B.K.P. Horn. *Robot Vision*. MIT Press, 1986.
- [Horowitz y Pavlidis, 1974] S.L. Horowitz y T. Pavlidis. Picture segmentation by a directed split-and-merge procedure. En *Second International Joint Conference on Pattern Recognition*, páginas 424–433, 1974.
- [Horowitz y Pavlidis, 1976] S.L. Horowitz y T. Pavlidis. Picture segmentation by a tree transversal algorithm. *Journal of the ACM*, 23:368–388, 1976.
- [Hotter y Thoma, 1988] M. Hotter y R. Thoma. Image segmentation based on object oriented mapping parameter estimation. *Signal Processing*, 15:315–334, 1988.
- [Hsu *et al.*, 1984] Y.Z. Hsu, H-H. Nagel, y G. Rekers. New likelihood test methods for change detection in image sequences. *Computer Vision, Graphics and Image Processing*, 26:73–106, 1984.
- [Huber, 1964] P.J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35:73–101, 1964.
- [Ip y Shen, 1998] H.H.S. Ip y D. Shen. An affine-invariant active contour model (ai-snake) for model-based segmentation. *Image and Vision Computing*, 16:135–146, 1998.
- [Irani *et al.*, 1992] M. Irani, B. Rousso, y S. Peleg. Detecting and tracking multiple moving objects using temporal integration. En G. Sandini, editor, *Proceedings, Second European Conference on Computer Vision, (Santa Margherita Ligure)*, páginas 282–287. Springer-Verlag, Mayo 1992.
- [Isard y Blake, 1996] M. Isard y A. Blake. Contour tracking by stochastic propagation of conditional density. En Jan-Olof Eklundh, editor, *7th European Conference on Computer Vision*, páginas 343–356, Cambridge, Abril 1996. Springer-Verlag.
- [Jacobs, 1993] O.L.R. Jacobs. *Introduction to Control Theory*. Oxford University Press, second edition, 1993.
- [Jahne *et al.*, 1999] B. Jahne, H. Haubecker, y P. Geibler, editores. *Handbook of computer vision and applications*. Academic Press, 1999.
- [Jain *et al.*, 1995] R. Jain, R. Kasturi, y B.G. Schunck. *Machine Vision*. Computer Science. McGraw-Hill International, 1995.

- [Jain *et al.*, 1996] A.K. Jain, Y. Zhong, y S. Lakshmanan. Object matching using deformable templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(3):267–278, 1996.
- [Jain y Jain, 1981] J.R. Jain y A.K. Jain. Displacement measurement and its application in interframe image coding. *IEEE Transactions on Communication*, 29:1799–1808, 1981.
- [Jain, 1985] R.C. Jain. Dynamic scene analysis. En *Progress in Pattern Recognition*, páginas 125–167, 1985.
- [Jain, 1989] A.K. Jain. *Fundamental of Digital Image Processing*. Information and System Sciences. Prentice Hall, Inc, New Jersey, 1989.
- [Jang *et al.*, 1997] D-S. Jang, G-Y. Kim, y H-I. Choi. Model-based tracking of moving object. *Pattern Recognition*, 30(6):999–1008, 1997.
- [Jang y Choi, 2000] D-S. Jang y H-I. Choi. Active models for tracking moving objects. *Pattern Recognition*, 33:1135–1146, 2000.
- [Jolly *et al.*, 1996] M.D. Jolly, S. Lakshmanan, y A.K. Jain. Vehicle segmentation and classification using deformable templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(3):292–308, 1996.
- [Kalman, 1960] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME- Journal of Basic Engineering*, páginas 35–45, Marzo 1960.
- [Karmann y von Brandt, 1990] K-P. Karmann y A. von Brandt. Moving object recognition using an adaptive background memory. En V. Cappellini, editor, *Time-Varying Image Processing and Moving Object Recognition*, volumen 2. Elsevier, Amsterdam, The Netherlands, 1990.
- [Kass *et al.*, 1988] M. Kass, A. Witkin, y D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.
- [Kervrann y Heitz, 94] C. Kervrann y F. Heitz. Robust tracking of stochastic deformable models in long image sequences. En *International Conference on Image Processing*, volumen 3, páginas 88–92, 94.
- [Kilger, 1992] M. Kilger. A shadow handler in a video-based real-time traffic monitoring system. En *IEEE Workshop on Applications of Computer Vision*, páginas 1060–1066, Palm Springs, CA, 1992.
- [Kittler *et al.*, 1987] J. Kittler, J. Eggleton, J. Illingworth, y K. Paler. An average edge detector. *Pattern Recognition Letters*, 6:27–32, 1987.
- [Kittler y Illingworth, 1986] J. Kittler y J. Illingworth. Minimum error thresholding. *Pattern Recognition*, 19:41–47, 1986.

- [Klausmann *et al.*, 1999] P. Klausmann, K. Kroschel, y D. Willersinn. Performance prediction of vehicle detection algorithms. *Pattern Recognition*, 32:2063–2065, 1999.
- [Koller *et al.*, 1993] D. Koller, K. Daniilidis, y H.H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10:257–281, 1993.
- [Koller *et al.*, 1994a] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, y S. Russel. Towards robust automatic traffic scenes analysis in real-time. En *12th International Conference on Pattern Recognition*, páginas 126–131, 1994.
- [Koller *et al.*, 1994b] D. Koller, J. Weber, y J. Malik. Robust multiple car tracking with occlusion reasoning. En Jan-Olof Eklundh, editor, *Proceedings, 5th European Conference on Computer Vision, Berlin*, páginas 189–196. Springer-Verlag, 1994.
- [Koller, 1993] D. Koller. Moving object recognition and classification based on recursive shape parameter estimation. En *12th Israel Conference on Artificial Intelligence, Computer Vision and Neural Networks*, páginas 359–368, Diciembre 1993.
- [Kottke y Sun, 1994] D.P. Kottke y Y. Sun. Motion estimation via cluster matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:1128–1132, 1994.
- [Lavagetto, 1990] F. Lavagetto. Infrared image segmentation through iterative thresholding. En *Real-Time Image Processing II*, páginas 29–38. The International Society for Optical Engineering, 1990.
- [Lee y Chen, 1997] C-H. Lee y L-H. Chen. A fast motion estimation algorithm based on the block sum pyramid. *IEEE Transactions on Image Processing*, 6(11):1587–1591, 1997.
- [Lef y Joshi, 1993] C-H. Lef y A. Joshi. Correspondence problem in image sequence analysis. *Pattern Recognition*, 26:47–61, 1993.
- [Leymarie y Levine, 1993] F. Leymarie y M.D. Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:617–634, 1993.
- [Li y Chakrabarti, 1996] H-L Li y C. Chakrabarti. Motion estimation of two-dimensional objects based on the straight line hough transform: a new approach. *Pattern Recognition*, 29(8):1245–1258, 1996.
- [Liu *et al.*, 1998] H. Liu, T-H. Hong, M. Herman, y R. Chellappa. Motion-model-based boundary extraction and a real-time implementation. *Computer Vision and Image Understanding*, 70(1):87–100, 1998.
- [Liu y Huang, 1991] Y. Liu y T.S. Huang. Determining straight line correspondences from intensity images. *Pattern Recognition*, 24(6):489–504, 1991.

- [Low, 1991] A. Low. *Introductory Computer vision and Image Processing*. McGraw-Hill, 1991.
- [Mae *et al.*, 1996] Y. Mae, Y. Shirai, J. Miura, y Y. Kuno. Object tracking in cluttered background based on optical flow and edges. En *13th International Conference on Pattern Recognition*, páginas 196–200, 1996.
- [Maravall, 1993] D. Maravall. *Reconocimiento de Formas y Visión Artificial*. Paradigma. RA-MA, 1993.
- [Mardia y Hainsworth, 1988] K.V. Mardia y T.J. Hainsworth. A spatial thresholding method for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:919–927, 1988.
- [Matthies y Kanade, 1989] L. Matthies y T. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3:209–236, 1989.
- [Maybank *et al.*, 1996] S.J. Maybank, A.D. Worrall, y G.D. Sullivan. A filter for visual tracking based on a stochastic model for driver behaviour. En Jan-Olof Eklundh, editor, *4th European Conference on Computer Vision*, páginas 540–549, Cambridge, U.K., April 1996. Springer-Verlag.
- [Maybeck, 1979] P.S. Maybeck. *Stochastic Models, Estimation and Control*. Academic Press, Inc, 1979.
- [McLauchlan *et al.*, 1992] P. McLauchlan, I. Reid, y D.W. Murray. Coarse image motion for saccade control. En *Proceedings of the 3rd British Machine Vision Conference*. BMVA Press, 1992.
- [McLauchlan y Malik, 1997] P. McLauchlan y J. Malik. Vision for longitudinal vehicle control. En *Proceedings of the British Machine Vision Conference*, páginas 260–269. BMVA Press, 1997.
- [Medioni y Nevatia, 1984] G. Medioni y R. Nevatia. Matching images using linear features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):675–685, 1984.
- [Mehrotra, 1998] R. Mehrotra. Establishing motion-based feature point correspondence. *Pattern Recognition*, 31(1):23–30, 1998.
- [Metaxas y Terzopoulos, 1993] D. Metaxas y D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:580–591, 1993.
- [Meyer y Bouthemy, 1994] F. Meyer y P. Bouthemy. Region-based tracking using affine motion models in long image sequences. *Computer Vision, Graphics and Image Processing*, 60(2):119–140, 1994.

- [Milgram, 1979] D.L. Milgram. Region extraction using convergent evidence. *Computer Graphics and Image Processing*, 11:1–12, 1979.
- [Mitiche y Bouthemy, 1996] A. Mitiche y P. Bouthemy. Computation and analysis of image motion: a synopsis of current problems and methods. *International Journal of Computer Vision*, 19(1):29–55, 1996.
- [Morefield, 1977] C.L. Morefield. Application of 0-1 integer programming to multitarget tracking problems. *IEEE Transactions on Automatic Control*, 22(6):302–312, 1977.
- [Muerle y Allen, 1968] J.L. Muerle y D.C. Allen. Experimental evaluation of techniques for automatic segmentation of objects in a complex scene. En G.C. Cheng, editor, *Pictorial Pattern Recognition*, páginas 3–13. Thompson, Washington, 1968.
- [Murray *et al.*, 1987] D.W. Murray, S.A. Castellow, y B.F. Buxton. From an image sequence to a recognized polyhedral object. En *3rd Alvey Vision Conference*, páginas 201–210, 1987.
- [Murray y Buxton, 1989] D.W. Murray y B.F. Buxton. Scene segmentation from visual motion using global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):200–228, 1989.
- [Nagel y Enkelmann, 1984] H.H. Nagel y W. Enkelmann. Towards the estimation of displacement vector fields by “oriented smoothness” constraints. En *Seventh International Conference on Pattern Recognition (Montreal, Canada, July 30-August 2, 1984)*, IEEE Publ. 84CH2046-1, páginas 6–8. IEEE, 1984.
- [Nagel y Enkelmann, 1986] H.H. Nagel y W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:565–593, 1986.
- [Nagel, 1981] H.H. Nagel. Image sequence analysis: What can we learn from applications? En T.S. Huang y S. Thomas, editores, *Image Sequence Analysis*, capítulo 2. Springer-Verlag, 1981.
- [Nagel, 1983a] H.H. Nagel. Overview on image sequence analysis. En T.S. Huang, editor, *Image Sequence Processing and Dynamic Scene Analysis*, volumen 2 de *NATO ASI Series, Computer and Systems Sciences*. Springer-Verlag, 1983.
- [Nagel, 1983b] H.H. Nagel. Displacement vectors derived from second order intensity variations in image sequences. *Computer Vision, Graphics and Image Processing*, 21:85–117, 1983.
- [Nagel, 1987] H.H. Nagel. On the estimation of optic flow: Relations between different approaches and some new results. *Artificial Intelligence*, 33:299–324, 1987.

- [Nagel, 1990] H.H. Nagel. Extending the “oriented smoothness constraint” into the temporal domain and the estimation of derivatives of optical flow. En O. Faugeras, editor, *Proceedings, First European Conference on Computer Vision, Antibes, France*, páginas 139–148. Springer-Verlag, Abril 1990.
- [Nakaya y Harashima, 1994] Y. Nakaya y H. Harashima. Motion compensation based on spatial transformations. *IEEE Transactions on Video Technology*, 4:339–356, 1994.
- [Nelson, 1991] R.C. Nelson. Qualitative detection of motion by a moving observer. En *Conference on Computer Vision and Pattern Recognition*, páginas 173–178, 1991.
- [Nesi, 1993] P. Nesi. Variational approach to optical flow estimation managing discontinuities. *Image and Vision Computing*, 11(7):419–439, Septiembre 1993.
- [Oh y Lee, 2000] H-S. Oh y H-K. Lee. Block-matching algorithm based on an adaptive reduction of the search area for motion estimation. *Real-Time Imaging*, 6:407–414, 2000.
- [Ong y Spann, 1995] E. Ong y M. Spann. Robust computation of optical flow. En *Proceedings of the British Machine Vision Conference*, páginas 573–582. BMVA Press, 1995.
- [Overington, 1987] I. Overington. Gradient-based flow segmentation and location of the focus of expansion. En *3rd Alvey Vision Conference*, páginas 167–177, 1987.
- [Pal y Pal, 1993] N.R. Pal y S.K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, 1993.
- [Park y Han, 1997] J.S. Park y J.H. Han. Estimating optical flow by tracking contours. *Pattern Recognition Letters*, 18:641–648, 1997.
- [Park y Han, 1998] J.S. Park y J.H. Han. Contour motion estimation from image sequences using curvature information. *Pattern Recognition*, 31(1):31–39, 1998.
- [Peterfreund, 1999] N. Peterfreund. Robust tracking of position and velocity with kalman snakes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(6):564–569, Junio 1999.
- [Pietikainen *et al.*, 1982] M. Pietikainen, A. Rosenfeld, y I. Walter. Split-and-link algorithms for image segmentation. *Pattern Recognition*, 15(4):287–298, 1982.
- [Pla y Bober, 1997] F. Pla y M. Bober. Estimating translation/deformation motion through phase correlation. En *Proceedings, International Conference on Image Analysis and Processing, Florence, Italy, September 1997*, páginas 653–660. IAPR, 1997.
- [Pratt, 1991] W.K. Pratt. *Digital Image Processing*. Wiley-Interscience. John Wiley and sons, inc, second edition, 1991.

- [Radeva *et al.*, 1995] P. Radeva, J. Serrat, y E. Marti. A snake for model-based segmentation. En *International Conference on Computer Analysis and Image Processing*, Praga, Czech Republic, 1995.
- [Radeva y Marti, 1995] P. Radeva y E. Marti. An improved model of snakes for model-based segmentation. En *International Conference on Computer Analysis and Image Processing*, Praga, Czech Republic, 1995.
- [Rao, 1992] B. Rao. Data association methods for tracking systems. En A. Blake y A. Yuille, editores, *Active Vision*, capítulo 6, páginas 91–105. MIT Press, 1992.
- [Reid, 1979] D.B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24:843–854, 1979.
- [Remagnino *et al.*, 1997] P. Remagnino, A. Baumberg, T. Grove, D. Hogg, T. Tan, A. Worral, y K. Baker. An integrated traffic and pedestrian model-based vision system. En *Proceedings of the British Machine Vision Conference*, páginas 380–389. BMVA Press, 1997.
- [Revol y Jourlin, 1997] C. Revol y M. Jourlin. A new minimum variance region growing algorithm for image segmentation. *Pattern Recognition Letters*, 18:249–258, 1997.
- [Roberts, 1977] L.G. Roberts. Machine perception of three-dimensional solids. En J.K. Aggarwal, R.O. Duda, y A. Rosenfeld, editores, *Computer Methods in Image Analysis*. IEEE Computer Society, Los Alamitos, California, 1977.
- [Robertson *et al.*, 1973] T.V. Robertson, P.H. Swain, y K.S. Fu. Multispectral image partitioning. Technical Report TR-EE-73-26, School of Electrical Engineering, Purdue University, August 1973.
- [Rosenfeld y Kak, 1982] A. Rosenfeld y A.C. Kak. *Digital Picture Processing*. Computer Science and Applied Mathematics. Academic Press, second edition, 1982.
- [Russ, 1995] J.C. Russ. *The Image Processing Handbook*. CRC Press, second edition, 1995.
- [Saber *et al.*, 1997] E. Saber, A.M. Tekalp, y G. Bozdagi. Fusion of color and edge information for improved segmentation and edge linking. *Image and Vision Computing*, 15:769–780, 1997.
- [Sahoo *et al.*, 1988] P.K. Sahoo, S. Soltani, A.K.C. Wong, y Y.C. Chen. Survey of thresholding techniques. *Computer Vision, Graphics and Image Processing*, 41(2):233–260, 1988.
- [Sarigianidis y Pycock, 1993] G.H. Sarigianidis y D. Pycock. Motion correspondence using a neural network. En *Proceedings of the British Machine Vision Conference*, páginas 649–658. BMVA Press, 1993.

- [Schunck, 1989] B.G. Schunck. Image flow segmentation and estimation by constraint line clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:1010 – 1027, 1989.
- [Scott, 1987] G.L. Scott. Four-line method of locally estimating optic flow. *Image and Vision Computing*, 5(2):67–72, 1987.
- [Seferedis y Ghanbari, 1993] V. Seferedis y M. Ghanbari. General approach to block-matching motion estimation. *Optical Engineering*, 32:1464–1474, 1993.
- [Segen y Pingali, 1996] J. Segen y S. Pingali. A camera-based system for tracking people in real time. En *Proceedings of the 13th International Conference on Pattern Recognition*, páginas 63–67, Viena, 1996. IEEE Computer Society Press.
- [Sethi y Jain, 1987] I.K. Sethi y R. Jain. Finding trajectories of feature points in a monocular image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):56–73, 1987.
- [Shapiro *et al.*, 1995] V. Shapiro, I. Backalov, y V. Kavardjikov. Motion analysis via interframe point correspondence establishment. *Image and Vision Computing*, 13(2):111–118, 1995.
- [Shapiro, 1995] L.S. Shapiro. *Affine Analysis of Image Sequences*. Cambridge University Press, 1995.
- [Singh, 1990] A. Singh. An estimation-theoretic framework for image-flow computation. En *International Conference on Computer Vision, Osaka*, páginas 168–177, Diciembre 1990.
- [Singh, 1992] A. Singh. *Optic Flow Computation: A Unified Perspective*. IEEE Computer Society Press, Los Alamitos, CA, 1992.
- [Siyal *et al.*, 2000] M.Y. Siyal, M. Fathi, y M. Atiquzzaman. A parallel pipeline based multiprocessor system for real-time measurement of road traffic parameters. *Real Time Imaging*, 6:241–249, 2000.
- [Smith y Brady, 1995] S.M. Smith y J.M. Brady. Asset-2– real-time motion segmentation and shape tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:814–820, 1995.
- [Smith y Buechler, 1975] P.L. Smith y G. Buechler. A branching algorithm for discriminating and tracking multiple objects. *IEEE Transactions on Automatic Control*, 20:101–104, 1975.
- [Smith, 1995] S.M. Smith. Asset-2– real-time motion segmentation and shape tracking. Technical Report TR95SNS2b, Defence Research Agency, UK, 1995.

- [Snyder y Rajala, 1983] W.E. Snyder y S.A. Rajala. Track acquisition of sub-pixel targets. En T.S. Huang, editor, *Image Sequence Processing and Dynamic Scene Analysis*, volumen 2 de *Computer and System Sciences*, páginas 502–507. Springer-Verlag, 1983.
- [Song *et al.*, 1990] S. Song, M. Liao, y J. Qin. Multiresolution image dynamic thresholding. *Machine Vision and Applications*, 3(1):13–16, 1990.
- [Sonka *et al.*, 1993] M. Sonka, V. Hlavac, y R. Boyle. *Image Processing, Analysis and Machine Vision*. Chapman & Hall Computing, 1993.
- [Stark y Ihle, 1997] K. Stark y T. Ihle. Visual tracking of solid objects based on an active contour model. En *Proceedings of the British Machine Vision Conference*, páginas 640–649. BMVA Press, 1997.
- [Steward *et al.*, 1994] B.D. Steward, I. Reading, M.S. Thomson, T.D. Binnie, K.W. Dickinson, y C.L. Wan. Adaptive lane finding in road traffic image analysis. En *Proceedings of the 7th International Conference on Real Traffic*, páginas 133–136, 1994.
- [Sudhir *et al.*, 1997] G. Sudhir, S. Banerjee, y A. Zisserman. Finding point correspondences in motion sequences preserving affine structure. *Computer Vision and Image Understanding*, 68(2):237–246, 1997.
- [Tabb y Ahuja, 1994] M. Tabb y N. Ahuja. A multiscale region-based approach to image matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, páginas 415 – 419, 1994.
- [Tan *et al.*, 1991] T.N. Tan, G.D. Sullivan, y K.D. Baker. A structure from constrained motion using point correspondences. En *Proceedings of the British Machine Vision Conference*, páginas 301–309. BMVA Press, 1991.
- [Tan *et al.*, 1994] T.N. Tan, G.D. Sullivan, y K.D. Baker. Fast vehicle localisation and recognition without line extraction and matching. En *Proceedings of the British Machine Vision Conference*, páginas 85–94. BMVA Press, 1994.
- [Tan *et al.*, 1998] T.N. Tan, G.D. Sullivan, y K.D. Baker. Model-based localisation and recognition of road vehicles. *International Journal of Computer Vision*, 27(1):5–25, 1998.
- [Teal y Ellis, 1996] M.K. Teal y T.J. Ellis. Spatial-temporal reasoning on object motion. En *Proceedings of the British Machine Vision Conference*, páginas 465–474. BMVA Press, 1996.
- [Tekalp, 1995] A.M. Tekalp. *Digital video processing*. Prentice Hall, 1995.
- [Terzopoulos y Szeliski, 1992] D. Terzopoulos y R. Szeliski. *Active Vision*. MIT Press, 1992.

- [Thévenaz, 1990] P. Thévenaz. Motion analysis. En *Pattern Recognition and Image Processing in Physics*, páginas 129–166. Scottish Universities Summer School in Physics, 1990.
- [Thompson y Pong, 1987] W.B. Thompson y T.C. Pong. Detecting moving objects. En *1st International Conference on Computer Vision*, páginas 201–208, 1987.
- [Torr y Murray, 1993] P.H.S. Torr y D.W. Murray. Statistical detection of independent movement from a moving camera. *Image and Vision Computing*, 11(4):180–187, 1993.
- [Tsai *et al.*, 1993] C.T. Tsai, T.N. Sun, y P.C. Chung. Minimizing the energy of active contour model using a hopfield network. *IEE Processings Part E: Computers and Digital Techniques*, 140(6):297–303, 1993.
- [Verri y Poggio, 1987] A. Verri y T. Poggio. Against quantitative optical flow. En *First International Conference on Computer Vision, (London, England, June 8- - 11, 1987)*, páginas 171–180, Washington, DC., 1987. IEEE Computer Society Press.
- [Wang y Adelson, 1994] J.Y.A. Wang y E.H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3:625–638, Septiembre 1994.
- [Wang y Brady, 1990] H. Wang y J.M. Brady. A fast algorithm for computing optic flow and its implementation on a transputer array. En *1st British Machine Vision Conference*, páginas 175–180, 1990.
- [Waterfall y Dickinson, 1984] R.C. Waterfall y K.W. Dickinson. Image procesing applied to traffic. *Traffic Engineering Control*, 25:60–67, 1984.
- [Welch y Bishop, 1995] G. Welch y G. Bishop. An introduction to the kalman filter. Technical Report TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3175, 1995.
- [Welch y Bishop, 1997] G. Welch y G. Bishop. An introduction to the kalman filter. <http://www.cs.unc.edu/welch/Kalman/Kalman.html>, 1997. Departement of Computer Science, University of North Carolina, Chapel Hill, NC 27599-3175.
- [Welch y Wood, 1993] P.H. Welch y D.C. Wood. Image tracking in real-time: a transputer emulation of some early mammalian vision processes. *Image and Vision Computing*, 11(4):221–228, Mayo 1993.
- [Weng *et al.*, 1993] J. Weng, T.S. Huang, y N. Ahuja. *Motion and Structure from Image Sequences*. Número 29 en Springer Series in Information Sciences. Springer-Verlag, 1993.
- [Wezka y Rosenfeld, 1979] J.S. Wezka y A. Rosenfeld. Histogram modification for thresholding selection. *IEEE Transactions on Sustems, Man and Cybernetics*, 9(1):38–52, 1979.

- [Wiklund y Granlund, 1987] J. Wiklund y G.H. Granlund. Image sequence analysis for object tracking. En *Proceedings of the 5th Scandinavian Conference on Image Analysis*, páginas 641–648, 1987.
- [Williams y Shah, 1992] D.J. Williams y M. Shah. A fast algorithm for active contours and curvature estimation. *Computer Vision, Graphics and Image Processing*, 55:14–26, 1992.
- [Wiskott, 1999] L. Wiskott. Segmentation from motion: Combining gabor- and mallat-wavelets to overcome aperture and correspondence problem. *Pattern Recognition*, 32, 1999.
- [Wixson, 1996] L. Wixson. Illumination assessment for vision-based traffic monitoring. En *13th International Conference on Pattern recognition*, páginas 56–62, Viena, Agosto 1996. IEEE Computer Society.
- [Worrall *et al.*, 1991] A.D. Worrall, R.F. Marslin, G.D. Sullivan, y K.D. Baker. Model-based tracking. En *Proceedings of the British Machine Vision Conference*, páginas 310–318. BMVA Press, 1991.
- [Worrall *et al.*, 1994] A.D. Worrall, R.F. Marslin, G.D. Sullivan, y K.D. Baker. Pose and structure recovery using active models. En *Proceedings of the British Machine Vision Conference*, páginas 137–146. BMVA Press, 1994.
- [Xu y Uberbacher, 1997] Y. Xu y E.C. Uberbacher. 2d image segmentation using minimum spanning trees. *Image and Vision Computing*, 15:47–57, 1997.
- [Young y Chellappa, 1990] G.S.J. Young y R. Chellappa. 3-d motion estimation using sequence of noisy stereo images: Models, estimation, and uniqueness results. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:735–759, 1990.
- [Zhang y Faugeras, 1992] Z. Zhang y O.D. Faugeras. Three-dimensional motion computation and object segmentation in a long sequence of stereo frames. *International Journal of Computer Vision*, 7(3):211–241, 1992.
- [Zhang, 1992] Z. Zhang. On local matching of free-form curves. En *British Machine Vision Conference*, páginas 347–356, 1992.
- [Zhang, 1997] Y.J. Zhang. Evaluation and comparison of different segmentation algorithms. *Pattern Recognition Letters*, 18:963–974, 1997.
- [Zheng y Chellappa, 1995] Q. Zheng y R. Chellappa. Automatic feature point extraction and tracking in image sequences for arbitrary camera motion. *International Journal of Computer Vision*, 15:31–76, 1995.
- [Zhou *et al.*, 1989] Y.T. Zhou, V. Venkateswar, y R. Chellappa. Edge detection and linear feature extraction using 2-d random field model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:84–95, 1989.

-
- [Zhu *et al.*, 2000] Z. Zhu, G. Xu, B. Yang, y X. Lin. VISITRAM: a real-time vision system for automatic traffic monitoring. *Image and Vision Computing*, 18:781–794, 2000.

Índice de Materias

A

accidente, 124
adaptación del fondo, 59
alerta, 122–124
análisis de movimiento
 aplicaciones, 2–3
 métodos, 44
 problemas, 1–2
apertura, 60, 151, 152
 problema, 48
asociación de datos, 90
atasco, 119

B

búsqueda por descenso, 55, 65
bloques, 52
bordes, 22

C

características
 métodos, 50–54
carril, 119
 detección automática, 60, 119
cierre, 60, 151, 153
cinemática, 90
clustering, 24–25
codificación, 3
correlación de fase, 51
correspondencia, 88
correspondencia de bloques, 52
correspondencia de puntos, 50
crecimiento de regiones, 22

D

descenso más profundo, 67
detección de carriles, 46
detección de movimiento, 2, 45, 57–61

diferencia de la imagen desplazada, 55,
 64, 65
diferencias de imágenes, 55
dilatación, 63, 151
división de regiones, 23
división y crecimiento de regiones, 23

E

elemento estructurante, 60, 151, 152
erosión, 151, 152
estimación, 155
estimación de movimiento, 64–69
estimadores por lotes, 91
estimadores recursivos, 91

F

facetas, 92
filtro de Kalman, 99–102, 155–157
flujo óptico, 2, 89
 ecuación, 48
 restricciones, 48

G

Gabor, filtro de, 51
gradiente
 métodos, 47–50
grafo de adyacencia, 72

H

histograma, 21

I

iluminación
 cambios de, 60
imagen de referencia, 60
imagen referencia
 actualización, 58

- integración temporal, 17, 88
integración temporal morfológica, 87–89, 128–129
- K**
k-medias, 24
Kalman, 93, 99–102, 109, 115, 155–157
- L**
lista maestra, 102–108
- M**
métodos basados en facetas, 92
memoria temporal, 12, 17, 102
meteorología, 3
modelos bidimensionales, 92
modelos de contornos activos, 92
modelos de movimiento, 90
modelos deformables, 92
modelos tridimensionales, 91
movimiento
 detección, 57–61
 análisis, 43
 see análisis de movimiento, 3
 estimación, 2, 64, 69
 métodos basados en características, 50–54, 148
 métodos basados en el gradiente, 47–50
 métodos basados en regiones, 53–54
 precisión subpíxel, 67
 segmentación, 2, 70–72
multiresolución, 67
- O**
oclusiones, 49
operaciones morfológicas, 60, 151–153
outliers, 50, 69
- P**
percepción de movimiento, 1
predicción de estados, 156
profundidad, 89
- R**
red vial, 118
- reestimación de movimiento, 72
región, 53, 93
 perdida, 107
 permanente, 106–107, 122
 reducción del número de regiones, 61–64
robótica, 3
robusto, 67
 estimador, 68
- S**
sección crítica, 126
segmentación, 19–29
 métodos, 21–25
 basados en bordes, 22
 basados en regiones, 22–25
 basados en umbralización, 21
 tipos de métodos, 21
segmentación de movimiento, 70–72
segmentación final, 108
segmentación virtual, 128
seguimiento, 2, 87, 89–97
semáforo, 126
snake, 90, 92
split-and-merge, 23
subpíxel, 67
substracción de imágenes, 60, 63
supercuadráticas, 90
- T**
tabla de vecindad, 72
thresholding, 21
tráfico
 Centro de Control de Tráfico, 6
 monitorización, 6, 118
 tareas, 15, 118–124
tracking, 87
 see seguimiento, 2
trayectoria, 121
- U**
umbralización, 21
- V**
vehículo parado, 122

velocidad de vehículos individuales, 13

velocidad media del tráfico, 13

volumen de tráfico, 119