

20/2002-03

**UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA  
UNIDAD DE TERCER CICLO Y POSTGRADO**

Reunido el día de la fecha, el Tribunal nombrado por el Excmo. Sr. Rector Magfco. de esta Universidad, el/a aspirante expuso esta TESIS DOCTORAL.

Terminada la lectura y contestadas por el/a Doctorando/a las objeciones formuladas por los señores miembros del Tribunal, éste calificó dicho trabajo con la nota de SOBRESALIENTE

" CUM LAUDE "

Las Palmas de Gran Canaria, a 3 de diciembre de 2002.

**El/a Presidente/a: Dr.D. Juan A. Méndez Rodríguez,**

**El/a Secretario/a: Dr.D. Antonio Falcón Martel,**

**El/a Vocal: Dr.D. José Andrés Moreno Pérez,**

**El/a Vocal: Dr.D. Casiano Rodríguez León,**

**El/a Vocal: Dr.D. Rafael Alejandro Montenegro Armas,**

**El Doctorando: D. Cayetano Nicolás Guerra Artal,**



**Universidad de Las Palmas de Gran Canaria**

DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS



**TESIS DOCTORAL**

**CONTRIBUCIONES AL SEGUIMIENTO VISUAL  
PRECATEGÓRICO**

**CAYETANO GUERRA ARTAL**

LAS PALMAS DE GRAN CANARIA

OCTUBRE 2002



# Universidad de Las Palmas de Gran Canaria

DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS



TESIS DOCTORAL TITULADA **CONTRIBUCIONES AL SEGUIMIENTO VISUAL PRECATEGÓRICO**, QUE PRESENTA D. CAYETANO GUERRA ARTAL, REALIZADA BAJO LA DIRECCIÓN DEL DOCTOR D. FRANCISCO MARIO HERNÁNDEZ TEJERA Y LA CODIRECCIÓN DEL DOCTOR D. JORGE CABRERA GÁMEZ

Las Palmas de Gran Canaria, octubre 2002

El doctorando

El director

El codirector

Cayetano Guerra Artal

Francisco Mario Hernández  
Tejera

Jorge Cabrera Gámez



## **Agradecimientos**

Quisiera expresar mi agradecimiento tanto al director de esta tesis, Francisco Mario Hernández Tejera, como al codirector, Jorge Cabrera Gámez, por el apoyo y los consejos recibidos para la elaboración de la misma. Desearía, además, agradecer especialmente la ayuda de Juan Méndez Rodríguez.

También quisiera agradecer a todos los miembros de la “División de Inteligencia Artificial y Sistemas” del “Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería” sus consejos y ayuda. No quisiera dejar de agradecer el apoyo recibido siempre de Roberto Esper-Chaín.

Esta memoria está dedicada a mis padres y, muy especialmente, a Beatriz, quien conoce bien el trabajo que ha supuesto su realización.

Gracias a todos.





*A mis padres*

*y*

*a Beatriz*



# Índice general

<b>1. Introducción</b>	<b>5</b>
1.1. Estructuración del documento . . . . .	8
<b>2. Seguimiento mediante aprendizaje basado en contexto</b>	<b>11</b>
2.1. Representaciones isomórficas de segundo orden . . . . .	14
2.2. Aprendizaje incremental mediante discriminación basada en contexto . . .	17
2.3. Aplicabilidad. Problemas y contextos . . . . .	21
2.4. Particularización de la búsqueda incremental basada en contexto . . . . .	23
2.4.1. Memorización de los patrones aprendidos. Banco de patrones . . . .	28
2.4.2. Algunas implicaciones del seguimiento con aprendizaje basado en contexto . . . . .	30
<b>3. Búsqueda de patrones</b>	<b>33</b>
3.1. Búsqueda basada en áreas . . . . .	34
3.2. Medidas de similitud . . . . .	35
3.2.1. Medidas de correlación . . . . .	37
3.2.2. Medidas de distancia de intensidad . . . . .	38
3.2.3. Medidas de detección secuencial de similitud . . . . .	41
3.2.4. Criterios de cambio de signo . . . . .	42
3.3. Actualización del patrón . . . . .	43
3.3.1. Mantenimiento del patrón constante . . . . .	43
3.3.2. Actualización constante . . . . .	43
3.3.3. Actualización de periodo constante . . . . .	45
3.3.4. Actualización mediante umbral constante . . . . .	45
3.3.5. Actualización basada en estadísticos . . . . .	46
3.3.6. Actualización basada en contexto . . . . .	46
3.4. Adecuación del patrón . . . . .	47

3.4.1.	Normalización de la iluminación . . . . .	47
3.4.2.	Actualización ponderada . . . . .	47
3.4.3.	Modificación mediante transformaciones afines . . . . .	48
3.4.4.	Patrones deformables . . . . .	48
3.4.5.	Búsqueda de múltiples patrones . . . . .	48
3.5.	Estrategias de búsqueda . . . . .	48
3.5.1.	Barrido espiral . . . . .	49
3.5.2.	Líneas epipolares . . . . .	49
3.5.3.	Filtros estimadores . . . . .	49
3.5.4.	Actualización del ganador, "Winner-Update" . . . . .	51
3.5.5.	Jerarquías . . . . .	51
3.5.6.	Transformación log-polar . . . . .	52
3.6.	Resumen de la clasificación . . . . .	54
3.7.	Búsqueda basada en características . . . . .	56
3.8.	Actualización basada en contexto . . . . .	60
3.8.1.	La necesidad de tomar en cuenta el entorno . . . . .	60
3.8.2.	Cómo tener en cuenta el entorno . . . . .	65
3.8.3.	Banco de patrones . . . . .	70
3.8.4.	Pérdida del objeto . . . . .	77
<b>4.</b>	<b>Sobre la arquitectura de un módulo genérico de seguimiento precate- górico</b> . . . . .	<b>79</b>
4.1.	Resultados del seguimiento . . . . .	80
4.2.	Estructura interna del módulo de seguimiento . . . . .	82
4.2.1.	Métodos de búsqueda . . . . .	83
4.2.2.	Bloque de control . . . . .	87
4.2.3.	Bloque de combinación . . . . .	88
4.2.4.	Información de la imagen y variables de cámara . . . . .	91
4.2.5.	Ciclo de trabajo . . . . .	91
4.3.	Capacidades del módulo de seguimiento . . . . .	92
4.3.1.	Comandos del módulo de seguimiento . . . . .	92
4.3.2.	Seguimiento de objetos . . . . .	93
4.3.3.	Seguimiento de múltiples objetos . . . . .	94
4.3.4.	Combinación de módulos de seguimiento . . . . .	97

4.3.5.	Adaptación del módulo de seguimiento a sistemas de visión activa	102
4.3.6.	Adaptación al estéreo	102
4.3.7.	Pérdida del objeto	103
4.3.8.	Recuperación	104
4.4.	Hardware de soporte	105
4.4.1.	Consideraciones sobre la integración	106
<b>5.</b>	<b>Evaluación experimental y aplicaciones</b>	<b>109</b>
5.1.	Caracterización del dispositivo de adquisición de imágenes	112
5.1.1.	Cálculo del nivel de ruido	112
5.2.	Actualización basada en contexto vs. patrón constante	116
5.3.	Actualización basada en contexto vs. actualización constante	116
5.4.	Actualización basada en contexto vs. actualización de periodo constante	118
5.5.	Actualización basada en contexto vs. actualización umbral constante	118
5.5.1.	Experimento 1	119
5.5.2.	Experimento 2	119
5.6.	Actualización basada en contexto vs. actualización basada en estadísticos	123
5.6.1.	Experimento 1	125
5.6.2.	Experimento 2	128
5.6.3.	Experimento 3	128
5.7.	Conclusiones	131
5.8.	Utilización de bancos de patrones	133
5.8.1.	Experimento 1	135
5.8.2.	Experimento 2	136
5.8.3.	Experimento 3	139
5.8.4.	Experimento 4	141
5.8.5.	Conclusiones	144
5.9.	Detección y seguimiento de objetos: DESEO	147
5.9.1.	Consideraciones previas para el diseño de SVC	147
5.9.2.	CAV	149
5.9.3.	Integración de módulos	156
5.9.4.	Conclusiones	160
<b>6.</b>	<b>Principales aportaciones, conclusiones y trabajo futuro</b>	<b>163</b>

6.1. Principales aportaciones . . . . .	163
6.2. Conclusiones . . . . .	165
6.3. Trabajo futuro . . . . .	167
<b>A. Tecnologías hardware de soporte</b>	<b>169</b>
A.1. ASIC . . . . .	170
A.2. FPGA . . . . .	171
A.3. Sistemas basados en DSP . . . . .	173
A.3.1. Arquitectura del TMS320C80 . . . . .	174
A.3.2. Arquitectura de la placa de procesamiento Precision MX Video Engine . . . . .	177
A.4. Procesadores de propósito general . . . . .	179
A.4.1. Familia Intel Pentium . . . . .	180
<b>B. Algoritmo de búsqueda sobre una arquitectura DSP multiprocesadora con memoria compartida</b>	<b>187</b>
<b>C. Algoritmo de suma de diferencias absolutas para el Intel Pentium III</b>	<b>193</b>
<b>D. Comandos del módulo de seguimiento</b>	<b>199</b>
<b>E. Soporte Experimental: WinVision</b>	<b>203</b>
E.0.2. Microsoft DirectShow . . . . .	203
E.0.3. Filtro WinVision . . . . .	205

# Capítulo 1

## Introducción

La presente tesis plantea como hipótesis la posibilidad de diseñar y construir un módulo de seguimiento visual versátil, autónomo y precategórico. La versatilidad del módulo pretende que éste sea fácilmente integrable dentro de sistemas de visión más complejos o de propósito más general, que hagan uso del módulo para la tarea de seguimiento. El módulo de seguimiento ofrecerá al sistema que lo integra el mantenimiento de un cierto objeto de interés dentro de su campo visual a lo largo de un determinado periodo de tiempo. Esto permitirá al sistema realizar algún tipo de procesamiento posterior sobre este objeto de interés.

Se entiende por autonomía del módulo de seguimiento al funcionamiento del mismo sin ningún tipo de ajuste manual o empírico previo. El módulo de seguimiento debe ser capaz por sí mismo de adaptarse a las condiciones visuales y de contexto en su entorno de trabajo.

El carácter precategórico del módulo es necesario debido a que debe permitirse el seguimiento de cualquier tipo de objeto, con lo que no es posible mediante sistemas de bajo nivel establecer categorías ni realizar procesos de identificación de objetos.

El planteamiento del desarrollo de un módulo de seguimiento tiene como origen las muchas aplicaciones en las que la tarea de seguimiento visual supone una parte importante de su solución. Por tanto, consideramos que estas aplicaciones pueden llevarse a cabo o ser mejoradas mediante la utilización de módulos como el expuesto en este trabajo, ofreciendo funciones de seguimiento independientemente de su finalidad. Dado que entre los problemas observados dentro de los sistemas de visión por computador en general se encuentran la falta de modularización y reutilización de los diferentes subsistemas que los conforman, el módulo de seguimiento desarrollado pretende ser una contribución para superar esta limitación ya que posee condiciones esenciales para posibilitar su

rápida integración y fácil manejo.

La consecución de módulos de seguimiento autónomos y reubicables pasa por el cumplimiento de dos condiciones indispensables. En primer lugar, los métodos de búsqueda empleados para la realización del seguimiento deben caracterizarse por ser en si mismos adaptables al objeto de interés. En esta tesis se propone una teoría de aprendizaje incremental basado en contexto que, aplicada sobre diversos métodos de búsqueda, permiten a estos llevar a cabo tal adaptabilidad de forma automática. La segunda condición consiste en la definición de una arquitectura interna del módulo y de una interfaz de manejo del módulo, válida para poder llevar a cabo cualquier integración del mismo de manera rápida y fácil. Las aportaciones teóricas y prácticas de la presente tesis ofrecen soluciones para conseguirlo.

Un módulo de seguimiento con las características descritas puede encontrarse empotrado en el marco de muy diversos contextos de aplicación, de los que aquí se enumeran tan sólo algunos de ellos a modo de muestra:

- **Navegación.** De la misma manera en que las personas utilizan referencias visuales para caminar o rodear objetos, un robot puede hacer uso de ellas para obtener de la misma manera su posición y orientación relativa a ellas. Con un sistema de seguimiento multiobjetivo, es posible no sólo usar una simple marca para realizar la tarea sino varias marcas para obtener una mejora en la fiabilidad y hacer más sencilla la obtención de la posición y orientación. Balkenius y Kopp (Balkenius y Kopp, 1997) utilizan un esquema de búsqueda elástica de patrones donde cada patrón está codificado con un número de características junto con su relaciones espaciales, de esta forma la localización de una característica restringe la búsqueda de la siguiente característica. Murray et al. (Murray et al., 1996) proponen un conjunto de reglas basadas en la fijación para determinar el movimiento de un robot sin ningún tipo de representación de su entorno.
- **Lectura.** Seguir algo no significa necesariamente seguir un determinado objeto y no variar el punto de fijación sobre éste. Cuando la fijación recae sobre algunas letras o una palabra de una línea de un texto, estas pueden ser seguidas mientras existe un movimiento relativo entre el soporte que las contiene y el observador. En este tipo de aplicación, el módulo de seguimiento mantiene dentro del campo visual el conjunto de letras seguido, mientras otro módulo categórico tipo OCR realiza su reconocimiento. Posteriormente, y a diferencia de un sistema de fijación estática, la fijación se desplaza ligeramente a la derecha hasta obtener el siguiente conjunto de caracteres, donde el proceso comienza de nuevo.



- **Reconocimiento de caras.** Esta ha sido una de las aplicaciones seminales de esta tesis y fuente de una gran parte del trabajo empírico. El reconocimiento de caras es un problema complejo del que existen numerosos trabajos en la bibliografía que tratan de identificar personas en entornos controlados y con vistas frontales de las caras. El problema se hace más difícil cuando la persona a reconocer no se encuentra inmóvil sino que se mueve de forma natural frente a las cámaras. Stillman et al. (Stillman et al., 1998) utilizan un sistema con múltiples cámaras y basan su módulo de reconocimiento en el *FaceIt Developer Kit 2.0* de *Visionics Corporation*. Hernández et al. (Hernández et al., 1999) desarrollaron DESEO, una plataforma de visión activa capaz de realizar el seguimiento de personas con objeto de posibilitar su posterior reconocimiento. Este sistema hace uso de los trabajos de esta tesis y se encuentra ampliamente descrito en el Capítulo 5.

Realizar el seguimiento de caras puede tener otras utilidades, como puede ser la lectura de labios (Loy et al., 2000) o el seguimiento y extracción de gestos.

- **Realimentación visual para brazos robot.** Controlar brazos robot por medio de seguimiento visual es una técnica útil para interactuar con entornos dinámicos (Allen et al., 1994). Un resumen sobre control visual puede encontrarse en (Hutchinson et al., 1996). Kragić y Christensen (Kragić y Christensen, 2000) presentaron un sistema que no depende de marcas para realizar el seguimiento sobre dispositivos mecánicos sino que usan una aproximación basada en regiones implementando básicamente dos métodos: Flujo óptico combinado con una búsqueda basada en SSD (suma de diferencias cuadradas). Esta aproximación permite no depender de ningún tipo especial de características.
- **Vigilancia.** Esta aplicación ha sido de los primeros usos prácticos de algoritmos de seguimiento y aún mantiene su interés por su alta utilidad, (Collins et al., 1999). Normalmente, los sistemas de vigilancia típicos constan de cámaras estáticas con módulos de detección que activan la grabación de secuencias. Sistemas algo más sofisticados con cámaras motorizadas incorporan módulos de detección que realizan la misma función que las anteriores y además ejecutan un proceso de seguimiento comandando los motores de las cámaras.
- **Seguimiento del movimiento de personas.** En este tipo de aplicaciones no se sigue a la persona en si sino a una parte de su cuerpo. Esto permite, mediante la interpretación de gestos del cuerpo y las manos, su utilización para el control de diferentes elementos sin la presencia física de ningún dispositivo de interfaz mecánico, (Darrell et al., 1996). En (O'Hagan y Zelinsky, 1997; Hall y Crowley,

1999) el seguimiento de los dedos de las manos se utiliza como ratón o teclado para interactuar con un ordenador. Otros trabajos realizan el seguimiento múltiple de diferentes partes de los labios para detectar fonemas (Stiefelhagen et al., 1997; Loy et al., 2001).

- **Cotrol del tráfico.** La monitorización de vehículos en carreteras y autopistas es otra de las aplicaciones prácticas del seguimiento, (Barattin et al., 1998). Con ella se pueden detectar colisiones y maniobras peligrosas. Normalmente, los sistemas dedicados a estas tareas se aprovechan de fondos o entornos inmóviles para llevar a cabo una fácil detección.

## 1.1. Estructuración del documento

Esta sección describe brevemente los diferentes capítulos en los que se divide esta tesis:

*Capítulo 2: Seguimiento mediante aprendizaje basado en contexto.* Constituye el núcleo fundamental y principal aportación teórica de esta tesis. Se describe el concepto de *aprendizaje incremental basado en contexto* mediante el uso de espacios de representación isomórfica de segundo orden. Este concepto destaca la importancia del entorno en la localización de objetos en escenas, permitiendo la automatización de diversos aspectos de los diferentes algoritmos de búsqueda convencionales.

*Capítulo 3: Búsqueda de patrones.* Se compone de una primera parte dedicada a la revisión y clasificación de las diferentes técnicas de búsqueda basadas en correspondencias existentes para la localización de patrones sobre imágenes. Posteriormente, una segunda parte describe de manera algorítmica y metodológica las aportaciones de esta tesis destinadas a la mejora en la robustez y automatización de los procesos de búsqueda. Para ello, se hace uso de las ideas contenidas en el Capítulo 2, demostrando cómo el seguimiento basado en contexto mejora de manera sustancial el rendimiento y robustez de los métodos de búsqueda.

*Capítulo 4: Sobre la arquitectura de un módulo genérico de seguimiento precategórico.* La obtención de módulos de seguimiento robustos pasa por la utilización de diversos métodos de búsqueda concurrentes cuya unión de resultados ofrezca mayores garantías al seguimiento. Este capítulo describe, de manera teórica, los componentes y la estructura

interna del módulo de seguimiento autónomo propuesto, que hace uso de la fusión de los resultados de diversos métodos de búsqueda para llevar a cabo su tarea. Otra de las cualidades descritas de este módulo se encuentra en su cómoda reutilización e integración en cualquier sistema que necesite realizar el seguimiento de objetos para llevar a cabo su tarea.

*Capítulo 5: Evaluación experimental y aplicaciones.* Se detallan los experimentos realizados durante la elaboración de esta tesis. Todos ellos demuestran las mejoras en los resultados introducidas mediante las aportaciones metodológicas descritas en los capítulos anteriores. Así mismo, se describe la aplicación DESEO como utilización práctica del módulo de seguimiento desarrollado.

*Apéndice A: Tecnologías hardware de soporte.* Se enumeran en este apéndice las diferentes tecnologías hardware disponibles en la actualidad para la construcción de sistemas de visión económicamente viables. Dichas tecnologías se clasifican aquí según un criterio de rendimiento vs. utilidad. Cuanto mayor es el rendimiento de una arquitectura específica menor es la flexibilidad que permite su reutilización en otras aplicaciones distintas y viceversa. De estas tecnologías se presta una mayor atención a las utilizadas en el trabajo experimental de esta tesis, es decir, a los sistemas DSP y microprocesadores de propósito general. Dentro de los DSP se incluye una descripción detallada del TMS320C80 de Texas Instruments, el cual ha sido utilizado para la construcción de uno de los sistemas de seguimiento experimental. Así mismo, el Pentium III de Intel ha sido el microprocesador de propósito general empleado en las restantes aplicaciones experimentales desarrolladas, con lo que también se ha incluido una descripción arquitectónica interna detallada.

*Apéndice B: Algoritmo de búsqueda sobre una arquitectura DSP multiprocesadora con memoria compartida.* Detalla la implementación del algoritmo de suma de diferencias absolutas que se propone y que se ha utilizado para llevar a cabo la búsqueda de patrones en imágenes sobre el DSP TMS320C80 de Texas Instruments. Esta implementación es transportable a cualquier otra arquitectura multiprocesadora con memoria compartida.

*Apéndice C: Algoritmo de suma de diferencias absolutas para el Intel Pentium III.* Recoge la implementación del núcleo del algoritmo de búsqueda basado en la suma de diferencias absolutas sobre un microprocesador Pentium III de Intel utilizando la potencia ofrecida por el repertorio de instrucciones MMX.

*Apéndice D: Comandos del módulo de seguimiento.* Describe todos los comandos del módulo de seguimiento propuesto. Estos comandos permiten el control completo sobre el módulo de seguimiento autónomo desarrollado y utilizado en esta tesis.

*Apéndice E: Soporte Experimental: WinVision.* Este apéndice ofrece una introducción al soporte software experimental desarrollado para realizar las distintas medidas sobre el rendimiento y precisión de los diferentes algoritmos estudiados.

## Capítulo 2

# Seguimiento mediante aprendizaje basado en contexto

El hecho de *ver* implica la interpretación, en algún grado, de las imágenes del mundo externo que se captan mediante sensores visuales. Es posible afirmar, según ciertas teorías, que interpretar lo que hay en una imagen es asignar una representación simbólica a cada objeto plasmado en la misma. Se puede entonces definir un *objeto visual* (o *distal*) como una entidad física distinguible de otras del mundo a la que se le puede asociar una representación simbólica propia.

Se denomina en este trabajo al mundo externo donde se hallan los objetos visuales *espacio distal*, en el sentido dado por Edelman (Edelman, 1997). Estos objetos visuales pueden evolucionar en este espacio desplazándose o transformándose de manera que sus vistas, captadas mediante sensores visuales, varíen a lo largo del tiempo. Como criterio de diseño operativo, estas imágenes diferentes deben corresponder a la misma representación para el mismo objeto independientemente de su vista.

Uno de los procesos que los sistemas de visión deben realizar para llevar a cabo

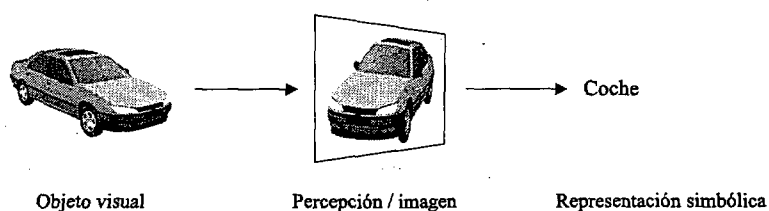


Figura 2.1: Un objeto visual en el espacio distal tras un proceso de percepción es transformado mediante un esquema de representación en un símbolo ubicado en un espacio proximal.

su función es transformar las imágenes adquiridas en una representación interna, sobre la cual puedan llevar a cabo las tareas encomendadas. Esto es así ya sea por razones de ejecución práctica o por consideraciones de diseño y desarrollo. Dependiendo de la finalidad del sistema de visión, esta representación será más o menos elaborada o compleja y, por tanto, su nivel de abstracción o simbolismo será mayor o menor.

Tras los procesos de representación y categorización, el objeto visual situado en el mundo exterior o espacio distal se asocia con un símbolo que lo representa. Independientemente de la estructura o complejidad que posea, este símbolo existirá en un espacio interior de representación, el cual es denominado espacio proximal. La Figura 2.3 muestra cómo un objeto visual, en este caso un coche, en el espacio distal tras un proceso de percepción llega a un espacio de representación simbólica que lo categoriza.

*"La representación convierte elementos del mundo externo o distales en símbolos internos de los mismos. Estas representaciones simbólicas internas de los objetos visuales se denominan objetos proximales."*

Existe consenso entre los investigadores en cuanto a la necesidad de la representación, explícita o implícitamente, para posibilitar cualquier actividad visual útil de mínima complejidad. Sin embargo, existen diferentes posturas en cuanto a cómo llevarla a cabo. Las diferentes propuestas toman evidencias provenientes de analogías con el sistema visual humano, aunque la manera real en que el ser humano realiza la interpretación de su percepción visual permanece aún desconocida en toda su complejidad. A continuación se enumeran varios paradigmas al problema de la representación de objetos y escenas:

- **Reconstruccionismo.** Esta teoría debida a Marr (Marr, 1982) plantea como objetivo la obtención de una descripción tridimensional completa del entorno para proceder posteriormente al reconocimiento. Los esfuerzos realizados durante la década de los ochenta en solventar las enormes dificultades encontradas llevaron a plantear serias dudas sobre la factibilidad de esta aproximación como método para obtener sistemas de visión viables con las restricciones de las aplicaciones reales.
- **Descomposiciones estructurales.** En este caso la representación de un objeto se realiza mediante la relación espacial de determinadas formas básicas que lo compondrían, Biederman (Biederman, 1987). Su principal dificultad radica en la detección de las partes que conforman el objeto. Una interesante discusión sobre las aproximaciones a la representación mediante descripciones estructurales vs. basadas en vistas se encuentra en (Hummel, 2000).

- Teorías basadas en restricciones geométricas (Bolles y Cain, 1982). Basan su representación en la consistencia del establecimiento de la correspondencia entre las características de un objeto y sus respectivas restricciones de ubicación en la imagen. En otras palabras, la representación se basa en una descripción cuantitativa de la geometría del objeto.
- Rutinas visuales. Ullman (Ullman, 1984; Ullman, 1987) introdujo el término *Visual Routines* para hacer referencia a la propuesta de un modelo de visión destinado a unir el procesamiento visual de bajo y alto nivel. Se planteó un modelo constituido por dos etapas. En una primera etapa se realiza una representación básica de la escena mediante el cómputo de características locales elementales de la imagen, llevándose a cabo de forma continua y en paralelo.

Una segunda etapa hace uso de esta representación básica para lograr una interpretación global de la escena, tomando en cuenta las relaciones espaciales que no pueden ser inferidas en la primera etapa y con un conocimiento de los objetivos del sistema. Ullman planteó esta interpretación global de la escena mediante el uso de procesos visuales secuenciales que llevan a cabo una descripción iterativa e incremental de la misma. Estos procesos los denominó *Rutinas Visuales*.

Las rutinas visuales están constituidas por *operadores elementales*, formados por procedimientos que incluyen el trazado de rectas, detección de intersecciones, trazado de contornos, marcado de objetos e inundación de regiones. Los objetivos del sistema y sus percepciones pasadas influyen en la selección de las rutinas visuales a ejecutar en cada momento y en la interpretación realizada por cada rutina visual de la escena. Las rutinas visuales se aplican sólo a los objetos que el sistema considere importantes y necesarios.

- Espacios de características multidimensionales. Formalmente todas las aproximaciones a la representación anteriores pueden ser expresadas como espacios de características multidimensionales. Sin embargo, la concepción de estos métodos se basa en la idea de que todos los modelos de objetos asumen que son representados y almacenados en términos de sus coordenadas de características en un determinado espacio de referencia. Las características concretas empleadas para su clasificación generan diversas técnicas para el reconocimiento o categorización de los objetos.

Es necesario destacar en este punto la relación existente entre los espacios de características multidimensionales y las representaciones basadas en vistas. Las técnicas basadas en vistas se fundamentan en la apariencia visual del objeto bajo diferentes poses y condiciones de iluminación. De esta manera se parte del aspecto

bidimensional del objeto en la imagen para extraer características normalmente no ligadas a las propiedades geométricas cualitativas o cuantitativas. Entre estas técnicas se pueden citar:

- Técnicas basadas en "histogramas multidimensionales" donde características del objeto como su histograma de color (Swain y Ballard, 1991) o de niveles de grises pueden representar a un objeto a efectos de su identificación.
- Técnicas basadas en pixels, Poggio y Edelman (Poggio y Edelman, 1990) mostraron cómo objetos tridimensionales pueden ser reconocidos mediante el uso de suficientes vistas bidimensionales de los mismos.
- Técnicas que hacen uso de métodos algebraicos con el fin de reducir la dimensionalidad original de los espacios de características. El análisis de los componentes principales (eigenvectors) muestra cómo problemas que usan representaciones originales con una alta dimensionalidad pueden ser transformados a una representación de dimensionalidad intrínsecamente baja, (Moghaddam, 1999). Turk y Pentland (Turk y Pentland, 1991a) demostraron que las caras humanas pueden ser representadas y reconocidas por medio de componentes principales de las mismas (eigenfaces). Murase y Nayar (Murase y Nayar, 1993) desarrollaron un método de componentes principales paramétrico para reconocer objetos tridimensionales directamente a partir de su apariencia.

En (Edelman, 1997) se encuentra una revisión de las técnicas de representación expuestas aquí brevemente.

Una vez decidida la representación de los objetos es posible llegar al reconocimiento o identificación de los mismos mediante el planteamiento teórico de las tareas a realizar por el sistema de visión. La representación escogida estará relacionada con los objetivos del sistema.

## 2.1. Representaciones isomórficas de segundo orden

Es posible concebir la generación de una representación como el proceso de establecimiento de una correspondencia  $S$  entre el conjunto de los objetos distales  $\mathcal{D}$  y el conjunto de los elementos proximales (Shepard y Chipman, 1970; Edelman et al., 1996; Choe, 2002).

$$S : \mathcal{D} \rightarrow \mathbb{R}^n$$



Se entiende por elementos proximales a unidades o entidades propias en el espacio de representación. Estos últimos pueden ser simples puntos en un espacio  $\mathfrak{R}^n$  sin ningún componente estructural añadido que se asocian a símbolos. Los diferentes tipos de correspondencias que se pueden dar son útiles para el tratamiento de diferentes clases de problemas.

- Relación uno a uno: establece un isomorfismo entre ambos conjuntos. Cada objeto visual del espacio distal posee su propia representación o símbolo proximal.
- Relación muchos a uno: establece la relación entre muchos objetos distales y una misma representación proximal. Esto equivale a asociar un mismo símbolo a varios objetos proximales, lo que supone asignar un objeto visual a una clase o categoría de objetos perceptuales.
- Relación uno a muchos: en general deben ser evitadas ya que originan ambigüedad al asignar diversas representaciones al mismo objeto distal.

Para dotar de mayor utilidad a la representación es posible establecer relaciones entre los elementos de un mismo espacio. Una de las posibles relaciones entre ellos son las similitudes o distancias entre los mismos.

En este punto se debe aclarar la aplicación de los conceptos de "similitud" y de "distancia" (ver Sección 3.2 en el Capítulo 3) en los espacios distal y proximal respectivamente. Aunque en principio ambos conceptos puedan hacerse equivalentes no consideramos muy apropiado, desde un punto de vista perceptual, utilizar el concepto "distancia" entre objetos distales sino el de "similitud", puesto que el parecido entre dos objetos distales es altamente dependiente del observador y del criterio utilizado para comparar. Por tanto, no es posible a priori cuantificar de forma absoluta el parecido entre ellos tal y como lo haría una medida de distancia. No obstante, sí es posible establecer el parecido relativo de dos objetos con respecto a un tercero, mediante las relaciones *ser más parecido que* o *ser menos parecido que*.

De la misma forma, desde un punto de vista analítico no es apropiado utilizar el concepto de "similitud" entre objetos proximales puesto que, en este caso, carecen de cualquier estructura o característica a comparar. Son tan sólo símbolos o puntos en un espacio  $\mathfrak{R}^n$  cuya única singularidad proviene exclusivamente de su situación en un lugar determinado de ese espacio. Por tanto, las similitudes entre los objetos distales resultan más adecuadamente expresadas de forma analítica como las distancias entre sus respectivas representaciones en el espacio proximal.

Tal y como se ha expuesto en la relación uno a uno, del concepto de isomorfismo se deduce que una correspondencia isomórfica distal-proximal por si misma permitiría a un sistema decidir si dos estímulos son o no idénticos comparando sus símbolos internos. Pero no, por ejemplo, si dos de estos estímulos son, de alguna forma, más parecidos entre si que cualesquiera otros dos estímulos. Para lograr esto parece entonces razonable pensar que las representaciones proximales de objetos distales similares deberían poseer una distancia entre ellas menor que en el caso de que fueran representaciones de objetos visuales muy distintos.

Esta nueva condición exige el establecimiento de un *isomorfismo de segundo orden* entre el espacio distal y el proximal (Edelman, 1999). Ello quiere decir que si la similitud entre dos objetos del mundo exterior o distal  $A$  y  $B$  es mayor que entre los objetos  $B$  y  $C$ , entonces la distancia entre sus representaciones proximales respectivas ( $A', B'$  y  $C'$ ) cumplirá que:  $d(A', B') < d(B', C')$ . Esto se ilustra en la Figura 2.3.

Como ya se mencionó anteriormente, la similitud entre objetos en el espacio distal es altamente dependiente de la percepción del observador y del criterio usado para comparar, lo cual le confiere un carácter algo subjetivo. Sin embargo, las vistas del mismo objeto del mundo real adquiridas en dos instantes de tiempo arbitrariamente cercanos pueden considerarse parecidas para casi cualquier observador. Cuanto más breve sea el intervalo entre las adquisiciones de las vistas mayor ha de ser el parecido entre ellas. Con lo cual, la propiedad fundamental de una representación basada en una correspondencia isomórfica de segundo orden reside en que pequeñas transformaciones en la percepción de un objeto distal, (rotaciones, deformaciones, etc.) deben corresponder a pequeños desplazamientos sobre las coordenadas de localización de ese objeto en el espacio de representación proximal. Esta observación conduce a la definición del siguiente principio:

***Principio de continuidad:*** *Dos vistas incrementalmente cercanas de un objeto real que evoluciona transformándose visualmente respecto a su observador en el espacio distal se corresponden a dos representaciones incrementalmente cercanas en el espacio proximal.*

Supóngase un método de representación basado en espacios de características multidimensionales y más concretamente basado en vistas. Entre las diferentes técnicas de representación existentes es posible escoger una de las más sencillas, en la cual el espacio de todas las posibles imágenes de  $m \times n$  pixels estaría definido por  $\mathcal{R}^{m \times n}$ . Por tanto, un objeto cualquiera representado por una imagen de  $m \times n$  pixels corresponderá a un único punto de ese espacio. Según el principio de continuidad, las pequeñas transformaciones

del objeto corresponderán a puntos muy próximos en el espacio de representación. Estas transformaciones posibles del objeto se ven considerablemente reducidas debido a las restricciones del mundo físico, que no permiten más de unos ciertos grados de libertad en las transformaciones de los objetos. De esta forma, del espacio  $\mathbb{R}^{m \times n}$  completo, tan sólo un subespacio del mismo acogerá los puntos correspondientes a vistas físicamente posibles mediante las diferentes transformaciones visuales del objeto. Este subespacio poseerá una dimensionalidad igual al número de grados de libertad del objeto y generará una *variedad* topológica en el espacio proximal asociada al objeto que se definirá como:

*Variedad asociada a un objeto es el subconjunto de dimensionalidad reducida del espacio proximal que corresponde a los lugares o puntos pertenecientes a las representaciones de las diferentes vistas que el objeto pueda ofrecer mediante un cierto número de grados de libertad permitidos.*

Esta variedad puede ser concebida de manera sencilla como una hipersuperficie paramétrica alabeada en un espacio, donde cada parámetro se corresponderá con un grado de libertad concreto del objeto, (Hernández, 1987).

Durante el proceso de seguimiento de un objeto dentro de un periodo de tiempo dado, éste no ofrece todas las posibles vistas de si mismo incluidas en su variedad sino un subconjunto del mismo cuya forma será la de una curva paramétrica alabeada, donde el parámetro corresponderá al tiempo. Esta curva se denominará *Curva de Transformación Visual del Objeto*, véase Figura 2.2.

## 2.2. Aprendizaje incremental mediante discriminación basada en contexto

Cada objeto visual y sus transformaciones sufridas a lo largo del tiempo crean una curva de transformación visual dentro de su variedad en el espacio proximal. Cada punto de esa curva de transformación visual se corresponde con una vista de ese objeto en un instante dado. Seguir ese objeto visual en el espacio distal implica seguir los distintos puntos de esa curva en el espacio proximal. Desafortunadamente, al iniciar el seguimiento de un objeto visual, tan sólo se conoce el punto inicial o extremo de esa curva, el resto de los puntos deberán ser aprendidos autónomamente.

En un entorno real existen, además del objeto de interés, más objetos visuales que proyectan sus representaciones en el espacio proximal y que poseen sus respectivas variedades. En este sentido denominamos:

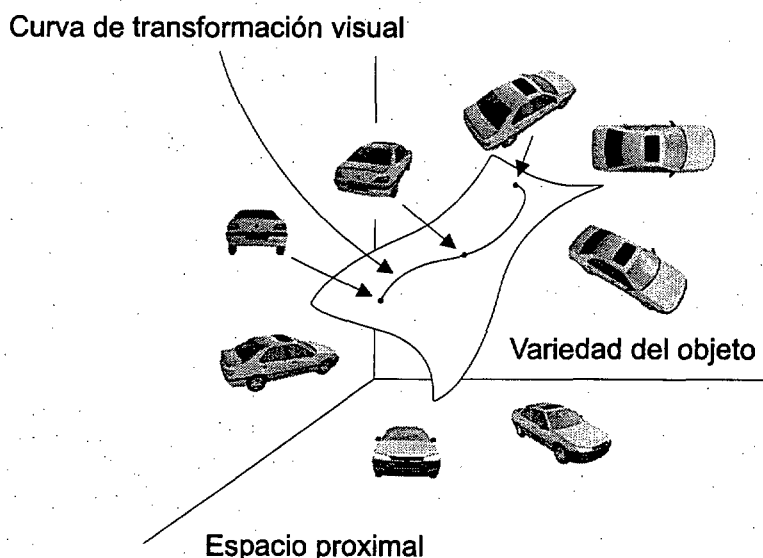


Figura 2.2: La figura muestra la variedad de un objeto dentro de un espacio de representación proximal. La variedad del objeto es un subconjunto de ese espacio en donde se encuentran todas las representaciones posibles del objeto. La curva de transformación visual constituye un subconjunto lineal de esa variedad y corresponde a las representaciones que se dan del objeto a lo largo de un periodo de tiempo determinado, normalmente durante un proceso de seguimiento.

*Objetos del Contexto* al conjunto de los objetos visuales presentes en una escena diferentes del objeto de interés.

Resulta razonable que, al ser objetos diferentes, las correspondencias proximales de estos otros objetos visuales estén alejadas de los puntos de la variedad de interés. Al igual que el objeto de interés, durante un intervalo de tiempo concreto cada objeto del contexto también genera su propia curva de transformación visual incluida en su variedad.

Según se deduce de lo expuesto, la única pista que permite, en un instante dado, clasificar una proyección proximal de un objeto visual como objeto de interés es su cercanía al segmento actual de la curva de transformación visual de interés. Expresado de otra forma, del conjunto de todas las proyecciones proximales pertenecientes a todos los objetos visuales en la escena en un momento dado, la proyección proximal perteneciente al objeto de interés será aquella con menor distancia a la curva de transformación visual de interés generada hasta ese momento.

El análisis de una escena sencilla ayuda a clarificar lo expuesto anteriormente. Supóngase que, en principio, en el espacio distal sólo existe el objeto visual de interés. Al comienzo del proceso de seguimiento únicamente se tendrá la proyección de ese objeto sobre el espacio proximal, con respecto al cual se deberán calcular las distancias

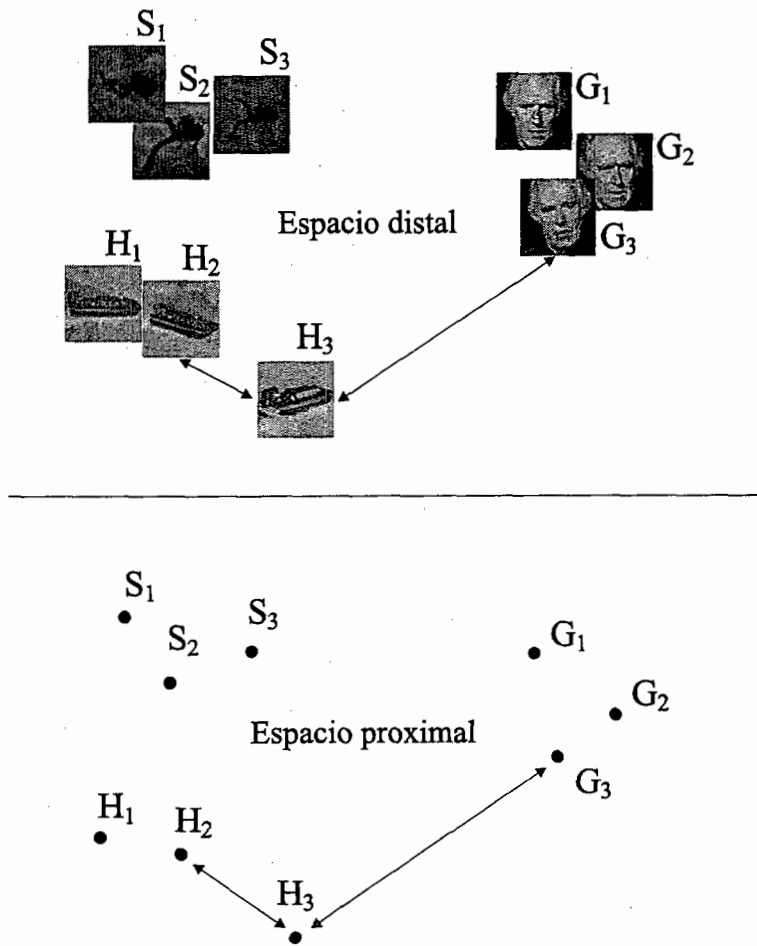


Figura 2.3: Relación isomórfica de segundo orden entre un espacio distal y un espacio proximal. Esta correspondencia debe cumplir ciertas relaciones de similitud entre objetos. Por ejemplo, si la similitud en el espacio distal entre el objeto  $H_2$  y  $H_3$  es mayor que entre los objetos  $H_3$  y  $G_3$ , entonces en el espacio proximal esta relación debe mantenerse. Es decir, la distancia entre  $H_2$  y  $H_3$  debe ser menor que entre  $H_3$  y  $G_3$  en el espacio proximal.

de las demás proyecciones futuras. Sin embargo, en este caso no tiene sentido utilizar una función de distancia para calcular la cercanía o lejanía de una nueva proyección de un objeto visual sobre el espacio proximal puesto que, al no existir otro objeto en el entorno distal, cualquier punto de representación proximal pertenecerá obviamente al objeto de interés. Por el contrario, si existieran otros objetos visuales en el espacio distal, las representaciones de estos podrían, en principio, estar arbitrariamente próximas.

En el caso de una escena con la presencia de objetos visuales de contexto sí resulta necesaria una clasificación de las diferentes representaciones proximales de los diferentes objetos. Supóngase una escena con dos objetos visuales suficientemente distintos, uno es el objeto de interés y el otro un objeto del contexto, de manera que ambos proyectan

sus representaciones proximales con una cierta distancia entre ellas (véase Figura 2.4 en  $t_1$ ). Estos dos objetos cambiarán la forma en que son vistos a lo largo del tiempo de alguna manera, ya sea porque se alejan, acercan, rotan, se deforman o como se representa en la Figura 2.4, sufren *morphing* hasta transformarse en otros objetos. De acuerdo con el principio de continuidad, cualquiera que sea el tipo de cambio aplicado sobre los objetos, éste debe ser gradual y suave. Se establecerá correctamente entonces que la representación proyectada en  $t_2$  del objeto de interés será aquella con menor distancia o más cercana a la representación del objeto de interés en  $t_1$ , que es el punto de representación proximal de referencia.

De nuevo en el instante  $t_3$ , se aprecia que en las nuevas proyecciones proximales correspondientes hay una más cercana a la proyección de interés en  $t_1$ , la cual será tomada nuevamente como perteneciente a la curva de transformación visual del objeto de interés. El proceso seguirá llevándose a cabo correctamente de esta manera durante los siguientes instantes  $t_4$  y  $t_5$ . Durante este intervalo de tiempo, tanto el objeto de interés como el otro objeto del contexto han ido evolucionando hasta hacerse relativamente distintos de como eran o se veían originalmente en el instante  $t_1$ , cuyo representación proximal sigue utilizándose como referencia. De esta manera, en el instante  $t_6$ , ocurre que la proyección del objeto de interés se encuentra a mayor distancia o más alejada de su proyección en  $t_1$  que el objeto de contexto de la escena. Luego la proyección del otro objeto de la escena será tomada erróneamente como perteneciente a la variedad del objeto de interés; y por tanto provocará un fallo del proceso de seguimiento.

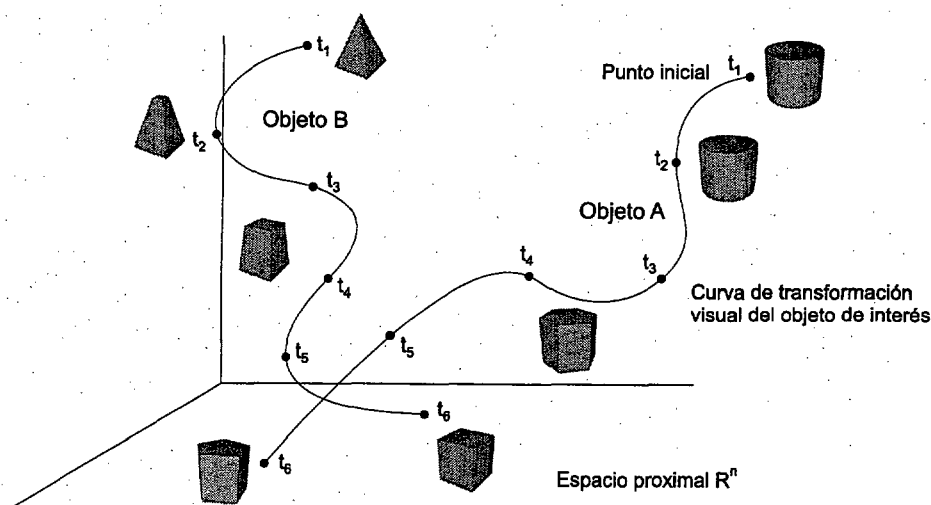


Figura 2.4: La figura muestra dos curvas de transformación visual pertenecientes a dos objetos distintos de la imagen, uno es el objeto de interés (objeto A) y el otro un objeto del contexto (objeto B).

Evidentemente, dentro del esquema de seguimiento presentado, el problema podría haberse solucionado tomándose en cada momento  $t$  la última representación proximal del objeto de interés como referencia para la siguiente, es decir tomar como referencia  $t - 1$ . Sin embargo, esto no siempre es posible o aconsejable mediante el uso de algunas técnicas de búsqueda como puede ser la búsqueda de patrones (véase Capítulo 3).

### 2.3. Aplicabilidad. Problemas y contextos

Como se desprende del análisis de las curvas de transformación visual y variedades asociadas a los objetos visuales de la escena, para cualquier esquema de representación escogido que incluya una correspondencia isomórfica entre sus espacios distal y proximal es siempre recomendable realizar el seguimiento de un objeto visual teniendo en cuenta los demás objetos de su contexto que puedan distraer la atención del sistema. Es necesario destacar ahora la importancia que esto tiene debido a que no solamente se incrementa la fiabilidad y robustez ante errores sino que son muchas sus implicaciones computacionales prácticas. Dependiendo de estos objetos del contexto y del método de búsqueda utilizado se pueden establecer diferentes optimizaciones que mejoran considerablemente el rendimiento computacional. A continuación se comenta la utilidad que tiene el control del entorno sobre algunas de las técnicas de búsqueda más comunes.

- Para un algoritmo de búsqueda basado en seguimiento de contornos de Hausdorff, (Sánchez-Nielsen y Hernández, 2000), el control del entorno se puede traducir en qué fracción de su perímetro es necesario seguir en cada momento. El cálculo del contorno del objeto de interés se basa en la transformación afín del mismo para intentar adaptarlo al movimiento propio del objeto en instantes posteriores. El método será más robusto cuantas más alternativas de encaje lleve a cabo, lo cual supone un costo computacional alto. Sin embargo, la cantidad o longitud de contorno a seguir puede ser menor dependiendo de los contornos de los objetos del contexto que se hallen cerca, lo que permite plantear una heurística específica que puede mejorar el comportamiento computacional en estos casos. En la Figura 2.5 se muestra muy simplificada cómo para un objeto de interés concreto, en este caso el cuadrilátero, la cantidad de contorno necesario para identificarlo varía en función de los objetos que lo rodeen. Por ejemplo, en el caso *A* la cantidad de contorno requerida para diferenciarlo de un círculo (señalado aquí con un trazo más grueso) es menor que para diferenciarlo de un triángulo rectángulo, caso *B*.
- Para un algoritmo sencillo de búsqueda por color, el control del contexto significa

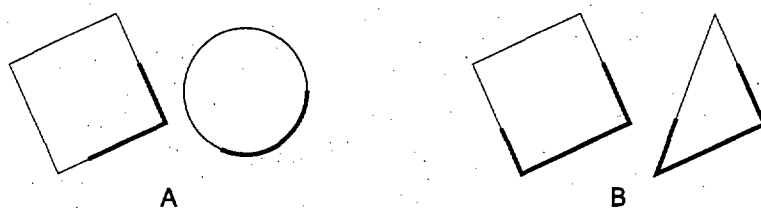


Figura 2.5: La figura B muestra simplídicamente la necesidad de tomar más cantidad del contorno de los objetos con respecto a la figura A con el fin de poder distinguirlos. Como se aprecia, el mismo cuadrilátero requerirá de mayor o menor contorno para ser localizado correctamente dependiendo de los objetos del contexto.

saber qué precisión debe tener el rango o intervalo de color buscado. Supóngase que se tiene un único objeto en la escena, éste posee un color rojo que puede variar ligeramente dependiendo de la orientación de las normales de sus caras respecto a la fuente de luz. Además, todo el fondo de la escena es negro. En esta situación el intervalo de color a buscar centrado en el rojo puede ser muy amplio, debido a que prácticamente todo lo que no sea negro va a pertenecer al objeto de interés. Si ahora existiese, junto al objeto rojo de interés, un objeto de tono rosado, el rango de color a buscar estaría bastante más limitado para no incluir al tono rosado del objeto contextual.

- Para un algoritmo de búsqueda de patrones, como el tratado en este trabajo, significa saber cuándo ha de tomarse un nuevo patrón para llevar a cabo la comparación. Esto se demuestra con mucho mayor detalle en la siguiente sección.

En general se puede afirmar que:

*"Para correspondencias isomórficas de segundo orden en las representaciones proximales de los objetos visuales, la búsqueda de un objeto en un determinado escenario deberá ser más precisa en función de la similitud de los objetos cercanos a éste. Esto permite definir una heurística de búsqueda de las nuevas versiones de los patrones con costos computacionales reducidos."*

Computacionalmente esto puede entenderse como una metaheurística que proporciona una serie de optimizaciones aplicables a los procedimientos de búsqueda. De esta forma, cualquier algoritmo de búsqueda con un esquema de representación isomórfica que tome en cuenta el contexto de la manera antes descrita obtendrá las siguientes mejoras:



- Establecimiento de manera automática de los diferentes parámetros del método de búsqueda utilizado, haciendo al método más general y aplicable a otros entornos nuevos sin ningún tipo de ajuste manual o empírico previo.
- Adaptación de la potencia y tiempo de cómputo a las necesidades reales del seguimiento en ese momento y de esta manera poder liberar los recursos computacionales aplicables para otras tareas.
- Variación de forma automática de la frecuencia de funcionamiento del método de búsqueda. De esta manera el ciclo de trabajo se intensifica en función de la complejidad de la escena.

Hacemos notar que, desde un punto de vista general, la representación proximal, a efectos de los planteamientos aquí realizados puede ser cualquier espacio de imágenes (monocroma, color, etc.) de propiedades (contornos, texturas, etc.) resultado de una transformación del dominio de las imágenes [p.e. PCA (Turk y Pentland, 1991a; Turk y Pentland, 1991b), ICA (Bartlett y Sejnowski, 1997), etc.] Sin embargo, se deben cumplir los criterios relativos de isomorfismo de segundo orden y, de cara a su ejecución en tiempo real, la transformación al espacio proximal debe hallarse ajustada a los requisitos de tiempo del sistema específico en desarrollo.

Como conclusión se obtiene que la aplicación de la teoría del aprendizaje incremental mediante discriminación basado en contexto proporciona mejoras sustanciales en la robustez de los métodos de búsqueda y optimización de los recursos computacionales.

## 2.4. Particularización de la búsqueda incremental basada en contexto

Partiendo de la concepción del aprendizaje incremental de las curvas de transformación visual basado en esquemas de representación isomórfica, esta sección describe una particularización de esta teoría, donde ahora la búsqueda se soporta en la correspondencia de patrones. Con tal motivo se establecen los espacios distal y proximal para llevar a cabo la representación de objetos incluidos dentro de un patrón de dimensiones concretas.

Supóngase que el objeto de interés se encuentra caracterizado en un patrón de  $m \times n$  pixels, que se ordena lexicográficamente en un vector  $P$  de  $m \times n$  elementos. Sea además  $\mathbb{I}$  un espacio  $\mathcal{R}^{m \times n}$  que conformará el espacio proximal, del cual  $P$  es un punto del

mismo. De esta forma,  $\mathbb{I}$  contiene todas las posibles imágenes que se puedan formar con una resolución de  $m \times n$ . En otras palabras, el generador de la representación proximal es el propio sistema de adquisición de imágenes. Por tanto, un objeto externo en el espacio distal, una vez digitalizado por las cámaras y localizado dentro de una ventana de dimensiones  $m \times n$  pixels, tiene su representación proximal en un espacio  $\mathbb{R}^{m \times n}$ , que se denominará *espacio de las imágenes*. Los valores de cada uno de los pixels  $v_i$  (estando  $i$  comprendido entre 1 y  $m \times n$ ) de la ventana corresponden a la proyección del punto  $P$  en cada una de las  $m \times n$  coordenadas. El punto  $P$  se corresponde a un objeto dado en un momento determinado. Al evolucionar este objeto, su ubicación en el espacio de las imágenes varía. Debido a la naturaleza enteramente digital de las imágenes, el espacio  $\mathbb{I}$  no es continuo sino discreto, esto hace que las  $m \times n$  coordenadas de este espacio no sean reales sino un subconjunto finito del conjunto de los enteros. En imágenes comunes de escala de grises, estos valores estarían comprendidos entre 0 y 255 ambos inclusive.

Como en cualquier espacio geométrico, en el espacio  $\mathbb{I}$  se pueden establecer distancias entre cualquier par de puntos pertenecientes al mismo. Se define aquí la distancia entre dos puntos como la norma  $L_2$ , dado que sus características como métrica se adecúan suficientemente a las necesidades del problema tratado.

$$D(P_1, P_2) = \|P_1 - P_2\|$$

En (Russell y Sinha, 2001) puede encontrarse un estudio sobre diferentes métricas para calcular la distancia entre objetos con relación a la similitud perceptual que poseen.

Como la correspondencia resultante entre el espacio distal y el espacio proximal conforman un isomorfismo de segundo orden, se cumple el principio de continuidad enunciado anteriormente. Según este principio, el objeto que evoluciona a lo largo del tiempo (se desplaza, rota, se deforma, etc.) debe describir una curva de transformación visual alabeada en el espacio proximal, véase Figura 2.6 Sin embargo, el tiempo es otro de los factores discretizados. En sistemas de adquisición reales los periodos de muestreo típicos son de 25 ó 30 imágenes por segundo, lo que corresponde con el periodo de muestreo o discretización de la curva de transformación visual.

En cada cuadro de una secuencia que incluye un objeto distal que se desea seguir se proyectan gran cantidad de vectores que pertenecen a distintas áreas de comparación, de dimensión igual al patrón a buscar, generados por el proceso de búsqueda. Si el patrón  $P$  de tamaño  $m \times n$  ha de buscarse en una imagen de resolución  $m' \times n'$ , donde por supuesto  $m < m'$  y  $n < n'$ , la cantidad de puntos  $N$  generados pertenecientes al espacio

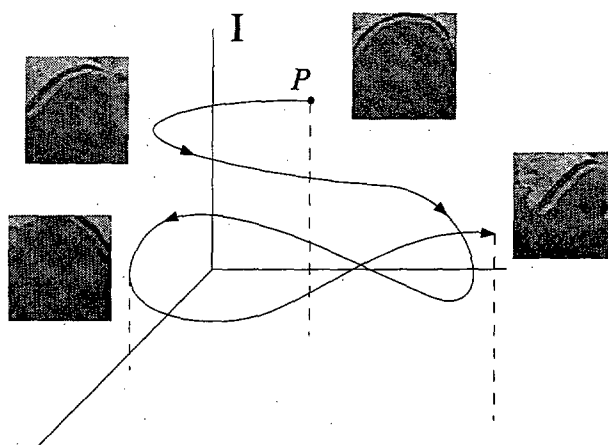


Figura 2.6: La variación suave en el tiempo del objeto a seguir se ve reflejado en el espacio de las imágenes  $\mathbb{I}$  como una curva alabeada formada por los puntos correspondientes a las diferentes vistas del objeto. Dado el carácter discreto de las imágenes digitales, la curva no será conexa como tal sino una sucesión de puntos en un espacio discreto.

$\mathbb{I}$  será:

$$N = (m' - m + 1) \times (n' - n + 1)$$

Entre ellos se encuentra un punto proximal que representa a ese objeto distal en el espacio  $\mathbb{I}$  en ese determinado instante, que es el que hay que identificar a lo largo de la secuencia para realizar el seguimiento.

Este punto elegido se corresponderá a la ventana de tamaño  $m \times n$  de la imagen más parecida al patrón. En la Figura 2.7 se visualizan las distancias de estos  $N$  puntos como una función de dos variables. Las dos variables independientes corresponden a las coordenadas de la imagen donde se aplicó la función de similitud. Se observa cómo la función contiene un mínimo absoluto y varios mínimos locales. El mínimo absoluto está rodeado por laderas que crecen a medida que esas áreas de la imagen se parecen cada vez menos al patrón. La existencia de más mínimos locales implica que se hallan objetos en otras áreas de la imagen similares al objeto del patrón en algún grado. Por tanto, de los  $N$  puntos generados en cada cuadro son tan sólo algunos los que se toman en consideración, estos son los mínimos locales. Considerados ahora como objetos también pertenecientes a la escena y cuyo valor en la función de similitud representa el grado de parecido con respecto a objeto seguido. Estos objetos, de igual forma que el objeto a seguir, muestran una trayectoria alabeada en el espacio proximal formada por los puntos correspondientes a las distintas proyecciones que generan sus diferentes vistas a lo largo de la secuencia. Estas trayectorias estarán incluidas en sus respectivas variedades.

En la gráfica a) de la Figura 2.8 se muestra de forma esquemática todos los puntos

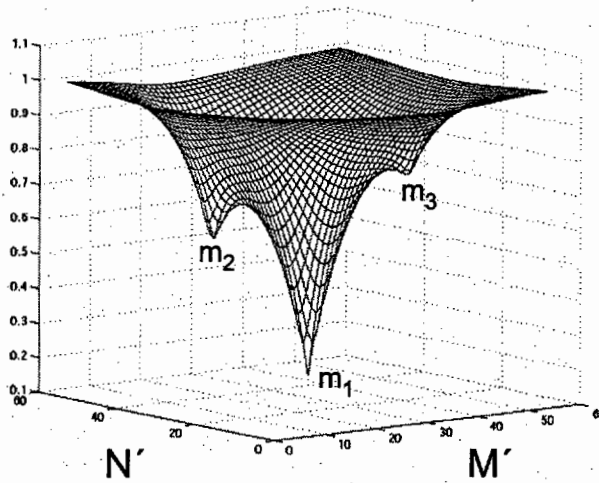


Figura 2.7: La figura muestra la superficie generada por la búsqueda de un patrón sobre una imagen. Cada valor de la superficie corresponde a la distancia (o similitud) entre el patrón y esa zona concreta de la imagen de la misma dimensión del patrón. Se observa cómo existe un mínimo absoluto  $m_1$  que corresponderá al punto de la imagen donde se ha localizado el objeto de interés, y dos mínimos locales  $m_2$  y  $m_3$  correspondientes a dos objetos del contexto en la imagen.

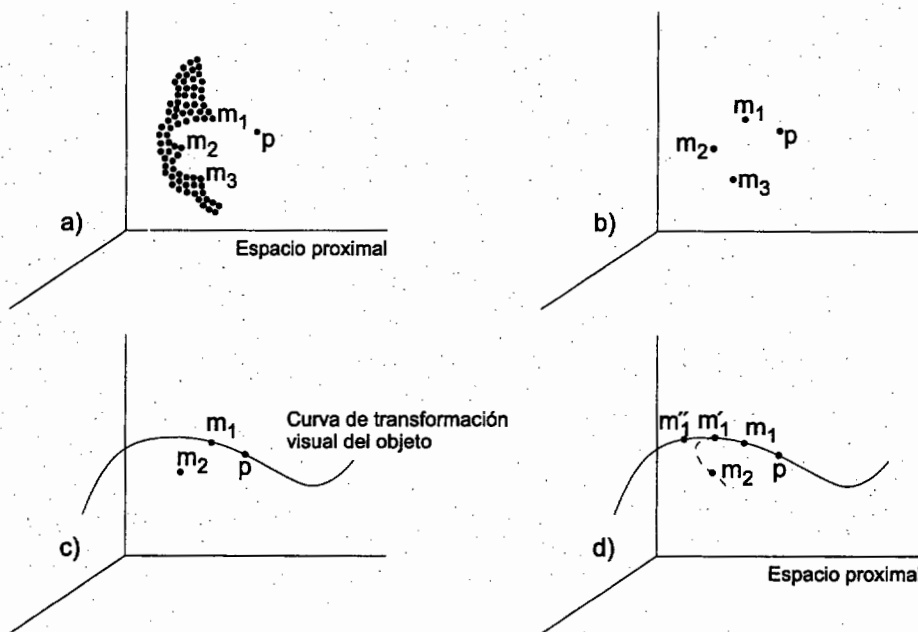


Figura 2.8: Las figuras corresponden a un espacio proximal donde se hallan diferentes representaciones de objetos. Aunque por motivos de visualización el espacio proximal se vea reducido a un espacio  $\mathcal{R}^3$ , su dimensión real es  $\mathcal{R}^{m \times n}$ . Las figuras c) y d) muestran la curva de transformación visual generada por un objeto a lo largo de la secuencia.

generados en el espacio proximal tras la aplicación de la función de similitud entre la imagen actual y el patrón a buscar cuya representación proximal corresponde al punto  $p$ . Esta función genera ciertos puntos que son, además del mínimo absoluto  $m_1$ , mínimos locales,  $m_2$  y  $m_3$ . En la gráfica *b)* se muestran estos puntos aisladamente. De todos ellos, tan sólo el segundo de los mínimos locales (punto con menor distancia a  $p$  después del mínimo absoluto),  $m_2$ , será tomado en cuenta para la determinación del próximo punto a tomar como referencia de la curva de transformación visual. En la gráfica *c)* se observa cómo el mínimo absoluto  $m_1$  se asume como punto correspondiente a la curva de transformación visual del objeto seguido. Si se supone, por simplicidad, que el objeto distal del contexto que corresponde a la proyección proximal  $m_2$  no cambia, mientras que el objeto a seguir se deforma con el paso de los sucesivos cuadros, se observa cómo se suceden los puntos de su curva de transformación incluida en su variedad. Todos ellos son, en principio, mínimos absolutos en sus respectivos cuadros con respecto al patrón  $p$ . Sin embargo, puede verse que las distancias  $d(m_1, p) < d(m_2, p)$  y  $d(m'_1, p) < d(m_2, p)$ , pero  $d(m''_1, p) > d(m_2, p)$ , con lo que no actualizar el patrón a un nuevo punto de referencia de la curva de transformación visual en algún instante anterior conllevaría a una pérdida de éste, debido a la generación de una curva errónea, representada en la gráfica *d)* mediante la línea segmentada.

Evitar esto significa que debe ser observado el entorno del objeto de interés para controlar las posibles interferencias que puedan causar otros objetos del contexto. Para tal fin, debe actualizarse el patrón original con otros puntos pertenecientes a la curva de transformación de éste, con lo que se deben aprender qué puntos son los que deben ser escogidos. Esta pista la dan precisamente los demás puntos en cada cuadro pertenecientes a las demás curvas de transformación de los otros objetos en la escena. En un cuadro determinado, un punto de la curva de transformación del objeto de interés debe tomarse como nuevo patrón cuando la distancia de un punto perteneciente a otro objeto distinto sea menor que una cierta distancia umbral. Esta distancia umbral se define en el momento en que se toma un nuevo punto de la curva de transformación del objeto de interés como patrón. Este nuevo patrón estará representado por un punto en el espacio proximal, luego este valor umbral se determinará como un valor no mayor que la distancia a la que se encuentra el objeto del contexto más cercano. Normalmente, este valor se puede fijar a la mitad de esa distancia. Esto hace que el muestreo de patrones del objeto sobre la variedad dependa enteramente del entorno donde se halle.

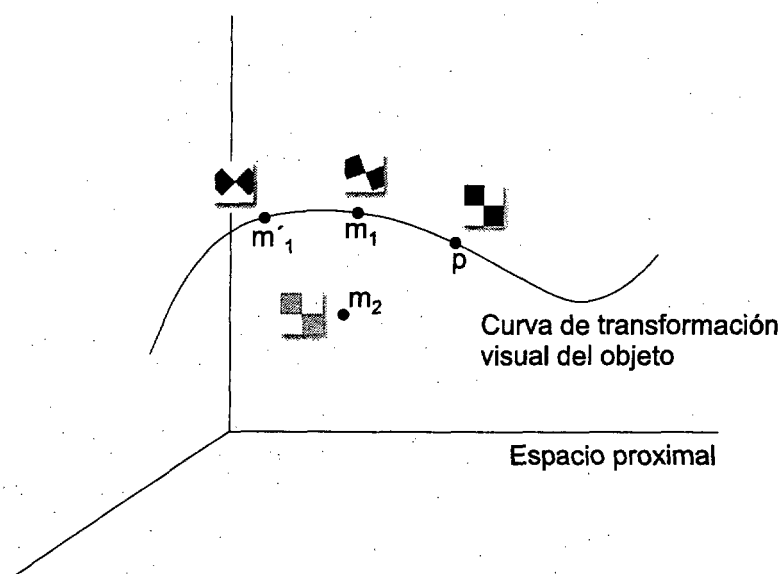


Figura 2.9: La figura muestra un ejemplo de objeto concreto cuya rotación hace que su representación proximal vaya localizándose en distintos puntos de su curva de transformación visual. El punto  $p$  corresponde al patrón actualmente utilizado para buscar el objeto en los diferentes cuadros,  $m_1$  y  $m_1'$ . Se observa también cómo el objeto con representación proximal  $m_2$ , que se encuentra también en la escena, se halla relativamente próximo al punto  $p$ . Con lo cual, a medida que el objeto de interés vaya rotando cada vez más, llegará un momento en que su distancia a  $p$  sea mayor que la distancia entre  $m_2$  y  $p$ .

### 2.4.1. Memorización de los patrones aprendidos. Banco de patrones

Es común realizar el seguimiento sobre objetos que muestran un comportamiento cíclico en cuanto a la vista ofrecida por éste hacia el observador. Por ejemplo, una persona cuya cara se encuentra frente al observador y que vuelve periódicamente la cara hacia la derecha e izquierda, Figura 2.10. Las vistas ofrecidas por las distintas poses serán diferentes entre ellas pero, por ejemplo, la pose con vista frontal será prácticamente igual antes y después de haber girado la cabeza hacia la derecha. Probablemente, alguna vista frontal, alguna lateral derecha y alguna lateral izquierda sean en determinados momentos los patrones escogidos y aprendidos para llevar a cabo el seguimiento. Estas vistas proyectarán su representación respectiva en determinados puntos del espacio proximal, todos ellos pertenecientes a la variedad de la cara a seguir.

Según esto, es posible plantear el almacenamiento en una memoria visual de los patrones aprendidos mediante sus representaciones proximales del objeto en diferentes vistas o poses. Estas vistas pasadas pueden ser útiles en el caso de que el objeto vuelva

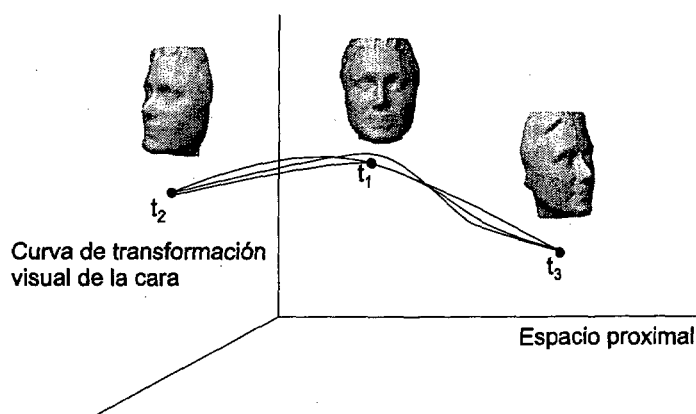


Figura 2.10: Cuando la vista ofrecida por el objeto de interés muestra un comportamiento cíclico, su representación recaerá repetidamente sobre la curva de transformación visual o muy cerca de ella.

a mostrar el mismo aspecto que anteriormente.

En el caso de un objeto seguido cualquiera que mostrara una transformación cíclica en su apariencia, es decir, si por ejemplo rotara sobre su eje periódicamente, volvería a mostrar una cierta vista cada cierto tiempo. La Figura 2.11 muestra este caso. Trascurrido un ciclo completo y en el caso de condiciones de entorno estables, no sería necesario adquirir nuevos patrones porque todos estarían ya aprendidos. Sin embargo, al ser el aprendizaje de una variedad generada por un objeto visual dependiente de las condiciones del entorno, si estas cambiaran, el aprendizaje sobre la variedad debería proseguir. Por esta razón, si en el entorno aparecen objetos similares al objeto visual de interés, obligaría a cambiar la frecuencia del muestreo, y por tanto, se necesitaría adquirir y aprender nuevos patrones.

La utilidad de recuperar vistas o representaciones pasadas radica en que constituye una cierta garantía de fiabilidad puesto que volvemos sobre un subconjunto de la variedad del objeto ya visitada. Como se muestra en el Capítulo 3 aprender un nuevo patrón implica la posible deriva del objeto visual de interés.

Disponer de una memoria que almacene las representaciones proximales de cada nuevo patrón aprendido conlleva dos inconvenientes asociados. Por un lado se encuentra la limitación de la memoria física disponible en los computadores reales, ya que en principio estos requerimientos de memoria podrían ser arbitrariamente grandes. Por otro lado, y como principal dificultad, se encuentra el costo de procesamiento que implica comparar todos los patrones previamente almacenados. Todo esto hace que el número de representaciones almacenadas deba ser finito.

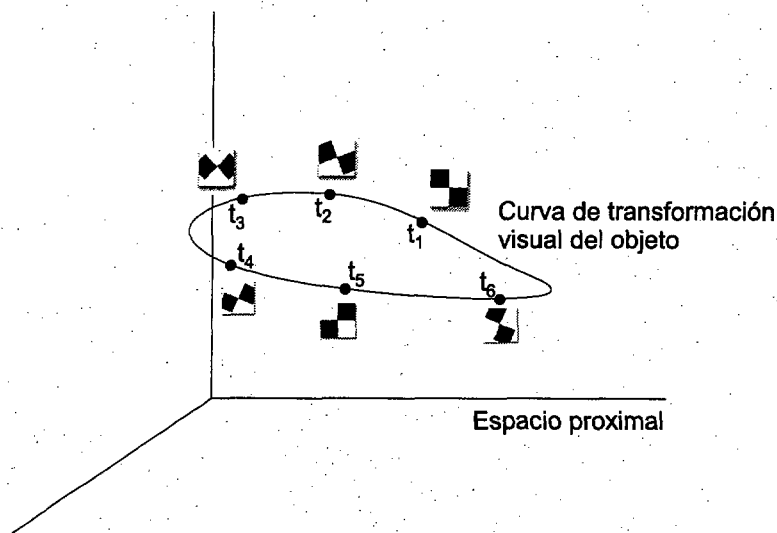


Figura 2.11: Curva de transformación visual correspondiente a la rotación de un objeto. En la curva alabeada cerrada se encuentran todas las vistas del objeto ofrecidas durante un ciclo. Los instantes  $t_1$  a  $t_6$  podrían corresponder a los patrones elegidos para realizar el seguimiento según unas condiciones de entorno.

Al ser la memoria de representación limitada, cuando esta se llena, es necesario sustituir algunas representaciones almacenadas por otras. En el Capítulo 3 se describe el procedimiento práctico adoptado para realizar la actualización de las representaciones almacenadas.

### 2.4.2. Algunas implicaciones del seguimiento con aprendizaje basado en contexto

Además de las ventajas que posee este método, ya expuestas a lo largo de este capítulo, existen tres implicaciones reseñables en la utilización de este modelo de memorización: una relativa a la representación basada en vistas, otra relativa a la extracción de vistas como muestreo y otra tercera relacionada con las competencias del asentamiento perceptual. Sobre ellas se reflexiona a continuación.

#### Sobre la representación basada en vistas

En los últimos años se han desarrollado diversas teorías, modelos computacionales y métodos de visión artificial basados en apariencia (appearance-based), también denominados basados en vistas (view-based) o basados en imágenes (image-based). Todos ellos asumen que los objetos se representan y comparan en términos de sus representa-



ciones en un marco espacial de visualización referido al sensor (Edelman, 1998; Edelman y Weinshall, 1991; Poggio y Edelman, 1990; Ullman, 1989; Ullman y Basri, 1991). La idea central de esta aproximación es que los objetos se representan en memoria como vistas, de manera que a partir de operaciones en las coordenadas de las características en estas vistas se obtienen nuevas vistas y se registran en memoria. La naturaleza de estas operaciones es el foco de atención principal de muchas teorías basadas en vistas que forman parte del conjunto de las técnicas básicas más utilizadas en visión artificial (Rao y Rajes, 1997; Turk y Pentland, 1991a; Murase y Nayar, 1995; Rao y Ballard, 1995; Huttenlocher et al., 1996; Black y Jepson, 1996; Leonardis y Bishof, 1996). Asimismo, la aproximación presenta implicaciones a nivel de neurofisiología de la visión, en tanto que constituye una de las aproximaciones que intentan explicar cómo funcionan los mecanismos biológicos del reconocimiento.

### **Sobre el muestreo de la variedad**

El proceso de extracción de vistas puede entenderse como un muestreo de la variedad que representa a la forma tridimensional en el espacio distal por un criterio de relevancia establecido en función de lo distinguible del objeto de interés respecto al contexto. Este criterio puede basarse en diversas medidas de disimilitud, y permite extraer una representación primaria del objeto con un conjunto de vistas (o muestras) representativas. Estas vistas serán suficientemente significativas en relación con su contexto visual, en lugar de simples criterios de muestreo constante. Entre los criterios comunes de muestreo constante se pueden citar los criterios temporales, de diferenciación sólo respecto a la vista actual o puramente geométricos de ángulos constantes de muestreo en el ámbito de la esfera de visualización, (Murase y Nayar, 1995; Murase et al., 1996).

### **Sobre el asentamiento perceptual**

La tercera implicación se refiere a las posibilidades que ofrece un módulo de seguimiento precategórico que implemente un mecanismo basado en contexto con memorización de vistas, en relación con su utilización en el desarrollo de sistemas dotados de competencias de Asentamiento Perceptual, tanto en su planteamiento simbólico (Harnad, 1990) como en su planteamiento físico (Brooks, 1990).

Entiéndase asentamiento perceptual como aquel conjunto de procesos que posibilitan a un sistema extraer y fijar conceptos perceptuales a partir de datos provenientes de los sensores, lo que permite establecer fases primarias del aprendizaje y la adaptación. Los dos planteamientos de asentamiento mencionados anteriormente se corresponden

con las aproximaciones principales en la concepción, interpretación y desarrollo de sistemas inteligentes artificiales. De esta forma, mientras la primera se corresponde con el conjunto de mecanismos que permiten ligar los datos perceptuales aferentes con el dominio simbólico en sistemas desarrollados partiendo de la Hipótesis del Sistema de Símbolos Físicos (Newell, 1980); la segunda se corresponde a la referenciación perceptual, que se efectúa en el ámbito de los sistemas desarrollados según la Hipótesis de los Fundamentos Físicos de la Inteligencia, en el paradigma de la denominada Nueva Inteligencia Artificial o también denominada Inteligencia Artificial Subsimbólica, (Brooks, 1990; Pfeifer y Scheier, 1999). En esta última, el conocimiento del entorno lo adquiere el sistema a partir de la interacción y no se realiza declaración simbólica explícita alguna (Vogt, 2000; Vogt, 2002; Sun, 2000). Al proceso de crear y mantener en el tiempo la conexión entre representaciones abstractas y representaciones a nivel físico en el ámbito de sistemas robóticos algunos autores (Coradeschi y Saffiotti, 2000) lo denominan *Anclaje Perceptual* (*perceptual anchoring*).

El diseño de un módulo de seguimiento de objetos de esta naturaleza incorpora normalmente mecanismos de atención, ya sea basados en movimiento, como en DESEO, o por la variación de otra pista visual proveniente de otro canal sensorial. Si el mecanismo de atención correspondiente señala al módulo de seguimiento el inicio del proceso, éste se activará y seguirá al objeto visual de interés hasta su pérdida, si es el caso. Durante el tiempo de su actividad, el módulo de seguimiento memorizará los patrones o vistas significativas. Por tanto, entre el instante de detección y el de pérdida se dispondrá de un conjunto de vistas significativas, respecto al contexto, del objeto visual en su evolución espacio-temporal. Esas vistas se corresponderán a un muestreo del concepto visual primario que representan a un objeto visual significativo en el contexto actual del sistema. Si el banco de patrones o vistas se suministra a un módulo superior, éste recibirá la definición del concepto visual correspondiente, que puede ser elaborada convenientemente para generar un Asentamiento del Concepto Perceptual correspondiente. Dicho módulo puede estar constituido arquitectónicamente según diversas concepciones. Por ejemplo, según un modelo conexionista (Greco et al., 1998), que se adapta bien al tratamiento en niveles subsimbólicos para la realización de abstracciones.

Por tanto, la utilidad de un módulo de seguimiento desarrollado según estas ideas resulta clara para la construcción de sistemas dotados, no solo de robustez en el seguimiento, sino también de aprendizaje de conceptos visuales.

## Capítulo 3

# Búsqueda de patrones

La búsqueda de patrones <sup>1</sup> es una técnica de visión por computador ampliamente utilizada para la identificación o localización de elementos u objetos en imágenes en diversos campos como: robótica, imágenes médicas, astronomía, proceso de señal de radar, etc. En esta técnica, un patrón (definido normalmente como una imagen de dimensiones reducidas que contiene una representación de un objeto o parte de un objeto) se busca sobre una imagen o parte de esa imagen de dimensiones mayores. En el caso de buscarse sólo sobre parte de la imagen a ésta se la denomina *Ventana de Búsqueda*. Para realizar la búsqueda se desplaza el patrón sobre la imagen aplicando una función de coincidencia que tomará un valor dependiendo del grado de semejanza entre patrón e imagen.

Entre las diversas formas existentes en la bibliografía para buscar semejanzas entre el patrón y la imagen, se puede establecer una primera clasificación entre las técnicas de búsqueda en dos grandes grupos. En un primer grupo se encuentran aquellos procedimientos que usan matrices bidimensionales de pixels sobre los que realiza algún tipo de comparación entre los pixels del patrón y de la ventana de búsqueda. En un segundo grupo se encuentran aquellas técnicas que llevan a cabo un refinamiento mayor dado que tanto dentro del patrón como en la ventana de búsqueda se realiza una extracción previa de características que son las que luego se han de comparar. Estos grupos se denominan como técnicas de búsqueda basadas en áreas y técnicas de búsqueda basadas en características respectivamente.

---

<sup>1</sup>Entendemos "búsqueda de patrones" en el contexto de este documento como la traducción del término anglosajón "pattern matching".

### 3.1. Búsqueda basada en áreas

Los algoritmos de búsqueda de patrones basados en áreas son métodos desarrollados a partir de técnicas para el proceso de señal y fundamentados en medidas estadísticas. Son además la base en la que se inspiran los métodos más complejos basados en búsqueda de características.

Su funcionamiento básico se ilustra en la Figura 3.1. Un patrón se desplaza a lo largo y ancho de la ventana de búsqueda y a medida que recorre todos los pixels se genera un valor dado por una función de similitud que evalúa el parecido existente entre el patrón y la región de la imagen sobre la que se encuentra éste, la cual será de igual tamaño que el patrón. Al final del proceso, el punto con valor máximo corresponderá con la zona de la ventana de búsqueda más parecida al patrón.

Existen en la bibliografía múltiples propuestas que mejoran sustancialmente dos aspectos fundamentales en la búsqueda: robustez frente a las transformaciones visuales del objeto de interés e incremento de la velocidad de procesado. Estas mejoras dan lugar a un esquema de funcionamiento de los procedimientos de búsqueda basados en áreas que se puede dividir en cuatro etapas sucesivas:

- *Medidas de similitud.* Consisten en las funciones que determinan la similitud o disimilitud entre patrón e imagen. Diferentes propuestas sobre la forma de computar la similitud conllevan a una mejora de la velocidad de procesado.
- *Actualización del patrón.* Comprenden los diferentes criterios sobre cuándo actualizar el patrón actual por otro nuevo, incidiendo de forma directa en la robustez y adaptabilidad de la búsqueda.
- *Adecuación del patrón.* Antes de llevar a cabo la comparación de cualquier imagen y/o patrón es posible realizar algún tipo de ajuste previo en sus niveles de gris o color con objeto de eliminar efectos debidos a las condiciones ambientales donde se realiza el seguimiento.
- *Estrategias de búsqueda.* Dado que los procesos de búsqueda de patrones requieren una alta capacidad computacional, estas estrategias persiguen la disminución de los cálculos necesarios mediante la introducción de diferentes heurísticas de recorrido sobre la ventana de búsqueda.

A continuación se describen diferentes aproximaciones existentes en la bibliografía sobre cada una de las etapas expuestas.

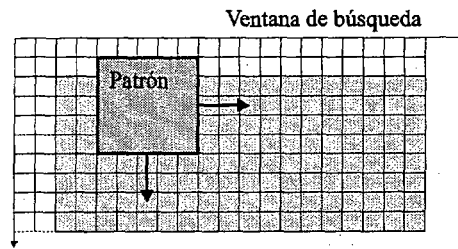


Figura 3.1: Búsqueda del patrón sobre la región de interés. La zona en gris de la región de interés corresponde a los puntos donde se calculó un valor de la función de similitud.

### 3.2. Medidas de similitud

La definición de similitud<sup>2</sup> entre patrón y ventana de búsqueda es primordial dentro de los algoritmos de búsqueda. La función de similitud establece una métrica que evalúa el parecido entre imágenes. Normalmente esta medida se hace 0 cuando ambas imágenes son iguales y se incrementa al aumentar la diferencia entre ellas con lo que formalmente se debería de hablar de una medida de disimilitud. Hagedoorn (Hagedoorn, 2000) ha realizado un extenso trabajo dedicado al estudio de la búsqueda de patrones usando diversas medidas de similitud.

Antes de definir formalmente el concepto de similitud entre objetos, se ha de considerar primeramente el concepto de distancia como paso previo, (Spath, 1980). Sea un conjunto, finito o infinito, de elementos  $U \subseteq \mathbf{R}^n$  que se corresponden con los vectores de características, en el campo real  $\mathbf{R}$ , asociadas a las muestras de un problema dado. Se denomina función de distancia  $D$  a una correspondencia  $D : U * U \rightarrow \mathbf{R}$ , la cual para todo par arbitrario de elementos  $x, y \in U$  se cumple que:

- 1)  $D(x, y) \geq D_0$
- 2)  $D(x, x) = D_0$
- 3)  $D(x, y) = D(y, x)$

Donde  $D_0$  es un número real finito arbitrario, que incluso puede ser negativo. La primera propiedad indica que la función de distancia posee una cota inferior. La segunda de las propiedades obliga a que la distancia sea mínima para el caso en que los elementos sean idénticos. La tercera es una propiedad de simetría, en caso de no cumplirse, la función pasaría ser una función de pseudodistancia. Se dice que la función de distancia es *métrica* si además cumple las siguientes propiedades:

<sup>2</sup>En numerosos textos en español se traduce el término anglosajón "similarity" como similaridad. Sin embargo, tal término es inexistente en el idioma español, por lo que su traducción correcta es similitud.

- 4)  $D(x, y) = D_0 \Rightarrow x = y$
- 5)  $\forall x, y, z \in U : D(x, z) \leq D(x, y) + D(y, z)$

La cuarta propiedad obliga a que siempre que el funcional sea mínimo, los elementos deben ser idénticos. La quinta propiedad corresponde a la desigualdad triangular. Si además  $D_0 = 0$  se obtiene el concepto de métrica del análisis funcional.

Una vez definido el concepto de distancia y dado que se ha planteado partiendo de vectores de características de los objetos, es posible establecer el concepto de disimilitud entre dos objetos como la distancia que los separa tal y como se ha definido previamente. Por tanto, las funciones de distancia asignan un valor numérico a la noción de disimilitud entre dos formas.

El concepto de similitud se entiende como el parecido entre dos objetos, de tal forma que a mayor similitud menor distancia. Formalmente y por analogía con la definición de distancia o disimilitud se define la similitud como la correspondencia  $S : U * U \rightarrow \mathbf{R}$  con las propiedades siguientes:

- 1)  $S(x, y) \leq S_0$
- 2)  $S(x, y) = S_0$
- 3)  $S(x, y) = S(y, x)$

Donde  $S_0$  es una cota superior a cuyo valor se llega únicamente cuando dos objetos son iguales.

La relación, entonces, entre distancia y similitud es cualitativamente inversa pero funcionalmente indeterminada, de tal forma que:

$$1) D(x, y) > D(x, z) \Leftrightarrow S(x, y) < S(x, z)$$

Aunque el concepto de similitud está presente a lo largo de éste y muchos otros trabajos, las funciones usadas para llevar a cabo las implementaciones y cálculos reales suelen ser las de distancia o disimilitud. Esto es así puesto que resulta más cómodo e intuitivo asignar el valor 0 a dos objetos iguales e incrementar ese valor a medida que los objetos se diferencien. Debido a esto y por simplicidad se utiliza el término *similitud* para referenciar o englobar tanto a la propia similitud (p.e. la correlación) como al concepto de disimilitud (p.e. el error cuadrático) y distancia.

Habiéndose entonces definido el concepto de similitud y matizado su uso, corresponde ahora el análisis de los distintos tipos de funciones utilizadas para su cálculo.

Cox (Cox, 1995) y Brown (Brown, 1992) presentan sendas revisiones sobre distintas funciones para el cálculo de distancias. Una taxonomía de estas medidas de distancia se ha llevado a cabo mediante su división en cuatro categorías: Medidas de correlación, medidas de distancia de intensidad entre pixels, medidas de detección de similitud secuencial y criterios de cambio de signo.

### 3.2.1. Medidas de correlación

El término *medidas de correlación* es usado para designar muchas más técnicas de búsqueda que las propiamente de correlación. Este término se ha generalizado debido a que las primeras técnicas de búsqueda hacían uso de las funciones de correlación unidimensional propias del proceso de señal. La correlación ha sido usada en visión por computador desde sus comienzos (Pratt, 1974; Moravec, 1977). Tomándose  $I$  como la matriz de puntos de la imagen y  $P$  la del patrón, la correlación entre ambas se define como:

$$CC(u, v) = \sum_{j=0}^{sizey} \sum_{i=0}^{sizex} I(u+i, v+j)P(i, j) \quad (3.1)$$

la cual es la sumatoria de los productos de los valores de los pixels del patrón y de la imagen. Esta función genera un rango de valores cuyo máximo corresponde al punto de mayor similitud.

La correlación cuenta entre sus ventajas con la de poder calcularse en el dominio de la frecuencia muy rápidamente, ya que una operación computacionalmente costosa de tipo convolución es transformada en una simple multiplicación en el dominio frecuencial. Entre sus inconvenientes cabe destacar que no siempre un máximo valor corresponde con una similitud real entre imagen y patrón. En ciertos casos extremos, imágenes con valores muy altos, es decir, muy claras, luminosas o incluso saturadas en gran medida pueden dar resultados mayores que los de la propia coincidencia. Para intentar resolver estos problemas, debidos a los distintos niveles de luminosidad entre patrón e imagen, se introducen distintos tipos de normalizaciones de la medida obteniéndose entonces *correlaciones normalizadas*.

#### Correlación normalizada

En este caso el valor de la correlación limita los posibles valores al intervalo cerrado  $[0,1]$  con la condición de que ninguno de los valores de la imagen ni del patrón

puedan ser cero.

$$\begin{aligned}
 E_I^2(u, v) &= \sum_{j=0}^{sizey} \sum_{i=0}^{sizex} I^2(u+i, v+j) \\
 E_P^2 &= \sum_{j=0}^{sizey} \sum_{i=0}^{sizex} P^2(i, j) \\
 NCC(u, v) &= \frac{CC(u, v)}{\sqrt{E_I^2(u, v) \cdot E_P^2}} \quad (3.2)
 \end{aligned}$$

Desafortunadamente, esta normalización impide una transformación directa al dominio frecuencial.

### Coefficiente de correlación

La covarianza normalizada corresponde con el coeficiente de correlación lineal estadístico.

$$\bar{I}(u, v) = \frac{1}{sizex \cdot sizey} \sum_A I(u+i, v+j)$$

donde

$$\begin{aligned}
 \sum_A &= \sum_{j=0}^{sizey} \sum_{i=0}^{sizex} \\
 \bar{P} &= \frac{1}{sizex \cdot sizey} \sum_A P(i, j) \\
 ZNCC(u, v) &= \frac{\sum_A^{sizex} [(I(u+i, v+j) - \bar{I}(u, v)) \cdot (P(i, j) - \bar{P})]}{\sqrt{\sum_A^{sizex} [(I(u+i, v+j) - \bar{I}(u, v))^2] \cdot \sum_A [(P(i, j) - \bar{P})^2]}} \quad (3.3)
 \end{aligned}$$

Esta medida es también llamada *correlación normalizada de media cero*. El esquema de normalización es el mismo que en la correlación normalizada, sin embargo, antes de que se realice ninguna otra operación se substraen, tanto en la imagen como en el patrón, las respectivas medias previamente calculadas. Este tipo de medidas de similitud mejoran el comportamiento frente a posibles cambios de luminosidad.

### 3.2.2. Medidas de distancia de intensidad

Estas clases de medidas se basan en el cálculo aditivo de las diferencias en intensidad entre los pixels de la imagen y el patrón. Muchas de estas medidas son equivalentes unas con otras, sin embargo el criterio de utilización de una u otra en particular radica en el tipo y características especiales del procesador que las compute. Es decir, existen



procesadores que incluyen entre su juego de instrucciones alguna instrucción que realiza directamente, y por tanto en pocos ciclos de reloj, una operación necesaria para este cálculo.

### Suma del valor absoluto de las diferencias

Se calcula como el valor absoluto de la diferencia entre los valores de los pares de pixels del patrón y la imagen. También se denomina métrica  $L_1$ .

$$D_{svad}(u, v) = \sum_A |I(u + i, v + j) - P(i, j)| \quad (3.4)$$

### Suma del cuadrado de las diferencias

Su formación es similar a la de una métrica  $L_2$ , sin embargo no se precisa la raíz cuadrada del resultado.

$$D_{scd}(u, v) = \sum_A (I(u + i, v + j) - P(i, j))^2 \quad (3.5)$$

### Raíz del error cuadrático promedio

Esta métrica es equivalente a la anterior, sin embargo requiere algo más de computación.

$$D_{rcm}(u, v) = \sqrt{\frac{1}{n} \sum_A (I(u + i, v + j) - P(i, j))^2} \quad (3.6)$$

### Varianza de las diferencias

Esta distancia es la varianza de las diferencias pixel a pixel entre la imagen y el patrón.

$$D_{vd}(u, v) = \frac{n \sum_A (I(u + i, v + j) - P(i, j))^2 - (\sum_A I(u + i, v + j) - P(i, j))^2}{n^2} \quad (3.7)$$

### Varianza de las diferencias cuadradas

Esta distancia es un versión al cuadrado de la varianza de las diferencias.

$$D_{vdc}(u, v) = \frac{n \sum_A (I(u+i, v+j) - P(i, j))^4 - (\sum_A (I(u+i, v+j) - P(i, j))^2)^2}{n^2} \quad (3.8)$$

### Varianza del valor absoluto de las diferencias

Es la versión en valor absoluto de la varianza de diferencias.

$$D_{vva}(u, v) = \frac{n \sum_A |I(u+i, v+j) - P(i, j)|^2 - (\sum_A |I(u+i, v+j) - P(i, j)|)^2}{n^2} \quad (3.9)$$

### Filtrado previo

Algunas de las diferentes variaciones introducidas en las funciones de medida expuestas anteriormente intentan de alguna forma aislar el efecto producido por las variaciones de iluminación que, como ocurre en las funciones de medida más simples, hacen que el patrón y la imagen no se asemejen. Un esquema alternativo a los anteriores consiste en el filtrado previo de las imágenes y del patrón para luego aplicar alguna medida simple de las expuestas previamente. Para llevar a cabo este esquema se utilizan frecuentemente versiones discretas del gradiente o laplacianas. Una vez aplicado alguno de estos filtros se obtiene una imagen de contornos, la cual, en principio, está libre de variaciones producidas por la iluminación. No sólo se evitan efectos aditivos o multiplicativos de esta sino también direccionales, como ilustra la Figura 3.2. Una aproximación al gradiente discreto se expone a continuación:

$$G_x = \frac{\partial f}{\partial x} = f(x-1, y) - f(x+1, y)$$

$$G_y = \frac{\partial f}{\partial y} = f(x, y-1) - f(x, y+1)$$

$$\|G(x, y)\| = \sqrt{G_x^2 + G_y^2}$$

La laplaciana es considerada por muchos investigadores como la mejor transformación para obtener invarianza frente a cambios de la intensidad luminosa. Martin y Crowley (Martin y Crowley, 1995) mostraron analíticamente que sumas pesadas de segundo orden

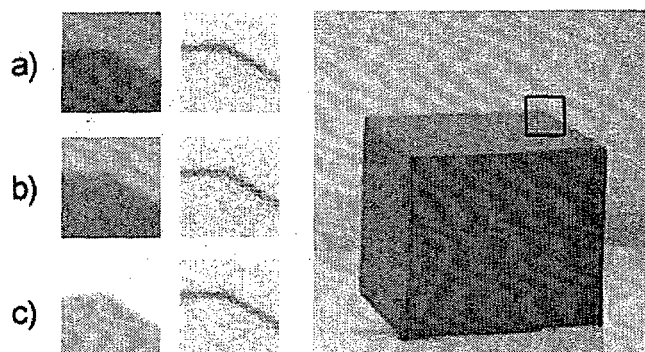


Figura 3.2: La figura muestra como diferentes intensidades luminosas sobre un objeto son convertidas a contornos similares tras la aplicación de una laplaciana. Los tres pares de patrones *a)*, *b)* y *c)* muestran el patrón y su imagen de gradiente respectiva.

a diferentes ángulos proporcionaban una excelente invarianza frente a las rotaciones.

$$\nabla^2 = \begin{vmatrix} -1 & -2 & -1 \\ -2 & 12 & -2 \\ -1 & -2 & -1 \end{vmatrix}$$

Entre otros filtrados previos hay que señalar al SLOG o signo de la laplaciana de la gaussiana, el cual reduce aproximadamente seis veces el tiempo de cómputo de una SSD o suma del cuadrado de diferencias u otras implementaciones de la correlación. Este operador es similar a convolucionar una imagen con una gaussiana y posteriormente con un operador laplaciana, para finalmente extraer los cruces por cero. De esta forma, la imagen resultante es una imagen binaria donde un bit representa un pixel. El filtro SLOG se computa usando una diferencia de gaussianas, en lugar de una laplaciana de la gaussiana, véase Figura 3.3. Esto es debido a que la diferencia de gaussianas permite la separación del kernel, lo cual se transforma en una computación más rápida. Aunque la laplaciana de la gaussiana y la diferencia de gaussina no son numéricamente idénticas sí poseen propiedades similares.

### 3.2.3. Medidas de detección secuencial de similitud

Las medidas de similitud vistas hasta ahora son computacionalmente costosas, las medidas de detección secuencial de similitud fueron propuestas por Barnea y Silverman (Barnea y Silverman, 1972) con el objeto de reducir los tiempos de procesamiento. La

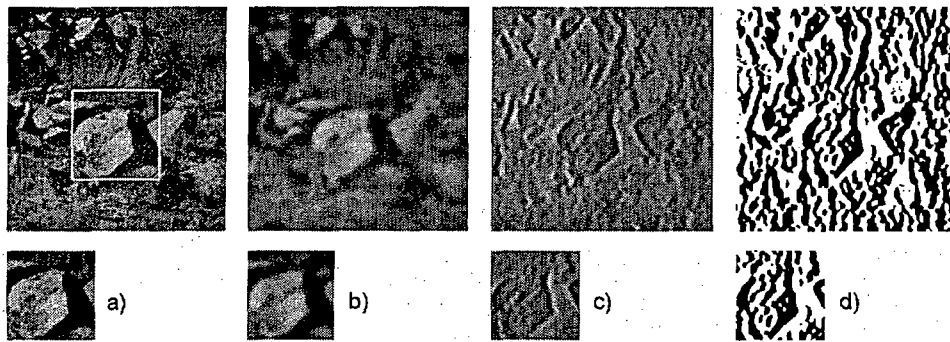


Figura 3.3: Las imágenes superiores representan la región de interés a lo largo de los sucesivos pasos del filtrado. Las inferiores muestran el patrón buscado. La imagen *b*) es la misma imagen que *a*), la original, después de un filtrado paso bajo, esto se realiza para evitar cruces por cero debido a las altas frecuencias que pudieran aparecer como ruido en el resultado final. La imagen *c*) representa la diferencia de gaussianas de la imagen *b*). Finalmente, la imagen *c*) es binarizada haciendo corresponder los pixels con valores positivos con un uno y los negativos con un cero, dando como resultado la imagen *d*).

idea es sencilla; la distancia entre cada par de pixels del patrón y la imagen se considera como un error que se va actualizando sucesivamente a medida que se van calculando más errores. Si este error acumulativo traspasa cierto umbral ya no es necesario seguir actualizando el error y la función de similitud en ese punto toma entonces un valor determinado.

### 3.2.4. Criterios de cambio de signo

Fue propuesto por Venot (Venot et al., 1984). El esquema de detección es el siguiente: supóngase que tanto el patrón como la imagen se diferencian únicamente debido a un ruido aditivo de media cero y distribución simétrica, entonces si se calcula la diferencia pixel a pixel entre estas:

$$D(i, j) = I(i, j) - P(i, j)$$

y se toma la matriz  $D$  como un vector ordenado por filas y columnas, se obtiene que si las dos imágenes se parecen, el número de cambios de signo entre elementos adyacentes de la matriz  $D$  tomada como un vector será elevado. Si las imágenes tienen diferencias mayores que las propias del ruido, el número de cambios de signo será mucho menor. Por tanto, será el número de estos cambios de signo lo que establecerá el grado de similitud entre patrón e imagen.

### 3.3. Actualización del patrón

El objeto a seguir varía normalmente su aspecto a lo largo del tiempo debido tanto a acercamientos, alejamientos, rotaciones y deformaciones del mismo como a cambios en la intensidad luminosa o dirección de la luz. Todas estas circunstancias provocan una alteración de los valores de los pixels de la imagen del objeto de interés. Puesto que estos valores cambian, el patrón original no sería localizado en los sucesivos cuadros, si estos eventos ocurren. Por lo tanto, el patrón ha de adaptarse a tales alteraciones progresivas actualizándose a medida que estas van teniendo lugar. El problema que han de resolver estas estrategias es "cuándo" ha de llevarse a cabo esta actualización del patrón.

Con tal motivo han sido propuestas varias estrategias de actualización de los patrones para tratar los cambios en la imagen del objeto. Entre las más utilizadas en la bibliografía se encuentran: mantenimiento del patrón constante, actualización constante, actualización de periodo constante, actualización mediante umbral constante, actualización basada en estadísticos y la propuesta en esta tesis: la actualización basada en la consideración del efecto del contexto.

Es importante destacar que estas estrategias tan sólo abordan el problema de en qué momento realizar una actualización del patrón y llevarla a cabo tomando el nuevo patrón directamente de la imagen o cuadro actual sin más transformación.

#### 3.3.1. Mantenimiento del patrón constante

Mantener el mismo patrón a lo largo de todo el proceso de seguimiento implica que no puede haber variaciones significativas en el objeto seguido o de lo contrario el objeto se perderá irremediabilmente. Por tanto, este tipo de esquema de mantenimiento del patrón sólo es posible en escenas donde el objeto de interés no altera su aspecto visual, sino tan sólo su posición bidimensional en la imagen mediante transformaciones de traslación.

#### 3.3.2. Actualización constante

La forma de actualización más sencilla consiste en el cambio constante del patrón. Es decir, el patrón es buscado sobre la imagen o cuadro actual. Una vez encontrado, el área de la imagen coincidente o más parecida con el patrón es tomado como nuevo patrón. En otras palabras, se sustituye el patrón por este área de imagen del mismo tamaño del patrón, y el proceso se repite en cada cuadro de la secuencia. La actualización

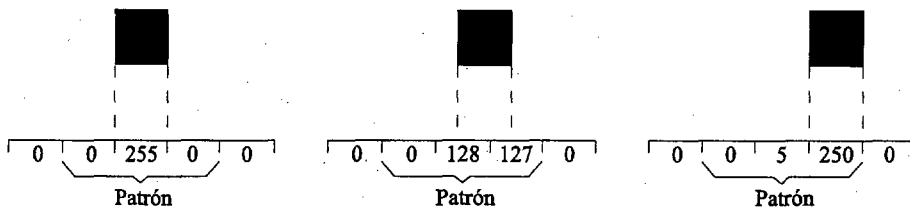


Figura 3.4: Ejemplo de deriva del patrón que se produce por la actualización constante del mismo.

constante conlleva un importante problema asociado. Cuando en el movimiento del objeto relativo a la cámara se dan ciertas circunstancias se provoca una deriva del objeto con la consiguiente pérdida de éste.

Puede verse esto gráficamente mediante la Figura 3.4 donde se expone un caso de deriva muy simplificado. La figura representa una proyección unidimensional de un caso de seguimiento con actualización de patrón constante. El objeto a seguir se muestra con un cuadrado negro cuya proyección en la imagen corresponde con el tamaño de un píxel. En este caso el patrón posee solamente tres píxeles, que en el primer paso tiene los valores  $(0,255,0)$ , puesto que el objeto proyecta su imagen sobre el píxel central del patrón. Si el objeto posee un movimiento relativamente lento hacia la derecha, al siguiente cuadro puede proyectar su imagen aproximadamente a partes iguales entre dos píxeles adyacentes. Al buscar el patrón anterior sobre la nueva imagen se obtiene que la mejor coincidencia se da entre las tripletas  $(0,255,0)$  y  $(0,128,127)$ . Con lo cual el nuevo patrón ahora se convierte en  $(0,128,127)$ . Posteriormente, con su movimiento hacia la derecha, el objeto en el tercer cuadro se proyecta prácticamente sobre un solo píxel  $(0,5,250)$ . Al realizar la búsqueda, la mejor coincidencia del patrón  $(0,128,127)$  se obtiene en el área de la imagen  $(0,5,250)$  con lo que este área pasa a convertirse en el nuevo patrón. Como se ha visto, el objeto ha pasado de estar en el centro del patrón a estar en un margen del mismo. Si el movimiento hacia la derecha del objeto continúa de igual forma, en los próximos cuadros se habrá perdido. Hay que destacar que la posición relativa del patrón dentro de la imagen no se ha movido en los sucesivos cuadros.

Según lo expuesto, en la actualización constante existe un problema de deriva del patrón debido a un efecto de digitalización del objeto, el cual se desplaza a una cierta velocidad en la imagen. No obstante, se han propuesto algunos esquemas de filtrado previo de las imágenes para evitar estos efectos, (Kishore y Rao, 2001).

### 3.3.3. Actualización de periodo constante

Este esquema de actualización puede verse como una generalización del esquema anterior de actualización constante. En este caso, la actualización no se lleva a cabo en cada cuadro de la secuencia sino una vez cada cierto periodo de tiempo constante, esto intenta retrasar, aunque no impide, la aparición de la deriva del patrón ocurrida en la actualización constante (Reynolds, 1998). La principal desventaja que posee este esquema consiste en que las actualizaciones no tienen lugar de una manera oportunista y precisa sino totalmente periódica. Esto hace que puedan darse dos hechos distintos: Por un lado, es posible que se realicen actualizaciones irrelevantes. Es decir, actualizaciones que no son necesarias y que pueden empeorar el comportamiento del seguimiento posibilitando la deriva. Por otra parte, puede darse el caso en que sea absolutamente necesario que deba realizarse una actualización y que el sistema se halle en ese momento en mitad del periodo de actualización, y por tanto esta no se lleve a cabo con la consiguiente pérdida del objeto.

### 3.3.4. Actualización mediante umbral constante

Evitar el problema de la deriva pasa por no actualizar el patrón en cada cuadro, puesto que actualizar el patrón es siempre permitir potencialmente la deriva del mismo. Sin embargo, no actualizarlo nunca conlleva la posible pérdida del objeto en poco tiempo si éste cambia su aspecto con cierta rapidez. Una solución intermedia y muy utilizada consiste en establecer un umbral que permita el cambio sólo cuando el grado de disimilitud lo supere. En otras palabras, el patrón se actualiza cuando el mínimo de la función de similitud traspasa ese valor umbral. La principal dificultad que posee este método radica en la asignación del valor a este umbral, que implica siempre un compromiso entre posible deriva y pérdida. Un valor muy pequeño supone una casi continua actualización, con lo que se tendría de nuevo el problema de deriva anterior. Por el contrario, un valor excesivamente elevado haría que el objeto se perdiera antes de que se hubiese actualizado correctamente. En este tipo de solución el establecimiento de umbrales se realiza basado en algún tipo de heurística, con lo que valores adecuados para un tipo de escenario y condiciones pueden resultar completamente inválidos para otros.

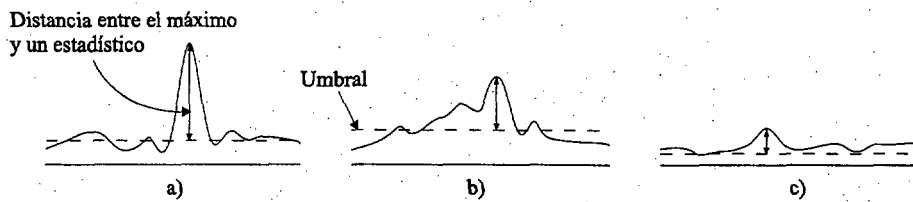


Figura 3.5: La figura muestra tres ejemplos de calidad del máximo. La calidad viene dada por la distancia de éste a un umbral, que ahora se establece mediante un estadístico concreto, por ejemplo la media del resto de los puntos. Cuanto mayor sea esta distancia mayor calidad.

### 3.3.5. Actualización basada en estadísticos

El sólo traspaso de un umbral por parte de un variable concreta, como es el índice de similitud, no contiene información adicional acerca de lo que rodea a ese máximo o mínimo, es tan sólo información puntual. Sin embargo, la calidad de un máximo o mínimo viene condicionada por los valores que le rodean. Por ello, Inoue y otros (Inoue et al., 1992) utilizan una medida del máximo o mínimo con relación a la media del resto de los valores, véase Figura 3.5. Si el máximo (o mínimo) está suficientemente diferenciado de la media del resto de los valores, será un máximo (o mínimo) de calidad. En particular se utiliza una medida de fiabilidad del mínimo, obtenido mediante una correlación, definida como:

$$Fiabilidad = \frac{C_{media} - C_{minimo}}{C_{media}}$$

Donde  $C_{media}$  es la media de todos los valores obtenidos mediante la correlación. De esta forma, la actualización se realiza no por el valor absoluto de un mínimo sino del valor relativo de un mínimo con respecto a su entorno.

### 3.3.6. Actualización basada en contexto

Este método para llevar a cabo la actualización del patrón se encuentra dentro de las aportaciones de este trabajo de tesis. Su fundamento teórico se describe en el Capítulo 2, donde el aprendizaje incremental basado en contexto tiene como particularización la actualización del patrón en la búsqueda de patrones. La exposición exhaustiva de este método se halla en la Sección 3.8 de este mismo capítulo. Su introducción aquí obedece al mantenimiento de una exposición ordenada de estos métodos.



## 3.4. Adecuación del patrón

Una vez tomada la decisión de actualizar el patrón, este nuevo patrón puede ser modificado o generado a partir del propio cuadro actual o de antiguos patrones. Por tanto, el patrón empleado no corresponde en ningún momento a una imagen real del objeto. Entre las transformaciones más comunes aplicadas a los nuevos patrones se encuentran aquellas que tienden a eliminar factores que influyen en la iluminación del mismo, factores de escala, factores de rotación o deformación, etc.

### 3.4.1. Normalización de la iluminación

Estas formas de adaptación del patrón han sido ya implícitamente expuestas en secciones anteriores. Las medidas de similitud que hacen uso de la intensidad media de la imagen para asemejar el patrón a este nuevo nivel de luminosidad son un ejemplo de adecuación del patrón. Debido a que esta adaptación se ocupa exclusivamente de los cambios de iluminación a que se pueda ver sometido el objeto a seguir, este tipo de adecuación va acompañada normalmente por algún método de adaptación a la forma del objeto.

### 3.4.2. Actualización ponderada

Este tipo de actualización puede ser vista como un filtro temporal IIR sobre cada pixel del patrón. Mediante esta aproximación el patrón no es sustituido completamente por uno nuevo, sino que el nuevo patrón se genera como combinación del patrón actual y la información proveniente del nuevo aspecto del objeto dada por la imagen actual, (Reynolds, 1998). Normalmente, esto se hace asignando un porcentaje de la nueva vista al patrón actual. Este porcentaje puede ser constante o puede basarse en un coeficiente de certidumbre  $C$ , que indica qué nivel de certeza existe acerca de que la nueva vista del objeto sea realmente la buscada.

$$\text{Nuevo patrón} = \text{Patrón actual}(1 - C) + \text{Nueva vista}(C)$$

El nivel de certidumbre está normalizado en el intervalo  $[0, 1]$ .

### 3.4.3. Modificación mediante transformaciones afines

El patrón puede ser transformado para adaptarse al nuevo aspecto que tome la imagen del objeto debido a ciertas deformaciones. Para ello, el uso de transformaciones afines sobre el patrón consigue actualizarlo satisfactoriamente sólo bajo ciertas circunstancias. En el caso de objetos bidimensionales o proyecciones ortográficas típicas de escenas aéreas, los resultados pueden ser válidos (Hager y Belhumeur, 1996). Es posible también las transformaciones más complejas mediante el uso de hiperplanos que permitan mayor libertad de movimientos a objeto de interés planos (Jurie y Dhome, 2001; Jurie y Dhome, 2002). Sin embargo, muchas de las transformaciones que sufren los objetos situados en escenas de interiores a las distancias habituales en estos contextos no pueden ser resueltas mediante este tipo de transformaciones afines, ya que los objetos no suelen ser bidimensionales y en numerosas ocasiones sus rotaciones se efectúan con un eje de rotación orientado hacia el plano de imagen de la cámara.

### 3.4.4. Patrones deformables

Un paso más allá en la adecuación del patrón para su búsqueda consiste en la deformación del mismo con el fin de obtener un modelo con la forma o contorno aproximado del objeto de interés, (Tagare, 1997). En estos casos el patrón deja de ser rectangular para adoptar una forma curva arbitraria. En otros casos se emplean mallas triangulares que, a modo de regiones activas (active blobs), se adaptan dinámicamente al objeto de interés, (Sclaroff y Isidoro, 1998).

### 3.4.5. Búsqueda de múltiples patrones

Varios patrones pueden ser usados simultáneamente para seguir el mismo objeto. En este caso los diferentes patrones se ajustan a diferentes partes de ese objeto con el propósito de aumentar la robustez global. Balkenius y Kopp (Blakenius y Kopp, 1996; Balkenius y Kopp, 1997) desarrollaron un método basado en varios patrones para seguir partes del mismo objeto. De esta forma, sus posiciones relativas entre ellos configuran una nueva pista para el seguimiento.

## 3.5. Estrategias de búsqueda

Uno de los principales inconvenientes que poseen los métodos de búsqueda de patrones basados en áreas es su coste computacional. Por ello, han sido numerosas las diferentes propuestas realizadas por los investigadores para reducir este coste evitando la búsqueda exhaustiva del patrón en la imagen. Básicamente, esta reducción se lleva a cabo o bien aplicando ciertas heurísticas para hacer el barrido de la búsqueda o bien mediante el establecimiento de diferentes modelos de tamaño del patrón e imagen. A continuación se presentan algunas de estas soluciones:

### 3.5.1. Barrido espiral

Esta estrategia parte de la suposición de que durante el seguimiento el objeto variará su posición ligeramente entre sucesivos cuadros por lo que la búsqueda sobre la imagen no comienza en la esquina superior izquierda de esta sino en la última posición detectada del objeto, es decir de su posición en el cuadro anterior. Luego, el recorrido que se lleva a cabo durante la búsqueda sigue un camino en espiral (Doi y Hashimoto, 2001). En cuanto la función de similitud devuelve un resultado aceptable, el procedimiento finaliza.

### 3.5.2. Líneas epipolares

Este tipo de estrategia se realiza fundamentalmente en la búsqueda de patrones en imágenes estereoscópicas. Cuando se conoce la dirección y orientación de las cámaras o par estereoscópico es posible establecer una correspondencia entre un objeto que se encuentra en una de las imágenes y la recta o lugar geométrico donde se debe encontrar ese mismo objeto en la otra imagen del par (Faugeras, 1993). Con lo cual, el problema se reduce de una búsqueda bidimensional a una búsqueda unidimensional. Esta reducción es crucial en entornos estereoscópicos para aminorar el costo computacional y, sobre todo, para evitar posibles ambigüedades durante la búsqueda. Ver Figura 3.6.

### 3.5.3. Filtros estimadores

Los filtros estimadores realizan aproximaciones basadas en la "teoría de sistemas", es decir, consideran al objeto y su movimiento como un sistema con sus propias variables. Para ello se crea un modelo, el cual puede ser muy simple o estar muy adaptado a algún

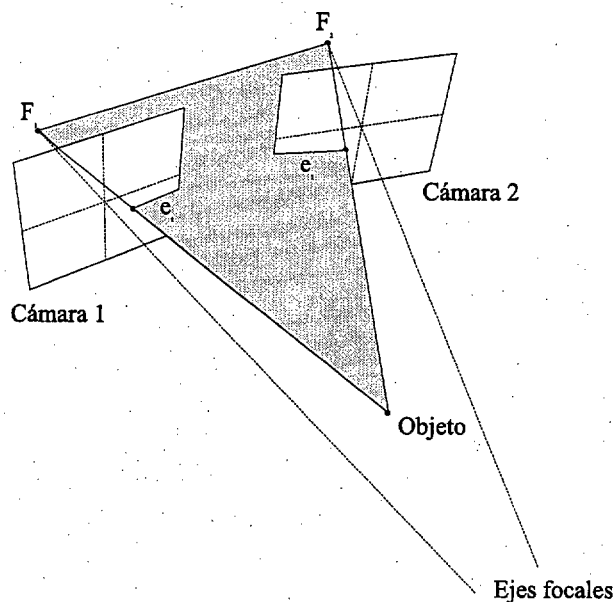


Figura 3.6: El plano formado por el objeto y los puntos focales  $F_1$  y  $F_2$  se conoce como plano epipolar. La intersección de este plano con cada uno de los planos focales de las dos cámaras originan las líneas epipolares  $e_1$  y  $e_2$ .

tipo de movimiento concreto, como parábolas (para objetos que caen por efecto de la gravedad), u otros. Existen numerosos tipos de filtros que pueden ser clasificados básicamente en cuatro categorías: Filtros de primer orden variantes en el tiempo, filtros de orden reducido, filtros de Kalman y filtros no lineales. De entre estos filtros, es el de Kalman el más empleado (Kalman, 1960). El filtro de Kalman es una técnica de estimación recursiva, basada en la predicción y actualización, para seguir un conjunto  $N$  de propiedades visuales de un objeto móvil. Cada una de las  $N$  propiedades se definen mediante vectores de estado. Generalmente, se emplean dos vectores de estado (Gil et al., 1994), denominados  $X_k^n$  y  $Y_k^n$  que representan respectivamente la posición y la velocidad. Para cada uno de los vectores de estado se definen un conjunto de ecuaciones. En el caso de la posición, el vector de estado se refiere a las coordenadas estimadas 2D de las  $N$  propiedades en la imagen, es decir:

$$\hat{X}_k = (X_k^1, Y_k^1, X_k^2, Y_k^2, \dots, X_k^n, Y_k^n)^T$$

El filtro de Kalman se basa en una estrategia de tres etapas más una inicialización previa, como se muestra en la Figura 3.7. Estas etapas son:

- Predicción, en este paso se lleva a cabo, según el estado de las variables actuales del filtro, la estimación del objeto para el instante actual.

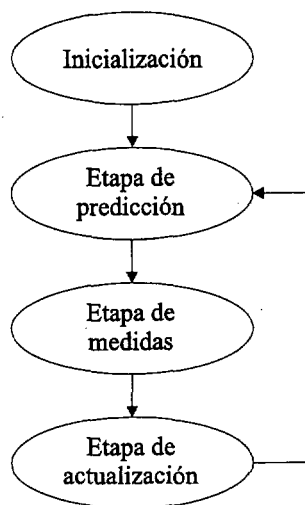


Figura 3.7: Etapas del filtro de Kalman.

- Medición, una vez obtenida la posición y demás variables reales del objeto seguido se realiza una comparación con las estimaciones previas para tener el error de medida.
- Actualización, el error de medida previamente calculado permite la actualización de las distintas variables del filtro con el fin de realizar una corrección del mismo.

#### 3.5.4. Actualización del ganador, "Winner-Update"

Este tipo de recorrido realiza una búsqueda progresiva del objeto de interés por medio de la actualización de los valores que la función de error vaya teniendo en cada una de las posiciones del área de búsqueda definida, (Chen et al., 2000; Chan-Hung, 2000). En búsquedas de patrones en imágenes esta estrategia de búsqueda consigue una rebaja considerable del tiempo de cómputo normal requerido.

#### 3.5.5. Jerarquías

Los métodos jerárquicos se basan en la utilización de imágenes a diferentes resoluciones, normalmente en una progresión de un factor  $\times 2$ , (Arbeiter y Bessler, 1986; Kishore y Rao, 2000). Esto genera un estructura de imágenes piramidal en donde se realiza una aproximación burda o inicial sobre la de menor resolución hasta que se obtiene, mediante las sucesivas imágenes a resoluciones mayores, un localización precisa del objeto buscado en la última imagen. Esto es equivalente a un submuestreo en métodos de



Figura 3.8: La figura representa a una misma imagen a resoluciones distintas. La búsqueda del objeto se realiza partiendo de la imagen de menor resolución (imagen más pequeña) donde se localiza burdamente el área donde se encuentra el objeto de interés para progresivamente ir refinando esta localización en las sucesivas imágenes de mayor resolución.

búsqueda basados en áreas donde la función de similitud es aplicada cada 1, 2, 4, 8, etc. pixels, para volver posteriormente sobre el área que generó los mejores índices y buscar con un muestreo menor. Ver Figura 3.8.

### 3.5.6. Transformación log-polar

La transformación log-polar muestra una nueva forma de distribución de los pixels de la imagen. En este caso los pixels son distribuidos de una manera 'foveal' (Panerai et al., 1995; Guerra, 1996). En los sistemas visuales biológicos la fóvea es la región de la retina donde se encuentra una concentración mayor de células fotorreceptoras. Tomando esta disposición biológica los pixels son distribuidos de una forma incremental desde el perímetro hasta el centro del campo visual. Básicamente, la transformación log-polar establece una correspondencia desde pixels de un plano cartesiano hacia pixels en un plano logarítmico-polar. Su motivación viene inspirada por los sistemas visuales biológicos donde los movimientos oculares tratan de mantener al objeto de interés siempre centrado en la retina, de tal forma que sólo se aprecian los detalles del objeto que se sitúan justo en la fóvea, (Bernardino y Santos-Victor, 1999). Esta correspondencia se define como:

$$(\xi, \eta) = (\log(\rho), \theta) \quad (3.10)$$

donde  $\rho$  y  $\theta$  son las coordenadas log-polar de los pixels en el plano cartesiano. Debido a la expresión logarítmica que conforma la primera coordenada, existe una singularidad en el punto  $(0, 0)$ , por tanto la correspondencia log-polar no está definida en la vecindad

del centro de coordenadas. De modo que la coordenada  $\xi$ , en imágenes discretas, sólo está definida en el intervalo:

$$\xi = [\log(\rho_{min}), \log(\rho_{max})]$$

Siguiendo esta nueva distribución, aplicar un algoritmo de búsqueda tipo *suma de diferencias cuadradas* o SSD entre imágenes log-polar supone algunos efectos sobre los resultados. Como se puede ver en la expresión 3.10 y en la Figura 3.9, un píxel de la parte baja de la imagen log-polar corresponde con varios píxels en la imagen cartesiana y píxels en la parte superior pueden tener una correspondencia uno a uno entre ambas imágenes. Debido a este hecho, cada píxel en la imagen log-polar posee un peso implícito. Luego un algoritmo de búsqueda tipo SSD en una imagen log-polar corresponde con un algoritmo SSD sobre una imagen cartesiana con una distribución de pesos en los diferentes píxels. Según (Bernardino y Santos-Victor, 1999), si  $I'_1$  e  $I'_2$  son dos imágenes log-polar, e  $I_1$  e  $I_2$  dos imágenes cartesianas, entonces la aplicación de un algoritmo SSD sobre  $I'_1$  e  $I'_2$  es equivalente a un algoritmo SSD ponderado sobre  $I_1$  e  $I_2$ .

$$SSD(I'_1, I'_2) = SSD\left(\frac{I_1(x, y)}{\sqrt{x^2 + y^2}}, \frac{I_2(x, y)}{\sqrt{x^2 + y^2}}\right) \quad (3.11)$$

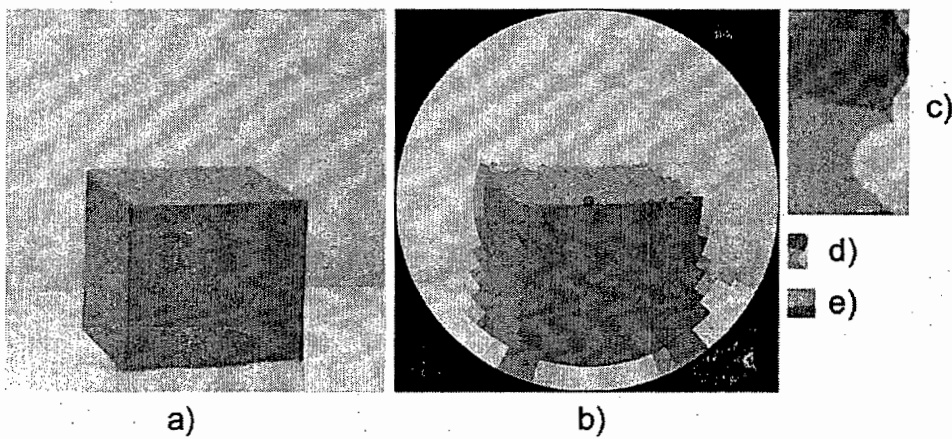


Figura 3.9: La imagen *a)* muestra una distribución cartesiana o normal de un cubo. La imagen *d)* representa el mismo cubo reducido a una distribución log-polar. Debido a su tamaño proporcional con respecto a *a)*, la imagen *c)* muestra una ampliación de *d)*. La imagen *b)* muestra una imagen reconstruida a partir de *d)*. El lado izquierdo de la imagen *c)* corresponde al área central de la imagen *b)* y el lado derecho corresponde a la periferia. Finalmente, *e)* es un fragmento de *a)* con el mismo número de pixels que *d)* expuesto aquí por razones de comparación únicamente. La transformación log-polar también disminuye el tamaño de la imagen original reduciendo progresivamente la resolución desde la región central hacia la periferia.



### 3.6. Resumen de la clasificación

Como resumen de la clasificación expuesta, la Figura 3.10 muestra esquemáticamente las etapas en que puede ser dividido un proceso de seguimiento mediante la búsqueda de patrones basado en áreas y los posibles algoritmos o medidas disponibles en cada una de ellas. En primer lugar se encuentran las medidas de similitud, las cuales proporcionan unas coordenadas correspondientes a donde se ha encontrado el objeto seguido y, en ocasiones, un valor de confianza o probabilidad acerca de que el objeto encontrado sea realmente el objeto de interés seguido. También proporcionan información sobre el entorno del objeto de interés. Una vez encontrado el objeto de interés basado en el parecido a un patrón de referencia, este patrón debe ser actualizado para acomodarse al nuevo aspecto visual del objeto. La etapa de actualización puede considerarse como la más crítica dentro del seguimiento puesto que es donde la posibilidad de error y pérdida del objeto es mayor. Dentro de los métodos de actualización se encuentra la actualización basada en contexto que forma parte de las aportaciones del presente trabajo. Dentro de los métodos de adecuación del patrón tiene lugar la manipulación del patrón e imagen con el fin de añadir robustez al proceso de búsqueda. Su inclusión dentro de un proceso de seguimiento no es estrictamente necesaria. Por último se encuentra la fase o estrategias de búsqueda cuya función principal es la de optimizar los recursos computacionales para disminuir la alta demanda computacional de estos métodos de búsqueda. Es oportuno comentar que cada método o estrategia de las distintas fases del procedimiento completo de búsqueda es cooperante con los métodos o estrategias de las otras fases. Es decir, cualquier medida de similitud puede trabajar con cualquier método de actualización, y a su vez, cooperar con algún método de adecuación del patrón con una determinada estrategia de búsqueda.

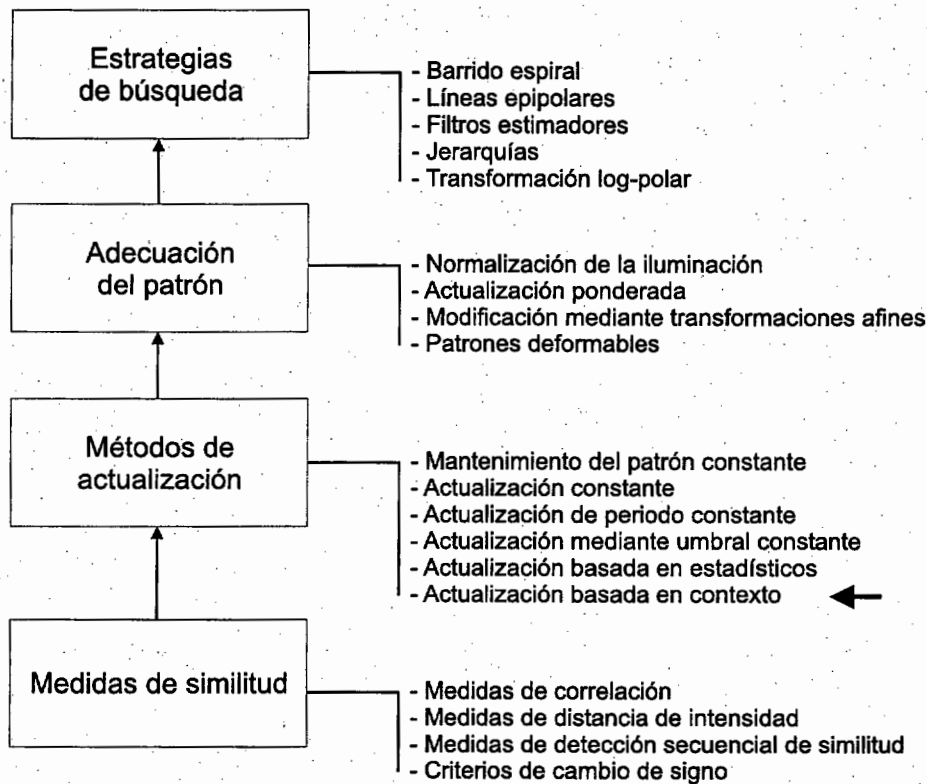


Figura 3.10: Clasificación de las etapas en que puede ser dividido un proceso de seguimiento mediante la búsqueda de patrones basado en áreas. La entrada marcada con una flecha señala la fase del procedimiento de búsqueda en la que se encuentra ubicado el método de actualización propuesto en esta tesis.

### 3.7. Búsqueda basada en características

Este tipo de búsqueda se basa en la extracción de características o primitivas destacadas del objeto de interés en la imagen mediante algún operador. Una vez almacenadas son buscadas en las sucesivas imágenes de la escena, intentando establecer una correspondencia entre las características almacenadas y las del objeto de interés que se encuentra en la nueva imagen.

En principio, este tipo de técnicas mejoran la robustez y flexibilidad del seguimiento al permitir ciertas transformaciones geométricas como rotaciones y cambios sensibles en la iluminación. Así mismo, dependiendo del nivel de elaboración de las primitivas se permite algún grado de ocultación del objeto. Sobre el objeto de interés se crea un modelo de objeto que es conformado por las diversas primitivas geométricas que se extraen de él. El problema en cada aplicación particular radica en saber qué primitivas son las idóneas o, al menos, las más factibles. La elección de las primitivas que definen el modelo deben cumplir una serie de requisitos mínimos para que el proceso de seguimiento pueda ser realizado eficientemente:

- *Consistencia*: las primitivas que se van a usar, deben ser detectadas consistentemente en el dominio espacio-temporal, lo que implica que deben ser lo suficientemente representativas y estables para que puedan ser localizadas en cada escena.
- *Complejidad*: la complejidad computacional es un factor muy importante en aplicaciones de tiempo real. La elección de primitivas complejas repercute directamente en el tiempo de cómputo necesario para detectarlas.

Existen una serie de factores que dificultan el proceso del seguimiento basado en características. Entre estos podemos destacar los siguientes como los más importantes:

- La presencia de otros objetos móviles o estáticos, ya que esto da lugar a entornos complejos que generan gran cantidad de primitivas, ralentizando la tarea del establecimiento de la correspondencia. Además se debe tomar en cuenta la presencia de primitivas generadas debido al ruido.
- La posibilidad de que parte del objeto aparezca oculto por el resto de los elementos de la imagen. Mientras un número suficiente de características o primitivas del objeto buscado se encuentren visibles, la detección y posterior seguimiento de éste es posible.

- La ausencia de primitivas del objeto debido a fallos en el proceso de su detección. Este tipo de fallos puede deberse a múltiples factores, como iluminación deficiente o borrosidad del objeto debido a su propio movimiento.
- La imprecisión en las posiciones de las primitivas debido a la inexactitud en los procesos de detección.
- Las diferentes posibles deformaciones de los objetos no rígidos pueden causar la pérdida de posiciones de referencia relativas entre las distintas primitivas del objeto.
- El uso de cámaras móviles puede conllevar la pérdida de primitivas por el emborronamiento en la imagen producido por su propio egomovimiento junto a una baja velocidad de obturación en la adquisición.

Los modelos bidimensionales de los objetos se construyen generalmente a partir de primitivas locales que describen propiedades más complejas del objeto que las que se pueden obtener a partir de primitivas globales. Entre las diferentes primitivas locales que constituyen los diferentes modelos bidimensionales destacan contornos, esquinas, segmentos de línea, curvas y puntos interesantes. Las primitivas globales no se suelen emplear en métodos de seguimiento, ya que el objeto que se está moviendo, normalmente modifica su forma por lo que, características tales como área o perímetro no son medidas consistentes para localizar y seguir un objeto móvil. Aquellos métodos que emplean primitivas locales para realizar el seguimiento de un objeto móvil, se les conoce con el nombre de *Métodos Estructurales*.

Definir una medida de similitud para este tipo de búsqueda es más complicado ya que la medida debe estar basada en los atributos de las características. En la mayoría de los algoritmos de búsqueda basados en características, las diferencias geométricas se combinan con heurísticas y umbrales para computar la función de similitud, denominada normalmente como función de coste o beneficio. Mientras que una función de costo debe ser minimizada, una función de beneficio debe ser maximizada para lograr una buena coincidencia.

Entre algunas de las técnicas mejor estudiadas para realizar seguimiento de objetos mediante la búsqueda de características se pueden destacar:

## Hausdorff

La distancia de Hausdorff es una medida definida entre dos conjuntos de puntos, en este caso imagen y patrón. Esta medida es robusta frente a la presencia de incertidumbre como puntos lejanos o ruido. Al contrario de otras aproximaciones, la distancia de Hausdorff no establece una correspondencia uno a uno, sino que es una medida de proximidad entre conjuntos de puntos, (Sánchez-Nielsen y Hernández, 2000).

## Snakes

El desarrollo de los contornos activos surge del trabajo de Kass y otros (Kass et al., 1987). Entre sus posibles aplicaciones se encuentran la segmentación de imágenes y el seguimiento de objetos no rígidos. El modelo de contornos activos o *snakes* emplea modelos dinámicos y elásticos y una combinación de fuerzas basada en un esquema de minimización de energía con la idea de ajustar el contorno activo a una cierta disposición de la imagen o mapa de propiedades. El mínimo local de esta función de energía corresponde a las propiedades del objeto buscado.

## Histogramas de color

La información proporcionada por los histogramas de color ha sido ampliamente utilizada debido a su robustez y eficiencia localizando e identificando objetos incluso cuando sus formas no son rígidas. Swain y Ballard (Swain y Ballard, 1991) desarrollaron un procedimiento basado en la retroproyección del histograma de color para una rápida localización del objeto buscado. Entre los inconvenientes que poseen las técnicas basadas en histogramas de color cabe destacar su alta sensibilidad a cambios de iluminación.

## Líneas

La detección de líneas ha sido una de las principales características a buscar en imágenes mediante la visión por computador. La literatura ofrece múltiples métodos de detección, como la transformada de Hough (Illingworth y Kittler, 1987) o la detección de lados de Canny (Canny, 1986). Los segmentos de líneas rectas pueden utilizarse para representar objetos, especialmente si han sido fabricados por el hombre Andersson (Andersson, 1994), y de esta manera realizar un seguimiento sobre ellos utilizando además información bidimensional y tridimensional, (Clark et al., 1996).

## Esquinas

Son fundamentalmente la intersección de líneas en sus extremos. Uno de sus atributos más importantes es el ángulo entre las dos líneas intersectadas, (Deriche y Giraudon, 1993).

## 3.8. Actualización basada en contexto

Como una de las principales aportaciones de esta tesis, se introduce a continuación un nuevo método para llevar a cabo la actualización del patrón. Este método se fundamenta en los planteamientos teóricos basados en representación y contexto expuestos en el Capítulo 2. Mediante esta teoría, la búsqueda de patrones toma en cuenta su entorno para realizar la actualización del patrón, y por tanto evitar la intervención manual en el ajuste del umbral. Esto consigue, como principal ventaja, que el método sea capaz de lograr una adaptación automática completa frente a cualquier tipo de escenario.

A lo largo de esta sección se detallarán, en primer lugar, los problemas asociados al método más comúnmente utilizado de actualización de patrón basado en umbrales estáticos, con el fin de hacer ver de manera más clara la necesidad de esta nueva forma de actualización propuesta. A continuación se expondrá detalladamente el método propuesto mediante ejemplos sintéticos y reales. Posteriormente se introducirán diferentes ampliaciones al mismo que pueden mejorar su robustez. Concretamente la incorporación de un banco de patrones en el algoritmo de búsqueda colabora, a modo de memoria visual a corto plazo, en la mejora de los resultados del método.

### 3.8.1. La necesidad de tomar en cuenta el entorno

Como se ha visto anteriormente, la sustitución del patrón es necesaria para no perder el objeto seguido que evoluciona con el tiempo. Una sustitución por cuadro provoca una posible deriva y por tanto una pérdida. Las sustituciones mediante el traspaso de un umbral conlleva la decisión de establecer un valor adecuado para éste. Una elección no apropiada de un umbral constante ocasiona pérdidas del objeto debido a la siguiente situación: La Figura 3.11 muestra cuatro cuadros de una secuencia de ejemplo en la que el objeto seguido, patrón negro y blanco, rota. A medida que el objeto va rotando la función de similitud cambia progresivamente. La figura también muestra, de manera simplificada, el estado de la función de similitud para cada uno de los cuatro cuadros de la secuencia. En el primer cuadro *a)* el objeto a seguir aparece idéntico al patrón, esto hace que la función de similitud devuelva un cero como resultado. De esta forma, la localización del objeto buscado, marcado con una equis, recae sobre el objeto correctamente. Es también visible un segundo mínimo generado por un objeto parecido al patrón aunque con un tono más claro. El umbral de cambio de patrón es constante y fijado a un cierto valor, que en este caso ha quedado por encima del segundo mínimo,

lo cual va a provocar la pérdida.

En el cuadro *b)* el objeto ha empezado a rotar, con lo que la función de similitud comienza a cambiar, y en este caso, el mínimo aumenta su valor a medida que la diferencia entre la imagen y el patrón va siendo mayor. El segundo mínimo aparece constante ya que el objeto al que pertenece no cambia. Aún así la localización del objeto se realiza correctamente puesto que, aunque el objeto haya rotado, sigue siendo lo más parecido al patrón.

Durante el cuadro *c)* se repiten las mismas circunstancias que en el anterior. Sin embargo, la disimilitud del objeto con el patrón es mayor y su valor se sitúa casi a la misma altura que la del segundo mínimo.

Al llegar al cuadro *d)* la situación ha cambiado. El objeto ha continuado rotando y su mínimo asociado ha quedado en este momento por encima del hasta ahora segundo mínimo. Esto provoca que la localización del objeto se sitúe sobre el otro objeto de la imagen (marcado con una equis) el cual es ahora más parecido al patrón original, habiéndose entonces perdido el objeto de interés y provocado la conmutación de la atención al segundo objeto visual. Esto ha sucedido debido a una mala elección del valor del umbral de cambio. Al estar situado por encima del segundo mínimo, no ha dado opción a una actualización del patrón antes de producirse la pérdida. Si el umbral hubiese estado situado entre el primer y el segundo mínimo, el patrón se habría actualizado antes de perderse el objeto. Sin embargo, situar el umbral constante correctamente para varios tipos de escenarios no es posible ya que las condiciones varían considerablemente. Un umbral podría estar fijado adecuadamente para un escenario pero ofrecer malos resultados en otro escenario distinto.

Este fenómeno, descrito aquí de manera simplificada, es normal que ocurra en esquemas de actualización de patrón mediante umbral fijo. Luego, el umbral de cambio debería estar situado lo suficientemente bajo para no incluir segundos mínimos y lo suficientemente alto para no realizar actualizaciones del patrón innecesarias. Este compromiso tan sólo se puede alcanzar mediante el cálculo dinámico del umbral, y es esto lo que la solución aportada en esta tesis propone.

Como se ha visto, la asignación del valor del umbral depende del contexto donde se halle el objeto de interés, ya que es el entorno el que va a generar segundos mínimos (o mínimos locales) alrededor del primero (o mínimo absoluto). La Figura 3.12 muestra un ejemplo real de esto. La secuencia corresponde a dos monedas de dos y cinco céntimos de euro que se alejan de la cámara. El primer cuadro corresponde al comienzo de la secuencia, donde la moneda pequeña se toma como patrón. La superficie generada por



el algoritmo de búsqueda se refleja debajo del primer cuadro donde se aprecia el mínimo con valor cero y un segundo mínimo correspondiente a la moneda mayor. En este caso al umbral se le ha asignado manualmente un valor mayor que el segundo mínimo, lo cual creará el mismo problema que el caso sintético anterior. Al alejarse la cámara el tamaño relativo de la moneda mayor se hace cada vez más pequeño hasta, como muestra el segundo cuadro, equipararse al tamaño de la moneda en el patrón. Al estar situado el patrón por encima de los dos mínimos no se produce ninguna actualización del patrón y por lo tanto, en el tercer cuadro, la moneda mayor se vuelve más parecida a la moneda del patrón. Se produce una pérdida en este caso por la no actualización del patrón.

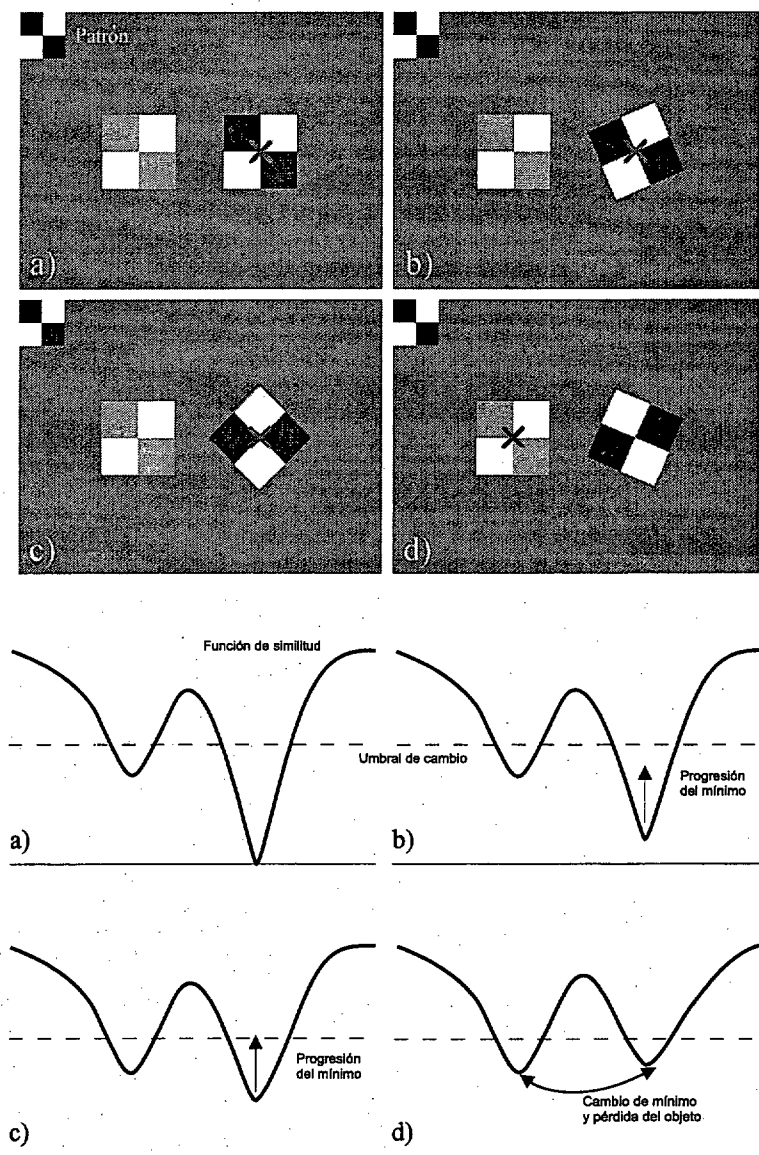


Figura 3.11: Representación de cuatro cuadros de una secuencia con actualización del patrón mediante umbral constante y los valores de similitud en la búsqueda.

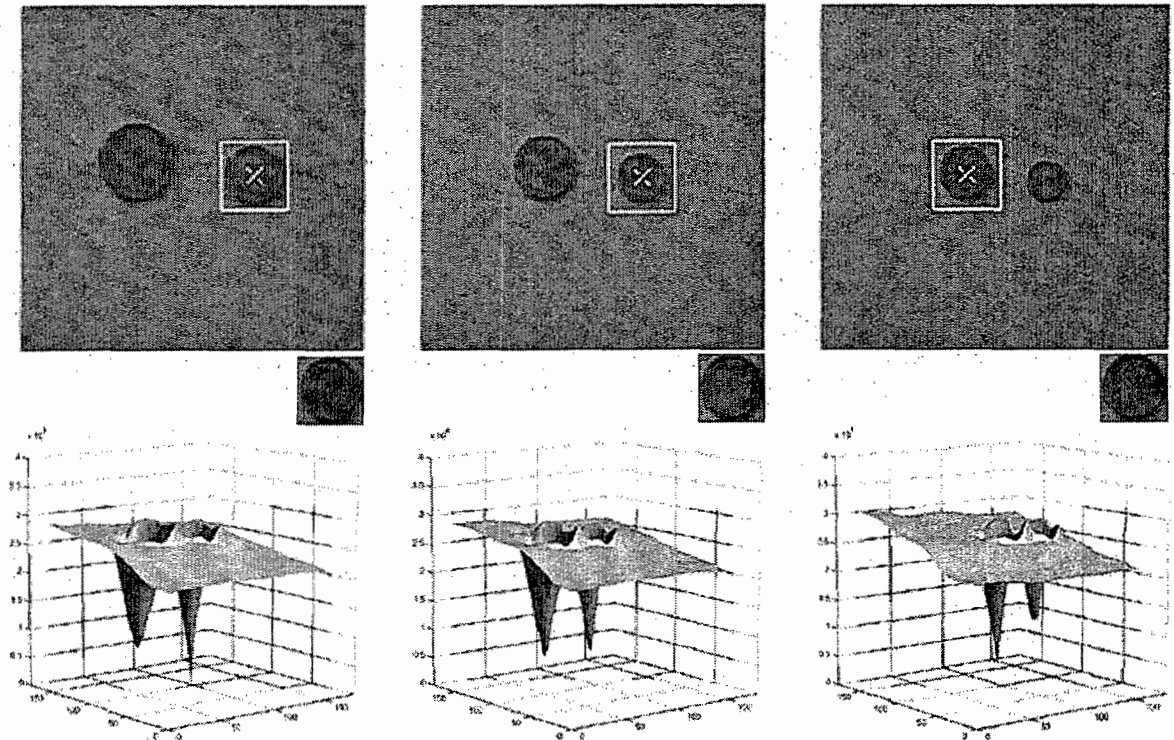


Figura 3.12: La figura muestra como al comienzo de la secuencia el patrón corresponde a la moneda pequeña. Esto genera un mínimo absoluto sobre la moneda pequeña y un segundo mínimo o mínimo local sobre la moneda mayor. Cuando la cámara se aleja y el patrón no se actualiza la moneda mayor se vuelve progresivamente más parecida al patrón. Esto se aprecia en el segundo cuadro de la figura, donde el segundo mínimo decrece y el mínimo absoluto aumenta su valor. En el tercer cuadro la atención conmuta hacia la moneda mayor puesto que es esta la que ofrece una vista más parecida al patrón original.

### 3.8.2. Cómo tener en cuenta el entorno

Utilizando el ejemplo de la secuencia sintética anterior, la inicialización del proceso del método propuesto comienza, como primer paso, con la adquisición de un nuevo patrón tomado de la imagen actual o inicial de la secuencia. La función de similitud basada en este primer patrón aplicada sobre la misma imagen devuelve, obviamente, como mínimo un cero en el lugar de donde se extrajo el patrón. El siguiente paso consiste en obtener el segundo mínimo. Expresado de otra forma, el objeto dentro de la imagen actual más parecido al objeto de interés, que se denominará *objeto principal del contexto*. En el tercer paso se calcula el umbral de actualización como la mitad del valor de este segundo mínimo, o lo que es lo mismo, el umbral de actualización se establece en el punto medio entre el primer y el segundo mínimo. Todo esto se representa en el cuadro *a)* de la Figura 3.13.

Una vez el objeto de interés comienza a rotar de la misma forma que anteriormente, se aprecia cómo el mínimo que genera también comienza a ascender, cuadro *b)* de la Figura 3.13. Sin embargo, el mínimo aún no ha traspasado el umbral de cambio y por tanto no se lleva a cabo ninguna actualización.

En el cuadro *c)* se observa que, al haber seguido rotando el objeto, el mínimo llega a traspasar el umbral, con lo que se requiere una actualización del patrón que conlleva de nuevo a una inicialización del algoritmo de la misma manera que ya se ha comentado.

En el cuadro *d)* el patrón ha sido actualizado (el nuevo patrón aparece representado en la esquina superior izquierda del cuadro) y su mínimo es ahora cero puesto que el nuevo patrón se ha tomado de la misma imagen. De nuevo se busca el segundo mínimo para volver a calcular un nuevo umbral que dependa de lo que rodea al objeto de interés.

El ejemplo descrito muestra de manera sencilla cómo tomar en cuenta el entorno a la hora de establecer los umbrales y la forma en que se previene la pérdida del objeto seguido dado que nunca se permite que existan objetos del contexto por debajo del umbral de actualización.

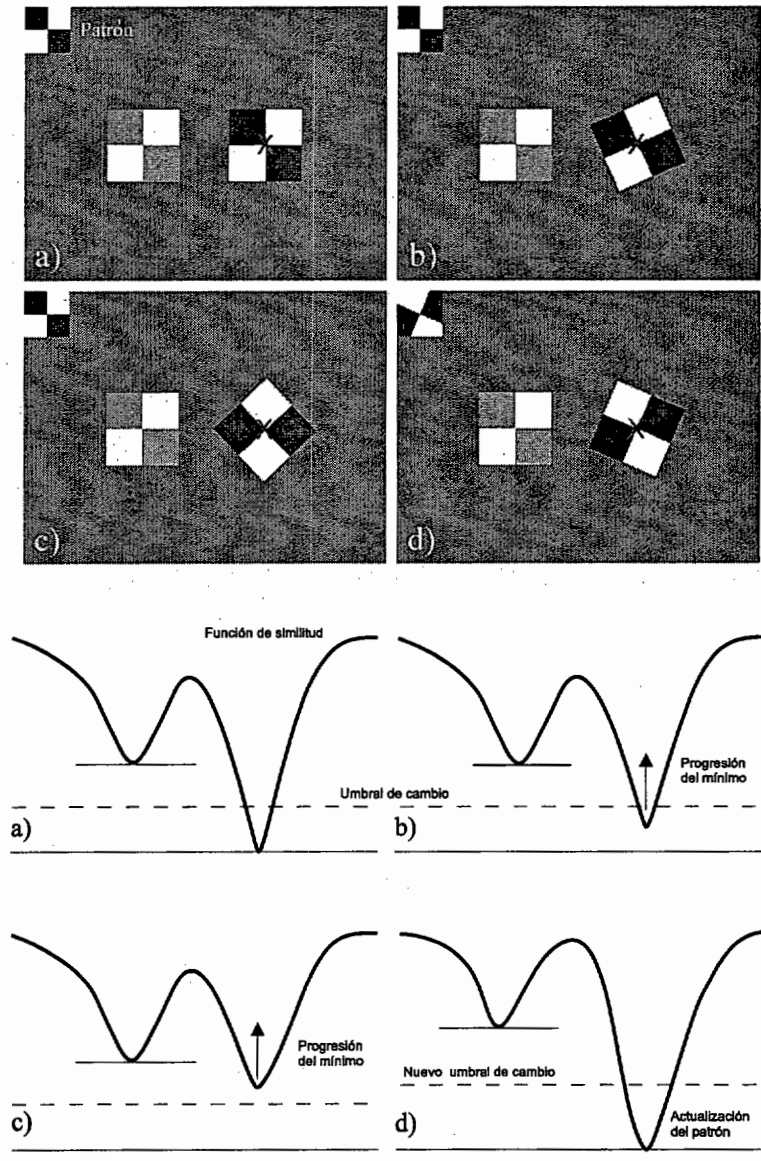


Figura 3.13: Representación de cuatro cuadros de una secuencia con actualización del patrón mediante umbral basado en contexto.

### Por qué no actualizar el umbral en cada cuadro

El factor que establece el valor del umbral es la distancia del mínimo de la función de similitud al segundo mínimo u objeto principal del contexto. Este umbral debe tomar un valor situado entre ambos, de tal manera que el patrón, a medida que se suceden los cuadros y cambia la apariencia del objeto, se actualice antes de que el mínimo perteneciente al objeto de interés se sitúe por encima del segundo mínimo. Parece entonces razonable pensar que si el umbral se establece como valor medio entre el mínimo y el segundo mínimo, debería mantenerse ese criterio a lo largo de toda la sucesión de cuadros siguientes. Es decir, el valor del umbral debería actualizarse en cada cuadro, Figura 3.14. Sin embargo, esto no está exento de problemas. En primer lugar, actualizar

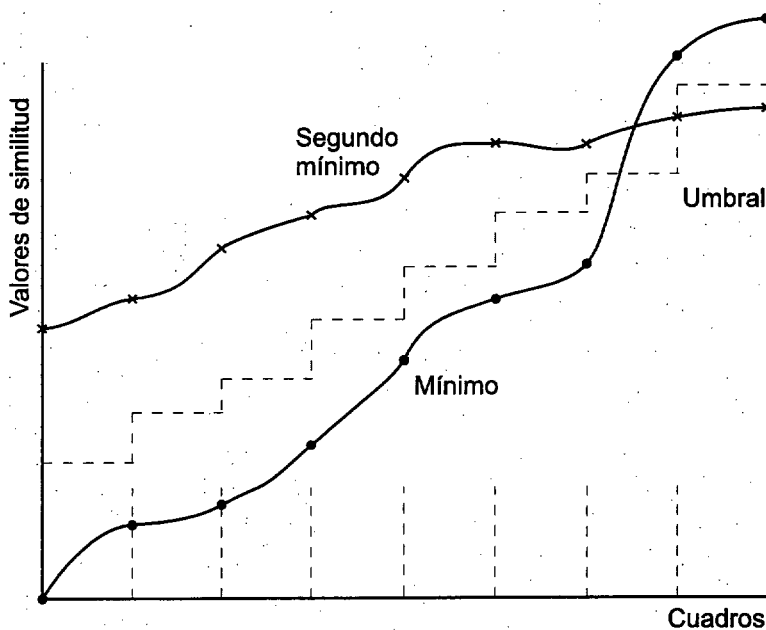


Figura 3.14: La figura representa en el eje horizontal el tiempo o los cuadros de la secuencia. El eje vertical el valor de la función de similitud. Las dos curvas corresponden al primer y segundo mínimo, la función escalonada entre ambos corresponde al umbral de cambio como valor medio entre los mínimos. Se observa que si este umbral de cambio se actualiza en cada cuadro es posible que el umbral se encuentre en un margen muy estrecho entre los mínimos que imposibiliten saber si eventualmente se han cruzado, como ocurre al final de la secuencia.

el umbral cada cuadro implica que en ciertos momentos el mínimo y segundo mínimo podrían estar arbitrariamente cerca, lo cual hace que en ciertas ocasiones no se pueda afirmar con seguridad si el mínimo perteneciente al objeto de interés sigue siendo el menor o ha pasado a ser el segundo mínimo. Debido a esto habría que establecer algún otro umbral de cercanía, que alertara sobre la proximidad entre los dos mínimos. En la

práctica es posible seguir este esquema, sin embargo en este trabajo se ha optado por no introducir nuevos umbrales prescindibles.

### Segundos mínimos bajo el umbral de cambio

Ocurre en algunos escenarios que los segundos mínimos decrecen hasta alcanzar el umbral de cambio. Esto implica que existen otros objetos que rodean al objeto seguido que se van asemejando progresivamente a este último. Esto implica que llegado un cierto momento sería imposible distinguir un objeto de otro.

Es común la siguiente situación: Supóngase el inicio de una escena en la que se ha seleccionado un objeto a seguir. En un principio no existe nada parecido alrededor del objeto con lo que el segundo mínimo posee un valor muy elevado, por tanto el umbral de cambio se sitúa muy alto. Esto permite al objeto de interés variar significativamente su aspecto sin requerir una actualización. A medida que transcurren los cuadros el objeto va cambiando su vista original.

Después de un cierto tiempo, la vista actual del objeto no se parece a la vista original, es decir el mínimo tiene un valor bastante alto, y por supuesto el segundo mínimo tampoco se parece. Sin embargo, puede llegar a existir casi la misma distancia entre la vista actual del objeto y la vista original, que entre el segundo mínimo y la vista original del objeto seguido. Esto se debe a que el umbral de cambio se situó considerablemente alto en un principio. Para evitar esta situación, se establece una nueva condición de actualización del patrón, la cual se lleva a cabo tan pronto como un segundo mínimo alcance el umbral de cambio, véase Figura 3.15.

### Qué ocurre cuando no existen segundos mínimos

Es posible no obtener segundos mínimos en determinadas ocasiones si no hay nada alrededor del objeto de interés que los genere. Realmente, este sería un caso ideal puesto que la localización del objeto estaría asegurada. De cualquier manera, en esta situación el cálculo del umbral de cambio se retrasaría hasta que existiera un segundo mínimo.

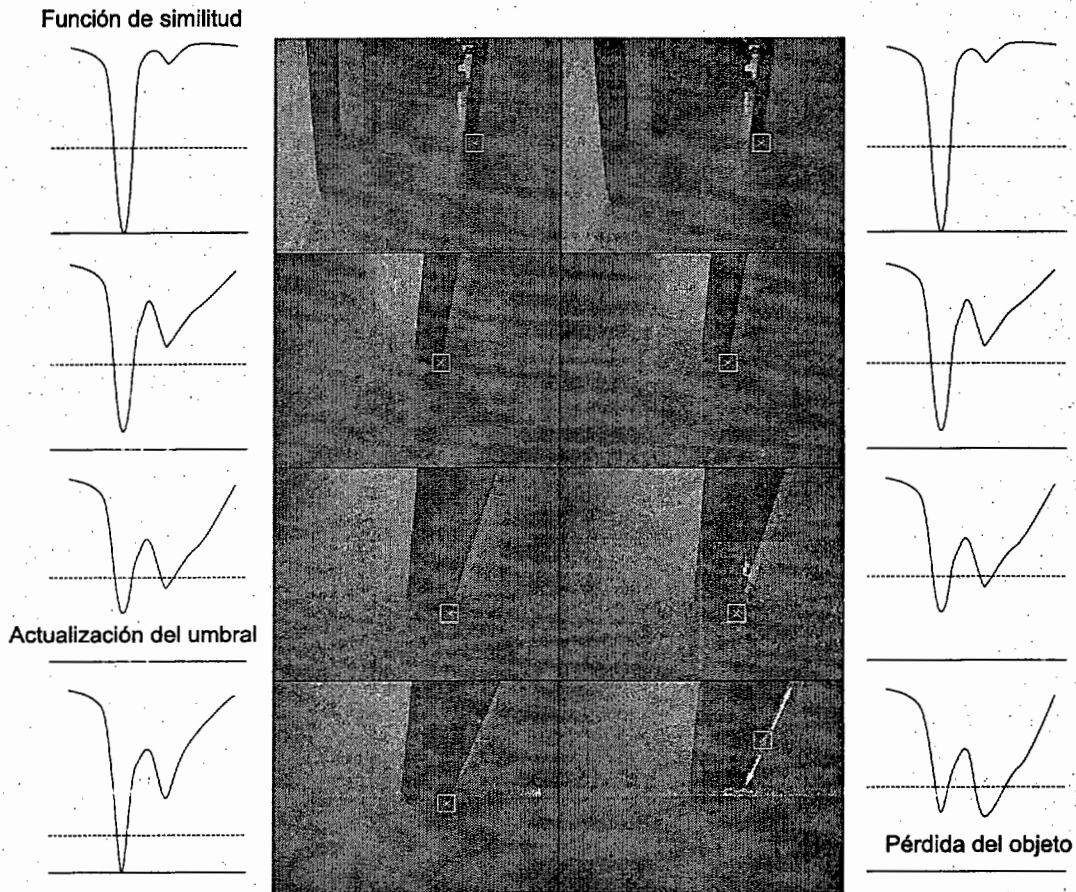


Figura 3.15: Secuencia que muestra como el paso del segundo mínimo bajo el umbral de cambio puede ocasionar una pérdida del objeto. La columna de cuadros de la derecha y de la izquierda constituyen la misma secuencia las cuales han sido procesadas teniendo en cuenta la posible incursión del segundo mínimo bajo el umbral (columna izquierda) y sin tenerlo en cuenta (columna derecha). El objeto de interés está marcado con un cuadrado blanco. El último cuadro de la columna derecha muestra como el patrón se desplaza arriba y abajo a lo largo del marco de la puerta puesto que esa región de la imagen contiene múltiples mínimos.





Figura 3.16: La figura muestra como un sistema de seguimiento precategórico no puede por si solo recuperar vistas de patrones anteriores con el fin de evitar derivas del objeto de interés tras una transformación momentánea de éste.

### 3.8.3. Banco de patrones

Al evolucionar visualmente un objeto sobre su entorno, es más que probable una deriva progresiva del objetivo a medida que los patrones se actualizan. Un módulo de seguimiento precategórico como el expuesto aquí no puede garantizar que el seguimiento se mantenga centrado a través de las sucesivas actualizaciones sobre el objeto de interés, puesto que no se realiza ningún tipo de identificación del objeto previa a la actualización del patrón a seguir. Puede ocurrir, en ciertas aplicaciones, que el objeto de interés vuelva a ofrecer repetidamente vistas anteriores tras algún cambio momentáneo.

Se dan estos casos, por ejemplo, en caras de personas que miran al frente, durante un instante giran ligeramente la cabeza hacia un lado y posteriormente vuelven a mirar al frente, como ocurre con interlocutores que hablan frente a una cámara. Con una alta probabilidad, el patrón de la cara con vista frontal utilizado al principio no es exactamente igual al patrón de vista frontal utilizado después de haber girado la cabeza. Es probable que este patrón tenga un leve desplazamiento. Como se muestra en la Figura 3.16 de ejemplo, al comienzo del seguimiento el patrón se encuentra situado centrado en el ojo izquierdo. Al ladear la cabeza hacia la izquierda el ojo izquierdo ha quedado prácticamente oculto por lo que el patrón se actualiza ahora tomando un área centrada en el ojo derecho. Cuando la cara vuelve a recuperar su posición frontal el patrón continúa centrado en el ojo derecho, habiéndose entonces producido un error con respecto a la posición original del patrón.

Una sucesión de estos movimientos de cabeza repetidos un determinado número de veces puede hacer que, finalmente, el patrón resultante diste considerablemente del primero. Esto es un comportamiento del seguimiento completamente esperable puesto que los parámetros del seguimiento evolucionan junto con el objeto seguido. Si parte del objeto que se emplea por el patrón para realizar el seguimiento desaparece o sufre una ocultación se debe entonces llevar a cabo la actualización a otra zona visible del objeto

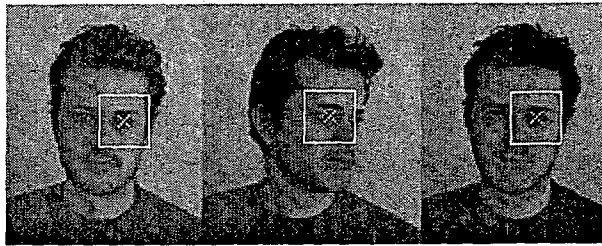


Figura 3.17: Cuando se utiliza un banco de patrones como memoria visual a corto plazo en sistemas de seguimiento precategórico, es posible evitar errores de deriva del patrón mediante la recuperación de vistas previas almacenadas en el banco.

con el fin de no perderlo. Si la zona oculta anterior se hace de nuevo visible no existe ningún mecanismo que lo detecte y retome esa zona como nuevo patrón, por tanto la deriva se ha producido.

Sin embargo, pueden ser recuperados patrones anteriores cuando la cabeza vuelva a ofrecer vistas previas mediante la utilización de una memoria visual a corto plazo que almacene vistas anteriores del objeto que podrían volver a ser utilizadas en un futuro próximo. Para ello se propone disponer de un banco de vistas que almacena los  $n$  últimos patrones usados por el seguimiento, de forma que puedan ser recuperados para su utilización futura. En la Figura 3.17 se muestran los mismos cuadros de la secuencia anterior pero ahora se ha utilizado un banco de patrones que ha permitido evitar el error recuperando la vista original. Cuantos más patrones se almacenen más probabilidades existen de que se recuperen vistas anteriores y se eviten derivas. Por otro lado, ha de tenerse en cuenta que estos patrones deben ser buscados en la imagen de la misma manera que el patrón actual, con lo que un mayor número de patrones significa un mayor tiempo de procesamiento.

### Similitud entre patrones

Dentro del banco de patrones se almacenan diferentes vistas del mismo objeto, y en principio es deseable que sean lo más significativas y frecuentes posibles, es decir que los patrones correspondan a las vistas del objeto más comunes o que más se den. En principio, deberían ser lo suficientemente diferentes entre si para abarcar el mayor número posible de poses del objeto a seguir. Dos patrones casi iguales o muy parecidos en el mismo banco no resultaría conveniente ya que consumirían recursos y no aportarían información nueva sobre el objeto. Por lo tanto, resulta lógico plantear si es necesario incluir algún tipo de filtro que permita o impida la inclusión de un nuevo patrón en caso de encontrarse a éste muy parecido con algún patrón ya almacenado en el banco. La respuesta es que realmente el propio algoritmo de actualización del patrón y cálculo del umbral implícitamente impide que ocurra esto. El algoritmo de cálculo del umbral establece su valor en función de lo parecido que sea el entorno al objeto. Es decir, el umbral será mayor si el entorno no se parece al objeto y será menor si existe algún objeto del contexto que pueda provocar una pérdida del objeto de interés. Cuanto más se parezca el objeto del contexto al objeto de interés más bajo será el umbral para realizar una actualización del patrón que evite una pérdida y más parecidos serán los patrones del banco.

Supóngase que se tiene un banco de tres patrones disponibles al comienzo el proceso. El primero de ellos es ocupado por la primera vista o patrón del objeto de interés. Esto generará un umbral que estará en función del segundo mínimo u objeto del contexto en la imagen. Cuando transcurran los sucesivos cuadros y el valor de correlación del patrón original sobre la imagen actual supere el umbral previamente establecido, un nuevo patrón será capturado. Este nuevo patrón difiere del primero un valor igual al de correlación que rebasó el umbral. Se generará un nuevo umbral y ambos patrones serán usados para buscar el objeto de interés en los siguientes cuadros. Si el objeto vuelve a mostrar su vista original (como aparecía al comienzo del proceso) el valor de correlación del primer patrón quedará por debajo del segundo.

La Figura 3.18 muestra cómo progresarían las funciones de similitud de ambos patrones dentro de un movimiento cíclico. En el caso de que ninguno de los valores de correlación de los patrones que están en el banco se sitúe por debajo del umbral querrá decir que el nuevo patrón que ahora se debe tomar diferirá lo suficiente de todos los patrones del banco.

El número de patrones que identifican un objeto dependerá del entorno donde se encuentre. Si no se halla nada parecido al objeto quizá no sea necesario actualizar

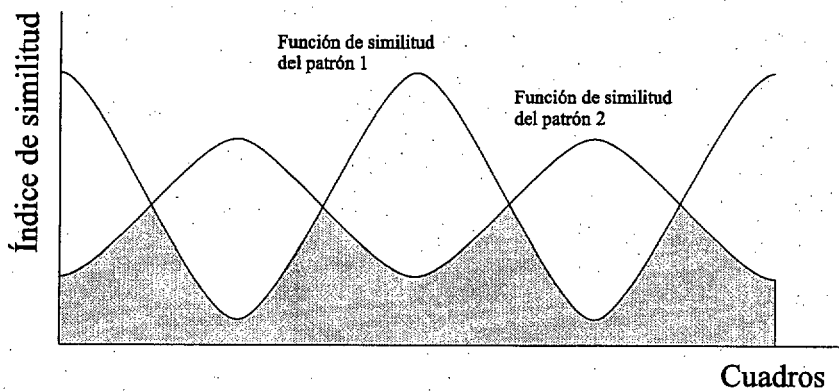


Figura 3.18: Figura esquemática en la que se muestra como progresarían en los sucesivos cuadros las funciones de similitud de dos patrones almacenados en el banco de patrones. Alternativamente irían creciendo y decreciendo si el objeto de interés tuviera un movimiento cíclico de algún tipo. El patrón escogido en cada momento sería el de menor valor en su función de similitud.

el patrón ni tener un banco de vistas o patrones del mismo. Por otro lado, si existen objetos del contexto muy parecidos al objeto de interés, sí será necesario identificar a este objeto mediante varias vistas del mismo.

### Índice de utilidad

Como se ha expuesto en secciones anteriores, el número de patrones que pueden ser almacenados en el banco debe ser limitado. Por tanto, ha de definirse algún criterio mediante el cual el algoritmo de seguimiento pueda gestionar automáticamente los patrones contenidos en el banco, eliminando los menos útiles para dejar espacio a los nuevos. Este criterio debe tomar en cuenta los factores de una determinada vista que muestren la utilidad de esta. Dos de estos factores relevantes son la *persistencia* y la *obsolescencia*.

Se entiende por *persistencia* de un patrón al tiempo o número de cuadros seguidos que éste es tomado como patrón actual. Es decir, el intervalo continuo de tiempo o ciclos de trabajo que un único patrón ha tomado para llevar a cabo el seguimiento. El factor de persistencia refleja muy claramente cuándo una determinada vista del objeto seguido es una vista muy común del mismo.

Por otro lado, la *obsolescencia* de un patrón se define como el tiempo transcurrido desde la última vez que fue usado como patrón actual. Se asume con este factor que un patrón con obsolescencia alta implica que la probabilidad de que se vuelva a utilizar es pequeña. Tanto la persistencia como la obsolescencia de cada patrón son actualizados en

### 3. Búsqueda de patrones

cada ciclo de funcionamiento de seguimiento. Por tanto, cada patrón del banco contiene un *índice de utilidad* que regula su relevancia o importancia. Cuando ha de extraerse un patrón del banco, se escoge aquel que posea el índice de utilidad menor. Este índice se define como:

$$U(t, i) = \frac{t_p(i)}{1 + t - t_s(i)}$$

Donde los diferentes parámetros son: el tiempo  $t$ , el número de patrón  $i$ , o índice dentro del banco, la persistencia del patrón  $i$ -ésimo  $t_p(i)$  y la obsolescencia del mismo  $t_s(i)$ . De la fórmula se extrae que la utilidad de un determinado patrón  $i$  crece a medida que su persistencia es mayor. Por el contrario, la utilidad se ve reducida cuando se incrementa su obsolescencia, que es medida en la expresión anterior como el tiempo actual  $t$  y el momento en que el patrón fue utilizado por última vez,  $t_s(i)$ . El denominador se incrementa en 1 dado que el patrón actual tiene obsolescencia 0 y generaría un error de cálculo de la función. Nótese también que el tiempo se mide a través de un contador con incrementos enteros.

En ocasiones un determinado patrón interesa, por diversas causas, que sea almacenado y no sustituido debido a su menor índice utilidad. Para ello se utiliza un indicador de bloqueo con el fin de impedir que el patrón en memoria no pueda ser extraído y sustituido por otro.

#### Criterios de elección de patrones

En determinadas aplicaciones o entornos existen características en una vista del objeto que impiden su aceptación como patrón, y por tanto tampoco se llega a la inclusión de éste en el banco. Estas características hacen referencia a posibles "inconvenientes" que la vista presenta y que podrían, en teoría, confundir o hacer fallar al seguimiento. No es posible afirmar que este tipo de características sean siempre perjudiciales para el seguimiento, aunque siempre causan alguna limitación en la precisión o en la fiabilidad. A continuación se presentan algunos casos en los que se dan estas situaciones:

- **Contraste.** El patrón elegido debe poseer un contraste suficiente. Los patrones planos, es decir sin variaciones marcadas del color o niveles de gris, por lo general no permiten encontrar el objeto buscado con suficiente precisión. En la Figura 3.19 se observa como el patrón  $A$  puede encontrarse realmente sobre casi cualquier punto de la superficie de color plano sobre la que está. En ocasiones esto puede resultar aceptable, ya que de todas formas el patrón está sobre el objeto de interés, en este ejemplo el camión. Sin embargo hay ocasiones en la que esta situación provoca

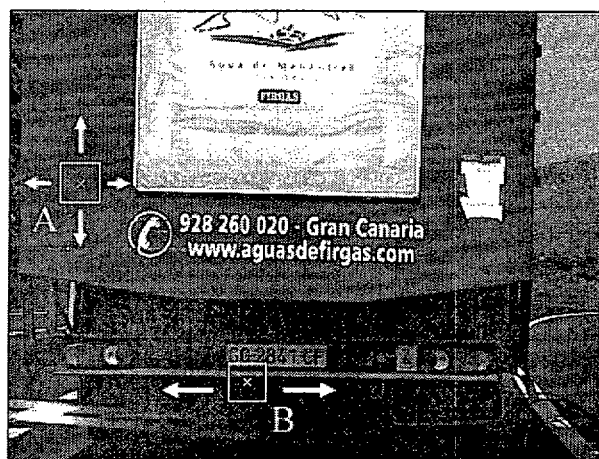


Figura 3.19: La figura muestra dos patrones con diferentes problemas. El patrón A consiste en un patrón plano, es decir, sin ninguna marca o referencia que lo haga fijarse en alguna zona precisa de la imagen. En este caso el patrón se mueve libremente por la superficie homogénea de la parte trasera del camión. El patrón B presenta un problema conocido como el problema de la apertura y que es similar al anterior ya que ahora no existe una referencia horizontal que lo limite. El patrón se moverá libremente a lo largo del parachoques del camión.

una pérdida del objeto a seguir debido a la propia incertidumbre de la posición del patrón al estar poco contrastado respecto al área que lo circunda.

- **Problema de la apertura.** También relacionado con lo anterior se encuentra el problema de la apertura, conocido clásicamente en problemas como la detección del movimiento y flujo óptico, (Beauchemin y Barron, 1995), puesto que en este caso no existe variación del color o nivel de gris en una de las dos dimensiones del patrón. Si se toma como patrón algún objeto alargado con continuación fuera del área del patrón será altamente probable que el algoritmo de seguimiento no pueda establecer la posición correcta en sucesivos cuadros puesto que no existirá referencia a tomar sobre una de las dimensiones de la búsqueda, lo que redundará en la incertidumbre de la posición. La Figura 3.19 en el patrón B muestra este efecto. El patrón puede moverse libremente a lo largo del parachoques trasero del camión del ejemplo.
- **Completitud del patrón.** El área del patrón debe ser completada únicamente con área del objeto a seguir sin incluir partes del fondo. En la Figura 3.20 se observa como, en los cuadros A y B, las personas que se encuentran caminando no cubren completamente el patrón y eso hace posible que el patrón se quede "enganchado" a algún elemento del fondo. Evidentemente, al no cubrir el objeto



Figura 3.20: Los dos primeros cuadros de la figura muestran cómo el patrón no incluye solamente áreas pertenecientes al objeto de interés sino también del fondo. Esto puede ocasionar problemas durante el seguimiento, ya que cuanto más porcentaje de fondo se incluya en el patrón más posibilidades de error existe. El tercer cuadro muestra un patrón completado solamente con el objeto de interés.

la totalidad del patrón, ello da oportunidad a que el fondo adquiera peso dentro del algoritmo de búsqueda. Siendo este peso mayor cuanto menos área del patrón cubra el objeto de interés.

- **Velocidad del objeto.** Este problema concierne principalmente al ajuste de parámetros de la cámara que tienen influencia sobre la calidad de adquisición de los patrones, ya que a ciertas velocidades relativas del objeto a seguir con relación a la cámara, los patrones pueden capturarse borrosos. Este tipo de defectos se pueden corregir cuando las condiciones de iluminación lo permiten mediante el incremento de la velocidad de obturación. De cualquier forma, como regla práctica no debe permitirse la adquisición de patrones mientras las cámaras se hallen en movimiento rápido o sacádico.

Todos estos casos expuestos han de ser tenidos en cuenta por un componente extra al módulo de seguimiento que realice un análisis más detallado del objeto a seguir, ya que en muchas ocasiones se requiere un conocimiento más preciso de las características del objeto a seguir, (Shi y Tomasi, 1994). El carácter precategórico del módulo de seguimiento hace difícil el control de estas situaciones.

### Múltiples bancos de patrones

El algoritmo de búsqueda descrito puede ser ampliado para llevar a cabo un seguimiento de varios objetos concurrentemente, lo cual tiene útiles aplicaciones. Por ejemplo, el seguimiento de un determinado objeto mediante el seguimiento de diferentes partes del mismo contribuye al incremento de robustez, ya que es posible prescindir de determinadas partes ocultas momentáneamente mediante el seguimiento de otras partes

visibles del objeto. Por tanto, cada objeto puede tener su propio banco de patrones con lo que cambiar de objeto seguido consiste en conmutar entre distintos bancos de patrones.

### 3.8.4. Pérdida del objeto

Saber que el objeto que se seguía se ha perdido es, en la mayoría de las aplicaciones, tan importante como seguirlo correctamente. Desafortunadamente no existe ningún elemento dentro del método expuesto de seguimiento precategórico que permita afirmar sin error alguno que el objeto que se ha estado siguiendo se ha perdido. Sin embargo, existen algunos índices que permiten aportar probabilidades de que así sea. Es necesario destacar que son tan sólo probabilidades lo que estos índices arrojan puesto que lo que puede plantearse en determinadas circunstancias como una pérdida puede ser tan sólo una interpretación casual incorrecta de una vista del objeto de interés.

A continuación se describen algunos de estos índices. Nótese que fundamentalmente estos se basan en el principio de continuidad y suavidad que los objetos seguidos deben mantener a lo largo del seguimiento y que proporcionan la coherencia visual entre las sucesivas vistas del objeto. En otras palabras, un objeto no puede variar súbita y bruscamente su aspecto en un breve intervalo de tiempo sin que ello se interprete como una pérdida del mismo.

- *Aumento súbito del mínimo por encima del umbral de actualización del patrón.* Este hecho claramente indica un salto cuantitativo y brusco de la disimilaridad entre el patrón actual y el punto de la imagen de mayor parecido. Sucede en cambios de orientación o deformación brusca del objeto y en cambios repentinos de la luminosidad de la escena. Cuando el aumento del índice por encima del umbral de cambio pasa de ser normal a ser considerado indicativo de pérdida es una cuestión de cantidades relativas medidas por factores como la velocidad de crecimiento del índice.
- *Cambio brusco en la trayectoria y superación del umbral del mínimo.* Estrechamente relacionado con el punto anterior se encuentra el hecho en el que el incremento súbito del índice de la función de similitud se acompañe de un cambio brusco en la trayectoria o posición con respecto a la posición anterior. Aunque depende de la aplicación, parece improbable la alteración no prevista de trayectorias o saltos de posición incoherentes.



- *Cercanía del segundo mínimo.* Es difícil asegurar el seguimiento en la proximidad de objetos muy parecidos. Aunque teóricamente el umbral de cambio se ajusta automáticamente para manejar estas situaciones, el ruido de los dispositivos de adquisición puede hacer imposible la determinación del objeto. Esto es debido a que es posible que el ruido inherente sea mayor que la distancia (en términos de función de similitud) entre los objetos similares. Este tipo de situaciones puede tener lugar también cuando se siguen objetos con patrones de textura.
- *Falta de un mínimo en la superficie de correlación.* Aplicar alguna medida de similitud sobre objetos sin ningún tipo de textura puede dar lugar a la falta de mínimo en la función. Es muy corriente este tipo de resultados cuando el seguimiento se hace en interiores y el patrón ha derivado, por ejemplo, hacia una pared monocroma y sin textura.

Como ya se ha indicado anteriormente, todos estos índices proporcionan una respuesta subjetiva o intuitiva al hecho de la pérdida. Sólo mediante el uso de métodos de identificación de mayor complejidad del objeto o categóricos se podría afirmar de manera fiable que el objeto seguido corresponde al objeto correcto.



## Capítulo 4

# Sobre la arquitectura de un módulo genérico de seguimiento precategórico

En el capítulo introductorio de este documento se planteó como hipótesis de partida la posibilidad de diseñar y construir un módulo de seguimiento visual precategórico, autónomo y adaptable. En base a las aportaciones al seguimiento visual descritas en los dos capítulos anteriores se cuenta con un marco teórico y práctico para el desarrollo de métodos de búsqueda precategóricos que puedan realizar la fijación de objetos de manera autónoma. El presente capítulo describe la adopción de tales métodos para proponer una arquitectura teórica de módulo genérico de seguimiento precategórico que reúna las condiciones de autonomía, robustez y adaptabilidad requeridas.

Es necesario destacar que las ideas contenidas en este capítulo pretenden proporcionar una colección de características deseables en un módulo de seguimiento surgidas por las necesidades encontradas a la hora de desarrollar procesos de seguimiento precategóricos. El modelo propuesto no persigue ser un modelo estricto de construcción sino establecer las líneas generales que, a nuestro entender, deben poseer esta clase de módulos.

El módulo de seguimiento se concibe como un componente cerrado dentro de un sistema de visión mayor que lo hospeda, y la adaptabilidad de este módulo se entiende como la facilidad ofrecida para su integración y utilización en dicho sistema de visión mayor. Tal adaptabilidad pasa por la definición de una interfaz de control sobre el módulo de seguimiento que debe ser lo suficientemente flexible como para posibilitar el gobierno del módulo por los diferentes sistemas de visión que lo utilicen, de manera que estos puedan obtener los distintos comportamientos y capacidades concretas requeridas al módulo de seguimiento. En otras palabras, la interfaz de control debe contar con un

conjunto suficiente de comandos para establecer el modo de trabajo demandado por el sistema de visión en el módulo de seguimiento.

Otro de los aspectos que se pretende alcanzar con el módulo de seguimiento es la robustez del comportamiento resultante. El seguimiento robusto por medio de procesos visuales precategóricos es un objetivo difícil de alcanzar. Los métodos de búsqueda enumerados en el capítulo anterior realizan un seguimiento fiable sólo bajo ciertas circunstancias del entorno que no siempre están presentes en contextos reales. No obstante, la combinación de varios de estos métodos de búsqueda puede ofrecer la fiabilidad demandada, (Pentland y Choudhury, 2000). La combinación o fusión de métodos permite a un sistema emplear la fuerza de cada uno de ellos mientras se minimiza el efecto de su debilidad (Toyama, 1998). En otras palabras, mediante la fusión se pretende dar preponderancia a los resultados de los algoritmos que mejor se adapten en cada caso a las condiciones visuales prevalecientes. Los métodos combinados crean una sinergia que hace factible la ejecución de tareas que serían muy complicadas de conseguir mediante métodos individuales.

La exposición del módulo de seguimiento en el presente capítulo se realiza mediante las siguientes secciones: Una primera sección detalla los resultados básicos que se espera que genere el módulo. Estos resultados son los valores obtenidos por el seguimiento y conforman los resultados de la operación normal del módulo. Una segunda sección muestra la arquitectura interna del módulo, analizando de manera teórica los diferentes componentes o bloques funcionales que se incorporan, así como su ciclo de trabajo. La tercera sección muestra las capacidades de que consta el módulo y las posibilidades que ofrece, comandos que se incluyen para su control, modos de operación o funcionamiento y facilidades de integración. Una última sección plantea distintas consideraciones sobre las características hardware para la implementación de módulos de seguimiento como el descrito en este trabajo.

## 4.1. Resultados del seguimiento

El funcionamiento básico del módulo de seguimiento se centra en generar las coordenadas de imagen donde se ha localizado el objeto de interés. Este resultado puede ser acompañado de más datos que enriquecen la información que se posee del objeto. A continuación se enumeran los distintos resultados que ofrece el módulo de seguimiento durante su funcionamiento normal:

- *Posición (x,y)* en coordenadas de la imagen procesada donde se ha detectado el

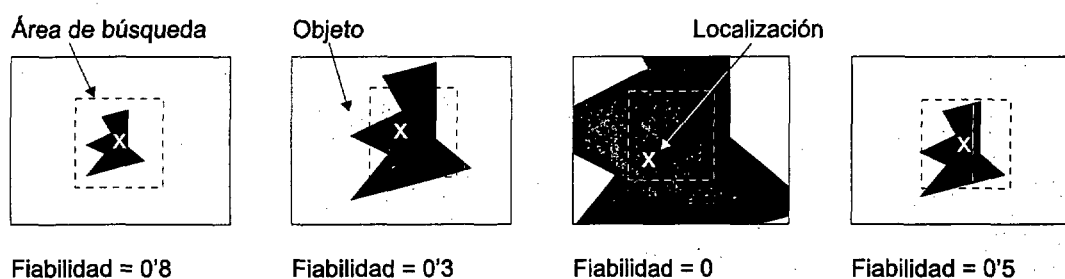


Figura 4.1: Figura esquemática que ilustra el resultado de fiabilidad devuelto por el módulo de seguimiento. Se muestran en cada cuadro diferentes valores orientativos de la fiabilidad que podrían tener varias vistas de un objeto. La primera vista posee el índice más alto debido a que el objeto puede fijarse establemente dentro del área de búsqueda. En la tercera vista es imposible fijar la posición del objeto dentro del área de búsqueda dado que el método de búsqueda no puede encontrar referencias visuales. Esto conduce a tener un índice de fiabilidad cero aunque el objeto no se haya perdido.

objeto buscado.

- *Vector bidimensional* que representa la dirección de movimiento del objeto en coordenadas de la imagen y cuyo módulo indica su velocidad en pixels de la imagen por segundo.
- *Estimación de la fiabilidad* de estos resultados. Su valor se encuentra comprendido en el intervalo cerrado  $[0, 1]$  donde el valor 0 representa una incertidumbre total sobre la posición actual del objeto, mientras que el valor 1 asegura completamente su localización. Es importante aclarar en este punto que una imprecisión total (fiabilidad cero) del conocimiento de la posición del objeto buscado no implica siempre la pérdida del mismo. En la Figura 4.1 se muestra simplificada cómo un objeto puede ser localizado en una imagen dentro de una determinada área de búsqueda. Cuando el objeto se acerca tanto que llena completamente el área de búsqueda puede que no existan referencias visuales por las cuales determinar alguna posición concreta dentro del objeto. Por tanto, en la tercera imagen la posición encontrada del objeto dentro del área de búsqueda puede ser cualquier punto dentro de la misma. Sin embargo, el objeto en si no se ha perdido aunque la posición devuelta no presente un comportamiento estable. La fiabilidad devuelta por el módulo de seguimiento resulta calculada como una combinación de los índices de fiabilidad que generan cada uno de los métodos de búsqueda incorporados en el módulo.
- *Condición de pérdida*, es un valor lógico que se vuelve verdadero cuando se ha producido la pérdida del objeto seguido por parte del módulo de seguimiento.

Si bien la posición del objeto buscado es un resultado básico del módulo de seguimiento, tanto el vector de velocidad y dirección, así como la estimación de la fiabilidad y la condición de pérdida no representan datos obligatorios en el funcionamiento del módulo, pudiendo en ocasiones no estar presentes en el resultado.

El módulo de seguimiento puede generar otros resultados adicionales además de los citados. Como se expondrá más adelante, existen comandos del módulo que permiten recuperar información más allá de la básica. Distintos parámetros internos que caracterizan al objeto de interés pueden ser extraídos del módulo y representar una información valiosa para el sistema externo que hospeda el módulo de seguimiento.

## 4.2. Estructura interna del módulo de seguimiento

El diseño de la arquitectura interna define y posibilita las capacidades que tiene el módulo de seguimiento durante su funcionamiento normal. Estas se orientan a las características que se han declarado resumidamente desde un principio como autonomía, robustez y versatilidad en un módulo de seguimiento precategórico.

Mediante el uso de las técnicas de Búsqueda Basadas en Contexto descritas en los capítulos anteriores es posible conseguir la autonomía deseada en el módulo de seguimiento. Estas técnicas permiten el funcionamiento del módulo sin ajuste manual previo a distintos tipos de escenarios.

Otro de los aspectos a tratar es la versatilidad o facilidad de integración del módulo en sistemas más complejos. Para ello, el módulo consta de unas salidas y entradas de datos definidas, así como de una librería de comandos que permiten su control total por parte de módulos superiores del sistema que lo hospeda.

Para conseguir estas capacidades se propone la arquitectura de bloques funcionales, cuyo diagrama se puede observar en la Figura 4.2. Dichos módulos se enumeran a continuación y se describen extensamente en las subsecciones siguientes:

- **Métodos de búsqueda.** Cada método es la implementación, siguiendo una definición establecida, de una técnica o algoritmo de búsqueda de objetos sobre imágenes como las expuestas en el Capítulo 3.
- **Bloque de control.** Este bloque realiza la gestión tanto de las peticiones de módulos externos como del funcionamiento interno del módulo.
- **Bloque de combinación.** Para obtener la robustez deseada en el módulo de

seguimiento es necesario combinar diferentes métodos de búsqueda con el fin de extraer las mejores cualidades de cada una de ellas en cada posible ámbito de aplicación.

- **Variables del módulo.** Estas variables recogen información útil para el funcionamiento de los diferentes métodos de búsqueda.

Por tanto, el módulo de seguimiento propuesto consta fundamentalmente de diferentes métodos o técnicas de búsqueda que son gestionados por un bloque de control integrado en el propio módulo de seguimiento. Los distintos métodos de búsqueda envían sus resultados a un bloque de combinación que es el encargado de su fusión en un resultado único.

Es importante recalcar que en la concepción del módulo de seguimiento, éste ha sido diseñado con la finalidad de su integración en diferentes tipos de sistemas de visión. Por ejemplo, tanto para sistemas basados en visión activa que exijan respuestas en tiempo real como para sistemas de vigilancia estáticos que tan sólo demanden uno o dos cuadros por segundo de procesamiento, el módulo de seguimiento ofrece funciones y características que permiten su plena adaptación.

Para la realización del capítulo experimental de este documento se han desarrollado módulos de seguimiento basados en las directrices expuestas en este capítulo. Lo que se presenta a continuación es la descripción de cada una de las partes propuestas para la construcción del mismo. Sin embargo, no debe interpretarse esta aportación del trabajo de tesis como un esquema a seguir riguroso para el diseño de módulos de seguimiento. Su finalidad se centra en la aportación teórica de ideas en forma de componentes y estructuraciones internas, que lleven a la consecución de las características deseables citadas anteriormente en un módulo genérico de seguimiento precategórico. Es decir, se expone un esquema de construcción válido para alcanzar las premisas de autonomía, robustez y versatilidad mediante la utilización de métodos de búsqueda precategóricos aunque abierto a diferentes modificaciones para adaptarlo a necesidades particulares.

#### 4.2.1. Métodos de búsqueda

Un método de búsqueda es básicamente la implementación de una técnica o algoritmo concreto de búsqueda de un objeto dado sobre una imagen. Para ello, cada método posee un conjunto de datos representativos del objeto a buscar que lo caracterizan e identifican, llamados aquí *descriptores*. Es decir, cada método de búsqueda consta del procedimiento de búsqueda que hace uso de estos descriptores para realizar

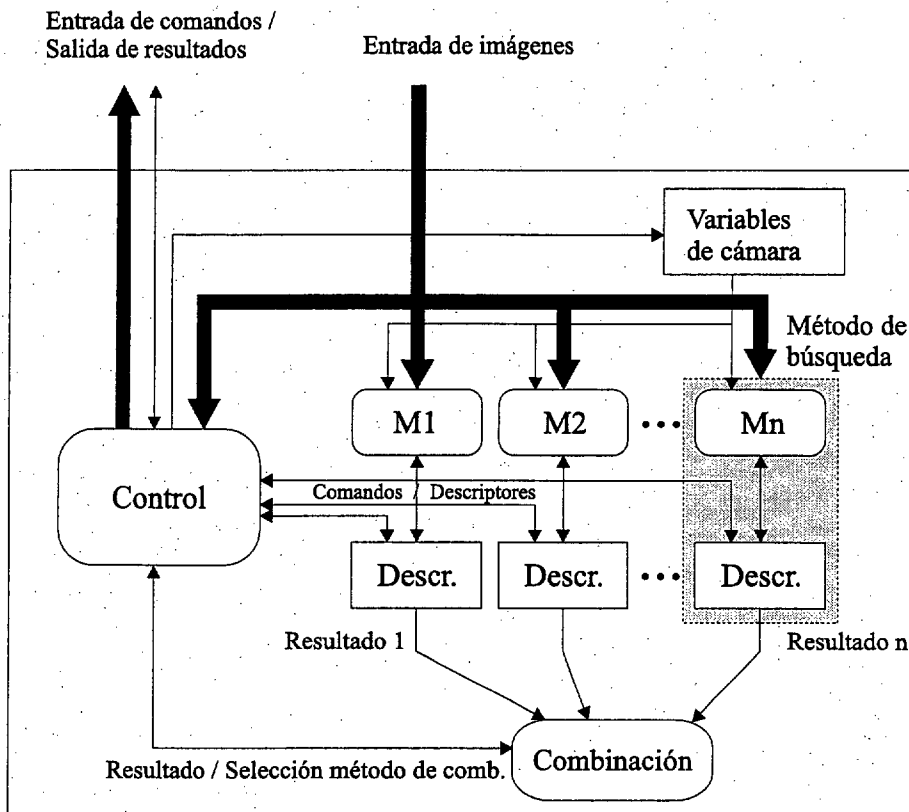


Figura 4.2: La figura representa el módulo de seguimiento propuesto. Posee un bloque de control, diferentes métodos de búsqueda que hacen uso de sus respectivos conjuntos de descriptores, un bloque de combinación de resultados y entradas y salidas para imágenes y entrada de comandos.

la localización del objeto de interés. Algunas de las técnicas más utilizadas para realizar la búsqueda son: la búsqueda de patrones (ampliamente descrito en el capítulo anterior), búsqueda por histogramas de color (Swain y Ballard, 1991), búsqueda de contornos por Hausdorff (Sánchez-Nielsen y Hernández, 2000), etc. Aunque las predicciones de localización mediante filtros predictivos como los de Kalman (Kalman, 1960), no son un método de búsqueda en sí, pueden ser incorporados como tales dentro de un módulo de búsqueda ya que proporcionan una estimación con características de resultado posicional del objeto buscado, al igual que los métodos de búsqueda propios.

Los métodos de búsqueda incluidos en el módulo de seguimiento propuesto deben caracterizarse por funcionar de manera autónoma. El Aprendizaje Incremental Basado en Contexto descrito en el segundo capítulo ofrece mecanismos para que los métodos de búsqueda posean esta cualidad.

Dentro del módulo de seguimiento no se tiene una apreciación tridimensional del entorno ni del objeto de interés. Por tanto, los procedimientos de búsqueda deben



devolver un resultado consistente en la posición, en coordenadas de la imagen, del objeto buscado, así como su velocidad y dirección también en coordenadas de imagen.

Es necesario destacar en este punto el problema que surge cuando el módulo de seguimiento toma sus imágenes desde cámaras con egomovimiento. Es decir, el movimiento bidimensional aparente del objeto en la imagen no corresponde exclusivamente a su propio movimiento sino a la combinación de éste con el movimiento inducido por la cámara. Existen métodos de búsqueda que requieren el conocimiento preciso del movimiento de la cámara para su compensación y de esta forma poder funcionar correctamente. Ejemplos de estos métodos son las técnicas de búsqueda basadas en flujo óptico (Mae et al., 1996), técnicas basadas en diferencias de imágenes para detectar objetos en movimiento o métodos de búsqueda predictivos basados en las posiciones anteriores del objeto de interés para generar una nueva estimación. Resulta entonces necesario que el método en cuestión disponga de la posición tridimensional y orientación de la cámara con respecto a un sistema de coordenadas fijo. Esta información se toma de las variables de cámara (descritas posteriormente) almacenadas en el propio módulo de seguimiento cuyos valores han sido introducidos por algún módulo externo del sistema.

Además de estos resultados, cada método debe generar una estimación de la fiabilidad de su búsqueda. Esta fiabilidad debe calcularse según determinadas consideraciones particulares y específicas para cada método de búsqueda. Así, la búsqueda de patrones puede generar su fiabilidad en función de la similitud encontrada, y un método de búsqueda por contornos de Hausdorff lo podrá hacer según la cantidad de contorno que consiga encajar. De la misma forma que el módulo de seguimiento devuelve un resultado de fiabilidad comprendido en el intervalo cerrado  $[0,1]$ , cada método de búsqueda deberá generar su valor de fiabilidad comprendidos dentro del mismo intervalo cerrado.

## Descriptores

Los descriptores son datos identificativos y caracterizadores del objeto a seguir, almacenados y utilizados por cada método de búsqueda y que permiten la localización y diferenciación del objeto con respecto al contexto. Entre los descriptores que se pueden encontrar en cada método de búsqueda particular se tienen: patrones, histogramas de color, perfiles o contornos, localizaciones o restricciones espaciales, etc. Expresados de otra forma, los descriptores se corresponden con los datos o variables de trabajo de los distintos métodos de búsqueda.

En determinados métodos de búsqueda es posible la incorporación de datos referentes al propio contexto de aplicación o entorno contenido en la imagen, ya que para el

funcionamiento robusto del método puede ser tan importante disponer de información del entorno como del propio objeto de interés. En otros métodos, dentro de los descriptores cabe también el almacenamiento de diferentes datos sobre características del objeto de interés correspondientes a vistas anteriores que son almacenadas en previsión de ser utilizadas nuevamente. Los bancos de patrones descritos en el capítulo anterior son un ejemplo de esto. Así mismo es también posible el almacenamiento de descriptores de otros objetos para un cambio rápido de objeto de interés durante el proceso de seguimiento. La definición expuesta sobre los descriptores conlleva a que el cambio de estos valores implica la búsqueda de otro objeto distinto.

Conceptualmente no es posible implícitamente compartir los descriptores de cada método de búsqueda con otros métodos. Sin embargo, nada impide a una realización particular el hacerlo.

### Ventana de búsqueda reubicable

Debido al coste computacional de muchos métodos de búsqueda, estos restringen el área de búsqueda a una ventana de la imagen completa. Una de las características importantes que deben poseer tales métodos de búsqueda consiste en la capacidad de reubicación de esta ventana.

En los sistemas de seguimiento con egomovimiento de cámaras, y en los sistemas de Visión Activa en mayor medida, la ventana de búsqueda reubicable permite al sistema mecánico reaccionar de manera más lenta a como lo hace el módulo de seguimiento.

La Figura 4.3 muestra en tres pasos cómo reacciona un sistema de seguimiento con cámaras motorizadas y ventana de búsqueda reubicable. Cuando el objeto se encuentra quieto *a*), el sistema lo centra sin mayor problema. Tan pronto como el objeto comienza a moverse con cierta velocidad *b*), el módulo de seguimiento lo mantiene localizado y desplaza la ventana de búsqueda para situarse sobre él en la imagen. El sistema mecánico posee, en la mayoría de las ocasiones, un comportamiento reactivo más lento que el del módulo de seguimiento, por lo que comienza a moverse más tarde. En el estado *c*), el sistema mecánico se adapta al movimiento del objeto.

La ventana de búsqueda reubicable evita pérdidas del objeto de interés debidas a la tardanza en la reacción electromecánica de los motores encargados de la orientación de la cámara.

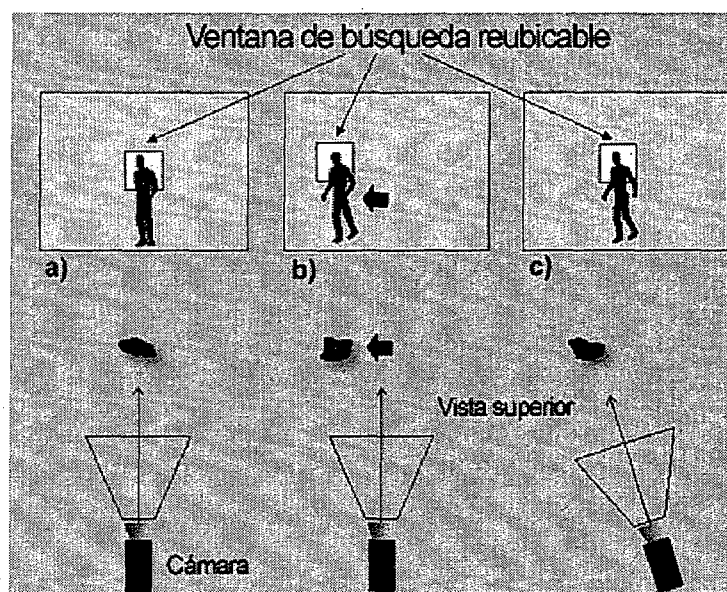


Figura 4.3: Desplazamiento de la ventana de búsqueda reubicable durante un proceso de seguimiento. Se observa cómo la ventana reubicable evita errores en la búsqueda por la reacción electromecánica más lenta de los motores de la cámara.

#### 4.2.2. Bloque de control

La activación y los comandos para cada uno de los métodos de búsqueda proviene del bloque de control, que es el encargado de recibir las diferentes órdenes que un módulo superior le comanda. También recoge y envía a este módulo superior los resultados del seguimiento.

Una de las funciones más elaboradas que debe realizar el bloque de control radica en secuenciar los comandos necesarios para llevar a cabo el seguimiento simultáneo de diferentes partes de un objeto que se halla en la imagen, (capacidad descrita posteriormente). Para realizar este proceso el bloque de control debe activar ordenadamente los métodos de búsqueda configurados para cada parte del objeto en cuestión y finalmente proporcionar un resultado único. Es decir, si el número de partes del objeto es  $n$ , entonces el bloque de control debe realizar  $n$  iteraciones configurando en cada ciclo todos los métodos de búsqueda para localizar una determinada parte del objeto. Una vez terminada la última iteración es también el bloque de control el encargado de generar una respuesta única de salida.

### 4.2.3. Bloque de combinación

Los resultados obtenidos por los métodos de búsqueda se introducen en un bloque de combinación, que realiza su fusión mediante alguna técnica definida, si se hacen colaborar al menos a dos métodos en el módulo. El resultado fusionado es enviado al bloque de control para hacerlo disponible externamente. El bloque de combinación de resultados de los diferentes métodos de búsqueda es un aspecto crucial en el módulo de seguimiento, puesto que es aquí donde se robustecen los resultados suministrados por el módulo.

Debido a la concepción del módulo de seguimiento, para diferentes tipos de aplicaciones y escenarios la combinación de los resultados puede ser también diferente en ellas. Es decir, entornos de aplicación diferentes pueden requerir tipos particulares de combinación. De la misma forma, es posible que se requiera una forma de combinación *ad hoc* para una aplicación concreta. Por ello, en el módulo de seguimiento se incorporan distintos métodos de combinación seleccionables según la aplicación y además se hace posible la introducción de algoritmos de fusión propios para un entorno de aplicación particular. La opción elegida de fusión (Krotkov, 1989; Abidi y Gonzalez, 1992; Parhami, 1994; Isard y Blake, 1998; Toyama, 1998; Triesch y von der Malsburg, 2000; Spengler y Schiele, 2001) depende de diferentes aspectos que se citan a continuación:

- *Los métodos de búsqueda incorporados en el módulo de seguimiento.* La naturaleza de los resultados de la localización del objeto de interés por parte de los métodos de búsqueda es decisiva para la elección de las técnicas de fusión adecuadas. Esta elección de técnicas se lleva a cabo en el momento de la definición del módulo de seguimiento por parte del diseñador, con lo que se incorporan antes de determinar la futura aplicación o entorno de funcionamiento del módulo.
- *La aplicación concreta donde funcione el módulo de seguimiento.* Una vez determinada una aplicación del módulo de seguimiento, esta puede requerir el uso de una u otra técnica de fusión. Condicionamientos como la rapidez o la variabilidad de las condiciones de trabajo pueden forzar la elección de sólo unas determinadas técnicas de fusión entre las disponibles. Esta elección es realizada por el módulo superior que utilice el módulo de seguimiento, a través de las funciones de control correspondientes.
- *Las condiciones de entorno de la aplicación.* Las condiciones de entorno de la aplicación pueden alterarse significativamente durante el funcionamiento del módulo. Cambios en las condiciones de iluminación o de contexto pueden demandar

el cambio dinámico en la técnica de fusión utilizada. Esta elección la realiza el propio bloque de combinación en función de los parámetros de fiabilidad que debe devolver cada método de fusión incorporado en el módulo.

- *La disponibilidad de resultados.* En ocasiones el módulo de seguimiento debe funcionar bajo demanda. Es decir, debe proporcionar un resultado en un determinado intervalo de tiempo. Es posible que no todos los métodos de búsqueda cumplan las restricciones temporales al servir resultados, con lo que estos deben ser descartados. Ello implica que la técnica de fusión empleada en ese momento deba variar.

### Técnicas de fusión de datos

La fusión de datos puede llevarse a cabo desde múltiples aproximaciones, dependiendo de la naturaleza de los datos a fusionar o de las necesidades impuestas. Esto conlleva a que los métodos de fusión en algunos sistemas pueden resultar bastante *ad hoc*. La literatura referente a las técnicas de fusión de datos es vasta y, a veces, caótica. Se pueden encontrar diferentes clasificaciones y extensos análisis de métodos de combinación de datos en Abidi y González (Abidi y Gonzalez, 1992) y en Joshi y Sanderson (Joshi y Sanderson, 1999). Es necesario añadir que estos estudios no sólo incluyen aspectos de la fusión de datos visuales sino también se analiza un amplio rango de diferentes tipos de sensores para la robótica, ofreciendo una idea de lo extraordinariamente amplio que es este campo.

En cuanto a las técnicas de fusión exclusivamente visual, según Clark y Yuille (Clark y Yuille, 1990), es posible establecer una primera clasificación en función del nivel de interrelación existente entre los distintos métodos de búsqueda a combinar. Esto da lugar a dos categorías de fusión: fusión de métodos débilmente acoplados o fuertemente acoplados. Las técnicas de fusión propuestas para el módulo de seguimiento están incluidas dentro de la primera categoría, lo que significa que las salidas de dos o más métodos de búsqueda producen información independientemente del resto de los métodos. No obstante, diseños particulares de módulos de seguimiento pueden concebir otro tipo de fusión para los resultados de sus métodos de búsqueda. Este tipo de fusión se ha escogido dado que permite la mayor independencia posible entre los diferentes métodos de búsqueda.

La independencia de los métodos es importante por varios aspectos que el bloque de combinación debe tomar en cuenta. Entre ellos se encuentran los distintos tiempos de respuesta de los diferentes algoritmos de búsqueda. Al poder constituirse el módulo

de métodos de búsqueda completamente diferentes, estos pueden ofrecer tiempos de respuesta distintos. Los requerimientos temporales del módulo de seguimiento establecen la frecuencia de trabajo del bloque de combinación. Es decir, si se le pide un resultado al módulo de seguimiento en un momento en el que no todos los métodos de búsqueda han finalizado su computación, a la hora de realizar la fusión, sólo se hará uso de los resultados presentes en ese momento, dejando sin fusionar los métodos de búsqueda que no han generado sus respuestas a tiempo. De esta forma, el bloque de combinación debe decidir la forma más apropiada de fusión en función de cuáles y cuántos son los resultados disponibles.

Entre los diferentes métodos de fusión de resultados que puede incorporar un módulo de seguimiento se pueden destacar:

- La *media ponderada* como uno de los métodos más directos e intuitivos consistente en tomar una media ponderada de la información redundante dada por los diferentes algoritmos y usarlo como un valor combinado.
- La *estimación bayesiana* que permite la fusión de datos mediante la utilización de las leyes de la teoría de la probabilidad.
- La fusión mediante *operaciones lógicas* es un método también común de fusión consiste en aplicar la combinación utilizando operaciones lógicas como AND u OR sobre las diferentes fuentes de datos.
- En las técnicas conocidas como de *votación*, diferentes métodos redundantes votan a un conjunto de posibles soluciones. Todos los votos se cuentan, tomándose la solución más votada. Existen en la literatura diferentes maneras de implementar un esquema de votación. Se puede encontrar una buena taxonomía de ellos en Parhami (Parhami, 1994).
- Los *filtros de Kalman* se utilizan de diferentes maneras en diferentes contextos. Una de ellas es para la combinación de varias medidas provenientes de múltiples sensores. Normalmente, estos sensores (métodos de seguimiento) no ofrecen resultados absolutamente coincidentes (se limitan a estimaciones, densidades de probabilidad, etc). Por ello, los filtros de Kalman ofrecen una manera óptima (en un sentido de mínimos cuadrados) de fusionar datos.
- En la fusión basada en *técnicas de inteligencia artificial*, los sistemas expertos basados en reglas pueden ayudar en la fusión de datos, ya que consideran aspectos más allá de los tradicionales, (Peers, 1998). El contexto de funcionamiento de

las aplicaciones reales ofrece una pista bastante rica para reducir y restringir el problema a sólo soluciones posibles.

- Por último, la técnica de fusión denominada *integración democrática* persigue lograr la detección robusta del objeto de interés mediante la integración auto-adaptativa de múltiples pistas (Triesch y von der Malsburg, 2000). El método no trata de seleccionar la pista más apropiada para cada situación sino utilizar las contribuciones de todas las pistas concurrentemente y de forma homogénea. En (Triesch, 2000) se exponen algunos fundamentos biológicos para este tipo de fusión. (Spengler y Schiele, 2001) muestran los resultados comparativos de la integración de múltiples pistas visuales mediante técnicas de integración democrática y "Condensation", (Isard y Blake, 1998).

#### 4.2.4. Información de la imagen y variables de cámara

Cada módulo de seguimiento debe poseer su propio flujo de entrada de imágenes con las que trabajar. El control de este flujo lo realiza el propio módulo, de tal forma que pueda suministrar imágenes o áreas de interés de esas imágenes a cada uno de los distintos métodos de búsqueda de los que disponga. En determinadas implementaciones el hardware de adquisición de vídeo, como digitalizadoras o captadoras, puede formar parte del propio módulo de seguimiento.

La posición, orientación y estado interno de las cámaras cuando se adquieren las imágenes es una información importante para algunos métodos de búsqueda. Por ello, estos datos deben ser recogidos, en el caso de que resulten necesarios, por el módulo de seguimiento, junto con la adquisición de cada nueva imagen a procesar, y almacenados en variables de estado internas de cámara. Estas variables deben encontrarse disponibles para cada método de búsqueda que las demande, y se corresponderán con diferentes parámetros intrínsecos y extrínsecos de las cámaras cuando se capturó el cuadro que se procesa en ese momento. Nótese que esta información no se corresponde con un modelo completo de la cámara ni de su soporte móvil. Su utilidad se justifica cuando el módulo de seguimiento incorpora métodos de búsqueda basados, por ejemplo, en técnicas de flujo óptico, substracción de imágenes o filtros predictivos donde el conocimiento del estado actual y pasado de la cámara es básico para su funcionamiento.

### 4.2.5. Ciclo de trabajo

El ciclo de trabajo del módulo de seguimiento se define como el tiempo transcurrido entre dos resultados sucesivos del módulo. En aplicaciones en tiempo real es común el requerir un determinado tiempo de respuesta a un módulo de búsqueda. En otras ocasiones, sin embargo, este tiempo de respuesta no es crítico y el propio módulo de seguimiento podría informar sobre la disponibilidad de un nuevo resultado robusto de seguimiento. Para contemplar ambas formas de funcionamiento, el módulo de seguimiento debe de poder configurarse para trabajar tanto bajo demanda como mediante la activación de llamadas o interrupciones que indiquen que se dispone de un nuevo resultado.

El control y la gestión no resultan complicados cuando el módulo de seguimiento puede tomarse el tiempo necesario para obtener una respuesta robusta a partir de todos los métodos de búsqueda incorporados. Por el contrario, si se exige un funcionamiento bajo demanda, es necesario tener en cuenta que el ritmo o frecuencia de trabajo de un módulo de seguimiento está estrechamente vinculado a distintos aspectos que se comentan a continuación:

Por un lado, en algunos métodos de búsqueda, tal y como se describió en el Capítulo 3, el entorno que rodea a los objetos de interés influye de manera decisiva sobre el tiempo necesario para llevar a cabo la búsqueda de forma robusta. Por otra parte, la misma concepción de algunos métodos de búsqueda hace que los tiempos de respuesta de los mismos sean variables dependiendo de la complejidad del objeto a seguir. Por todo ello, es tarea del bloque de combinación la gestión adecuada de los resultados disponibles en un momento determinado y proceder a su fusión. Se hace notar que la forma de combinar los resultados puede ser diferente en base a la cantidad o fiabilidad de los resultados disponibles en ese momento por parte de los distintos métodos de búsqueda.

## 4.3. Capacidades del módulo de seguimiento

El objetivo de la versatilidad obliga a aportar al módulo de seguimiento un conjunto lo más completo posible de capacidades, con el fin de hacerlo útil a la mayor cantidad de sistemas de visión que requieran tareas de seguimiento. A continuación se exponen las capacidades del módulo propuesto.



### 4.3.1. Comandos del módulo de seguimiento

Para la integración eficiente y cómoda del módulo de seguimiento dentro de un sistema mayor, el acceso a cada elemento o funcionalidad del módulo se lleva a cabo mediante el uso de una librería de comandos desarrollada al efecto. Esta librería soporta la interfaz del módulo de seguimiento hacia otro módulo o parte del sistema global que haga uso del seguimiento.

El lenguaje escogido para la implementación de la librería ha sido C++. La elección de este lenguaje viene motivada por estar ampliamente extendido y adecuado para el control y acceso a distintos elementos hardware del sistema. Además, su orientación a objetos lo hace conveniente para ocultar gran cantidad de detalles técnicos de la programación. De esta manera, el módulo de seguimiento puede ser visto como una clase en C++ con una serie de métodos que permiten controlar el seguimiento estableciendo los parámetros del mismo de una forma bastante precisa. De esta forma, por ejemplo, la decisión de comenzar el seguimiento de un objeto de la escena se realiza a través de la función de creación de nuevo objeto a seguir (función *CreateObject*), estableciendo los distintos parámetros o descriptores que identifiquen unívocamente al objeto en cuestión.

Todos los comandos del módulo de seguimiento propuesto se encuentran detallados en el apéndice D.

### 4.3.2. Seguimiento de objetos

El proceso de seguimiento de objetos visuales es la actividad principal del módulo. La primera capacidad con la que cuenta el módulo es la de comenzar el seguimiento de un determinado objeto de la escena de dos maneras distintas: automática y asistida. Mediante la primera opción, la captura de los diferentes descriptores que van a localizar en sucesivos cuadros al objeto de interés se lleva a cabo de manera interna al propio módulo, y más concretamente, a cada método de búsqueda incorporado. El único parámetro pasado al módulo al comienzo del seguimiento consiste en la posición en coordenadas de imagen del objeto a seguir. Con esa información cada método de búsqueda captura por sí solo las diferentes características del objeto necesarias en función del contexto de la escena, realizando su actualización de manera también autónoma durante todo el proceso de seguimiento.

La manera asistida de comenzar el seguimiento, por otro lado, se basa en que el módulo superior proporciona explícitamente los descriptores necesarios para realizar el seguimiento del objeto en la imagen. Esta capacidad es importante puesto que posibilita

la búsqueda del objeto en la imagen cuando no se conoce inicialmente su localización. Para iniciar este proceso, el módulo cuenta además con la opción de redimensionar explícitamente el tamaño del área de búsqueda en la imagen.

### 4.3.3. Seguimiento de múltiples objetos

El seguimiento de un objeto se lleva a cabo mediante la introducción de los descriptores correspondientes a ese determinado objeto. Cambiar esos descriptores por otros pertenecientes a otro objeto implica que se abandona el seguimiento del primero por el seguimiento de un segundo objeto distinto. Por tanto, esto conlleva a que pueda verse conceptualmente el cambio de objetivo a seguir como un simple cambio de contexto en las variables del módulo de seguimiento. Con esta facilidad para el cambio de objeto a seguir se pueden adoptar mecanismos que aprovechen las posibilidades que ofrece el seguimiento de múltiples objetos.

#### Seguimiento simultáneo de diferentes partes de un objeto

Una de las utilidades más inmediatas es el seguimiento de un objeto mediante el seguimiento de diferentes partes del mismo. Para posibilitarlo el módulo puede configurarse para obtener la localización de varios objetos durante un ciclo de trabajo. La condición necesaria para que esto pueda llevarse a cabo reside en que todas estas partes se encuentren dentro de la imagen a procesar. Una vez todas las partes del mismo objeto han sido encontradas, es necesario devolver un único resultado. Para ello se retorna la media geométrica de todos estos puntos.

Entre las ventajas que aporta el seguimiento de un objeto mediante varias de sus partes está el incremento en la robustez obtenido. Es posible de esta forma tratar de manera más eficiente la ocultación de parte del objeto. En la Figura 4.4 se muestra como se ha llevado a cabo el seguimiento de un coche mediante el seguimiento simultáneo de su parte delantera y trasera. Cuando en el Cuadro 3, la parte delantera es ocultada por la columna, la parte trasera es usada para no perder definitivamente el coche. Cuando la parte delantera aparece de nuevo, puede seguirse como antes. Es además necesario, en este ejemplo, un cierto conocimiento de la estructura del objeto a seguir puesto que en caso contrario el seguimiento de la parte delantera del coche quedaría en el Cuadro 3 fija sobre la columna sin generar ningún tipo de error. Por tanto, es necesario mantener la coherencia de una determinada relación espacial entre todas las partes que conforman el seguimiento del objeto.

Es importante destacar que el seguimiento que el módulo puede llevar a cabo de forma autónoma tiene lugar en un único ciclo de trabajo del mismo. Esto implica que todas las partes de un objeto único deban estar contenidas dentro de la misma imagen. Esto es así puesto que el módulo de seguimiento propuesto no posee mecanismos directos de control sobre la orientación de la cámara por la que recibe las imágenes.

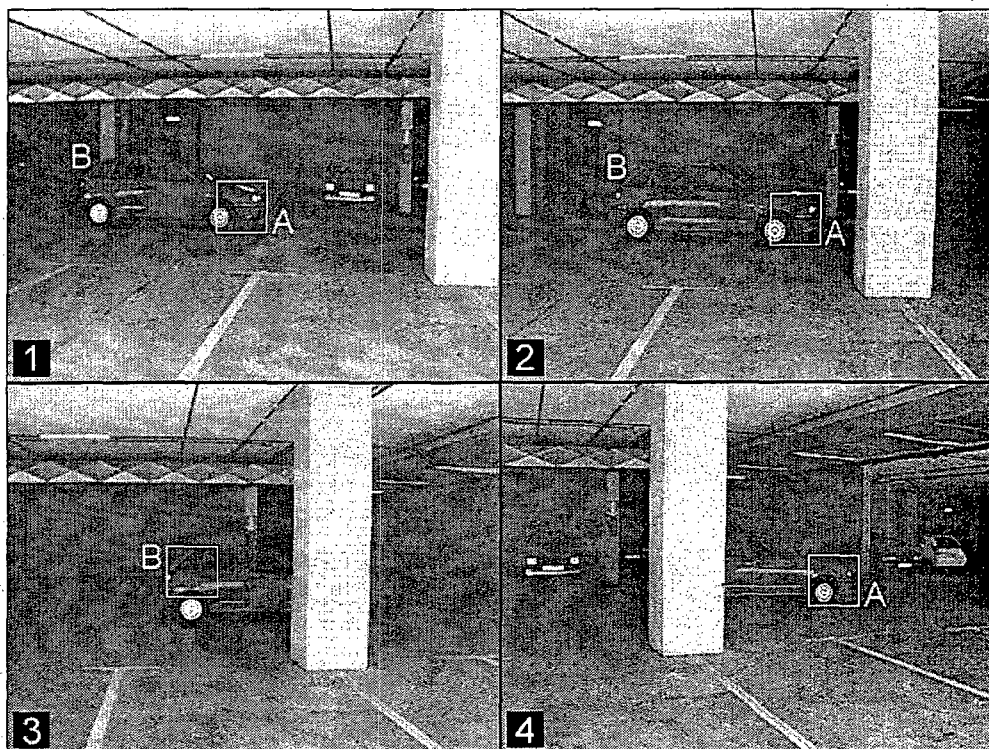


Figura 4.4: La figura muestra cuatro cuadros de una secuencia donde un coche es seguido utilizando diferentes partes de él.

### Seguimiento múltiple de diferentes objetos

El diseño del módulo de seguimiento ofrece tan solo un resultado de localización por ciclo de trabajo. No debe confundirse por tanto la localización de múltiples objetos que el módulo de seguimiento puede realizar internamente, como se ha descrito en la sección anterior, con el seguimiento de varios objetos individuales. Debido a esto, si los objetos a seguir se encuentran dentro de la misma imagen y se pretende un seguimiento independiente de cada uno de ellos, es decir se pretende que el módulo de seguimiento devuelva una localización para cada uno de los objetos, el procedimiento a tomar debe ser el seguimiento secuencial y cíclico de cada uno de ellos. De esta forma, la decisión de conmutación de la búsqueda entre ellos debe ser generada por un módulo superior

externo que comande este cambio de objetivo al módulo de seguimiento.

La secuencia de comandos a ejecutar por parte del módulo superior al módulo de seguimiento para seguir dos objetos sería: obtención de la localización del primer objeto, conmutación al segundo objeto mediante el cambio de descriptores, localización del segundo objeto y conmutación a los descriptores del primer objeto para seguir de nuevo con el primer paso. De esta forma se cerraría un ciclo de funcionamiento de un seguimiento de múltiples objetos dentro de la misma imagen.

### **Seguimiento de objetos que no se encuentran dentro del mismo campo visual**

Un problema ligeramente distinto a los expuestos anteriormente consiste en el seguimiento de objetos que no se encuentran dentro del mismo campo visual. Tanto si los diferentes objetos a seguir se corresponden con partes de un objeto como si son independientes, el sistema de visión debe realizar algún movimiento de cámara a fin de poder situar en el campo de vista las partes que no se encontraban previamente.

Estas acciones no pueden ser ejecutadas autónomamente por el módulo de seguimiento debido a que éste no posee control sobre los actuadores de las cámaras. Es decir, de forma parecida al apartado anterior, un módulo externo debe comandar una determinada secuencia de acciones que coordinen la búsqueda que realiza el módulo de seguimiento y el control del movimiento de la cámara.

Como ilustra la Figura 4.5 el sistema se compondría conceptualmente de los módulos que se relacionan a continuación, siendo el módulo superior el que controla toda la secuencia de acciones que se realizan. La primera acción que éste lleva a cabo es comandar al módulo de seguimiento para que localice y siga un determinado objeto dentro de la imagen. Una vez se tiene el objeto localizado, el módulo de seguimiento envía el resultado al módulo superior, que genera un comando de movimiento de la cámara al control de actuadores hacia el otro objeto que se está siguiendo junto con un comando de pausa para el módulo de seguimiento. Esto se debe a que durante el periodo de transición entre objetos o movimientos balísticos en sistemas de visión activa las imágenes son probablemente borrosas por defectos del mismo movimiento del sensor. Este hecho es lo que diferencia ligeramente al proceso de seguimiento de objetos dentro del mismo campo visual descrito anteriormente. Cuando la cámara es situada en su nueva posición, se reanuda la operación de seguimiento con el comando de localización del siguiente objeto al módulo de seguimiento.

El tiempo que el módulo de seguimiento está con cada objeto depende del módulo superior. Este tiempo puede ser de varios milisegundos a varios segundos dependiendo

de la aplicación concreta. En DESEO (Hernández et al., 1999), una de las aplicaciones desarrolladas y detallada en el capítulo correspondiente a los experimentos, se realiza el seguimiento de dos objetos, en este caso las caras de dos personas. El tiempo dedicado al seguimiento de cada cara es de un segundo. El tiempo de conmutación entre ambas depende básicamente del retardo mecánico de la cabeza motorizada sobre las que se sitúan las cámaras. En el caso de DESEO éste se encuentra en torno a la décima de segundo, véase Figura 4.6. Normalmente los objetos seguidos poseen movimiento propio, de tal forma que la precisión en el posicionamiento de la cámara puede mejorarse mediante la introducción de técnicas de estimación del movimiento de estos objetos. Sin embargo, mientras el objeto no es seguido, puede cambiar súbitamente su dirección de movimiento provocando que el posicionamiento de la cámara no sea totalmente correcto, lo que hace necesaria la ampliación del área de búsqueda durante la primera localización del objeto por parte del módulo de seguimiento.

Una visión detenida de la Figura 4.6 muestra como en el cuadro *a* una persona permanece quieta y la imagen se centra en su cara. En el cuadro *b* una nueva persona aparece, después de unos instantes y a petición de un módulo superior, el módulo redirige la atención hacia esta nueva persona. El cuadro *c* muestra lo que el módulo captura durante un movimiento sacádico. Estos cuadros borrosos son completamente inútiles como fuentes de datos y no son procesados. El tiempo tomado por un movimiento sacádico suele ser menor de un cuarto de segundo. El cuadro *d* representa el módulo de seguimiento centrado en la cara de la segunda persona. A partir de ahora, el sistema conmutará su atención entre las dos personas a una frecuencia de 1 Hz. En el cuadro *e*, la primera persona comienza a caminar a un paso normal, el sistema mantiene el seguimiento sin problemas. Después de un segundo, el sistema vuelve a la segunda persona y lo encuentra sin problemas, cuadro *f*. Después de otro segundo, el sistema tratará de nuevo de encontrar a la primera persona. En este caso, esta persona estaba andando mientras el sistema estaba ocupado atendiendo a la otra. Incluso utilizando técnicas de predicción, es altamente improbable que el sistema no alinee las cámaras justo en la cara, de esta forma, se utiliza un área de búsqueda mayor para su localización, cuadro *g*. Después de localizarla, el sistema se centra sobre ella (cuadro *h*) y el área de búsqueda recupera su tamaño normal. El cuadro *i* muestra el regreso a la segunda persona. El cuadro *j* es otra imagen borrosa también debida al movimiento sacádico. Mientras el sistema estaba en el cuadro *i*, la primera persona cambió su dirección de movimiento, lo que no puede ser controlado por la técnica predictiva implementada. El cuadro *k* muestra que el sistema se estabiliza con la persona muy cerca del borde de la imagen, sin embargo, con un área de búsqueda mayor es posible traer de nuevo la cara

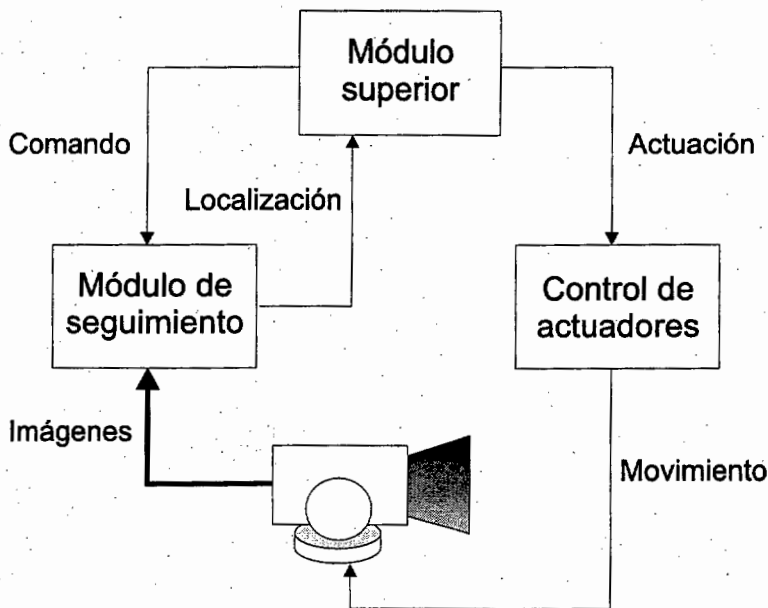


Figura 4.5: Esquema de componentes de un sistema de seguimiento utilizando una plataforma móvil para la cámara. Como se observa, el módulo de seguimiento no gestiona de forma directa el control de actuadores sobre los motores de la plataforma, sino que este control lo realiza externamente otro módulo del sistema.

al centro de la imagen (cuadro  $l$ ).

#### 4.3.4. Combinación de módulos de seguimiento

El módulo de seguimiento definido posee un diseño conceptual externo sencillo, de tal manera que resulta apropiado para aplicaciones que trabajen sobre entornos poco problemáticos. De la misma manera resulta suficientemente flexible y apto para formar módulos más complejos mediante la combinación de estos y con ello poder afrontar problemas de seguimiento sobre escenarios más complicados.

##### Combinación serie

Una de las posibles formas de combinación de diferentes módulos de búsqueda es la combinación en serie. Este tipo de combinación permite realizar esquemas de seguimiento de manera gradual en la precisión del resultado obtenido. Como ejemplo se puede destacar la utilización de módulos de seguimiento por color combinado serialmente con búsqueda de patrones. De tal forma que en un primer paso, la detección del objeto mediante el color devuelve una localización vaga o imprecisa que sirve como aproximación o definición de la zona de búsqueda de un segundo módulo de seguimiento basado en



Figura 4.6: En estos doce cuadros de una secuencia, tomada del sistema DESEO, es posible observar el comportamiento del módulo realizando el seguimiento de dos objetos (caras) con movimiento independiente a lo largo del tiempo.

búsqueda de patrones, cuyo resultado ofrece una mayor precisión en la situación del objeto seguido.

En la Figura 4.7 se muestra un ejemplo real de este tipo de combinación. Los resultados de la detección de una cara por parte de un módulo de seguimiento formado exclusivamente por un método de detección por color sirve como parámetro de área de búsqueda a otro módulo de seguimiento constituido por un método de búsqueda basado en correspondencia de patrones. Nótese que, en este caso, el resultado del módulo basado en color es el centro de la región o blob del color buscado, sin embargo la lectura de los descriptores ofrecen una información más adecuada como entrada al segundo módulo, ya que entre los descriptores del un módulo de seguimiento basado en color se halla el rectángulo que circunscribe la región (blob) de color detectada. Este rectángulo ahora tiene como misión definir el área de búsqueda del módulo basado en la correspondencia de patrones.

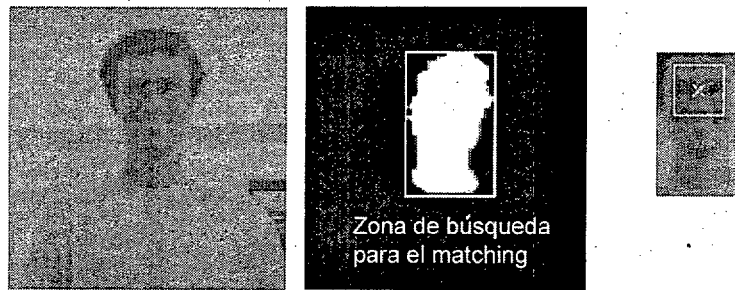


Figura 4.7: En la figura se observa cómo puede ser localizada una cara en una imagen mediante el uso secuencial de dos módulos de búsqueda, uno basado en color y otro basado en la correspondencia de patrones. En el primer lugar, una aproximación a la zona de la cara se lleva a cabo por medio de la búsqueda del color característico de la cara. En un segundo paso, utilizando la región de color devuelto por el primer módulo, el segundo módulo puede llevar a cabo una búsqueda más precisa sobre esa área.

Otro de los ejemplos que ilustran la potencialidad que ofrece la combinación en serie de módulos de seguimiento consiste en la estabilización previa de imágenes para otros métodos de búsqueda basados en detección de movimiento, que suelen trabajar mediante la substracción de imágenes o flujo óptico. Este tipo de situaciones se suele dar cuando el seguimiento se realiza con egomovimiento de la cámara y se pretende eliminar el movimiento inducido del fondo. Es necesario además que este fondo tenga un carácter bidimensional o al menos se le pueda aproximar como tal, de forma que no influya su perspectiva generando efectos no deseados. Este tipo de escenarios se pueden dar, por ejemplo, en secuencias terrestres tomadas desde el aire por aviones, satélites, etc. En este caso el terreno aparece prácticamente bidimensional. Una vez estabilizado el terreno se pueden extraer los objetos en movimiento.

### Combinación paralela

En secciones anteriores se expuso cómo el seguimiento de múltiples objetos era llevado a cabo por un único módulo de seguimiento. Este seguimiento era realizado realmente de manera secuencial. En este caso, la combinación de módulos de seguimiento en paralelo permite el seguimiento simultáneo de varios objetos. Ahora cada módulo realiza un seguimiento independiente de un objeto, generando por tanto una respuesta consistente en la localización del mismo, véase Figura 4.8. Todas estas respuestas juntas pueden ser enviadas a donde proceda según el esquema de funcionamiento del sistema completo. Las imágenes que los módulos de seguimiento reciben, pueden proceder de la misma fuente de imágenes para todos los módulos, o provenir de diversas cámaras. Esto



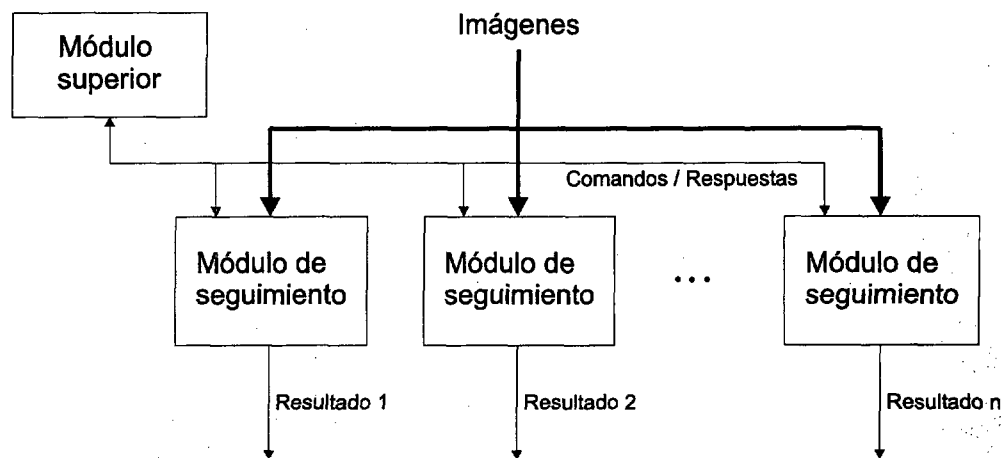


Figura 4.8: La figura muestra la combinación de varios módulos de seguimiento actuando en paralelo. Tanto las entradas como las salidas de estos módulos son independientes entre sí.

depende del propósito del sistema.

### Combinación recursiva

Otra de las características relevantes del módulo de seguimiento propuesto es su naturaleza recursiva, ya que cada módulo de seguimiento puede actuar a su vez como un posible método de búsqueda más dentro de otro módulo de seguimiento mayor. Esto se debe a que las entradas y las salidas o resultados de un método de búsqueda son básicamente equivalentes entre un método de búsqueda y un módulo de seguimiento. De esta forma, el bloque de control del módulo de seguimiento puede controlar un módulo de seguimiento completo como si se tratara de otro método de seguimiento más.

Este tipo de combinación resulta útil cuando se desea construir módulos de seguimiento complejos, donde, por ejemplo, se necesiten implementar distintos métodos de combinación de resultados concurrentemente, Figura 4.9.

#### 4.3.5. Adaptación del módulo de seguimiento a sistemas de visión activa

Se enumeran a continuación los diferentes aspectos que ofrece el módulo de seguimiento que facilitan su incorporación en sistemas de visión activa, (Aloimonos y Weiss, 1988). Todas estas características se hallan ampliamente descritas en sus respectivas secciones.

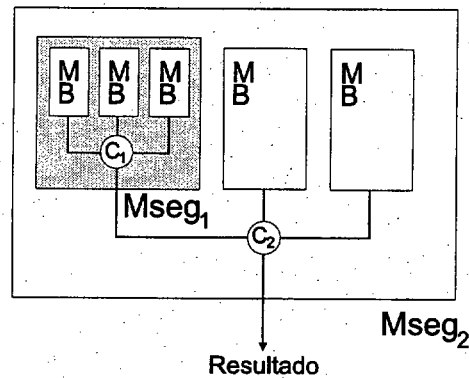


Figura 4.9: Representación esquemática de cómo llevar a cabo distintas formas de fusión ( $C_1$  y  $C_2$ ) de diferentes métodos de búsqueda (MB) mediante la combinación recursiva de módulos de seguimiento ( $Mseg_1$  y  $Mseg_2$ ).

- Una de las principales características que poseen los sistemas de visión activa es el estrecho control en bucle cerrado que requieren sobre el sistema de adquisición de imágenes. Este control conlleva la alteración constante de los parámetros intrínsecos y, fundamentalmente, extrínsecos de las cámaras. Por tanto, para realizar los procesos de seguimiento en sistemas con egomovimiento, el módulo de seguimiento tiene la capacidad de aceptar información sobre los parámetros de posición y de orientación con el fin de permitir la correcta realización de las funciones de búsqueda a los métodos sensibles a los cambios en los parámetros extrínsecos de las cámaras.
- Otro de los aspectos considerados en el diseño del módulo de seguimiento es la capacidad de ofrecer respuestas en periodos de tiempo definidos. Es decir, su operación interna permite generar una respuesta en un tiempo concreto, posibilitando su funcionamiento a frecuencias determinadas fijadas por el sistema de visión activa que integre el módulo.
- La utilización de un banco interno de objetos visuales al módulo de seguimiento permite al sistema de visión activa conmutar la atención entre diferentes objetos sin la necesidad de almacenar ninguna representación de los objetos de interés seguidos por parte del sistema de visión durante su funcionamiento normal.

Todas estas características son accesibles desde el conjunto de comandos que incluye el módulo de seguimiento.

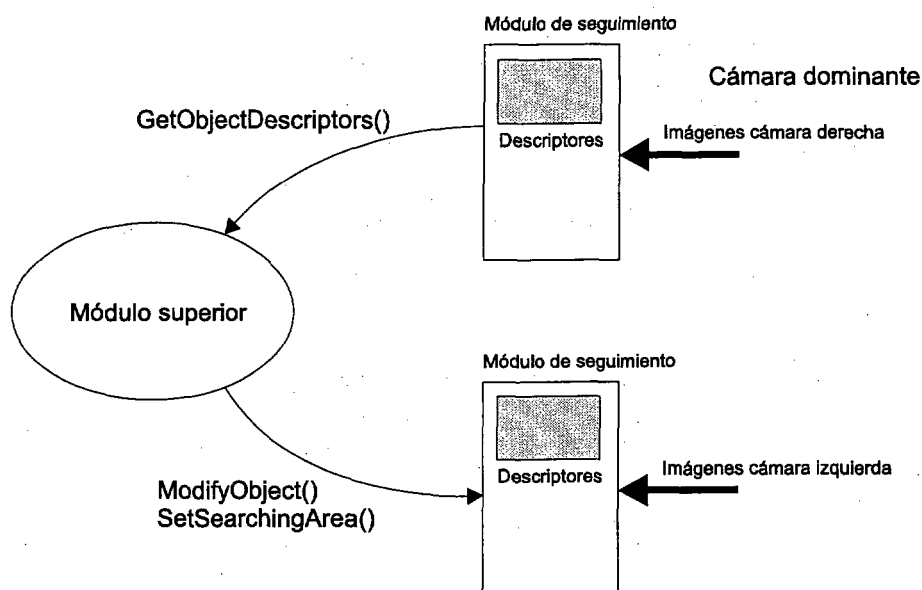


Figura 4.10: La figura muestra la combinación de dos módulos de seguimiento para realizar estereovisión. En este caso los descriptores del módulo dominante son pasados hacia el otro, de esta forma se asegura la concordancia de los objetos en ambos módulos. Además, se establece la zona de búsqueda sobre la línea epipolar del módulo no dominante.

#### 4.3.6. Adaptación al estéreo

En numerosas ocasiones, el seguimiento de objetos mediante sistemas de visión se realiza de forma binocular. Conceptualmente, esto se puede llevar a cabo mediante el seguimiento del mismo objeto por parte de dos cámaras, cada una de las cuales posee asociado un módulo de seguimiento. La principal dificultad radica en que ese objeto *debe* ser el mismo para un correcto funcionamiento del seguimiento.

Una de las formas de asegurar la concordancia entre los objetos seguidos por las dos cámaras es hacer que los dos módulos de seguimiento almacenen los mismos descriptores del objeto, además de obedecer a las restricciones epipolares impuestas. Esta compartición de descriptores es posible puesto que a partir de una determinada distancia del objeto a las cámaras, el ángulo de convergencia (*vergence angle*) se vuelve lo suficientemente pequeño como para que el cambio en la apariencia del objeto debido a la diferente localización de las dos cámaras resulte despreciable, véase Figura 4.10.

Es también posible no adoptar un esquema de ojo dominante y permitir que los dos módulos de seguimiento sean independientes en cuanto a sus descriptores, aunque aún así deben mantener las restricciones epipolares.

### 4.3.7. Pérdida del objeto

Una de las características que debe poseer un módulo de seguimiento es la detección de un posible error en el seguimiento. Es tan importante seguir un objeto correctamente como no seguir un objeto erróneo. Sin embargo, es difícil en algunas ocasiones concluir con una fiabilidad alta que el seguimiento de un objeto ha fallado. Es posible que la atención del módulo se haya centrado en otro objeto distinto del deseado de manera completamente desapercibida.

Esta pérdida puede haber sido provocada por diversos motivos vinculados con el método de búsqueda en si y por distintas características de la imagen de la escena. Por tanto, en los diferentes métodos de búsqueda se establecen distintos mecanismos para detectar errores en el seguimiento. Debido a que muchas de estas técnicas de búsqueda basan su funcionamiento en el establecimiento de una medida que indica el grado de similitud del objeto encontrado con el original de referencia, es normal el uso de un umbral que establezca el límite entre lo que se considera el objeto seguido o la pérdida de éste.

No sólo pueden emplearse los umbrales para detectar la pérdida de un objeto. Otros muchos factores pueden también indicarla. Cambios súbitos de posición del objeto en la imagen, aumentos bruscos en la medida de similitud del método de búsqueda, cambios globales en la escena, etc. son algunas de las pistas que pueden detectar un error en el seguimiento.

En técnicas genéricas de búsqueda precategóricas donde la detección de un objeto se basa en alguna característica concreta del mismo, la cual no resulta completamente invariable, no es posible afirmar con una certeza absoluta la posible pérdida en el seguimiento.

El módulo de seguimiento propuesto debe generar por tanto, junto con el resultado del seguimiento, un valor que estime la fiabilidad del mismo. Para ello, la fiabilidad de cada método de búsqueda resulta también fusionada de forma análoga al resultado, por el bloque de combinación, generando un único valor.

### 4.3.8. Recuperación

Cuando el módulo de seguimiento retorna activado el indicador de pérdida para señalar que el objeto seguido no se ha podido localizar, se debe iniciar un proceso de recuperación del mismo. Para ello, cada método de búsqueda resulta comandado por el bloque de control de forma especial para tratar este estado. Las causas de la pérdi-

estar basado en su propia placa de procesamiento independiente del hardware del sistema o, por el contrario, compartir los recursos computacionales con el resto de las tareas visuales del sistema. En cualquier caso, el módulo de seguimiento debe contar con un mínimo garantizado de recursos computacionales para llevar a cabo su función.

Para la implementación de los módulos de seguimiento realizados en esta tesis se ha recurrido a estos dos modos de computación. Por un lado, se ha construido un módulo de seguimiento sobre una placa DSP independiente del hardware del sistema. Esta opción permite un control total sobre los tiempos de cómputo y rendimiento del módulo de seguimiento. Por otro lado, el módulo de seguimiento construido sobre una arquitectura monoprocesadora compartida entre diferentes tareas hace preciso disponer de un sistema operativo que garantice la utilización del microprocesador por parte del módulo de seguimiento con una determinada frecuencia y durante un cierto periodo de tiempo.

- *Acceso al sistema de captura de imágenes.* Las diferentes tareas visuales de que pueda constar un sistema de visión necesitan tener acceso con una cierta frecuencia a las imágenes capturadas por el sistema de adquisición de imágenes.

En sistemas monoprocesadores, como estaciones de trabajo de propósito general, donde las diferentes tareas visuales se encuentran programadas como hebras o procesos, la opción más adecuada es, en muchas ocasiones, la de dejar disponibles las imágenes capturadas en zonas de memoria compartida. Esto permite a las tareas visuales acceder a las imágenes cuando las requieran. De no ser así, una tarjeta digitalizadora que funcione como sistema de adquisición se podría encontrar bloqueada por alguna tarea durante ciertos intervalos de tiempo, impidiendo la toma de imágenes a las demás tareas.

En otras ocasiones, el módulo de seguimiento puede constar de su propio recurso de adquisición de imágenes y de esa forma acceder a las imágenes directamente. Esto conlleva como beneficio el poder establecer ciertos parámetros de la cámara por el propio módulo de seguimiento.



## Capítulo 5

# Evaluación experimental y aplicaciones

El objetivo del apartado experimental de esta tesis es doble. Por un lado, la evaluación empírica y de manera exhaustiva de las técnicas de actualización de patrones, descritas en el Capítulo 3, mediante su comparación con el método de actualización basado en contexto. Por otro, la descripción y exposición de los resultados del desarrollo de una aplicación práctica basada en los conceptos previamente ensayados.

Dentro de los algoritmos de búsqueda basados en correspondencias de patrones, la actualización del patrón resulta, en la mayoría de los contextos de aplicación, fundamental para su correcto funcionamiento. Los experimentos descritos a continuación reflejan los comportamientos del mismo algoritmo de búsqueda basado en la comparación de patrones cuando se utilizan diferentes técnicas de actualización del patrón a buscar. La técnica de actualización del patrón basada en contexto se compara con las técnicas de patrón constante, actualización constante, actualización de periodo constante, actualización mediante umbral constante y actualización basada en estadísticos. De estos métodos, se realiza un análisis en profundidad de las técnicas de actualización basadas en umbral constante y en estadísticos, ya que son las más evolucionadas. Debido a la simplicidad del comportamiento de los tres primeros métodos hace innecesario un estudio mayor, y su comparación aquí se reduce a una descripción de sus importantes limitaciones.

Es necesario reiterar que todas las técnicas elegidas para la realizar la comparación son exclusivamente técnicas de actualización del patrón. En la bibliografía existen numerosos trabajos de seguimiento basados en la búsqueda de patrones que incluyen diferentes procedimientos adicionales con el fin de obtener mejoras en la robustez de la correspondencia, como pueden ser la normalización de los niveles de grises del patrón y de la imagen, escalados de los patrones, diferentes funciones de distancia, etc. Todos

estos trabajos tienen en su interior algún esquema de actualización del patrón, y es este esquema el objeto de experimentación aquí tratado.

En otras palabras, las mejoras introducidas por los diferentes trabajos existentes para realizar el proceso de búsqueda de patrones no corresponden a alternativas a los métodos comunes de actualización del patrón sino a complementos adicionales del método de búsqueda para incrementar su robustez. Es por esto que el proceso de búsqueda de patrones utilizando la actualización del patrón basada en contexto, propuesto en esta tesis, es también utilizable de la misma forma mediante la inclusión de los diferentes complementos adicionales comentados.

Resulta evidente que los múltiples factores que pueden influir en el seguimiento de objetos basados en la búsqueda de patrones (como cambios bruscos de luminosidad, proyecciones de sombras, movimientos rápidos, ocultaciones parciales, etc.) requieren algún tipo de procedimiento adicional que evite la pérdida puntual del objeto en el instante en que ese cambio brusco tiene lugar. Sin embargo, durante el seguimiento en condiciones normales del objeto, éste presenta unas transformaciones visuales que son perfectamente asumibles por un esquema de actualización del patrón para su correcto seguimiento. Este tipo de transformaciones visuales comunes del objeto seguido conforman el escenario de interés de estos experimentos.

La actualización del patrón es necesaria siempre que el objeto a buscar pueda cambiar visualmente de alguna manera. Además, esta actualización del patrón debe realizarse en los momentos adecuados para evitar los dos tipos de errores más comunes en el seguimiento basado en la búsqueda de patrones: *derivadas del patrón* y *errores por saltos en el objeto de interés*, véase Capítulo 3.

Los experimentos expuestos a continuación muestran la bondad de los diferentes métodos de actualización para adaptarse al cambio normal y progresivo del objeto. Esta adaptación se medirá en función de los errores de deriva, provocados por una excesiva actualización, y en función de los errores de salto, provocados por una falta de actualización. Nuestra hipótesis, posteriormente confirmada, se basa en un ajuste adecuado de la tasa de actualización. Proponemos que el mejor método de actualización será aquel que realice de forma correcta el seguimiento con el menor número posible de actualizaciones.

Esto es así debido a que el proceso de seguimiento basado en la búsqueda de patrones posee un funcionamiento de carácter totalmente precategórico. Es decir, en el momento de realizar una nueva actualización del patrón no se lleva a cabo una interpretación previa de la escena con el fin de escoger de forma segura la nueva vista del



objeto de interés. Por lo tanto, el seguimiento basa gran parte de su funcionamiento en la asunción de que el objeto a seguir muestra un comportamiento coherente en cuanto a sus transformaciones visuales a lo largo del tiempo. Esto hace que en cada actualización del patrón la fiabilidad de que el nuevo patrón corresponda realmente a la nueva vista del objeto de interés disminuya.

Antes de llevar a cabo la comparación del método de actualización basado en contexto con los demás métodos de actualización, es necesario caracterizar el dispositivo de adquisición de imágenes utilizado a lo largo de estos experimentos. Esta caracterización permitirá la obtención de ciertos elementos importantes para el análisis en profundidad de los algoritmos de búsqueda de patrones. Esta caracterización tiene que ver fundamentalmente con el ruido de digitalización del CCD de la cámara y de la posterior compresión digital. Esto es importante debido a que el ruido afecta de forma aditiva a las medidas de distancia empleadas durante la búsqueda de los patrones.

Como complemento a este capítulo y en el Apéndice E se describe el software desarrollado para la realización de los experimentos: WinVision. Este programa permite llevar a cabo la búsqueda de patrones sobre arquitecturas Pentium III y tomar las diferentes mediciones experimentales. En el Apéndice D se expone el código de búsqueda de patrones que utiliza instrucciones MMX. Este repertorio de instrucciones en ensamblador hace posible la ejecución de este algoritmo en tiempo real, una de las aportaciones de este trabajo en el que se consigue hacer seguimiento mediante arquitecturas de propósito general.

Una vez completado el análisis experimental comparativo de las técnicas de actualización del patrón, se incluye una siguiente sección dedicada al estudio experimental del comportamiento que posee el módulo de seguimiento cuando cuenta con un banco de patrones a modo de memoria visual, véase Capítulo 3.

Como comentamos en el primer párrafo de este capítulo y por último, se describe la utilización de un módulo de seguimiento construido de acuerdo a las directrices de diseño propuestas en el Capítulo 4. Este módulo de seguimiento se integra dentro del funcionamiento de un sistema completo que realiza la detección y el seguimiento de objetos denominado DESEO. Es necesario señalar que no se muestra de forma explícita ningún resultado experimental ni comparativo de este sistema, su interés radica en la descripción de la relación funcional y arquitectónica entre el módulo de seguimiento y el resto del sistema DESEO.

## 5.1. Caracterización del dispositivo de adquisición de imágenes

Las secuencias para la realización de los experimentos descritos en esta sección se adquirieron, en la mayoría de los casos, utilizando una cámara digital Sony DCR-PC5 DV. Entre sus características más notables hay que destacar: Grabación digital, 500 líneas horizontales de resolución, lentes Carl Zeiss Vario-Sonnar, zoom óptico 10X e interface para la conexión digital al ordenador iLINK (IEEE-1394) DV.

### 5.1.1. Cálculo del nivel de ruido

Las señales capturadas mediante dispositivos o sensores electrónicos contienen algún componente de ruido aleatorio asociado. Su origen se encuentra en el propio principio de operación del dispositivo y en este caso, los CCDs de las cámaras digitales no son una excepción.

En este punto se pretende exponer un sencillo experimento destinado a cuantificar el nivel de ruido generado por el sensor visual utilizado, ya que éste influirá sobre la elección de los umbrales mínimos de actualización del patrón. La función de similitud tomada para la realización de la búsqueda del patrón lleva a cabo el cálculo de la sumatoria en valor absoluto de las diferencias entre los pixels del patrón y de la imagen. El ruido generado por la cámara conlleva a un incremento en el resultado de este cálculo y que será proporcional al tamaño del patrón. Este nivel de ruido establece un mínimo para el umbral a elegir para la actualización del patrón, ya que por debajo de este valor la diferencia entre patrón e imagen puede deberse exclusivamente al ruido.

Cuando las imágenes son digitalizadas mediante CCDs, los valores de los pixels obtenidos están en proporción a la intensidad de luz recibida. Idealmente, si la intensidad de luz recibida permaneciera constante, es decir, no hubiera ningún tipo de movimiento ni variación de luz en la escena, entonces el valor de los pixels debería mantenerse inalterado. Debido al ruido esto no sucede así, los valores de los pixels varían ligeramente a lo largo de los diferentes cuadros. Los motivos del ruido son múltiples y corresponden a distintas etapas dentro del proceso de digitalización, algunos serán despreciables y otros tendrán algún tipo de relevancia dependiendo de la aplicación. Básicamente, todos ellos se pueden clasificar dentro de dos grupos: los que poseen motivación eléctrica y los que la tienen térmica.

Antes de caracterizar de forma experimental el ruido generado por el CCD es

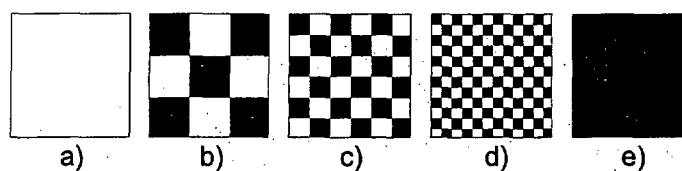


Figura 5.1: Patrones sintéticos para el cálculo de la cantidad de distorsión que induce cada uno mediante la compresión DV.

necesario introducir un nuevo factor de distorsión de la señal original que ha de añadirse a los anteriores, el efecto provocado por los codecs (codificadores-decodificadores) de la compresión DV. Tiene lugar debido a la reducción de precisión en las altas frecuencias que lleva a cabo el compresor para eliminar información con objeto de obtener un mayor ratio de compresión.

### Efecto de la compresión DV

Las altas frecuencias en la imagen corresponden a zonas de alto contraste donde la variación de niveles altos a bajos o viceversa es muy abrupta. Variaciones en el valor de un pixel provoca una alteración en los valores de los pixels vecinos que se acentúa en las zonas de alto contraste. En la Figura 5.1 se presenta una serie de patrones utilizados para ilustrar el efecto generado en función de la cantidad de contraste que contiene cada patrón. En la Tabla 5.1 se muestra que para los patrones planos *a* y *e* el efecto es menor que para los restantes patrones con mayor contraste de niveles. Los patrones *b*, *c*, y *d* muestran como a medida que la cantidad de contraste aumenta también crece el efecto de la compresión.

Patrón	Valor del efecto por pixel	Efecto de la compresión sobre un área de $24 \times 24$ pixels
a	1.41	815
b	3.4	1956
c	3.42	1971
d	4.7	2707
e	2.25	1298

Cuadro 5.1: Valores de la distorsión en los patrones inducida por la compresión DV. Se aprecia como a medida que el patrón muestra una densidad de cambio mayor el efecto generado aumenta. El efecto por pixel corresponde a la variación promedio en el nivel de gris de cada pixel.

El cálculo de esta distorsión inducida por la compresión se ha realizado generando sintéticamente los cinco patrones expuestos y posteriormente comprimiéndolos en

formato DV. El valor del efecto provocado en cada patrón consiste en la suma de las diferencias absolutas entre el patrón generado original y el patrón una vez codificado en DV.

### Ruido en patrones reales

En un caso real se puede observar cómo afecta el ruido a diferentes zonas de una escena como la que muestra la Figura 5.2. Las zonas marcadas con cuadrados negros representan las áreas donde se han tomado medidas del ruido. En el área central *A* obtenemos un valor de 2303, en el área *B* el valor desciende a 1529. En una zona sin textura como el área *C* se obtiene un mínimo de 1219. Por último, en la región *D* cuya densidad de cambio es mayor se obtiene un valor máximo del ruido de 2564. El ruido observable es ahora función tanto del inducido por el propio dispositivo como el efecto generado por la compresión DV.



Figura 5.2: Diferentes localizaciones dentro de la misma imagen ofrecen distinta cantidad de ruido. Esta diferencia es debida a cómo influye la cantidad de variación en los niveles de los pixels dentro de ese área de la imagen en la compresión digital DV.

Estos valores han sido calculados mediante la comparación de múltiples adquisiciones de la misma escena estática, posteriormente se han medido las diferencias entre pixels correspondientes en las distintas áreas seleccionadas y por último se ha obtenido la suma absoluta de estas diferencias.

**Conclusión:** La finalidad de estos experimentos de caracterización del dispositivo de adquisición de imágenes ha sido de la poder contar con unas medidas básicas de error que permitan establecer un umbral mínimo de trabajo para las medidas de similitud

Área	Error por pixel	Error sobre un área de $24 \times 24$ pixels
A	3,99	2303
B	2,65	1529
C	2,11	1219
D	4,45	2564

Cuadro 5.2: Valores del error. Al igual que en el caso sintético se aprecia también como a medida que el área muestra una densidad de cambio mayor el ruido generado aumenta. Además del ruido provocado por el compresor DV en las imágenes reales hay que contar con el error térmico propio del CCD.

empleadas en los posteriores experimentos. De esta forma es posible saber cuando una medida de similitud es debida a la diferencia real entre los pixels de los patrones y cuando puede ser debida en parte al propio ruido de adquisición. Para patrones de tamaño  $24 \times 24$  es posible, según estas pruebas de ruido, establecer un umbral mínimo de 3000. Por tanto, medidas de similitud menores a este valor pueden ser consideradas como debidas a ruido.

## 5.2. Actualización basada en contexto vs. patrón constante

La utilización del patrón constante puede funcionar de manera correcta únicamente en aplicaciones donde el objeto a seguir no altera excesivamente su aspecto visual. En otro caso, el patrón se perderá tan pronto como el objeto modifique, en una tasa superior a la permitida, su aspecto visual. Por tanto, la comparación entre el mantenimiento del patrón constante y el uso de la actualización basada en contexto tiene sentido, tal como se ha dicho, siempre y cuando el objeto a seguir no altere significativamente su apariencia. Mantener el patrón constante permite realizar el seguimiento de objetos que tan sólo varíen su posición  $x, y$  en la imagen.

**Conclusión:** Según esta condición, la comparación entre la actualización del patrón basada en contexto y el mantenimiento del patrón constante resulta trivial. Mientras el objeto de interés no modifique excesivamente su apariencia y no existan tampoco elementos del contexto que puedan confundir la búsqueda, ambos métodos ofrecerán un resultado equivalente.

## 5.3. Actualización basada en contexto vs. actualización constante

La actualización constante está basada en el cambio del patrón en cada cuadro de la secuencia. Es decir, el patrón obtenido en el cuadro anterior es el utilizado para llevar a cabo la búsqueda del objeto de interés en el cuadro actual. Una vez localizado éste es utilizado como patrón para la búsqueda en el siguiente cuadro. Este esquema de actualización es en teoría el que debería conseguir la mejor adaptación al objeto de interés en cada momento. Sin embargo, dada la naturaleza digital tanto de las imágenes como del patrón, tiene lugar un fenómeno de deriva provocado por la precisión en la digitalización de las imágenes, véase Capítulo 3.

La deriva es claramente observable en la secuencia mostrada en la Figura 5.3. En esta secuencia aérea el objeto a seguir corresponde a un trozo de terreno. El movimiento es constante en una dirección y provoca una deriva prácticamente constante. En otros contextos, cuando el movimiento del objeto es aleatorio o no tiene movimiento definido en ninguna dirección, el fenómeno de la deriva puede verse compensado y pasar inadvertido. Sin embargo, cuando el objeto tiene una marcada dirección de movimiento es



Figura 5.3: Deriva del patrón provocada por una actualización constante.

cuando más apreciable es la deriva debido a un exceso de actualizaciones, unido al error inherente de digitalización de los CCDs.

La Figura 5.3 muestra tres cuadros de la secuencia y los resultados del seguimiento mediante actualización del patrón constante. En el primer cuadro, el recuadro blanco indica el objeto de interés junto con una flecha que muestra la dirección de movimiento constante que posee la secuencia. En el segundo cuadro, la flecha indica la dirección de deriva que está teniendo el patrón y el recuadro blanco señala lo que en ese momento corresponde con el objeto de interés. El tercer cuadro se limita a marcar la situación del objeto de interés instantes más tarde.

La comparación entre la actualización constante y la actualización basada en contexto para la secuencia de la Figura 5.3 muestra como la actualización basada en contexto no encuentra la necesidad de realizar actualizaciones, ya que el objeto de interés no cambia su aspecto visual. Por tanto, mantiene como objeto de interés el mismo trozo de terreno durante toda la secuencia.

**Conclusión:** La actualización constante del patrón puede funcionar correctamente en un gran número de aplicaciones ya que la pérdida del objeto de interés por saltos hacia objetos del contexto es muy difícil. Sin embargo, cuando el movimiento del objeto de interés toma una dirección constante se manifiesta la deriva provocada por la digitalización.

## 5.4. Actualización basada en contexto vs. actualización de periodo constante

La actualización de periodo constante lleva a cabo una renovación del patrón cada cierto intervalo de tiempo constante. Es decir, el patrón es actualizado cada 10 cuadros, cada segundo, etc. Este método no tiene en cuenta ninguna condición de umbral ni de entorno. La actualización constante, por tanto, no es más que un caso particular de este método de actualización. Sin embargo, empíricamente se ha observado que para reducir el efecto de la deriva que presenta la actualización constante se puede realizar una actualización no tan frecuente. No obstante, la deriva puede reducirse pero no desaparecer. Nótese que en el caso de la secuencia aérea que ilustra el experimento incluido en la comparación con la actualización constante, si la actualización se realiza cada cinco cuadros el efecto de la deriva, medido en desplazamiento absoluto, se puede reducir aproximadamente a la quinta parte, pero no se elimina.

Otro de los inconvenientes que muestra este esquema se debe a que la actualización no siempre es oportuna. Es decir, es posible que se realice una actualización irrelevante en un momento dado de la secuencia y no se lleve a cabo una actualización absolutamente necesaria en otro momento. En este último caso este método actuaría de forma similar a como lo haría un esquema de patrón constante, pudiendo por tanto generar un error de salto en el objeto de interés.

**Conclusión:** Este tipo de actualización puede reducir el efecto de deriva que tiene lugar en la actualización constante pero mantiene la posible pérdida del objeto de interés de forma similar a cuando se mantiene el patrón constante.

## 5.5. Actualización basada en contexto vs. actualización umbral constante

La manera de actualización más utilizada en la bibliografía para solventar el problema de la pérdida cuando el patrón no es actualizado y de la deriva en la actualización constante es la actualización mediante umbral constante. Su funcionamiento está basado en la actualización del patrón cuando la diferencia entre el patrón y el punto más parecido en el área de búsqueda sea mayor que un cierto valor umbral.

Su principal inconveniente radica en la elección del valor umbral, puesto que el



mismo valor no ofrece resultados aceptables en todos los contextos. Es decir, determinados valores del umbral se adecuarán mejor a un tipo de escenas que a otras. Un valor de umbral bajo implicará una actualización muy frecuente y por tanto una posible deriva, mientras que un valor muy alto dificultará la actualización generando un posible fallo. Por tanto, la imposibilidad de ajustar un umbral adecuado para todos los casos hace este método inservible para un proceso de seguimiento adaptable automáticamente a distintos entornos de operación. A través de las dos secuencias experimentales siguientes se muestra cómo estos valores del umbral ajustados a una secuencia pueden ser completamente inadecuados para la otra y viceversa.

### 5.5.1. Experimento 1

Esta secuencia experimental muestra un ejemplo de seguimiento de una zona de una cara correspondiente al ojo derecho. La cara se mueve hacia adelante y atrás así como hacia los lados durante toda la secuencia cuya duración es de 41 segundos (1025 cuadros). La utilización del método de actualización basado en contexto ofrece un funcionamiento correcto del seguimiento, como puede observarse en la Figura 5.4.

La tabla 5.3 muestra los resultados obtenidos utilizando diferentes valores del umbral constante. Se puede observar como tan sólo en un estrecho rango alrededor del valor 5000 el seguimiento se lleva a cabo de una manera correcta. Para valores inferiores se produce deriva del patrón y con valores del umbral mayores tienen lugar errores de saltos en el objeto de interés por la no actualización del patrón. Estos errores se ilustran en las Figuras 5.5 y 5.6 respectivamente.

Valor del umbral	Resultado del seguimiento
2000	Deriva del patrón
4000	Deriva del patrón
5000	Seguimiento correcto
6000	Error en la búsqueda
8000	Error en la búsqueda
10000	Error en la búsqueda

Cuadro 5.3: Valores dados al umbral constante y resultado obtenido en el seguimiento.

### 5.5.2. Experimento 2

La secuencia de este experimento está tomada de la película de los Hermanos Marx *Una noche en la ópera*. La duración de esta secuencia es de aproximadamente 7

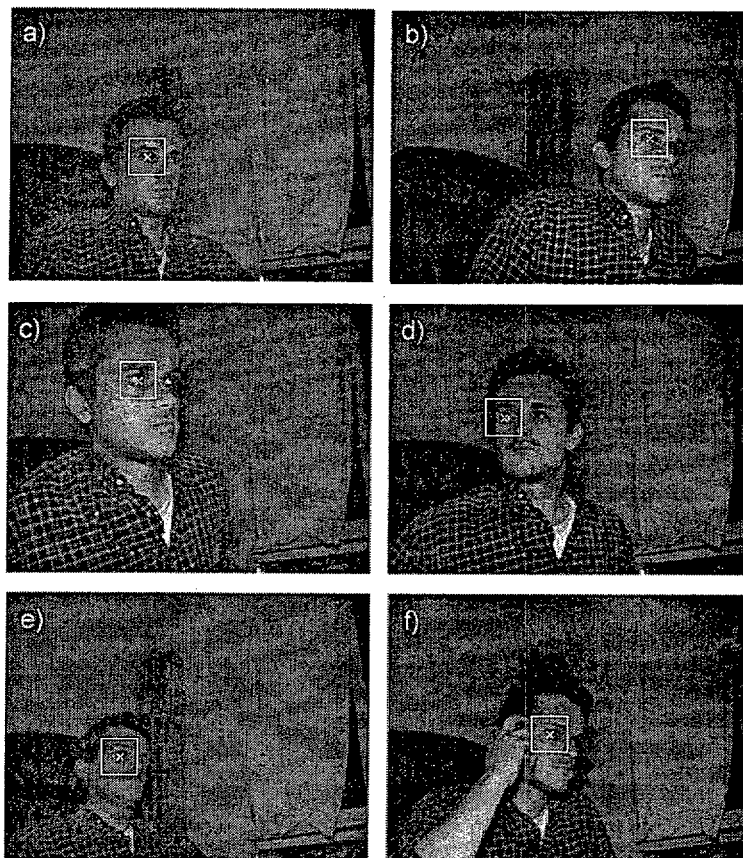


Figura 5.4: La secuencia muestra cómo es seguido el ojo derecho a lo largo de todo el tiempo mediante la actualización de patrones basada en contexto. El ojo derecho es mantenido como objeto de interés desde el primero hasta el último cuadro de la secuencia.

segundos y en ella, el objeto a seguir es la cara de Otis B. Driftwood (Groucho Marx). Como se puede ver en la Figura 5.7, la cara se mueve de arriba a abajo a lo largo de la secuencia al mismo tiempo que ofrece distintas vistas o poses. Los resultados del seguimiento obtenidos mediante el método de actualización basado en contexto se muestran superpuestos también en la Figura 5.7, donde el recuadro blanco representa el objeto seguido en cada momento. Como se puede observar, el resultado del seguimiento es correcto.

La Figura 5.8 representa la sucesión de valores del mínimo, del umbral y del segundo mínimo u objeto principal del contexto a lo largo de la secuencia. Se observa claramente en esta figura cómo los valores del umbral se adaptan automáticamente a las condiciones de la escena en cada momento, haciéndose el umbral más permisivo (alrededor del cuadro 70) o menos tolerante a cambios (sobre el cuadro 130), y todo esto debido siempre a las condiciones de entorno que rodean al objeto de interés.

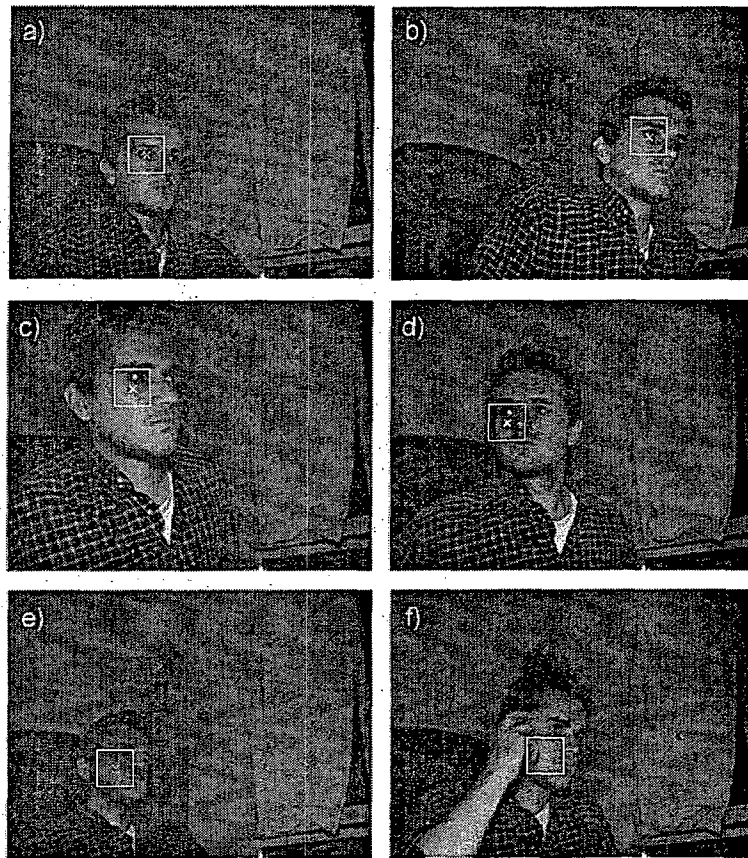


Figura 5.5: Un valor pequeño para el umbral genera un gran número de actualizaciones, lo cual combinado con el tipo de movimiento del objeto de interés provoca una deriva progresiva en el patrón, como se aprecia en la figura. A partir del cuadro c) comienza a apreciarse una ligera deriva que se hace mayor progresivamente. En el último cuadro de la secuencia, cuadro f), el objeto de interés ha derivado hasta la mejilla.

El método de actualización con umbral constante funciona en esta secuencia correctamente dependiendo de los valores del mismo. Valores del umbral comprendidos entre 7000 y 10000 ofrecen un resultado correcto. Cuando el umbral se sitúa por debajo del 7000 se produce una deriva debido a una actualización demasiado frecuente y al final de la secuencia el objeto a seguir termina siendo el sombrero. Con valores mayores de 10000, la actualización no es suficientemente rápida y el objeto se pierde a los pocos cuadros.

La obtención de los umbrales para la actualización basada en umbral constante y para los cuales el seguimiento es correcto, se ha logrado ejecutando el procedimiento de búsqueda repetidas veces con un rango amplio de valores del umbral.

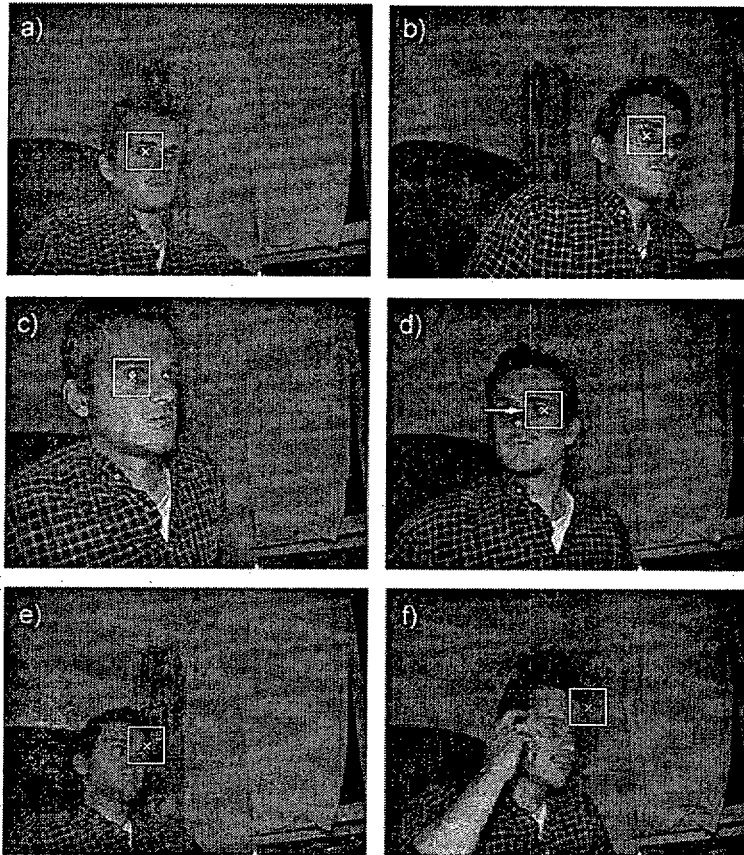


Figura 5.6: En la secuencia se observa cómo en el cuadro d) el patrón no es actualizado a tiempo y el método provoca un salto del objeto de interés hacia el ojo izquierdo, el cual es ahora la zona de la imagen más parecida al patrón actual. El salto en el objeto de interés ha sido debido a una elección del valor del umbral demasiado alto. A partir de este error se suceden una serie de saltos en el objeto interés que terminan en la elección de la zona superior de la frente como patrón actual, como muestra el cuadro f).

**Conclusión:** Como se ha mostrado con los resultados experimentales de estas dos secuencias, los valores del umbral en la actualización basada en umbral constante para los cuales el seguimiento es válido, no siempre son los mismos ni, frecuentemente, tampoco se solapan. Es decir, los valores del umbral apropiados varían de una aplicación a otra y no son, en principio, predecibles. Sin embargo, mediante la utilización de la actualización del patrón basada en contexto se logra un ajuste dinámico del umbral a las situaciones dadas en los experimentos.

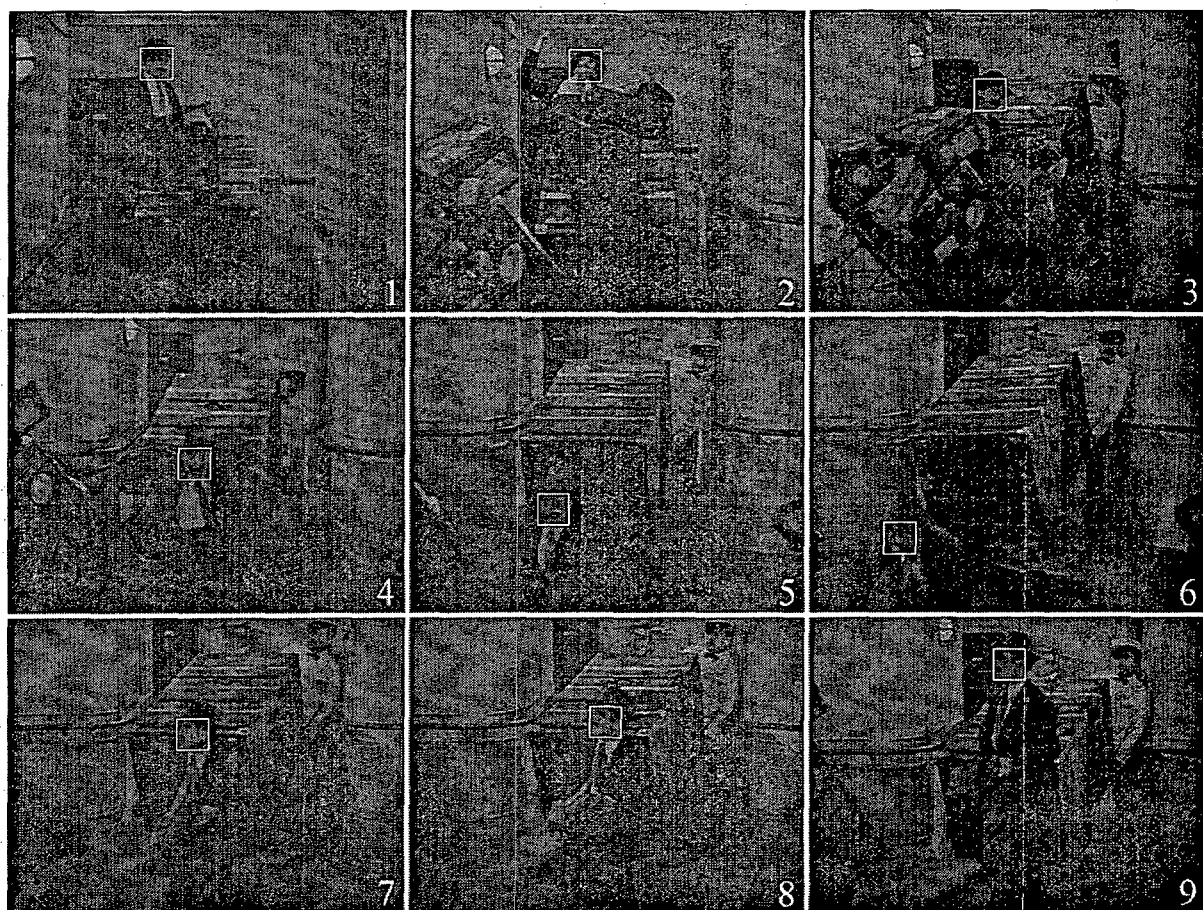


Figura 5.7: Secuencia extraída de la película *Una noche en la ópera* de los Hermanos Marx. Los diferentes cuadros muestran en un recuadro blanco el objeto seguido por el método de actualización del umbral basado en contexto.

## 5.6. Actualización basada en contexto vs. actualización basada en estadísticos

Dentro de los métodos existentes en la bibliografía dedicados a la actualización de patrones, los métodos basados en estadísticos son los que mejores resultados proporcionan debido a que toman en cuenta el entorno de una forma estadística para realizar la actualización. Como se explica en el Capítulo 3, el método basado en estadísticos lleva a cabo la actualización del patrón tomando como indicador la distancia del mínimo a la media del resto de valores de la función de comparación. Esto quiere decir que si la distancia entre el mínimo y la media es suficiente, el patrón que genera este mínimo es considerado como un buen patrón y por tanto será mantenido como patrón a buscar en los siguientes cuadros. Sin embargo, si esta distancia es pequeña implica que el mínimo

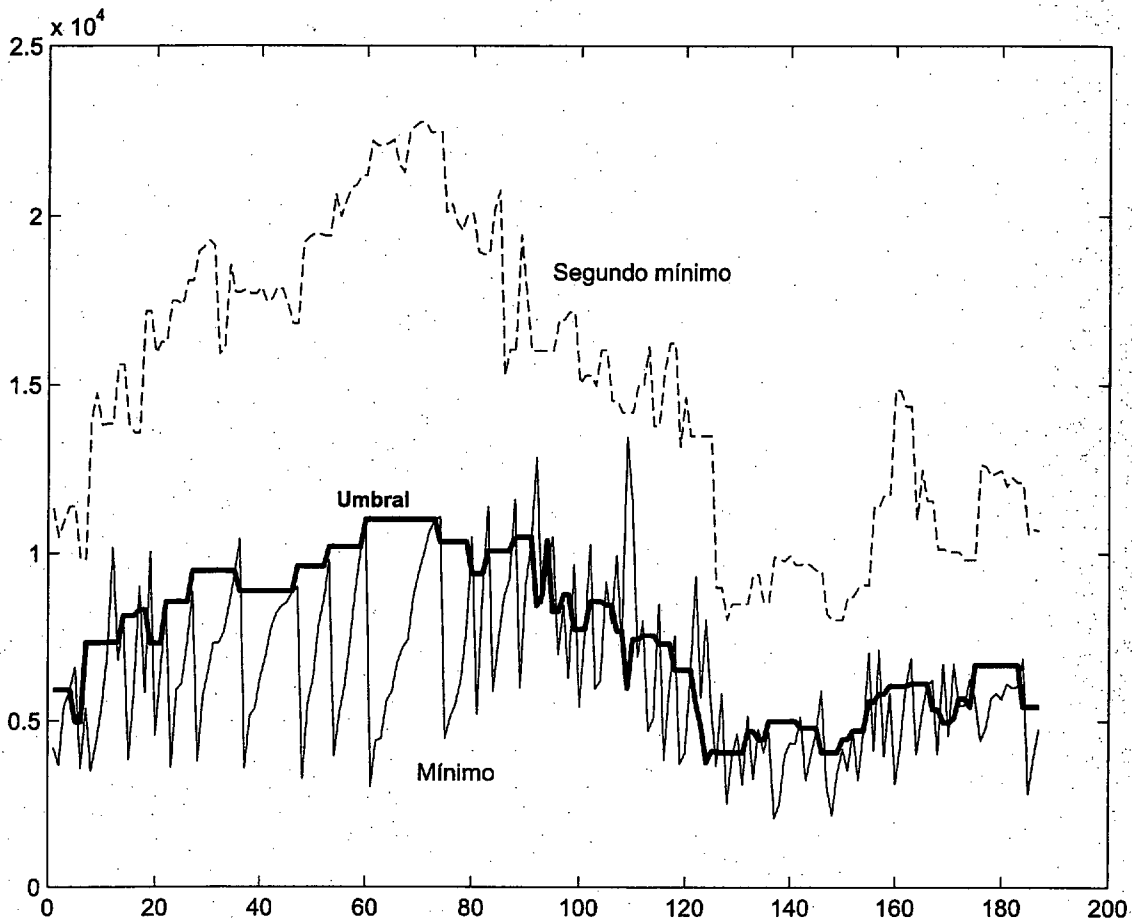


Figura 5.8: Traza del mínimo (línea continua) y el objeto principal del contexto o segundo mínimo (línea discontinua) a lo largo de la secuencia. La línea negra gruesa marca el umbral de actualización en cada momento. Se pueden apreciar las diferentes actualizaciones del patrón a lo largo de la secuencia y como, dependiendo del contexto, el umbral se vuelve más permisivo en el intervalo [40-80] y más restrictivo en torno al cuadro 140.

no se encuentra lo suficientemente destacado del resto de valores y por tanto será necesaria una actualización del patrón. Para obtener un índice que realice esta forma de actualización se emplea la fórmula:

$$Indice = \frac{C_{media} - C_{minimo}}{C_{media}} \quad (5.1)$$

Donde  $C_{media}$  es la media de todos los valores obtenidos mediante la función de disimilitud empleada y  $C_{minimo}$  es el menor valor de todos. El índice obtenido es considerado como el grado de fiabilidad que ha ofrecido la búsqueda del patrón en la imagen. Al ser el valor  $C_{media}$  siempre mayor que  $C_{minimo}$ , el índice se mantendrá siempre en el intervalo  $[0, 1]$  Cuanto más cercano a 0 sea el índice, menor será la fiabilidad obtenida

y viceversa.

No obstante, aunque el método basado en estadísticos actualice el patrón a buscar basándose en los demás valores del entorno, tales valores son tomados en conjunto. Es decir, no tiene en consideración determinados valores puntuales o aislados que pueden afectar a la fiabilidad del seguimiento. La primera secuencia experimental estudiada corresponde a una secuencia sintética que muestra claramente la ocurrencia de estos errores debidos a saltos en el objeto de interés ocasionados por valores aislados.

### 5.6.1. Experimento 1

La secuencia experimental incluye dos figuras cuadrangulares cuyo interior está dividido en cuatro partes también cuadrangulares. El objeto a seguir es la figura de la derecha y consta de cuadrados blancos y negros, mientras que el otro objeto de la escena sintética es una figura cuya diferencia con respecto al objeto de interés consiste en haber sustituido los cuadrados negros por otros de tonos grises. El patrón inicial original contiene el objeto de interés íntegro y sin rotaciones. En la secuencia generada tiene lugar una rotación de 30 grados en sentido antihorario del objeto de interés.

El resultado devuelto por el método de seguimiento basado en estadísticos viene reflejado en la Figura 5.9 por un cuadrilátero superpuesto con una equis en su centro. La figura también muestra diferentes cuadros junto con sus respectivas funciones de distancia. Se observa en el primer cuadro a) como el objeto de interés genera el mínimo de la función de distancia, pero también otras partes de la imagen generan mínimos locales más o menos pronunciados. Entre estos mínimos locales se encuentra el generado por el otro objeto de la imagen cuyo correspondiente mínimo es el siguiente en orden creciente, denominado objeto principal del contexto.

En este momento es importante destacar la forma que posee la función de distancias. Se puede observar claramente como la media de todos los puntos de esta función se halla en una zona marcada como un plano en el primer cuadro de la figura. Además se observa que la distancia de los dos mínimos comentados es considerable con respecto al valor de esta media. Expresado de otra manera, es posible afirmar que el mínimo tiene un alto grado de confianza según la función de fiabilidad expresada en la Fórmula 5.1.

Cuando el objeto de interés comienza a rotar, Figura 5.9 b), su mínimo asociado comienza a subir tal y como se observa en la figura. Aún en este momento su valor sigue siendo menor que el correspondiente al mínimo del otro objeto. Además su fiabilidad sigue siendo alta según la expresión 5.1, luego no se produce actualización del patrón. A

medida que el mínimo asciende debido a la continuada rotación del objeto, cuadro c), se acerca progresivamente al mínimo del otro objeto. La actualización sigue sin producirse ya que la fiabilidad del resultado de la búsqueda sigue siendo alta. En el cuadro d) el mínimo del objeto de interés ha quedado ahora por encima del, hasta ahora, segundo mínimo u objeto principal del contexto, provocando un salto del seguimiento hacia otro objeto distinto del objeto de interés y por tanto produciéndose un error ocasionado por la falta de actualización a tiempo. Este tipo de errores tienen lugar en secuencias donde el objeto a seguir se vea rodeado de otros objetos similares que generan mínimos cercanos al de interés.

En métodos de actualización basados en estadísticos se tiene en cuenta el entorno de manera global, es decir, no se tiene en cuenta las posibles ocurrencias de mínimos que puedan darse y que generen errores como el expuesto. Mediante la actualización basada en contexto se realiza un control de los segundos mínimos en la escena con el fin de llevar a cabo la actualización en el momento oportuno y evitar el tipo de error descrito.



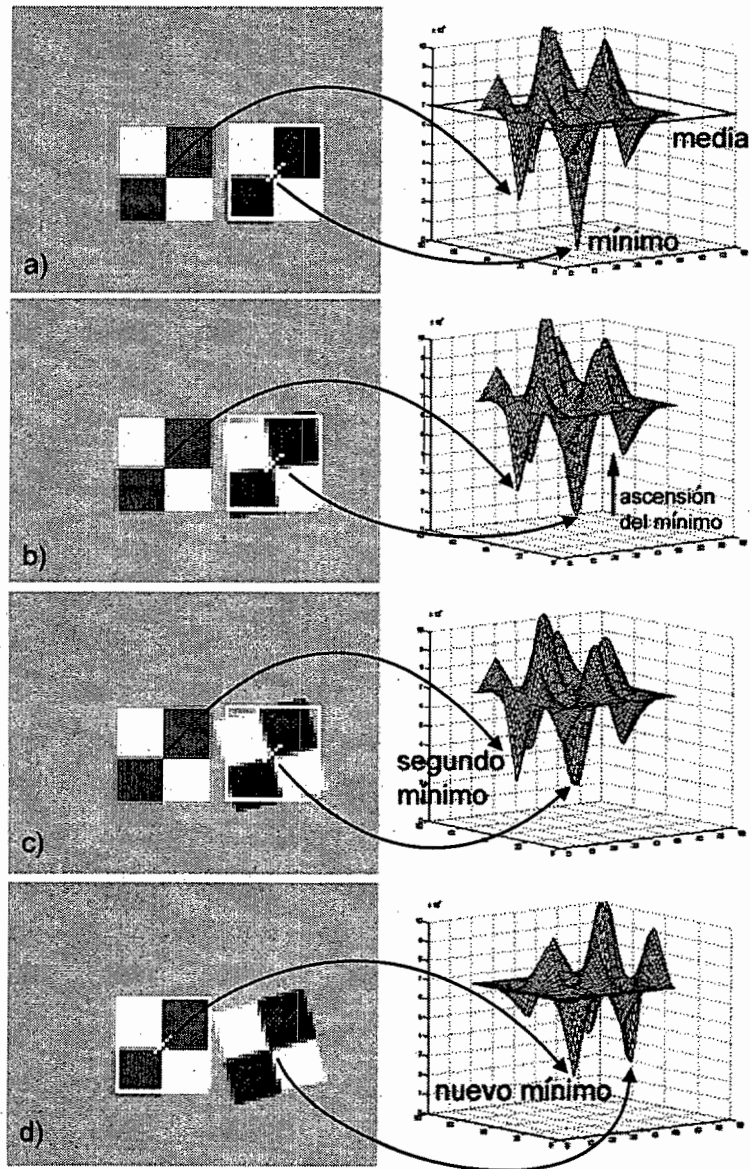


Figura 5.9: La figura muestra el salto provocado por el mínimo local generado por el objeto principal del contexto.

### 5.6.2. Experimento 2

La secuencia escogida para este experimento tiene una longitud de 100 cuadros (4 segundos) y muestra un banco de peces, de los cuales uno de ellos es el objeto de interés. En esta secuencia es claramente observable el proceso de pérdida explicado en el experimento anterior debido a la existencia de objetos similares al de interés.

Como se observa en la Figura 5.10 el pez de interés se indica mediante un cuadrado con una equis en su interior superpuesto a la imagen. A la derecha de cada cuadro se encuentra la función de distancias correspondiente. Se observa cómo en el cuadro a) se halla el objeto de interés marcado gracias a su mínimo. En este primer cuadro se observa también que a la izquierda del pez seguido se encuentra otro pez que ofrece una vista similar, y que por tanto genera también un mínimo local en la función de distancias. Sin embargo, estos mínimos locales no afectan suficientemente a la media generada por todos los demás valores y por tanto, para el método de actualización basado en estadísticos, estos objetos del contexto o generadores potenciales de errores pasan inadvertidos.

Unos milisegundos más tarde, cuadro b), el segundo mínimo está casi a la misma altura que el principal. Esto se explica porque el patrón no ha sido actualizado y la vista actual del objeto de interés comienza a ser ligeramente diferente a la que se halla en el patrón. Ya en el cuadro c) el objeto de interés muestra una vista menos parecida al patrón que la vista que muestra el pez más cercano, que constituye el objeto principal del contexto, y por tanto el seguimiento se centra ahora sobre éste. De nuevo y como en el experimento anterior, se ha producido un error de salto del objeto de interés debido a que no se ha actualizado el patrón a tiempo.

Dentro del esquema de actualización del patrón basado en estadísticos, este tipo de errores pueden ser corregidos aumentando el umbral de fiabilidad y haciendo, por tanto, que la frecuencia de las actualizaciones sea mayor, con la consecuente deriva potencial. La tabla 5.4 muestra diferentes valores del umbral de fiabilidad y sus resultados correspondientes para la actualización basada en estadísticos. En la Tabla 5.5 se presentan los datos del seguimiento mediante el método de actualización basado en contexto.

### 5.6.3. Experimento 3

No solamente se dan esta clase de errores de salto en secuencias con objetos del mismo tipo cercanos al de interés, como en el caso del experimento anterior. La similitud entre objetos puede venir dada por zonas de claros y oscuros en principio no demasiado

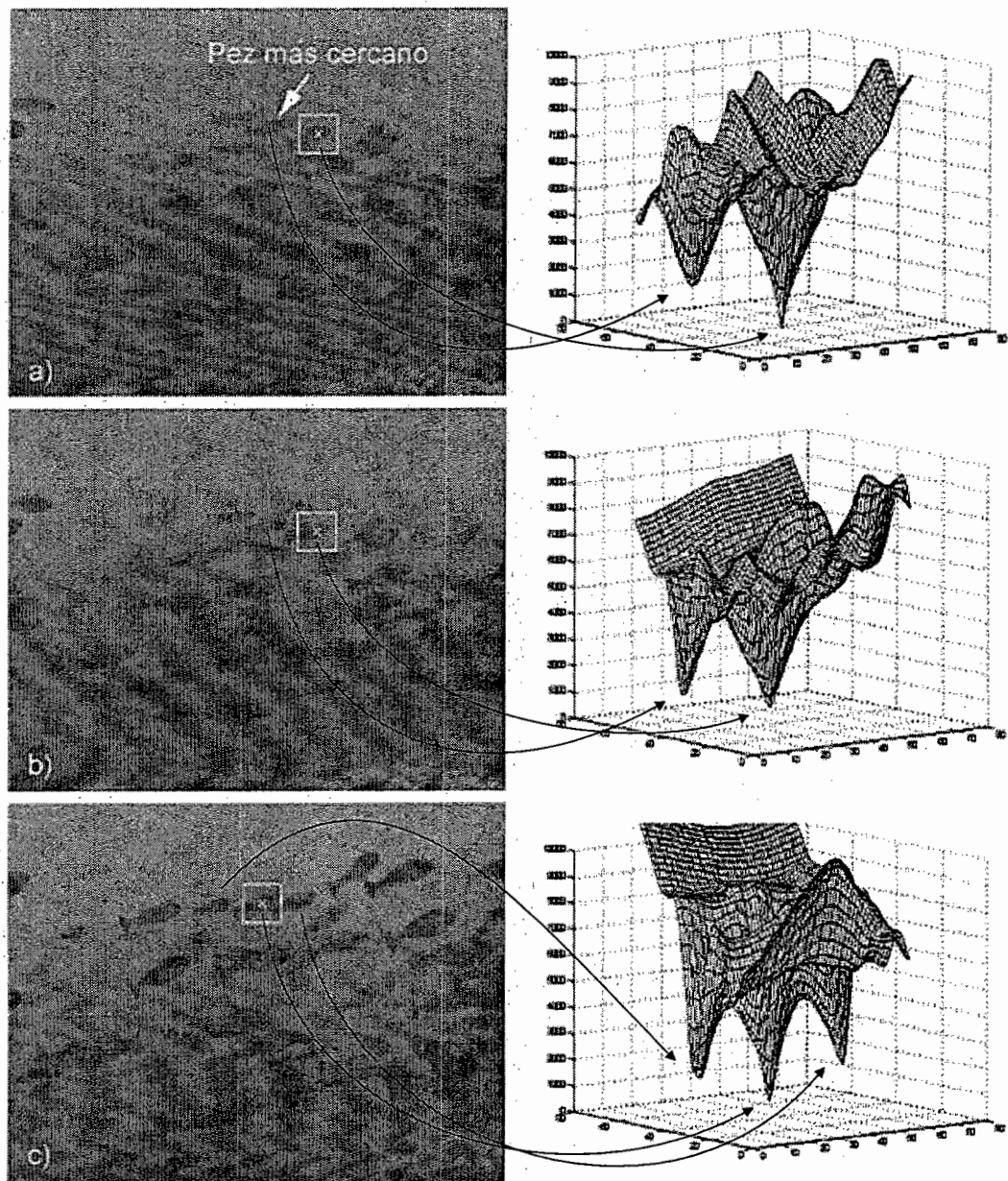


Figura 5.10: Tres cuadros que muestran cómo el método de actualización basado en estadísticos puede verse confundido por la presencia de objetos cercanos similares al objeto de interés. En el cuadro a) se observa que el objeto de interés, el pez marcado por un cuadrado blanco, genera el mínimo absoluto de la función. Sin embargo, el pez situado a la izquierda es muy parecido y por tanto genera también un mínimo muy bajo. Además es observable que estos dos mínimos son muy pronunciados, dejando el resto de los valores de la función muy altos. Esto hace que la media se sitúe muy separada de los dos mínimos menores, haciendo que el método de actualización no tenga la necesidad de actualizar el patrón. En el cuadro b) el pez de interés cambia su vista y esto provoca que su mínimo ascienda. En este momento los dos mínimos menores se hallan muy próximos. En el cuadro c) al se observa cómo al no haber tenido lugar ninguna actualización del patrón se ha producido una pérdida por un salto en el objeto de interés. Los tres mínimos que se aprecian corresponden al pez seguido y los peces situados a la derecha e izquierda de éste.

Umbral de fiabilidad	Número de actualizaciones	Errores por saltos
0,75	9	Sí
0,80	18	Sí
0,85	28	No

Cuadro 5.4: La tabla muestra el número de actualizaciones basadas en estadísticos y errores para diferentes umbrales de fiabilidad.

Número de actualizaciones	Errores por saltos
26	No

Cuadro 5.5: Número de actualizaciones para el seguimiento de la misma secuencia mediante el método de basado en contexto.

parecidas pero que pueden llegar a confundir al método de actualización debido a que la media producida por el resto de los valores de la imagen puede poseer un valor muy alejado del mínimo.

La secuencia que se muestra en la Figura 5.11 muestra este tipo de error para un contexto de seguimiento muy común: el seguimiento de la cara de una persona mientras camina. El error en el seguimiento viene provocado al final de la secuencia debido a que gran parte de la zona de la imagen que rodea a la cara es muy distinta al patrón y, por tanto, genera grandes valores en la función de distancia que elevan considerablemente la media haciendo que el patrón no se actualice. De esta forma, zonas parecidas a la cara seguida llegan a producir el mínimo absoluto de la imagen provocando la pérdida. La Figura 5.11 muestra gráficamente el origen de la pérdida, donde se aprecia en las funciones de distancia asociadas a cada cuadro la relevancia del mínimo absoluto con respecto al resto de los valores.

Para conseguir un correcto funcionamiento del método de actualización basado en estadísticos es necesario llegar a una frecuencia de actualización suficiente, lo cual se consigue elevando el nivel de fiabilidad hasta, como indica la tabla 5.6, un valor de 0.9. No obstante, esto implica realizar 468 actualizaciones frente a las 106 actualizaciones que se obtienen mediante la actualización basada en contexto como se indica en la tabla 5.7.

**Conclusión:** El método de actualización basado en estadísticos posee la facultad de tomar en cuenta el entorno para llevar a cabo la actualización del patrón. Sin embargo, esta consideración la realiza de manera global en la imagen. Es decir, sin tener en cuenta la existencia de mínimos locales que puedan confundir el seguimiento.

Umbral de fiabilidad	Número de actualizaciones	Errores por saltos
0,75	-	Sí
0,80	-	Sí
0,85	-	Sí
0,90	468	No

Cuadro 5.6: La tabla muestra el número de actualizaciones basadas en estadísticos y errores para diferentes umbrales de fiabilidad.

Número de actualizaciones	Errores por saltos
106	No

Cuadro 5.7: Número de actualizaciones para el seguimiento de la misma secuencia mediante el método de basado en contexto.

De los resultados experimentales realizados para comparar estos dos métodos de actualización se obtiene que el método de actualización basado en contexto se comporta mejor debido a que proporciona un seguimiento correcto con un menor número de actualizaciones, retrasando la aparición del error de deriva.

## 5.7. Conclusiones

Como conclusiones a los experimentos de comparación realizados se obtiene que el método de actualización basado en contexto propuesto en esta tesis mejora a los demás métodos de actualización referidos en la bibliografía en los siguientes aspectos:

- Minimiza el número de actualizaciones requeridas para el correcto seguimiento de los objetos, minimizando por tanto el riesgo de deriva en el seguimiento.
- Controla posibles objetos del contexto mediante los segundos mínimos locales que puedan llevar a error en el objeto seguido, provocando un salto de la atención entre el objeto de interés y otro objeto del contexto cercano similar según alguna función de similitud.
- Mejora la adaptación automática de la actualización del patrón a cualquier tipo de contexto con respecto a otros métodos de actualización automática que hacen uso de medidas estadísticas para llevar a cabo dicha actualización.

Por tanto, estos experimentos consiguen demostrar la mejor adecuación del método de actualización del patrón basado en contexto para ser utilizado como esquema de

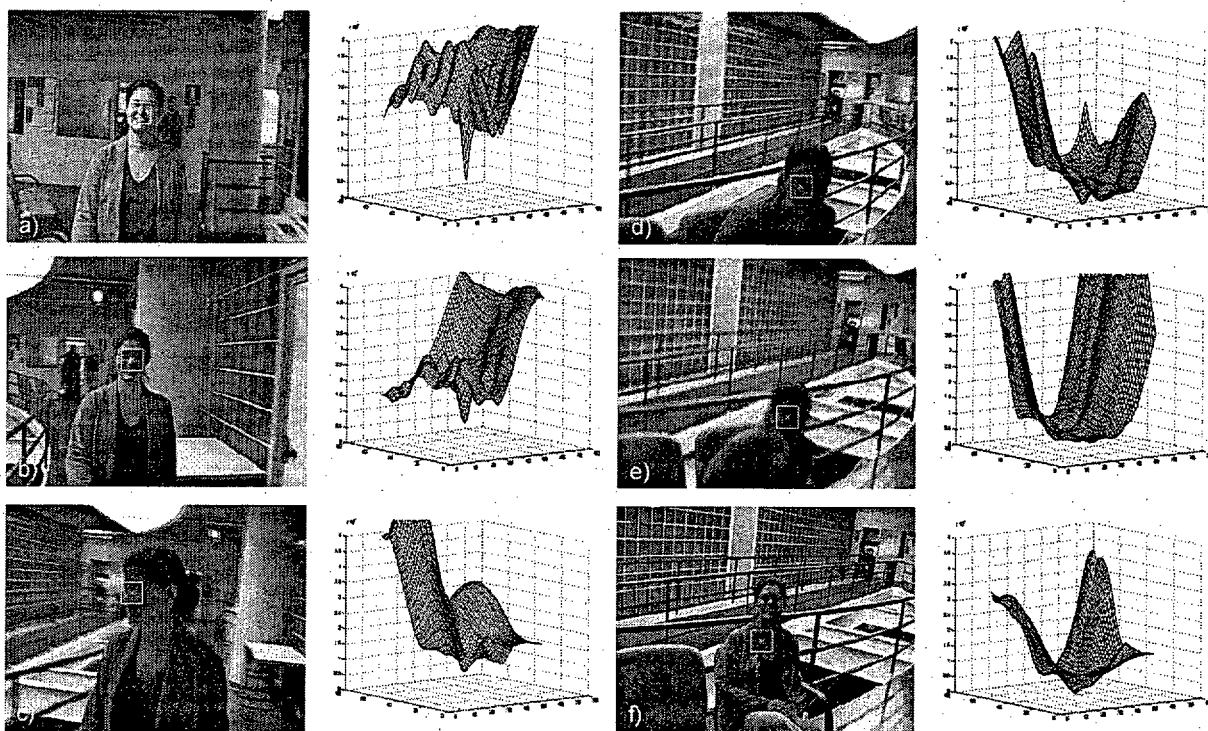


Figura 5.11: La figura muestra seis cuadros de la secuencia experimental donde el objeto seguido es una cara. En los dos primeros cuadros a) y b) se observa como el seguimiento no presenta ninguna dificultad debido a que la superficie de distancias muestra un mínimo destacado. En los dos siguientes cuadros c) y d) el mínimo ya no aparece tan destacado puesto que la cara tiene ahora mayor tamaño y genera mínimos locales próximos al mínimo absoluto. Este hecho se ve acentuado en el cuadro e) donde la imagen que rodea a la cara es muy distinta a la cara y por tanto eleva la media. Debido a la falta de actualización la cara termina por ser perdida a causa de otro mínimo local perteneciente a una zona ubicada entre el cuello y la chaqueta de la persona.

actualización de patrones en cualquier procedimiento de búsqueda mediante correspondencia de patrones. Todo ello con el fin de obtener módulos de seguimiento adaptables automáticamente y sin ningún tipo de ajuste manual previo a cualquier tipo de escenarios.

## 5.8. Utilización de bancos de patrones

El siguiente conjunto de experimentos se centran en el uso del esquema de actualización del patrón basado en contexto y muestran las mejoras en robustez obtenidas en los procesos de seguimiento al incluir un banco de diferentes vistas de un objeto a modo de memoria visual a corto plazo, véase Capítulo 3.

Como se ha expuesto, un banco de patrones permite recuperar ciertas vistas de un objeto ofrecidas por el mismo con anterioridad, con lo que retomar una vista previa aprendida como patrón actual refuerza la confianza que se concede al seguimiento y centra de nuevo el objeto de interés evitando la posible deriva que podría haberse acumulado. La reaparición de vistas previas de un objeto se da en muchos de los contextos de aplicación de tareas de seguimiento.

Llevar a cabo el seguimiento utilizando un banco de patrones implica, a priori, incrementar el número de correlaciones a realizar con respecto a la utilización de un solo patrón por parte del procedimiento de comparación. Son necesarias tantas correlaciones como patrones haya en el banco. Además, han de cumplirse las restricciones de tiempo real exigidas por el módulo de seguimiento. Por tanto, es necesario obtener medidas que caractericen la potencia de cálculo ofrecida por el sistema que compute las correlaciones.

Para los experimentos recogidos en este apartado se ha contado con un procesador Pentium III 500 MHz. con 384 MBytes de RAM ejecutando el sistema operativo de Microsoft Windows 2000 Professional. Las medidas de rendimiento tan sólo se refieren al algoritmo de la función de similitud dado por la suma de las diferencias en valor absoluto. No se incluye todo el proceso de captura, digitalización, comando de motores, etc. de un sistema de seguimiento completo. La Tabla 5.8 refleja los valores de tiempo de computación dados por la búsqueda de un patrón de  $24 \times 24$  pixels en diferentes tamaños de áreas de búsqueda. Por motivos de comparación se han incluido, dos columnas más con algoritmos con la misma funcionalidad escritos en ensamblador puro y C. El gran incremento de rendimiento del código realizado con instrucciones MMX con respecto al código C normal se debe a las características y naturaleza propias del algoritmo en cuestión. No cualquier algoritmo alcanza una mejora de rendimiento similar. En este caso, el algoritmo incluye una operación, que es fundamental en su ejecución, ya implementada en el propio microprocesador como instrucción MMX. Además la naturaleza SIMD del algoritmo se adapta perfectamente a la arquitectura MMX. La descripción detallada de este algoritmo en ensamblador con MMX y de su versión en C se encuentra en el Apéndice A.

Según estos datos es posible establecer un compromiso entre el número de patro-

Tamaño del área de búsqueda (pixels)	MMX (milisegundos)	Ensamblador (milisegundos)	C (milisegundos)
80 × 80	3.05	44.8	91.5
72 × 72	2.09	32.6	67.6
64 × 64	1.51	22.9	47.3
56 × 56	0.94	15.3	30.6
40 × 40	0.25	3.68	7.65
32 × 32	0.08	0.93	1.97

Cuadro 5.8: Tiempo de computación tomado por el núcleo de la función de similitud implementada en MMX, ensamblador y C respecto a un tamaño de área de búsqueda utilizando el microprocesador Pentium III 500 MHz. En el Apéndice C se encuentran los algoritmos de comparación desarrollados en ensamblador utilizando instrucciones MMX y en C.

nes del banco y el área de búsqueda usada. Con la utilización de este microprocesador es posible almacenar potencialmente hasta 13 patrones en el banco asignando un área de búsqueda de 80 × 80 pixels, manteniendo un tiempo de cómputo inferior a los 40 milisegundos.

La estrategia de inclusión y exclusión de patrones en el banco se encuentra recogida en la Sección 3.8.3 del Capítulo 3. Sí es necesario destacar que, al igual que el método de actualización basado en contexto, esta estrategia lleva a cabo su cometido de forma completamente autónoma, eliminando cualquier tipo de ajuste de parámetros manual previo, y de esta forma adaptándose automáticamente al entorno de trabajo.

A continuación se describen los experimentos incluidos en esta sección:

- En primer lugar, un experimento diseñado partiendo de imágenes sintéticas ilustra de manera muy clara los beneficios que aporta el banco de patrones. Se observa cómo la deriva debida a una rotación del objeto de interés se compensa posteriormente mediante la reutilización de vistas previas.
- Un segundo experimento realiza el seguimiento de un ojo de una cara durante cierto tiempo. En esta secuencia la persona gesticula y mueve la cabeza hacia los lados. Se observa la deriva ocurrida después de cierto tiempo cuando no se utiliza un banco de patrones y cómo la inclusión de éste la evita.
- En un tercer experimento se trata otro de los posibles marcos de aplicación de un banco de patrones. En este caso, escenas donde se da una desaparición momentánea del objeto buscado y es posteriormente recuperado gracias al banco de



patrones. En el experimento el objeto seguido es la línea derecha de una carretera y se dan recuperaciones del patrón del banco puesto que la línea se deforma cíclicamente hacia la derecha e izquierda. Además, este experimento muestra un interés adicional que podemos catalogar como primordial, relacionado con el efecto de la proyección de sombra causada por el paso del vehículo donde se encuentra la cámara bajo un puente, lo que altera significativamente el patrón actual.

- En el cuarto experimento se describe la corrección de un hecho que se da en cierto tipo de patrones cuando se utiliza un esquema de actualización del patrón basado en contexto que se comenta en la correspondiente sección.

### 5.8.1. Experimento 1

En este experimento se ha tomado una secuencia sencilla en la que un objeto rota 90 grados en sentido horario y de nuevo rota, ahora en sentido antihorario, 90 grados hasta situarse en la posición inicial. El patrón tomado se centra en la esquina superior derecha de este objeto. La Figura 5.12 representa el resultado del seguimiento cuando se utiliza únicamente un patrón. La imagen de la izquierda muestra la situación inicial. La imagen de la derecha representa el resultado o cuadro final de la secuencia, donde el patrón ha derivado hacia arriba y a la derecha.

Es importante destacar que la deriva provocada aquí no se debe al mismo proceso de digitalización descrito en el Capítulo 3 sino que es un efecto de la transformación del objeto de interés, debido a que una vez comienza la rotación se producen en el patrón una serie de actualizaciones que, evidentemente, no se centran en la esquina del objeto al existir una componente de traslación en el movimiento de la misma. Al volver a tomar el objeto la vista original, el proceso de seguimiento desconoce cuál era la posición o patrón inicial. Con esta configuración, si se repite de nuevo la secuencia el objeto se perdería completamente en la zona gris de la imagen.

Cuando se emplea un banco de patrones como memoria visual a corto plazo se obtienen, en muchas ocasiones, mejoras substanciales de la robustez del proceso de seguimiento. En este caso se ha utilizado un banco de ocho patrones o vistas del objeto, donde se almacenarán las sucesivas vistas que se vayan obteniendo a medida que el objeto va rotando. La Figura 5.13 muestra las situaciones inicial y final (las cuales coinciden) del proceso de seguimiento. También se muestra el contenido del banco de patrones durante la ejecución del seguimiento. A medida que se producen actualizaciones del patrón, estos nuevos patrones se almacenan en el banco. Cuando se termina la rotación en sentido horario y comienza la rotación en sentido antihorario, no tienen lugar nuevas

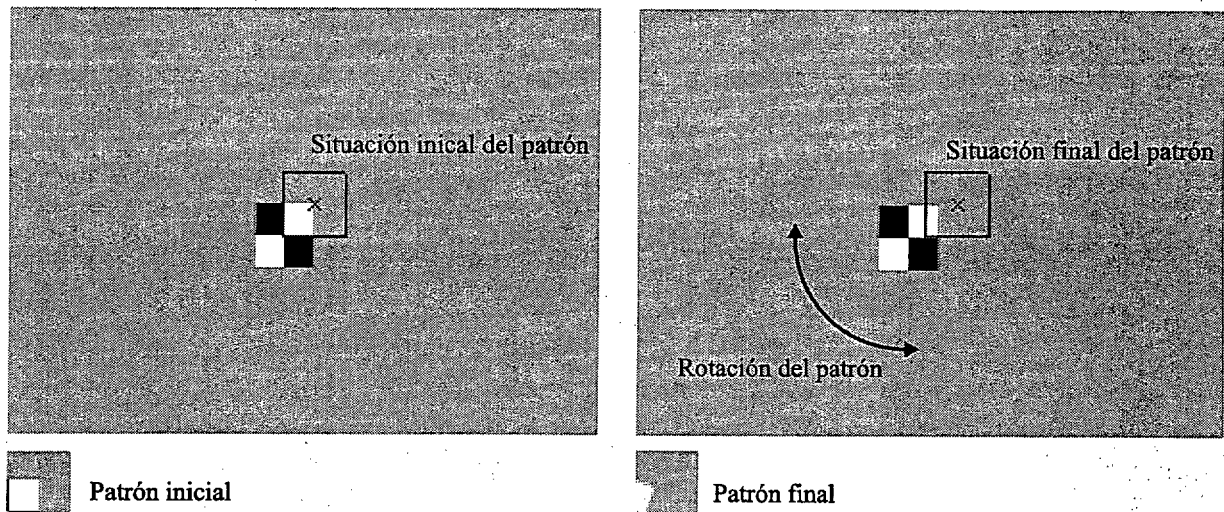


Figura 5.12: Seguimiento del objeto en rotación sin utilizar un banco de patrones. La imagen de la izquierda corresponde con el cuadro inicial y el patrón está centrado en la esquina superior derecha del objeto. La imagen de la derecha corresponde al cuadro final, donde se aprecia la deriva provocada por la rotación.

actualizaciones ya que la porción de curva de transformación visual, correspondiente a vistas anteriores del objeto, ya ha sido completamente aprendida y el objeto se sigue gracias a la recuperación de vistas previas almacenadas en el banco, con lo que se evita la deriva y posterior pérdida del objeto de interés.

### 5.8.2. Experimento 2

La secuencia de este experimento muestra un ejemplo típico donde se dan repeticiones del patrón en diferentes momentos. En este caso una persona estática en un escenario se encuentra hablando con iluminación constante y en varias ocasiones ladea ligeramente la cabeza para dirigirse a todo su auditorio. El objeto a seguir es su ojo derecho, que como muestra la Figura 5.14, muestra diferentes vistas. Puede observarse como existen vistas del ojo mirando hacia su izquierda, el ojo cerrado y el ojo en una vista frontal. Esta última vista es la más común y es recuperada en diferentes ocasiones del banco de patrones. La duración de esta secuencia es de 57 segundos y durante este tiempo se llevan a cabo numerosas actualizaciones y recuperaciones del banco de patrones. La Tabla 5.9 muestra el comportamiento del método con la utilización de distinta cantidad de patrones en el banco.

Hay que destacar que sin la utilización del banco de patrones, se produce una ligera deriva debido a los movimientos de la cabeza. Esta se debe a que el patrón se

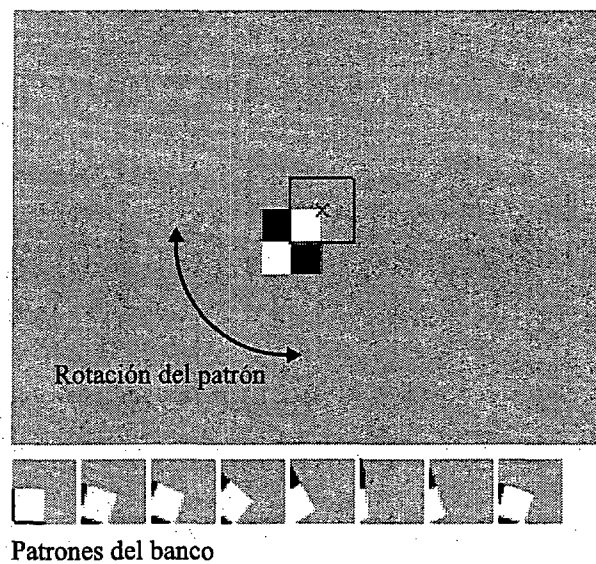


Figura 5.13: Seguimiento del mismo objeto en rotación pero utilizando un banco de patrones. La imagen muestra tanto el cuadro inicial como el final ya que no se ha producido deriva. En la parte inferior se muestran los ocho patrones almacenados en el banco durante el proceso de seguimiento.

actualiza para acomodarse a una nueva vista del objeto de interés, cuando el objeto ofrece de nuevo una vista anterior es imposible para el seguimiento ajustarse de nuevo al objeto si no dispone de ninguna referencia anterior. Con la inclusión de un banco de vistas anteriores es posible entonces recuperar las vistas previas y evitar este tipo de derivas. De esta forma se consigue mantener durante más tiempo un seguimiento correcto. La Tabla 5.9 recoge los valores obtenidos al realizar el seguimiento con diferente número de patrones en el banco. Sin la utilización de un banco de patrones se produce deriva dado que el objeto de interés se transforma y no es posible recuperar vistas anteriores. Mediante la utilización de un banco de patrones se observa como a medida que se incrementa el número de patrones en el banco, se reducen las actualizaciones y se incrementan las recuperaciones de vistas anteriores.

Patrones	Actualizaciones	Recuperaciones	Deriva
1	80	-	Sí
4	57	80	No
8	53	101	No

Cuadro 5.9: Número de actualizaciones y recuperaciones para el seguimiento de la misma secuencia mediante el método de basado en contexto sin y con el uso de un banco de patrones.

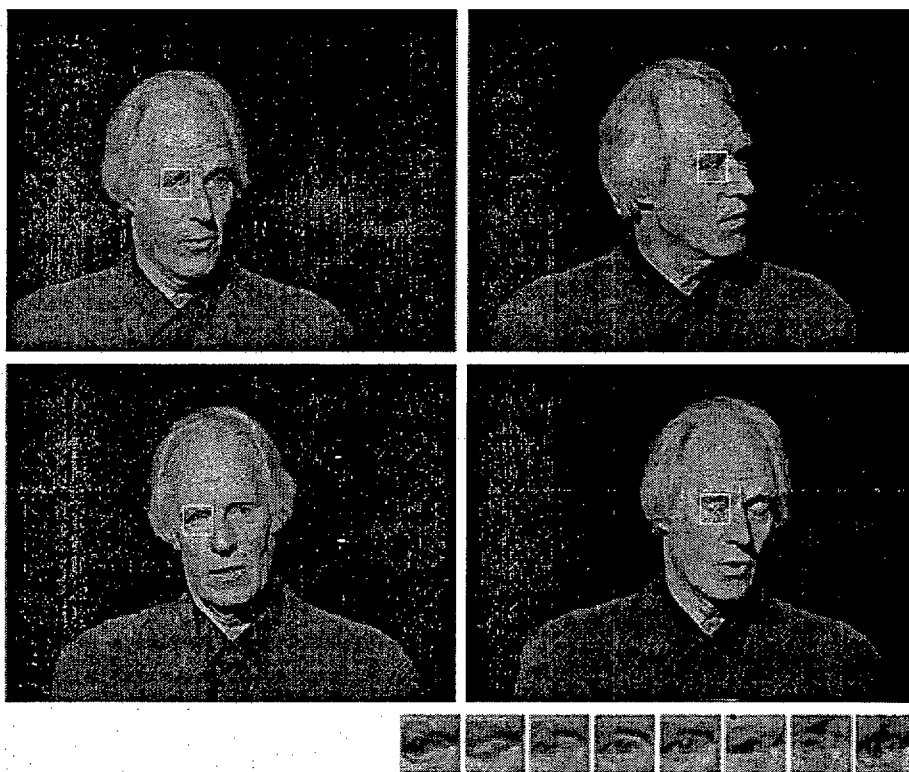


Figura 5.14: Distintos cuadros extraídos del seguimiento del ojo derecho utilizando un banco de ocho patrones. Durante la secuencia son recuperados los patrones incluidos en el banco con el fin de evitar derivas del objeto de interés debido a los sucesivos cambios en sus vistas. Se muestra también el contenido del banco de patrones en un momento dado del seguimiento. Se aprecia como el banco almacena distintas vistas significativas del ojo.



Figura 5.15: Cuadro final y posición resultante del seguimiento del ojo derecho cuando no se utiliza un banco de patrones (imagen izquierda) y cuando se utiliza un banco de ocho patrones (imagen derecha). Se aprecia claramente la deriva del patrón en la imagen izquierda.

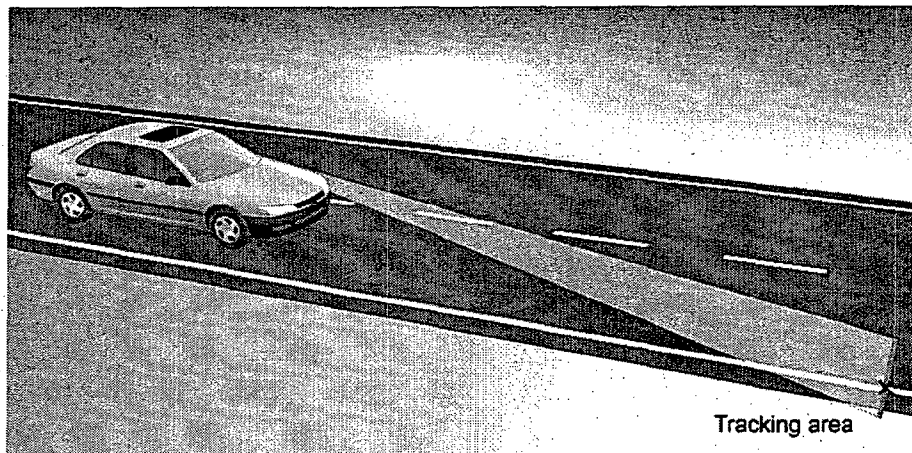


Figura 5.16: La figura muestra una simulación de seguimiento de un punto sobre la línea del margen de la carretera. Como se observa, la zona de búsqueda se limita a la línea representada aquí en color verde.

### 5.8.3. Experimento 3

Uno de los elementos visuales más útiles (en aplicaciones de conducción automática) para seguir cuando se conduce son las líneas blancas de los márgenes de la carretera. Estas líneas pueden ser vistas como objetos que se deforman durante la conducción. Son objetos relativamente fáciles de localizar en las imágenes mediante diferentes algoritmos clásicos de detección de bordes.

Este experimento consta del seguimiento de estas líneas blancas trazadas en los bordes de las carreteras. Realmente, no se sigue la línea completa sino sólo un segmento de ésta. Además, utilizando las funciones del módulo de seguimiento que permiten establecer el área de búsqueda del objeto en la imagen, esta se va a reducir a una búsqueda unidimensional. Ya que no existe posibilidad para movimientos verticales del patrón, tan sólo horizontales, la zona de búsqueda se extiende de derecha a izquierda pero no de arriba hacia abajo, con lo que la zona de búsqueda será sobre un área de anchura  $n$  y altura 1 pixel. Esto redundará en la limitación de costos computacionales, ya que buscar sobre una línea en lugar de buscar sobre un área consume mucho menos tiempo. Las Figuras 5.16 y 5.18 muestran cómo se realiza el seguimiento. La línea verde representa el área donde tiene lugar la búsqueda. Debido a que se conocen los parámetros de calibración de la cámara tanto intrínsecos como extrínsecos (distancias de la cámara verticalmente a la carretera, orientación y focal) es fácil obtener la posición tridimensional del punto de la línea seguida. Este punto se marca con una equis roja en las figuras.

En esta aplicación la línea de búsqueda tiene una longitud de 80 pixels y el patrón

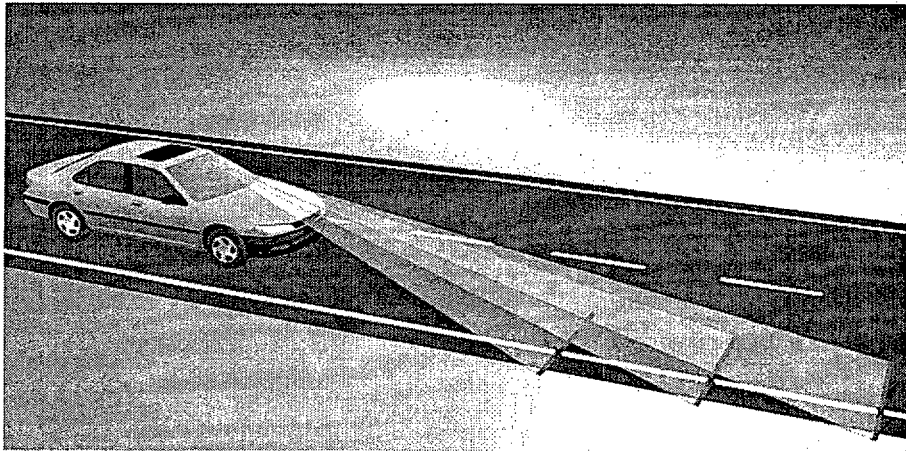


Figura 5.17: En esta ocasión el número de puntos a seguir es tres permitiendo un mayor grado de robustez, al mismo tiempo que ofrece una apreciación de la curvatura de la carretera.

cuadrado tiene 24 pixels de lado. Con estos parámetros, por ejemplo un Pentium III a 500 MHz, consume menos de un milisegundo en realizar la búsqueda. Estos resultados permiten situar varios puntos de seguimiento sobre la línea de la carretera manteniendo las restricciones de tiempo real. Esta estrategia ofrece dos ventajas principales: primera, permite obtener con costos asumibles en tiempo real una descripción de la curvatura de la carretera. Y en segundo lugar, situar múltiples puntos a lo largo de la línea mejora la robustez del seguimiento al introducir mayor grado de redundancia, como muestra la Figura 5.17.

En el experimento se han comparado los resultados realizando el seguimiento sin utilizar y utilizando un banco de patrones. Como se muestra en la Figura 5.18, la dificultad que presenta esta secuencia radica en dos puntos concretos: el paso bajo el puente y la zona de rayas transversales blancas destinadas a obligar a frenar el vehículo. En el primer punto, un cambio repentino de luminosidad cambia el aspecto del patrón de forma brusca. En el segundo punto, la línea blanca desaparece como patrón para pasar a ser un patrón enteramente blanco. Estos dos sucesos dentro de la secuencia tienen la particularidad común de que suceden en un breve intervalo de tiempo. En ambos casos la no utilización del banco de patrones conllevó a la pérdida del objeto. Sin embargo, la utilización de un banco de patrones permitió al módulo de seguimiento hacer uso de patrones anteriores para recuperar de nuevo el objeto. Concretamente, en el cuadro *g)* de la Figura 5.18 se pueden observar en la parte inferior de la imagen los patrones o vistas de la línea almacenados en el banco y que sirven para recuperar el objeto correctamente tras el paso bajo el puente. Así mismo, en el cuadro *h)* también se muestra el contenido

de los patrones durante el paso de las líneas blancas anchas transversales.

#### 5.8.4. Experimento 4

En ciertos contextos de aplicación se dan patrones que pueden dificultar el seguimiento de objetos mediante actualizaciones del patrón basadas en contexto. La dificultad se halla en que esta clase de patrones poseen una repetición de elementos o texturas en su vista. La Figura 5.19 ilustra gráficamente este hecho. Se puede observar como la repetición de bandas blancas y grises del patrón hace que se creen mínimos locales muy próximos, tanto en su valor como en su posición espacial dentro de la imagen, al mínimo absoluto. La cercanía del valor de estos mínimos locales hacen que el umbral dinámico de cambio que se genera en cada actualización sea sobrepasado muy frecuentemente provocando entonces repetidas actualizaciones del patrón que pueden ocasionar la deriva del mismo.

El siguiente experimento consta de una secuencia de 26 segundos de duración en la que un camión circula por la carretera y se acerca lenta y progresivamente a la cámara. El objeto a seguir en este caso es una zona de la matrícula. El patrón inicial escogido puede verse en la Figura 5.21. La dificultad de este patrón radica en que su contenido no abarca toda la matrícula sino parte de ella, de tal manera que la función de similitud genera múltiples mínimos debido al fenómeno mencionado en el párrafo anterior. A lo largo del resto de la matrícula se encuentran zonas similares al patrón que producen mínimos locales con valores muy cercanos al mínimo absoluto, en este caso, los demás caracteres de la matrícula.

Debido a esto, la función de similitud genera umbrales muy bajos. Esto es apreciable claramente en la Figura 5.20. Se puede observar cómo los restantes caracteres de la matrícula generan unos mínimos alineados en la dirección de la matrícula. A estos factores se añade el ruido inherente de digitalización de la cámara, que como ya se ha calculado se sitúa en torno a 3000 para el tamaño de patrón empleado. La consecuencia es que la actualización del patrón, y por lo tanto del umbral, se realiza muy frecuentemente. Cuanto mayor es la frecuencia de actualización del patrón mayor es la posibilidad de deriva de éste.

En la Figura 5.21 se muestran los cuadros inicial y final de la secuencia, así como la ubicación inicial y final del patrón buscado. Debido a la deriva, el objeto que se sigue al término de la secuencia viene a ser el parachoques del camión, cuya localización no es exacta sino que se establece aleatoriamente en cada cuadro a lo largo de todo el parachoques. Esto es así debido a la falta de una referencia horizontal del patrón que

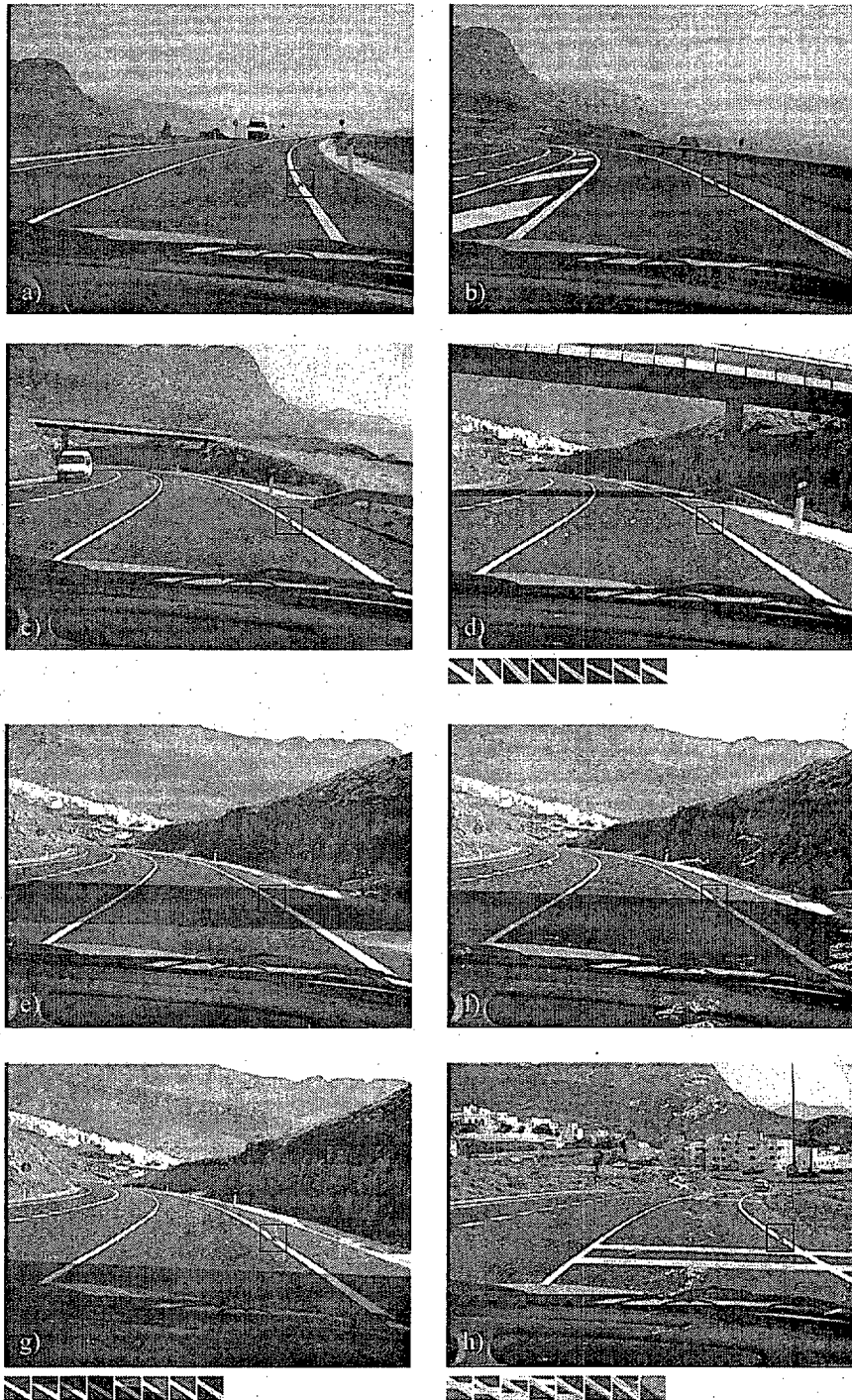


Figura 5.18: Ocho cuadros extraídos de una secuencia donde se ha realizado el seguimiento de un solo punto sobre línea del margen derecho de la carretera. La distancia recorrida es de dos kilómetros a una velocidad de 90 Km/h. En el cuadro g) se aprecian las vistas contenidas en el banco de patrones y que permiten recuperar instantes más tarde la línea derecha seguida.



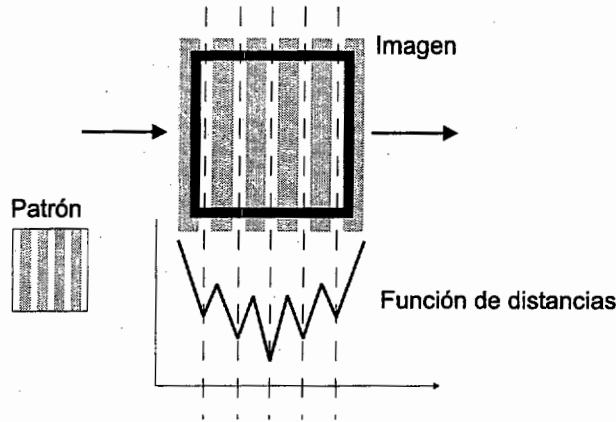


Figura 5.19: La figura muestra cómo un patrón con unas determinadas características puede generar múltiples mínimos muy cercanos. En el caso del patrón de la figura, éste contiene bandas grises y blancas que son las causantes de los mínimos locales que se generan en cada banda blanca.

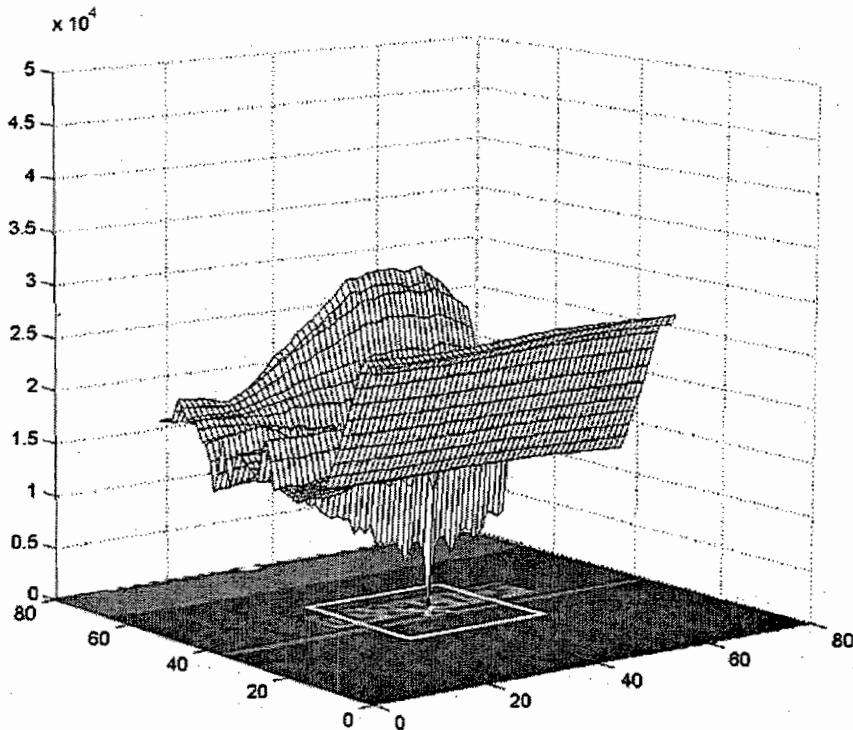


Figura 5.20: Función de similitud en el primer cuadro de la secuencia. Se observan el mínimo y los varios mínimos locales a lo largo de la región que corresponde al resto de los números de la matrícula.

lo fije en una posición concreta. Esta excesiva actualización se ilustra claramente en la Figura 5.23, donde se observa que el mínimo absoluto excede el umbral de cambio cada pocos cuadros, y en ocasiones en cada cuadro.

Con la utilización de un banco de cinco patrones se retrasa en gran medida la deriva y el resultado mejora sustancialmente, como se ilustra en la Figura 5.22. Esto se debe a que gran parte de las actualizaciones que se llevan a cabo no son sobre patrones nuevos extraídos de la imagen sino a patrones antiguos recuperados del banco de patrones. En esta secuencia experimental, el número de recuperaciones del patrón almacenados en el banco fueron 315. Por tanto, la utilización de un banco de patrones mejora el comportamiento del seguimiento con este tipo de objetos de interés, no decrementando la frecuencia de actualización del patrón sino recuperando los patrones previamente utilizados del banco, lo cual hace que la deriva se reduzca considerablemente.

Las frecuentes actualizaciones provocadas por este tipo de patrones no sólo son evitables mediante la utilización de un banco de patrones. Es posible detectar esta característica del objeto en el momento de realizar el seguimiento y seleccionar otra parte del objeto de interés para llevar a cabo el seguimiento, claro que esta solución exige la intervención de un módulo superior no precategorizado que garantice la idoneidad de la vista del objeto a seguir. Otra de las posibles soluciones consiste en agrandar el área de búsqueda de mínimos y de esta forma enmascarar o desestimar mínimos locales cercanos a otros mínimos menores. Una solución similar consiste en no considerar la existencia de mínimos locales a una menor distancia espacial determinada de la posición del mínimo absoluto u otro mínimo menor.

### 5.8.5. Conclusiones

La búsqueda de patrones utilizando antiguas vistas almacenadas en un banco de patrones permite, en numerosos escenarios de aplicación, recentrar el objeto de interés. Esta recuperación permite aumentar la confianza del seguimiento al retomar la curva de transformación visual que ya ha mostrado el objeto de interés previamente. De la misma forma, un banco de patrones permite salvar situaciones de ocultamiento o desaparición momentánea del objeto de interés.

El principal inconveniente de la utilización de un banco de patrones reside en que el tiempo de cómputo para realizar las búsquedas se multiplica por el número de patrones del banco.



Figura 5.21: Primer y último cuadro de una secuencia de 26 segundos donde el seguimiento se ha llevado a cabo sin la utilización de un banco de patrones. Como se observa el objeto de interés ha derivado hacia el parachoques trasero del camión donde la falta de referencia horizontal hace que el objetivo se mueva aleatoriamente de izquierda a derecha.



Figura 5.22: La misma secuencia que la anterior. En este caso, el seguimiento se ha hecho utilizando un banco de cinco patrones. El resultado, como se puede observar, es satisfactorio. La utilización del banco ha evitado, o al menos retrasado, la deriva del objetivo.

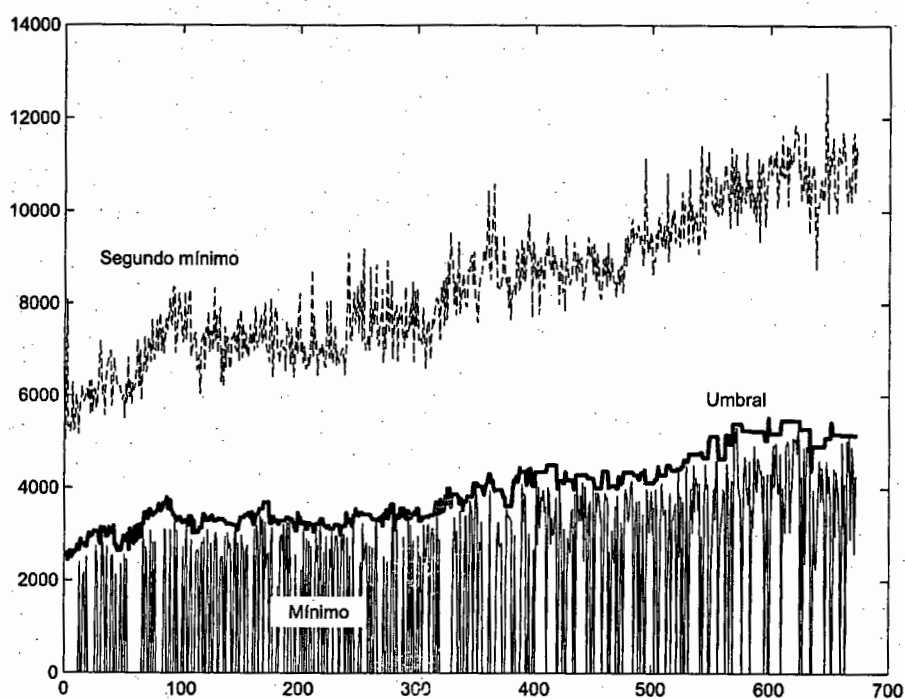


Figura 5.23: La figura muestra claramente la gran frecuencia con la que el patrón se debe actualizar. Aún así, se logra recuperar un gran número de veces el patrón del banco de patrones.

## 5.9. Detección y seguimiento de objetos: DESEO.

En las secciones experimentales anteriores se ha demostrado el mejor comportamiento de la actualización de los patrones en los métodos de búsqueda basado en áreas mediante la propuesta teórica de *Aprendizaje Basado en Contexto*, presentada en el Capítulo 2. Además, se ha introducido, en el Capítulo 4, un esquema de módulo de seguimiento que hace uso de estos resultados aplicándolos a los métodos de búsqueda que incorpora. Esta sección pretende, por tanto, constatar experimentalmente las cualidades de robustez e integración del diseño de módulo de seguimiento propuesto. Para ello, esta sección final muestra, de manera descriptiva, una aplicación práctica experimental que incluye la integración de un módulo de seguimiento, construido según los planteamientos expuestos en el Capítulo 4, en el marco de un sistema de visión más general.

Como consecuencia de este trabajo se exponen, así mismo, diversas consideraciones sobre la construcción de sistemas de visión modulares que, aunque no corresponden estrictamente con el objetivo de esta tesis, sí se considera interesante su inclusión en esta sección debido a que aclaran diversos aspectos concernientes al diseño de esta aplicación experimental denominada DESEO.

El sistema DESEO (DEtección y SEguimiento de Objetos), es un desarrollo basado en una arquitectura modular para el diseño de sistemas de percepción-acción, usando técnicas de Visión Activa (Aloimonos y Weiss, 1988) para la detección y seguimiento de objetos móviles en tiempo real. Su naturaleza modular facilita el reparto de actividades complejas sobre una red de módulos que implementan comportamientos más simples. DESEO integra un hardware consistente en una cabeza binocular comercial (Helpmate BiSight), una placa controladora de motores basada en DSP (Delta Tau PMAC) y dos placas de desarrollo basadas en el DSP de Texas Instruments TMS320C80 (C80), véase Figura 5.24. Todos estos elementos se integran a través de un PC dual ejecutando Windows NT 4.0. Debido a su naturaleza modular, el sistema puede ser reconfigurado para realizar tareas distintas.

### 5.9.1. Consideraciones previas para el diseño de SVC

La Visión por Computador parece tener una carencia de aproximaciones metodológicas comunes para el diseño de sistemas de visión, situación que es extensible en general al diseño de sistemas robóticos más complejos (Crowley y Bedrune, 1994). Esta carencia de metodología provoca normalmente que los sistemas de visión por computador (SVC) sean diseñados como sistemas cerrados que son difícilmente reusables para

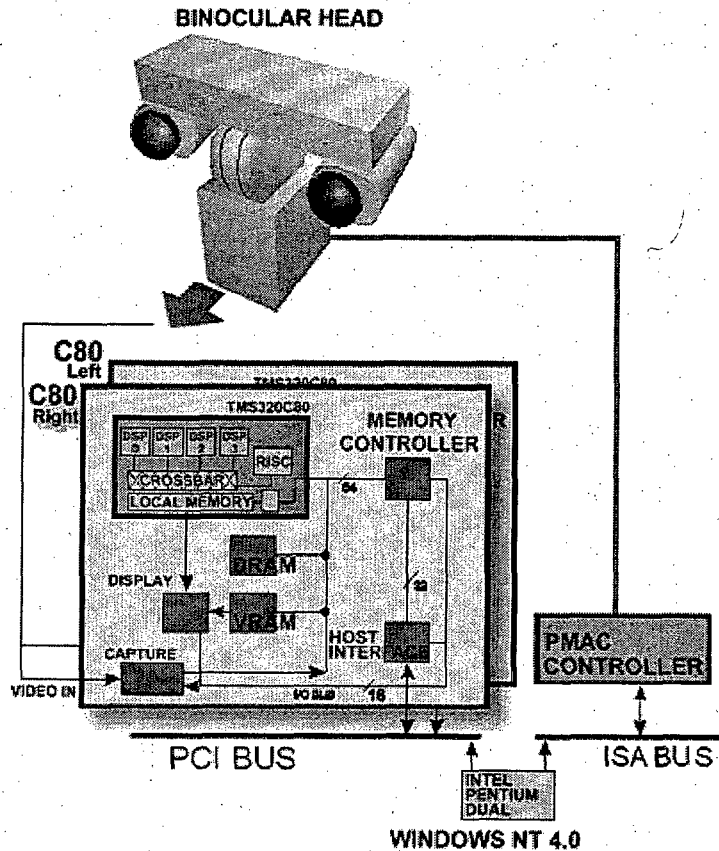


Figura 5.24: Representación esquemática de los componentes hardware de Deseo. El sistema está formado por una cabeza robótica Helpmate, dos placas de desarrollo TMS320C80, un controlador de motores Pmac y un PC ejecutando Windows NT 4.0.

otros problemas de visión, aún sólo ligeramente, distintos para los que fueron diseñados originalmente. En el proceso de diseño de un SVC es necesario tomar en cuenta el sistema dentro del marco de una metodología conceptual que asuma a priori consideraciones, objetivos y criterios de diseño. En este escenario, la disponibilidad de una arquitectura conceptual tiene la misión de salvar el vacío existente entre los recursos proporcionados por la tecnología asumida y las restricciones de diseño y requerimientos.

En base a estas ideas, se ha desarrollado DESEO de una forma sistemática tomando como premisa fundamental el que *los sistemas de visión por computador deben diseñarse para permitir su construcción de una manera incremental*. De esta forma, el concepto de módulo, como entidad lógica mínima que realiza alguna actividad útil, es el elemento básico. Además, para cumplir con este objetivo y favorecer la reconfigurabilidad del sistema en términos de módulos, estos deben compartir estructuras e interfaces comunes.

## 5. Evaluación experimental y aplicaciones

Posteriormente, la existencia de un marco conceptual para la integración de tales módulos facilitará en gran medida el desarrollo de estos sistemas de visión. Con tal fin se ha desarrollado CAV, una herramienta capaz de modelar los mecanismos de control necesarios para llevar a cabo la integración de diversos módulos que realicen las tareas visuales necesarias para el funcionamiento global del sistema, (Domínguez et al., 2000a; Domínguez et al., 2000b; Domínguez et al., 1999).

### 5.9.2. CAV

La *Arquitectura de Control para sistemas de Visión Activa*, CAV, modela el esquema de control como un conjunto de entidades asíncronas débilmente acopladas interactuando entre ellos, denominadas en esta tesis *Objetos de Procesamiento*. Este concepto permite modelar entidades activas heterogéneas cooperantes donde el procesamiento y el conocimiento pueden estar distribuidos. Además, cada objeto de procesamiento puede verse como una entidad auto-contenida que soluciona problemas y que exhibe las siguientes propiedades:

- **Autonomía**, debe ser capaz de realizar sus tareas sin intervención externa.
- **Capacidad de comunicación**, debe ser capaz de interactuar con otros objetos de procesamiento y su entorno.
- **Capacidad de respuesta**, debe poder percibir su entorno cambiante y actuar bajo restricciones temporales.
- **Proactividad**, debe ser capaz de mostrar un comportamiento dirigido a objetivos y de manera oportunista, tomando iniciativas cuando lo considere necesario.

Este concepto encaja convenientemente con entidades autónomas débilmente acopladas unas con otras, interactuando asíncronamente entre ellas y realizando diferentes tareas dentro de un sistema de visión completo como el que se concibe aquí. Sobre este concepto de objeto de procesamiento, se ha diseñado un esquema de control modular para sistemas de visión activa que se ha usado para desarrollar DESEO.

El objetivo de esta arquitectura es la de permitir la definición de un sistema con un comportamiento complejo, (Jones y Crowley, 1995), que emerja de la interacción de comportamientos individuales de cada objeto de procesamiento. Más específicamente, es posible modelar en cada objeto de procesamiento una actividad individual y su interacción con otros objetos de procesamiento. El comportamiento global se obtiene por la interacción de las actividades de los objetos de procesamiento.

Las señales constituyen el medio de interacción, esto es, el mecanismo para comunicarse los objetos de procesamiento unos con otros. Cada objeto de procesamiento puede emitir señales que transportan información y que cualquier otro objeto de procesamiento puede capturar y usar.

CAV sistematiza las declaraciones de los objetos de procesamiento, qué señales son enviadas y qué hacer cuando estas se reciben. Se modelan además las declaraciones de las señales y los datos que transportan. La arquitectura realiza transparentemente los accesos paralelos y/o concurrentes para compartir datos. De esta forma, los desarrolladores no necesitan preocuparse de estos aspectos, lo que redundará en una disminución de los tiempos de desarrollo.

La arquitectura utiliza recursos proporcionados por los sistemas operativos multitarea. El paralelismo entre objetos de procesamiento se consigue mediante concurrencia en sistemas monoprocesador. En sistemas multiprocesador, el paralelismo se obtiene dependiendo de cuántos procesadores están disponibles y cuántos objetos de procesamiento haya.

CAV permite asignar una prioridad a cada objeto de procesamiento, de tal forma que se puedan establecer jerarquías. Los objetos de procesamiento de bajo nivel pueden ser priorizados sobre los objetos de procesamiento de alto nivel. Expresado de otro modo, los objetos de procesamiento de frecuencia de funcionamiento alta obtendrán preferencia sobre los objetos de procesamiento de baja latencia.

En resumen, la arquitectura CAV permite modelar esquemas de control para sistemas de visión activa como objetos de procesamiento y señales. Además, proporciona plantillas de los objetos de procesamiento y las señales, de tal forma que los desarrolladores tan sólo deban rellenarlas con comportamientos y señales para tener operativo un sistema de visión.

### Tipología de los objetos de procesamiento

En CAV se distinguen los siguientes tipos de objetos de procesamiento:

- *Objetos de procesamiento fuente:* Sólo envían señales. No reciben, por tanto, ninguna señal excepto las de comienzo y finalización. Estos objetos de procesamiento están relacionados, normalmente, con los componentes perceptuales de un sistema.
- *Objetos de procesamiento de destino:* Estos objetos de procesamiento no emiten ninguna señal, sólo las reciben. Son objetos de procesamiento relacionados con los efectores del sistema.





Figura 5.25: Cabeza robot de DESEO

- *Objetos de procesamiento genéricos:* Envían y reciben señales. La arquitectura no impone ninguna restricción a estos objetos de procesamiento, por lo que pueden recibir datos de elementos perceptores y enviarlos a elementos efectores.
- *Objetos de procesamiento distribuidores:* Con esta clase de objetos de procesamiento pueden definirse "buses virtuales" en esta arquitectura. Su objetivo consiste en distribuir y repartir señales a través del sistema. Tienen sentido cuando una señal debe ser recibida por muchos objetos de procesamiento.
- *Objetos de procesamiento de red:* Estos objetos de procesamiento establecen "buses virtuales" entre diferentes estaciones de trabajo conectadas en red. Como los objetos de procesamiento distribuidores, su objetivo es sólo distribuir y repartir señales, aunque en este caso, a través de una red local.

### Tipología de las señales

CAV proporciona los siguientes tipos de señales:

- *Señales de comienzo y finalización:* Estas señales no transportan datos. La señal de comienzo se emite cuando el sistema comienza la ejecución, y la de finalización se emite cuando termina el funcionamiento del sistema. Ambas señales llegan a todos los objetos de procesamiento del sistema y son definidas por el sistema de forma transparente al usuario.
- *Señal de temporización:* Las acciones periódicas que posean los objetos de procesamiento pueden ser activadas mediante este tipo de señales. Esta señal especial

la posee cada objeto de procesamiento, no transporta datos y es definida transparentemente por la arquitectura.

- *Señal genérica o definida por el usuario:* Estas señales son usadas para señalar o transportar datos entre diferentes objetos de procesamiento.

El núcleo de cada objeto de procesamiento consiste en un bucle de proceso de señales donde se ejecutan acciones dependiendo de las señales recibidas. Se debe definir una acción para cada tipo definido de señal recibida.

### Implementación de DESEO: Definiciones de objetos de procesamiento y señales

En (Guerra et al., 1998a; Guerra et al., 1998b) pueden verse los criterios y metodología utilizados para analizar y diseñar el sistema DESEO, así como los objetos de procesamiento y señales identificados. La Figura 5.26 ilustra el conjunto de tareas y señales que conforman DESEO.

El estado inicial del sistema consiste en el análisis de las imágenes capturadas periódicamente tratando de localizar objetos que se muevan y que se pueda suponer que son individuos caminando normalmente. Después de la fijación sobre estos objetos, se intenta localizar la cabeza e identificar a la persona a lo largo de los sucesivos cuadros. Una posible variación de este comportamiento es comandar al sistema para que encuentre un individuo específico, aunque este comportamiento no ha sido implementado.

Los estados que conforman el sistema DESEO, y que aparecen en la Figura 5.26, se describen a continuación:

- **ADQ:** (Objeto de procesamiento de adquisición). Es un objeto de procesamiento genérico. Obtiene imágenes digitalizadas por ambos SDB C80 ya que se encuentra unido directamente a los elementos perceptuales (cámaras). Sirve a los objetos de procesamiento de localización y detección y su frecuencia de trabajo es variable.
- **DEL y DER:** (Objetos de procesamiento de detección derecho e izquierdo). Estos objetos de procesamiento genéricos realizan la detección de movimiento en cada cámara, computando las velocidades para cada región o "blob" movimiento independiente.
- **LOC:** (Objeto de procesamiento de localización). Este objeto de procesamiento genérico localiza cabezas y caras de personas para focalizar la atención sobre ellas.

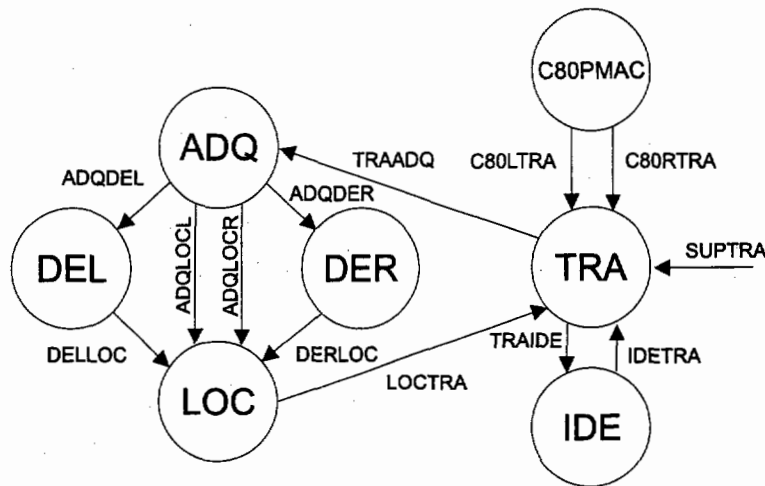


Figura 5.26: Esquema de objetos de procesamiento y señales del sistema DESEO

Este objeto de procesamiento utiliza pistas de color y movimiento provenientes de los objetos de procesamiento de detección.

- **TRA:** (Objeto de procesamiento de seguimiento). Objeto de procesamiento genérico que incluye el módulo de seguimiento para seguir las cabezas o caras localizadas con el fin de obtener buenas vistas de ellas para realizar la identificación con éxito. Controla además los motores de la cabeza robot y la frecuencia de adquisición. De esta forma, cierra el lazo de detección-localización del sistema de atención.
- **C80PMAC:** Objeto de procesamiento fuente que recoge información de nivel bajo referente al estado (posición y velocidad) de los motores que mueven la cabeza robot. Esta información tiene utilidad junto con las imágenes adquiridas en ese mismo momento.
- **IDE:** (Objeto de procesamiento de identificación). También un objeto de procesamiento genérico, es el objeto de procesamiento de más alto nivel. Realiza la identificación de caras de las personas seguidas.

Las señales que se procesan en DESEO son:

- **ADQDEL, ADQDER, ADQLOCL y ADQLOCR:** Estas señales transportan imágenes digitalizadas por las cámaras, sus tiempos de adquisición y los ángulos de cada motor de la cabeza robot.
- **DELLOC y DERLOC:** Transportan información sobre las regiones (*blobs*) con movimiento independiente desde los objetos de procesamiento de detección de movimiento hasta el objeto de procesamiento de localización.

- **LOCTRA**: Cuando se detectan las cabeza o caras, la atención se focaliza en ellas mediante esta señal. El objeto de procesamiento de seguimiento, y por tanto el módulo de seguimiento, comenzará a realizar el seguimiento sobre estos objetos de la escena. Obsérvese que esta señal transporta información que contiene el patrón y color a seguir y la posición donde debe empezar a buscar el módulo de seguimiento.
- **TRAADQ**: Señal que controla la latencia de adquisición de imágenes por parte del objeto de procesamiento de adquisición. Este señal conlleva la latencia del bucle de seguimiento.
- **C80LTRA** y **C80RTRA**: Estas señales proporcionan información de bajo nivel sobre el estado de los DSP y de cada motor de la cabeza robot.
- **TRAIDE**: Cuando se solicita la identificación de una cara por el usuario o periódicamente por el sistema en si, el objeto de procesamiento de seguimiento emite esta señal para indicar la petición al objeto de procesamiento de identificación.
- **IDETRA**: Señal de salida del objeto de procesamiento de identificación. Indica que alguien ha sido detectado.
- **SUPRA**: Señal del usuario para solicitar al sistema la identificación de la persona seguida.

### Cronogramas del sistema DESEO

Para cada objeto de procesamiento del sistema DESEO, es posible registrar todas sus acciones y las señales que envía y recibe. En la Figura 5.27 se expone un cronograma de algunos de los tiempos de procesamiento del sistema mostrando los periodos de trabajo de los objetos de procesamiento involucrados en llevar a cabo la detección de un objeto de interés, su localización, un movimiento sacádico para su fijación y, por último, el comienzo del proceso de seguimiento. El diagrama muestra cómo los objetos de procesamiento funcionan con su propia latencia y son activados asincrónamente por las señales.

La Figura 5.27 muestra, en primer lugar, el procesamiento del objeto de procesamiento de localización (**LOC**), marcado en la figura con la leyenda *DE*, para llevar a cabo la localización de un objeto detectado. Mientras tanto, el objeto de procesamiento de adquisición ejecuta el envío de señales a **LOC** con imágenes capturas durante las peticiones de localización, *AD*. Una vez localizado el objeto de interés se lleva cabo un movimiento sacádico sobre él por parte del objeto de procesamiento de seguimiento

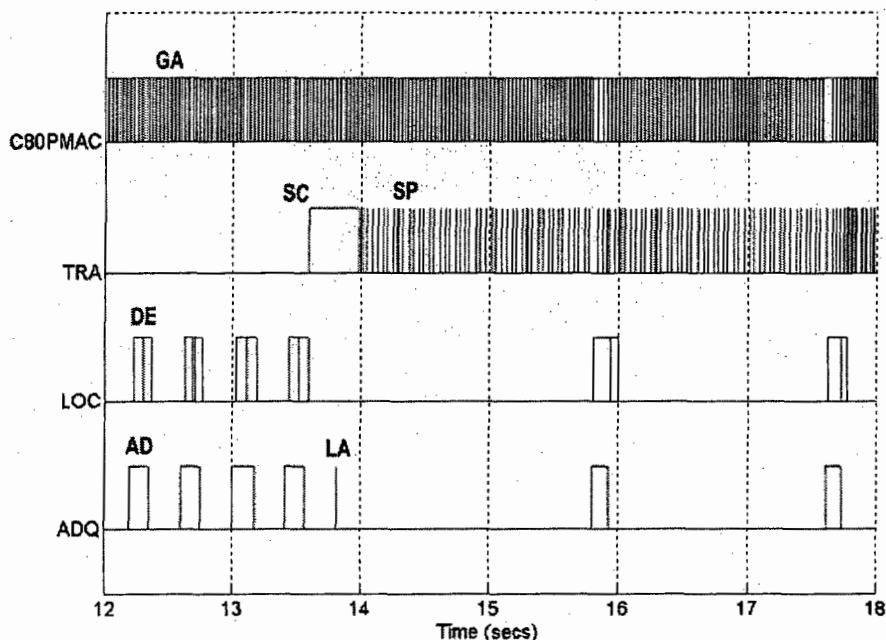


Figura 5.27: Cronograma de algunos de los procesos del sistema DESEO. Las leyendas en el gráfico corresponden a: **GA**, obtención de datos a bajo nivel. **SC**, movimiento sacádico y fijación. **AD**, proceso de adquisición de imágenes. **DE**, detección y localización de un objeto de interés. **SP**, seguimiento suave del objeto de interés. **LA**, cambio de latencia en el bucle de detección-localización.

(**TRA**). La duración de este movimiento se muestra en **SC**. Cuando el objeto de interés es centrado, se lleva a cabo el proceso de seguimiento suave en (**TRA**), el objeto de procesamiento envía ahora señales, **SP**, hacia el módulo de identificación para intentar su reconocimiento. Se observa que durante el seguimiento la frecuencia de trabajo de los objetos de procesamiento **LOC** y **ADQ** se reduce, obedeciendo la señal **LA** que establece ahora mayor prioridad al proceso de seguimiento. Sin embargo, durante todo este proceso, el objeto de procesamiento (**C80PMAC**) se mantiene trabajando independientemente a una alta frecuencia ofreciendo la información del estado de los motores de la cabeza robot, **GA**.

Hay que señalar que en la figura no están representados todos los procesos que se generan en un intervalo de funcionamiento del sistema como el expuesto, sino tan sólo los más significativos para una correcta comprensión del proceso de localización, seguimiento e identificación.

### 5.9.3. Integración de módulos

Una vez descrita la arquitectura CAV que soporta a DESEO se exponen más detalladamente algunos de los diferentes módulos que se insertan dentro de cada objeto de procesamiento. Concretamente, los tres módulos más importantes para el funcionamiento de este sistema son los módulos de localización, seguimiento y reconocimiento.

#### Módulo de localización

Este módulo se encuentra dentro del objeto de procesamiento de localización (LOC) y es realmente una secuenciación de dos etapas, dedicadas respectivamente a detectar primero áreas que exhiben movimiento coherente y producir luego una estimación de la posición de la cabeza dentro del área seleccionada. La detección de movimiento actúa como un mecanismo básico para dirigir la atención del sistema hacia áreas potencialmente de interés. Por simplicidad, en el contexto de esta aplicación se asume que las áreas con movimiento, a priori corresponden a individuos caminando. Esta hipótesis es confirmada o negada posteriormente basándose en evidencias, como se explica más adelante.

La detección de movimiento es realizada primeramente computando el flujo óptico y seguidamente segmentándolo para seleccionar las áreas que muestran un movimiento coherente. El flujo óptico se computa según la técnica presentada en (Campani y Verri, 1992), que aproxima el campo de flujo estimando el flujo óptico sólo en el centro de los fragmentos rectangulares en los que es dividida la imagen. La asunción básica de esta técnica reside en que dentro de cada fragmento rectangular se mantiene una aproximación lineal del campo de flujo. Claramente, el flujo óptico computado es una aproximación al campo de flujo "real". Para computar el flujo óptico en puntos distintos del centro de los fragmentos, se utiliza una interpolación lineal. Esta técnica ofrece una solución flexible que permite balancear precisión numérica con requerimientos computacionales variando simplemente el tamaño de los fragmentos. Esto hace posible computar el flujo óptico a diferentes niveles de precisión, por ejemplo a alta resolución en regiones de atención para un seguimiento preciso y un nivel más burdo en la periferia del campo de vista para detectar grandes movimientos de objetos (Murray et al., 1995).

Usando la cinemática conocida de la cabeza es posible sustraer del flujo óptico la componente de movimiento generada por el propio movimiento de la cabeza (Murray et al., 1995). Esto permite la detección de objetos en movimiento incluso cuando la cabeza también está en movimiento. En la segunda etapa de este módulo se utiliza un umbral del flujo óptico computado y detección de regiones o *blobs* para extraer las zonas

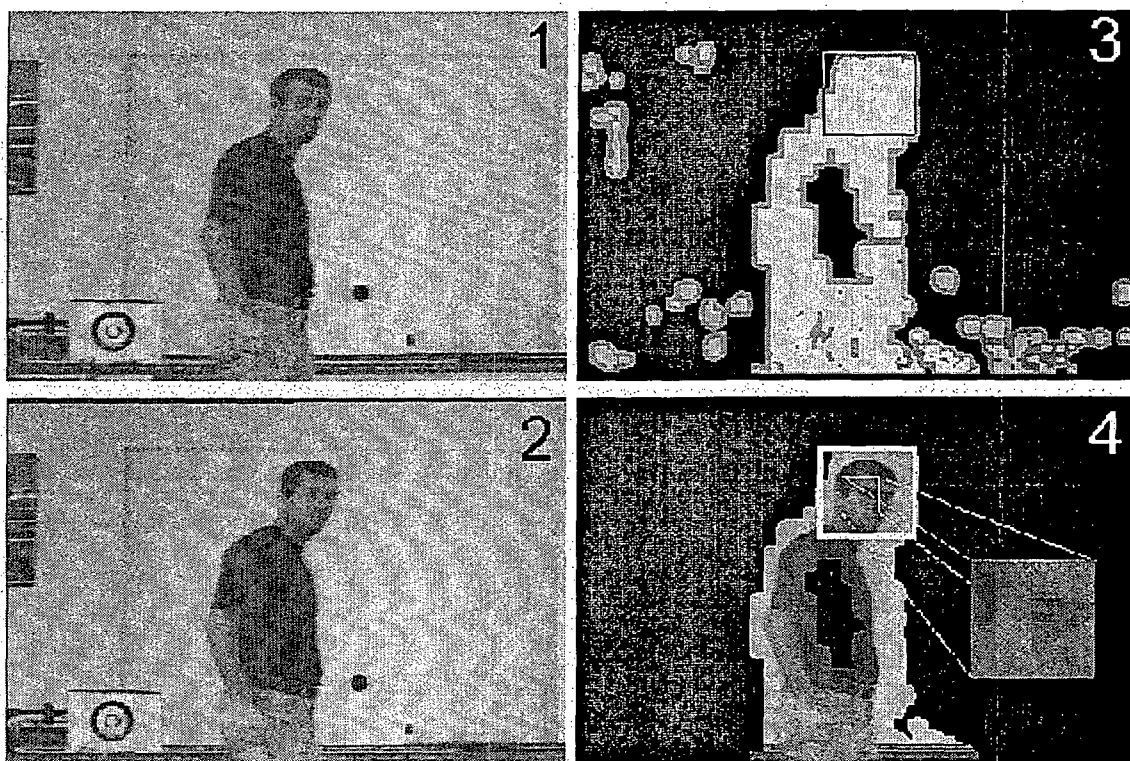


Figura 5.28: Imágenes de datos en un estado del análisis del flujo óptico. Las imágenes 1 y 2 representan dos cuadros consecutivos que son la entrada al módulo de detección de movimiento. La imagen 3 muestra una imagen de *blobs* proporcionada por el algoritmo de flujo óptico. La imagen 4 representa la cabeza localizada y el patrón extraído del algoritmo de correlación.

que muestren un movimiento coherente. Estas zonas son clasificadas de acuerdo a su tamaño, de tal forma que un área centrada en la parte superior de la zona más grande es extraída como una área de interés potencial. Este área es donde se espera encontrar la cabeza del individuo. Su tamaño y posición dentro del blob son determinados usando el conocimiento acerca de las proporciones de ancho y alto normales de una cabeza humana y la media de velocidad medida en esa zona, véase Figura 5.28. Una estrategia similar es usada en muchos otros sistemas, como en el sistema Perseus (Kahn et al., 1996).

### Módulo de seguimiento

El módulo de seguimiento se incluye dentro del objeto de procesamiento de seguimiento (TRA) y está organizado según las propuestas presentadas en el Capítulo 4, donde pueden encontrarse todos los detalles internos de la resolución de los problemas y restricciones planteadas del algoritmo de seguimiento utilizado por DESEO.

Los métodos de búsqueda utilizados por este módulo de seguimiento son: la búsqueda de patrones y el color. En cuanto al color su procesamiento se basa en la búsqueda de manchas suficientemente grandes de cierto color. El color se codifica en el espacio YUV. La componente de intensidad Y se elimina y la crominancia se clasifica en el plano UV, determinando aquel que pertenezca a la piel.

Los descriptores en este módulo están formados por un par de coordenadas  $x_u$ ,  $y_v$ , que representan un color determinado. De esta forma se etiquetan todos los pixels de la imagen como pertenecientes o no pertenecientes al objeto buscado. Una vez todos los pixels han sido etiquetados se aplica un algoritmo de extracción de regiones.

Se genera entonces un vector de regiones conexas existentes en la imagen. Los descriptores de color, además, poseen parámetros de la forma de las regiones buscadas. Esto significa que las regiones de un determinado color deben guardar una determinada relación ancho/alto con el rectángulo que las circunscribe para que sean consideradas como pertenecientes al objeto buscado. De tal forma, que las pistas realmente buscadas son: color y forma, véase Figura 5.29.

En cuanto al modo de operación establecido, el módulo de seguimiento se ha configurado para seguir varios objetivos de una manera secuencial. Cada objetivo es un elemento en una lista de Focos de Atención construida, en base a áreas de interés, por el módulo de detección. Para el foco de atención activo en un momento dado, la primera acción consiste en efectuar un movimiento sacádico que centre la imagen en la localización predicha del objetivo en el marco de referencia estático de la cabeza robot. Entonces se procede a un seguimiento normal hasta que el próximo foco de atención sea activado por el control visual.

Este foco de atención puede ser emplazado en cualquier parte del campo visual para poder seguir rápidamente al objeto, y de alguna forma, aliviar los tiempos de latencia debidos al sistema electromecánico de la cabeza robot. En paralelo, la cabeza es comandada a la posición esperada del objetivo usando un filtro predictivo alfa-beta, que toma en cuenta las latencias del proceso de seguimiento. La cabeza es comandada a nuevas posiciones cada 40 ms.

Una interesante característica es que se puede cambiar a un nuevo foco de atención cada ciclo de procesamiento si el foco actual y el nuevo son visibles dentro del mismo campo visual. La política de intercambio entre focos de atención está basada en una cola de prioridad. La longitud de la cola, o el número de objetos simultáneos que el sistema puede seguir simultáneamente, puede ser variable pero en los experimentos ha sido fijado a dos o tres.



Inicialmente, la lista de focos de atención está vacía hasta que el supervisor o el módulo de detección envía áreas de atención o alta relevancia al módulo de seguimiento para fijarlo. El manejo del foco de atención es llevado a cabo por un gestor que asigna una prioridad a cada foco de atención. Esta prioridad es computada en base a la velocidad estimada del objetivo y el error en la posición predicha.

Esto significa que incluso si existen otras áreas de atención pendientes de atención visual, el sistema puede no atenderlas si el foco de atención activo se mueve muy rápido o de una manera impredecible. En la práctica, seguir varios objetivos es sólo factible si se mueven lentamente y a velocidad constante. El cambio entre focos de atención se realiza mediante movimiento balísticos seguidos de fijación. El módulo de seguimiento debe asignar al foco de atención activo una rodaja de tiempo suficiente para realizar la sacada y estabilizarse en el objetivo para actualizar la estimación de su posición y velocidad. En la Figura 4.5 del Capítulo 4 se muestran diversos cuadros de una secuencia de funcionamiento del sistema DESEO.

## Comandos

El módulo de seguimiento posee una serie de comandos que son enviados por un nivel de control superior para controlar su funcionamiento. En el caso particular de DESEO, los objetos de procesamiento que solicitan el seguimiento de un determinado objeto comandan órdenes mediante determinadas señales al objeto de procesamiento de seguimiento. El objeto de procesamiento de seguimiento (**TRA**) traduce entonces las señales provenientes a comandos propios del módulo de seguimiento.

Entre los comandos existen órdenes para crear un nuevo objeto de interés para el seguimiento, cambiar sus propiedades, abandonar el seguimiento de un objeto, conmutar el seguimiento hacia otro objeto, etc. En el Apéndice D se detallan todos los comandos implementados en un módulo de seguimiento, los cuales corresponden también al módulo de seguimiento de Deseo.

## Reconocimiento

El módulo de reconocimiento, incluido dentro del objeto de procesamiento de identificación (**IDE**), opera sobre una pequeña ventana obtenida por el módulo de seguimiento, centrada en el punto de mejor coincidencia del patrón de la supuesta cara sobre la imagen. El procesamiento realizado en primer lugar consiste en localizar el centro de la cabeza en la ventana, véase Figura 5.30. Posteriormente, dadas las com-

plejas características de los procesos de reconocimiento y sus limitaciones inherentes, el reconocimiento es tratado como una cadena de clasificaciones oportunistas.

#### 5.9.4. Conclusiones

El objetivo de esta sección experimental ha sido la validación del modelo de módulo de seguimiento propuesto en el Capítulo 4. Para tal fin se ha diseñado y construido DESEO, que se plantea como un sistema de visión por computador que requiere, dada la finalidad de su cometido, unas elevadas prestaciones a los componentes que lo forman.

La validación del módulo de seguimiento se ha fundamentado en la verificación de las capacidades atribuidas a éste. Principalmente, la robustez de funcionamiento y la facilidad de integración.

La robustez mostrada por el módulo de seguimiento se ha evidenciado por su capacidad para realizar seguimiento en tiempo real con la posibilidad de conmutación entre distintos objetos de interés.

En cuanto a la facilidad de integración, el conjunto de comandos incorporado en el módulo de seguimiento se manifiesta completo y válido para permitir el funcionamiento y los requerimientos de un sistema como DESEO. Por otro lado, la autonomía en su funcionamiento constituye una ventaja considerable para su integración, dado que se reduce el control global del sistema completo.

Es importante destacar aquí que la robustez y autonomía ofrecidas por el módulo de seguimiento están implícitamente soportadas por el concepto de *aprendizaje basado en contexto*, expuesto en el Capítulo 2 de forma teórica.



Figura 5.29: La figura representa la detección del color de piel. En este caso, han sido detectados tres regiones o *blobs*. Sin embargo, sólo uno de ellos cumple con las restricciones de forma. De esta manera, el blob superior es considerado como el objeto a seguir.

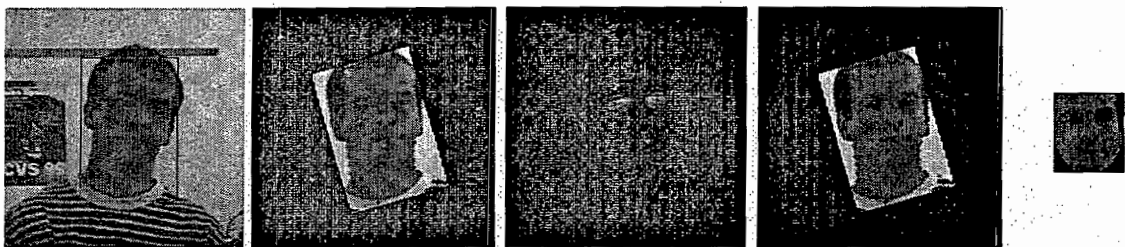


Figura 5.30: Proceso de normalización de caras previo al paso de indentificación. Como se observa en la figura, la cara es rotada convenientemente para obtener un rostro completamente vertical. Así mismo es también escalado y finalmente se elimina el resto de la imagen que no corresponda únicamente a la cara.



## Capítulo 6

# Principales aportaciones, conclusiones y trabajo futuro

### 6.1. Principales aportaciones

Las aportaciones principales de este trabajo de tesis incluyen tanto aspectos teóricos, dentro del campo de la representación visual, como metodológicos en el campo del diseño de módulos o componentes visuales para sistemas de visión más generales. Una enumeración de estas aportaciones con sus ventajas respectivas se presenta a continuación:

- Dentro de las aportaciones de este trabajo es necesario nombrar la realización de una clasificación y estructuración de los diferentes mecanismos empleados en la bibliografía en cuanto a los métodos de búsqueda mediante correspondencias de patrones basados en áreas. Este estudio ha permitido realizar un análisis en profundidad del problema del seguimiento mediante búsqueda de patrones.
- Dentro del campo de la representación visual se ha propuesto el uso de correspondencias isomórficas de segundo orden, en el sentido dado por Edelman (Edelman, 1997), con el fin de posibilitar el control del contexto o escenario de aplicación. De esta forma, se obtienen mejoras tanto en el rendimiento computacional como en la robustez alcanzada por los métodos de seguimiento.
- Mediante la particularización de las teorías propuestas acerca de la representación y el contexto en la búsqueda de patrones, se ha desarrollado un nuevo método para realizar de manera automática la actualización de los patrones a tomar como

referencia. Esta solución teórica al problema del seguimiento precategórico permite la adaptación del método a nuevos entornos de aplicación sin la necesidad de ningún ajuste manual o empírico previo.

- Se ha propuesto un nuevo método para el aprendizaje de un objeto a seguir que hace uso del contexto de aplicación y de una memoria visual que almacena las referencias significativas del objeto de interés. De esta forma, el seguimiento adquiere una mayor robustez con el modelo propuesto a efectos de:
  1. Relación de los puntos óptimos de muestreo respecto a un criterio de disimilitud mínima respecto al contexto.
  2. Descripción de objetos visuales por su apariencia
  3. Realización de un proceso automático y computacionalmente ligero de referenciación perceptual.

Todo esto dota de un mejor comportamiento y mayores posibilidades al módulo de seguimiento.

- Se ha mostrado la ventaja que supone la utilización de un módulo adaptable y autónomo de seguimiento en diversas aplicaciones de visión por computador. De esta forma se mejora la calidad del diseño, así como disminuyen los tiempos de desarrollo de sistemas que integren este tipo de módulos. Desde un punto de vista general se plantea la creación y desarrollo modular de subsistemas que procesen distintas tareas visuales para su uso en sistemas más complejos. Por tanto, se alienta la construcción de sistemas de visión que hagan uso de componentes ya definidos como metodología en la construcción de sistemas de visión por computador, y como único medio de producir desarrollos cada vez mayores y más complejos.
- Dentro del aspecto metodológico en la construcción de módulos de seguimiento se ha propuesto una estructuración interna de los mismos. Dicha estructura permite la cooperación de varios métodos de búsqueda y la posterior elección del esquema de fusión de resultados. Además, el diseño interno de este módulo de seguimiento permite la adopción de varias políticas de funcionamiento para llevar a cabo diferentes tipos de tareas, respuestas o comportamientos. Entre ellas, se encuentran la del seguimiento de múltiples objetos o la del seguimiento de un objeto por la fijación de varias de sus partes. Por último, se han descrito las funcionalidades que debe tener este tipo de módulos junto con los comandos que debe aceptar para su control total y ofrecer la máxima versatilidad.

- Se aporta la concepción de un esquema algorítmico paralelo que realiza la computación de operaciones de la familia de la correlación sobre arquitecturas hardware multiprocesador con memoria compartida de forma óptima en varios aspectos. En este trabajo se implementa el algoritmo de forma particular sobre una plataforma de desarrollo que integra un DSP multiprocesador.
- Paralelamente a este trabajo de tesis se ha desarrollado un software de seguimiento que hace uso de todos los puntos citados anteriormente. Este software permite su integración de forma fácil en prácticamente cualquier tipo de aplicación de visión en tiempo real y off-line. Basándose en este software, se han llevado a cabo la mayoría de los experimentos que se describen en este trabajo.
- Se ha evidenciado la viabilidad de los microprocesadores actuales de propósito general para su utilización en la ejecución de algoritmos de búsqueda. El rendimiento en tiempo real es perfectamente alcanzable por este tipo de microprocesadores, ofreciendo soportes económicos para el desarrollo de sistemas de visión.
- Experimentalmente se ha realizado un completo test de evaluación de las bondades de las distintas estrategias de actualización de patrones existentes en la bibliografía. Como resultado de estos experimentos se puede afirmar el mejor comportamiento de la estrategia de actualización basada en contexto propuesta en esta tesis.

## 6.2. Conclusiones

La construcción de módulos autónomos que realicen algún tipo de tarea visual ayuda de manera determinante al diseño de sistemas de visión complejos. De esta forma se consigue el aislamiento de ciertas tareas visuales en módulos cerrados, lo cual permite al diseñador obtener una apreciación y concepción del sistema completo más clara. Otra de las ventajas fundamentales que ofrece la construcción de módulos visuales concierne a las mejoras que progresivamente se pueden obtener en las técnicas de procesamiento incluidas en los mismos. Las optimizaciones del código o de los métodos en las tareas visuales aumentan el rendimiento del módulo de visión de forma interna. De esta manera, el sistema completo que hace uso de estos módulos ve incrementado su rendimiento sin la necesidad de modificaciones en su diseño. La modularización es también una característica básica para la posterior reutilización de los mismos componentes en otros sistemas o aplicaciones. Para ello es necesario una descripción de los interfaces necesarios para su control total.

Uno de los factores determinantes en la construcción de módulos visuales y en su posterior utilidad es el grado de autonomía de estos para llevar a cabo su tarea. Esto significa que es altamente recomendable que cada módulo visual realice su tarea de tal forma que no sea necesario llevar un control directo o intensivo sobre el funcionamiento del mismo.

Entre las distintas tareas visuales que pueden ser encomendadas a un módulo se encuentra la tarea del seguimiento visual. Los procesos de seguimiento están estrechamente ligados de una u otra forma a muchos tipos de aplicaciones o tareas visuales. El estudio de estos procesos y su incorporación de nuevas estrategias y métodos de búsqueda mejoran el rendimiento global de las aplicaciones completas.

La representación adecuada de los objetos de interés en los procesos de búsqueda, en especial las representaciones basadas en correspondencias isomórficas de segundo orden, permiten la mejora en el rendimiento y en la robustez del seguimiento ya que el contexto tiene marcada importancia dentro de la búsqueda del objeto de interés. De esta forma se evitan las pérdidas del objeto de interés debido a confusiones del método con objetos existentes en el escenario. Además permite la adecuación de los recursos computacionales a cada situación. En otras palabras, se minimiza el consumo de recursos computacionales, permitiendo que sean liberados para la ejecución de otras tareas. Esto conduce a la posibilidad de utilizar de manera factible equipamiento de cómputo de propósito general de bajo costo para ejecutar satisfactoriamente aplicaciones de visión por computador donde las tareas de seguimiento visual tengan una importancia crucial.

La aportación experimental de esta tesis demuestra de manera práctica y real las conclusiones aquí expuestas.

Como conclusión final es posible afirmar que la integración de tareas visuales en módulos o componentes se plantea como el camino a seguir en el desarrollo de sistemas de visión por computador. La integración de módulos visuales de terceros clarifica el diseño conceptual del sistema completo, mejora la calidad final y acelera el proceso de desarrollo. Además de la construcción modular de componentes visuales se precisa la definición de una arquitectura de soporte e integración de tales módulos. Estableciendo una correspondencia con el campo de la informática, se puede exponer el problema del avance en los sistemas de visión por computador como la necesidad de la definición de un "sistema operativo para la visión" que aúna y gestione los recursos o "módulos visuales" de manera eficiente sobre una plataforma común de desarrollo.



### 6.3. Trabajo futuro

Dentro de los posibles trabajos futuros que se puedan llevar a cabo para mejorar y ampliar el estudio de los planteamientos expuestos en esta tesis cabe destacar:

- *La integración y combinación de otras pistas visuales.* Es posible la ampliación a otras pistas o métodos de búsqueda que cooperen dentro del módulo de seguimiento mejorando la robustez obtenida. Los posibles mecanismos de combinación es, de esta forma, otro de los aspectos importantes a estudiar.
- *Experimentación en otros problemas y validación.* Aunque han sido numerosas las pruebas y experimentos realizados hasta la conclusión de este trabajo, es siempre aconsejable la verificación de la validez de los métodos expuestos mediante el uso de los mismos en más escenarios de aplicación y situaciones.
- *La integración del módulo de seguimiento con módulos categóricos con el fin de obtener un control de objetivos que mejore la robustez es otra de las propuestas futuras.* En diversas aplicaciones no sólo basta un sistema precategórico para llevar a cabo la tarea encomendada, es también necesario poder contar con módulos en el sistema que verifiquen, mediante procesos de alto nivel, el correcto funcionamiento de sistema completo.
- *Experimentación en referenciación perceptual integrando el módulo con otro de atención y un módulo categórico dotado de capacidad de referenciación, con el fin de evaluar las posibilidades de aprendizaje de conceptos visuales por parte del sistema completo.* Esta experimentación permitiría evaluar el proceso combinado para producción de sistemas con ciertos grados de autonomía y flexibilidad. Para aplicaciones en, por ejemplo, robótica o sistemas interactivos, como es el caso de las Interfaces Perceptuales de Usuario.



# Apéndice A

## Tecnologías hardware de soporte

El hardware juega un papel crucial dentro de la computación en tiempo real, no sólo porque el rendimiento del sistema recae sobre él sino también porque los costos de desarrollo varían sustancialmente entre los diversos tipos de plataformas existentes. Los diferentes aspectos hardware a considerar en el desarrollo de sistemas de visión por computador en tiempo real son el objetivo de este apéndice.

Dentro de la visión por computador en tiempo real básica, los algoritmos pueden ser clasificados en dos niveles: nivel bajo o de pixels y de nivel alto o de regiones. En el nivel bajo, las operaciones tienen lugar en pixels adyacentes o muy cercanos. Por otro lado, en los algoritmos de nivel de región, las operaciones se expanden sobre un mayor área o conjunto de pixels. Debido a la naturaleza de las operaciones de bajo nivel, éstas son idóneas para ser paralelizadas por medio de procesadores de tipo SIMD (Simple Instrucción Múltiples Datos) para alcanzar rendimientos elevados. Operadores como filtros, detección de bordes y sobre todo la búsqueda de patrones son enormes consumidores de recursos computacionales. Todas estas tareas de bajo nivel son frecuentemente requeridas como primeros pasos en sistemas visuales más complejos, y cuyo buen funcionamiento es crucial para el rendimiento del sistema completo. De esta forma, el seleccionar una arquitectura de computación acertada es una decisión crítica.

Se pueden realizar diferentes clasificaciones en el hardware disponible hoy día para el procesamiento de imágenes en tiempo real a bajo coste. Una forma útil de clasificación para sistemas de visión en tiempo real se realiza atendiendo a la especialización y flexibilidad ofrecidas por las diferentes arquitecturas. Alta especialización conlleva gran rendimiento en tareas específicas, sin embargo hace a estos sistemas inservibles para otras tareas diferentes. Por el contrario, los sistemas flexibles obtienen bajos rendimientos pero los hace reutilizables para otras aplicaciones. Cuanto más es-

pecializado es el hardware menos flexible y readaptable es, véase Figura A.1. Por tanto, atendiendo a este criterio de clasificación es posible establecer una taxonomía en las tecnologías disponibles de la siguiente forma:

- ASIC, Circuitos Integrados de Aplicación Específica: Esta tecnología es cara, no reutilizable y está únicamente justificada cuando otras tecnologías están muy por debajo del nivel de rendimiento exigido en el sistema o cuando los volúmenes de producción justifican los costos de desarrollo.
- FPGA, Field-Programmable Gate Arrays: Es una opción muy atractiva ya que el mismo componente hardware puede ser reconfigurado para tener una funcionalidad diferente. Sin embargo, esa tecnología es relativamente reciente y se encuentra en un nivel de desarrollo incipiente, con lo que desarrollar en esta tecnología implica un conocimiento y experiencia en VHDL y otras herramientas de desarrollo.
- DSP, Procesadores Digitales de Señal: Los DSPs han sido el recurso más comúnmente utilizado cuando se desea obtener una elevada capacidad computacional disponible manteniendo aún un alto grado de flexibilidad que los haga útiles para otro tipo de procesamiento posterior. Están normalmente adaptados para llevar a cabo operaciones tipo SIMD de manera muy optimizada sobre flujos de datos entrantes.
- Procesadores de propósito general: Tomando en cuenta su disponibilidad y bajo coste este tipo de procesadores son los elegidos cuando los requerimientos de tiempo de cómputo no son estrictos y tampoco se justifica la utilización de alternativas más caras. Este tipo de procesadores poseen el atractivo añadido del incremento sostenido de su rendimiento, que hace que procesadores de una generación a la siguiente puedan cuadruplicar su potencia de cálculo manteniendo los mismos precios.

Una descripción más detallada de cada una de las tecnologías presentadas se expone a continuación.

## A.1. ASIC

Los circuitos integrados de aplicación específica o módulos especializados se desarrollan para incrementar al máximo el rendimiento de una aplicación. Para conseguir esto, el algoritmo correspondiente se integra completamente en hardware, evitando, de

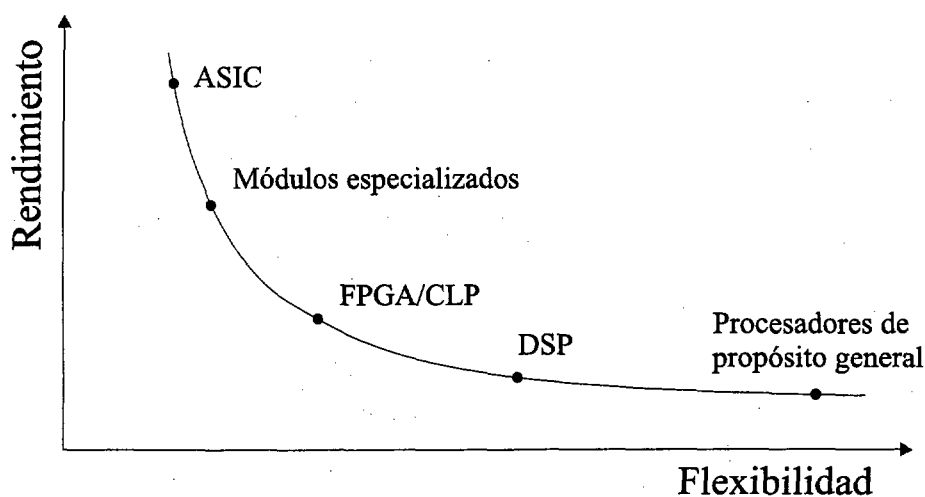


Figura A.1: Rendimiento vs. flexibilidad en las tecnologías usadas en sistemas de visión por computador.

este modo, el tiempo empleado en el nivel de decodificación de las instrucciones. Realizando además una optimización del hardware para esta determinada aplicación.

Estos diseños ad hoc son caros ya que son expresamente realizados para una tarea concreta. Sin embargo, algunos sistemas de visión hacen uso de módulos ASIC existentes y comercialmente disponibles que realizan ciertas tareas comunes entre diferentes aplicaciones, integrándolos como parte del sistema completo y reduciendo el coste de este sistema de visión considerablemente. (Inoue et al., 1992; Sawasaki et al., 1996) describen la utilización del chip SGS-Thompson STI3220, el cual fue originalmente diseñado para ser integrado dentro de videocámaras con el objetivo de realizar funciones de estabilización de imagen. Para llevar a cabo esta operación, el chip está optimizado para computar correlaciones locales basadas en diferencias absolutas de imágenes en 256 tonos de gris con patrones de  $8 \times 8$  o  $16 \times 16$  sobre áreas de búsqueda de  $23 \times 23$  o  $31 \times 31$  pixels respectivamente. Con áreas de búsqueda de  $16 \times 16$  el procesamiento puede ser realizado en 1.15 msec. Esto permite llevar a cabo varios procesos de correlación simultáneos sobre diferentes áreas de búsqueda manteniendo las restricciones de tiempo real (25/30 fps). Sin embargo, el sistema está limitado por patrones y áreas de búsqueda de tamaños fijos.

## A.2. FPGA

En 1985 Xilinx introdujo un dispositivo lógico reconfigurable denominado *field programmable gate array (FPGA)*, el cual hacía uso de una combinación de caracterís-

ticas tomadas de los *dispositivos lógicos programables* y de los *arrays celulares*. Debido a su gran flexibilidad, los FPGAs han sido rápidamente adoptados por investigadores y diseñadores, quienes han desarrollado sistemas completos utilizando sólo esta tecnología gracias al incremento progresivo en la densidad de puertas lógicas.

Básicamente, las FPGAs son dispositivos programables con un array interno de bloques lógicos, rodeados por un anillo de bloques de entrada/salida programables conectados entre sí. Un FPGA típico puede contener hasta decenas de miles de bloques lógicos e incluso un mayor número de flip-flops.

Esta tecnología combina el rendimiento de procesadores ASICs con la funcionalidad y versatilidad de los procesadores de propósito general. Se programan a muy bajo nivel, concretamente a nivel de puertas lógicas, lo cual permite asignar un mayor número de recursos lógicos donde las demandas de computación son mayores. De esta forma, el algoritmo es mapeado directamente en hardware.

Varios sistemas de visión, los cuales son aplicaciones con altas demandas de computación, se han desarrollado usando FPGAs ofreciendo resultados alentadores. Los algoritmos en este campo están caracterizados por operaciones repetitivas sobre grandes cantidades de datos, y por tanto, idóneos para ser paralelizados. Uno de estos sistemas es Splash 2 (Ratha y Jain, 1999), algunas de cuyas aplicaciones están cercanas al interés de este trabajo. Splash 2 es un sistema de visión desarrollado usando FPGAs donde se han mapeado, entre otros, algoritmos de seguimiento comunes. Cada placa de procesamiento Splash 2 consiste en un array de 16 FPGAs Xilinx 4010 (nombrados  $X_1$  a  $X_{16}$  y llamados *elementos de procesamiento* (PE)) y un decimoséptimo FPGA dedicado al control del flujo de datos dentro de la placa de procesamiento. Cada PE tiene sus propios 512 Kb. de memoria que es utilizada para el almacenamiento de datos temporales. Todos los PEs están conectados a través de un crossbar controlado por  $X_0$ . Estas placas están alojadas en una estación Sparc y son controladas por una placa de interface, la cual incorpora un bus de 36-bits (SIMD bus) funcionando a través de todos los PEs.

(Pudipeddi et al., 1997) describen un esquema de seguimiento visual generando un conjunto pirámidal de imágenes que abarca desde resoluciones gruesas a finas. Este esquema combina modelos de computación SIMD, pipelining y recirculación de datos con objeto de demostrar la potencia de las FPGAs (Splash 2) en tareas de análisis visual. El primer paso del algoritmo calcula una pirámide gaussiana de cinco niveles, comenzando con una imagen de  $512 \times 512$  pixels y terminando con una imagen de  $32 \times 32$ , la cual representa el nivel más burdo. Se utiliza la suma de diferencias absolutas para determinar la localización de la mejor coincidencia de una ventana de  $16 \times 16$  sobre un área de búsqueda de  $32 \times 32$ . Para evitar pérdidas debido a movimientos rápidos

del objeto seguido, el algoritmo comienza con una búsqueda sobre la imagen de menor resolución para obtener una primera aproximación a la localización del objeto, lo cual implica una búsqueda sobre la imagen completa. Una vez realizado esto, se lleva a cabo una búsqueda en el siguiente nivel sobre una área de  $32 \times 32$  centrada esta vez sobre la localización detectada en el paso anterior. Este proceso es repetido hasta que el nivel más fino en resolución proporciona una posición precisa del objeto en la imagen.

Entre las ventajas adicionales de las arquitecturas FPGAs se pueden mencionar: Se eliminan los tradicionales overheads de los procesadores de propósito general debido a la prebúsqueda y decodificación. Un diseño puede ser actualizado o mejorado sin reemplazar el hardware existente. Posibles errores de diseño no implican la fabricación nuevamente del hardware sino sólo la reprogramación.

La mayor desventaja que impide la adopción masiva de esta tecnología proviene de su dificultad de programación. Es necesario todavía varios pasos para que el tradicional código de programa pueda ser implementado en una FPGA. Se requieren lenguajes de descripción de hardware (VHDL, Verilog, etc.), herramientas de síntesis de hardware y herramientas de desarrollo de software para reconfigurar una FPGA. Este hecho supone un gran esfuerzo de aprendizaje para nuevos usuarios. Sin embargo, se han realizado algunos intentos de creación de compiladores para FPGAs, los cuales permiten configurar una FPGA directamente desde lenguajes de alto nivel como C++ o Java (Guccione, 1995; Guccione, 1997).

Otro importante inconveniente de los FPGAs es su actual imposibilidad para ser reconfigurados en tiempo de ejecución, ya que los FPGAs no han sido tradicionalmente concebidos para cumplir con este requerimiento. Sin embargo, están apareciendo recientemente nuevos diseños de FPGAs que combinan microprocesadores paralelos y reconfiguración parcial para posibilitar su reprogramación en tiempo de ejecución.

### A.3. Sistemas basados en DSP

Los DSPs han sido tradicionalmente los dispositivos más utilizados cuando se requiere aumentar la capacidad computacional disponible a costes moderados. Se han desarrollado numerosos sistemas de visión en ámbitos académicos y comerciales utilizando DSPs. Básicamente, poseen las características de los procesadores de propósito general ya que son tan flexibles como ellos para ser programados. Su diferencia radica en un aumento del número de unidades funcionales paralelas que mejoran sustancialmente el rendimiento.

Como parte de la plataforma experimental de esta tesis, se presenta a continuación una descripción detallada del hardware utilizado en el desarrollo del sistema DE-SEO. Este hardware consiste en una placa de desarrollo que integra un DSP TMS320C80 de Texas Instruments la cual lleva a cabo el proceso de seguimiento.

### A.3.1. Arquitectura del TMS320C80

El circuito integrado TMS320C80 es un procesador digital de señal diseñado por Texas Instruments (véase Figura A.2), el cual ofrece un gran rendimiento en tareas de proceso de imágenes. Entre sus componentes principales se incluyen:

- Cuatro DSP de 32-bits paralelos. Parallel Processor (PP).
- Un procesador RISC con unidad de coma flotante. Master Processor (MP).
- Dos controladores de video (VC).
- Una red de interconexión cruzada (crossbar switching network).
- Un controlador de transferencias (TC).
- 50KBytes de SRAM.

#### Los procesadores paralelos

Cada PP es un DSP de 32 bits avanzado con características especiales que mejoran el rendimiento en algoritmos de procesamiento de imágenes. Los cuatro PPs ofrecen la mayor parte de la capacidad de procesamiento del TMS320C80. Se encuentra dividido en tres unidades principales: unidad de datos, direccionamiento y control de flujo.

La concepción de unidad de datos es una de las características más destacadas de los procesadores paralelos (Texas Instruments, 1998b). Su flexibilidad permite manejar, procesar y direccionar varios datos en un solo ciclo de reloj. Ofrece la posibilidad de dividir la unidad de datos para procesar una palabra de 32 bits, dos enteros de 16 bits o cuatro bytes simultáneamente. Esta característica permite operaciones vectoriales sobre los datos. Debido a sus caminos de ejecución separados para el multiplicador y ALU de tres entradas, la unidad de datos es capaz de realizar multiplicaciones y operaciones en ALU simultáneamente en un ciclo de reloj. Cada PP posee un conjunto de 44 registros visibles, ocho de los cuales son de propósito general llamados registros D. Son también los registros primarios usados por la unidad de datos.



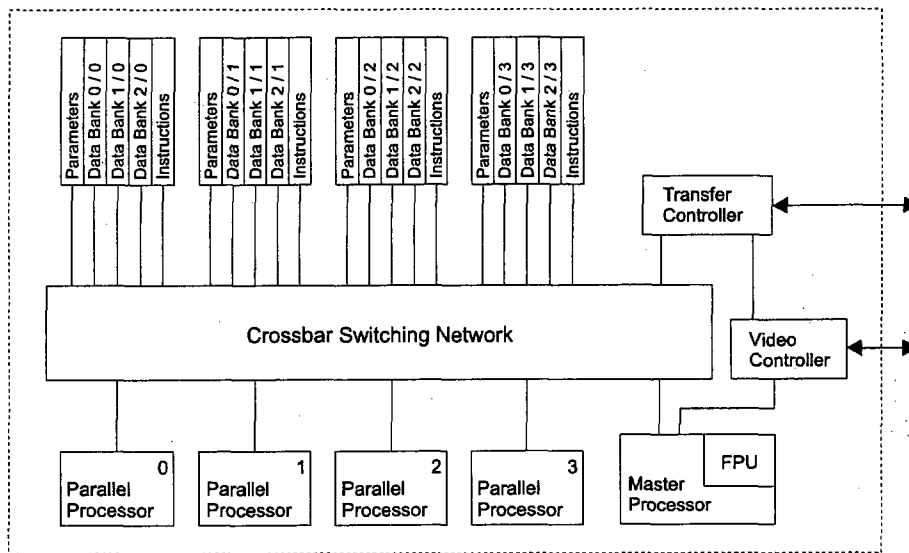


Figura A.2: Arquitectura del DSP de Texas Instruments TMS320C80. En la figura se puede apreciar los distintos elementos que incorpora: bancos de memoria, cuatro procesadores paralelos, procesador principal, red de interconexión, controlador de transferencias y controlador de vídeo.

Cada PP tiene dos unidades de direccionamiento, una unidad de direcciones globales y otra para direcciones locales. Son las encargadas de computar las direcciones para accesos a memoria y todos los movimientos de datos entre las diferentes unidades funcionales dentro del PP.

La unidad de control de flujo de programa realiza la búsqueda y decodificación de datos e instrucciones, controla el proceso de segmentación de las instrucciones y maneja las interrupciones para tareas de priorización.

### El Master Processor

El MP es un procesador segmentado de 32 bits con una unidad de punto flotante IEEE-754 (Texas Instruments, 1998a). La unidad de punto flotante, también segmentada, es capaz de comenzar una nueva instrucción de punto flotante en cada ciclo de reloj. Entre las funciones que posee el MP se encuentra la de controlar completamente cada uno de los PPs pudiendo inicializarlos, pararlos, reemplazar sus operaciones, etc.

### Controlador de vídeo

El TMS320C80 es un DSP orientado al procesamiento de imágenes, por esta razón incorpora un controlador de vídeo que es capaz de controlar dos subsistemas de

vídeo independientes (Texas Instruments, 1998*d*). Cada subsistema de vídeo consiste en una memoria de vídeo externa y una lógica de control.

### Red de interconexión (Crossbar Network)

Una red de interconexión de alta velocidad une estrechamente la memoria caché y los procesadores permitiendo un alto grado de paralelismo. El crossbar es autoconfigurable. Esto significa que cuando un procesador necesita acceder a un cierto banco de memoria de la caché, éste sólo tiene que direccionar el banco y la red conecta al procesador con el banco de la caché automáticamente. El crossbar tiene dos estados de segmentación, un acceso iniciado en un ciclo es finalizado dos ciclos después. Sin embargo, los accesos pueden ocurrir en cualquier ciclo. El crossbar permite hasta cinco búsquedas de instrucción y diez accesos paralelos por ciclo. Puede transferir hasta 1.8 GBytes/seg. de instrucciones y hasta 2.4 GBytes/seg. de datos. Un determinado banco de la caché puede ser accedido únicamente por un procesador en un determinado ciclo.

### Controlador de transferencias

Un controlador integrado de transferencias maneja las transferencias entre la memoria caché y memoria externa (memoria fuera del chip) (Texas Instruments, 1998*c*). Mover bloques de memoria desde la memoria externa hasta la caché proporciona una mayor rapidez de procesamiento de datos. Además, este controlador de transferencias está orientado al procesamiento de imágenes y puede ser programado para mover regiones rectangulares de imágenes, teniendo en cuenta las coordenadas de imagen y su direccionamiento lineal.

### Caché

Los 50 Kbytes de memoria caché están organizadas en 25 bancos de 2 Kbytes, permitiendo accesos simultáneos desde los procesadores. 32 de los 50 Kbytes de caché son compartidos entre los cinco procesadores, permitiendo una amplia variedad de configuraciones para algoritmos paralelos. No están permitidos los accesos simultáneos de dos procesadores a un mismo banco.

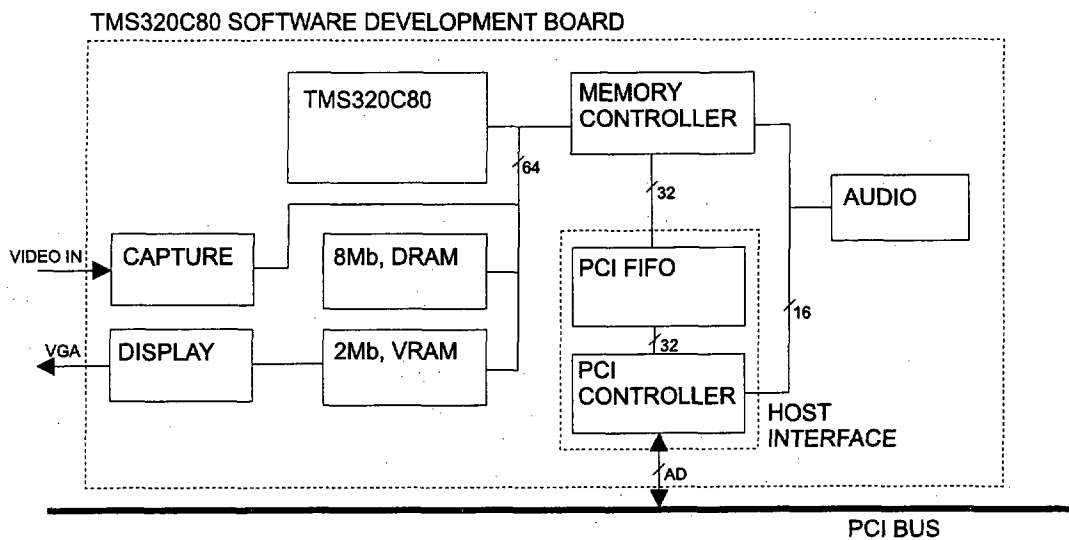


Figura A.3: Software Development Board, placa de desarrollo que integra el DSP TMS320C80 junto con diversos componentes más que permiten el desarrollo de aplicaciones sobre este DSP.

### A.3.2. Arquitectura de la placa de procesamiento Precision MX Video Engine

El chip TMS320C80 está contenido en una placa de desarrollo que posee más elementos útiles para el procesamiento digital de imágenes, esta placa es nombrada como *Precision MX Video Engine*. Además, la placa de desarrollo contiene un chip de video digitalizador/decodificador de Philips, AD 1848 Audio Codec, 8 MBytes de DRAM, 2 MB de VRAM y el TVP3020 Video Interface Palette (VIP), véase Figura A.3. Esta placa es conectada a una ranura PCI de un PC ejecutando Windows NT 4.0. Incluye una librería API para controlar el acceso a la placa. El MX Video Engine está dividido en seis bloques funcionales principales:

- TMS320C80
- Bloque de interface con el Host
- Controlador de memoria (MC) y DRAM/VRAM
- Bloque de captura de vídeo
- Bloque de visualización de vídeo
- Bloque de captura de audio y reproducción

## Host Interface Block

Las comunicaciones entre el MX Video Engine y su PC Host es llevada a cabo a través de este bloque funcional. La memoria principal, los registros y el propio TMS320C80 pueden ser mapeados en memoria y accedidos a través del interface PCI. Esta interface cumple con la especificación Bus Local PCI rev. 2.0 y 2.1. Además, soporta ciclos PCI esclavos y bus mastering. En ambos estados las transferencias en bloque son posibles sin estados de espera ofreciendo un pico de 132 MB/s. Cuando una transferencia tiene lugar entre el MX y el host todos los datos se encaminan a través del buffer FIFO. Este buffer está compuesto por 64 palabras de 32 bits. Existen realmente dos FIFOs, una para las transferencias entrantes y otra para las salientes desde o hacia el PC Host. Las transferencias simples en ambas direcciones son realizadas por dos mailboxes. Las transferencias de datos pueden ser realizadas mediante cinco formas. La tasa de transferencia varía dependiendo de la forma de transferencia elegida. Estas formas son:

- *Acceso al registro de interfaz PCI interno*, este es un método usado para leer y escribir en los registros de interface PCI del host.
- *Acceso al espacio de registro MX*, este método usa el bus IO para acceder a los diferentes registros de la placa MX. Esta comunicación es canalizada mediante los mailboxes.
- *Acceso a las transferencias de bloques del MX*, esta es la forma más general de realizar transferencias entre el MX y el PC Host. Es necesario tener un servidor corriendo en el MX para realizar el protocolo de transferencia. Este método es útil para implementar el paso de comandos y BLTs (transferencias de bloques). Para iniciar la transferencia de un bloque, la cabecera del BLT debe contener la siguiente información: Dirección del BLT, dirección de comienzo de la transferencia, tamaño de la transferencia y el pitch en el caso de transferencias de imágenes.
- *Bus mastering desde el MX a PCI*, este método permite al MX iniciar transferencias bus master hacia el host. La interface del host escribe datos a las direcciones PCI desde la FIFO.
- *Acceso al espacio de direcciones del MX entero*, permite al PC Host leer y escribir una palabra de 32 bits en cualquier dirección de la memoria principal del MX.

## Controlador de memoria

El controlador de memoria (MC) es el encargado de todas las comunicaciones entre el TMS320C80, la memoria principal, la memoria de vídeo y el resto de los componentes del MX. También se encarga de controlar las interrupciones y las condiciones de interrupción.

## Bloque de captura de vídeo

Este bloque realiza el proceso de captura de vídeo. El vídeo entrante es digitalizado por los convertidores 8-bit A/D TDA8708A y el TDA8709A. El decodificador SAA7196 decodifica y escala el vídeo y está directamente conectado a la FIFO de vídeo y el buffer de registro de 64-bits. Una vez una línea de vídeo es capturada, es sacada al bus de datos del MX para ser posteriormente ser procesada por el TMS320C80 y el controlador de memoria.

## Bloque de captura de audio y reproducción

La placa MX es también útil para propósitos multimedia. Por esta razón, la placa incorpora un bloque de captura y reproducción de audio ofreciendo la posibilidad de manejar y tratar audio.

## A.4. Procesadores de propósito general

Hasta hace poco tiempo, las estaciones de trabajo eran completamente ineficientes para llevar a cabo cualquier tarea de visión por computador en tiempo real. Sin embargo, a medida que los ordenadores de sobremesa y estaciones de trabajo basados en procesadores de propósito general comunes han ido incrementando su potencia, se han empezando a aplicar en cada vez más sofisticados algoritmos de visión. Obviamente, no alcanzan los índices obtenidos por los sistemas basados en ASICs pero su relación potencia vs. precio hace de ellos una alternativa cada vez más atractiva.

Los procesadores de propósito general han comenzado, desde hace algún tiempo, a incorporar potentes características como supersegmentación, caché de segundo nivel, predicción de saltos, estructuras superescalares, paralelismo a nivel de instrucción, y en especial nuevos conjuntos de instrucciones multimedia, etc. las cuales han mejorado enormemente su rendimiento. Sin embargo, todos estos avances no corresponden por

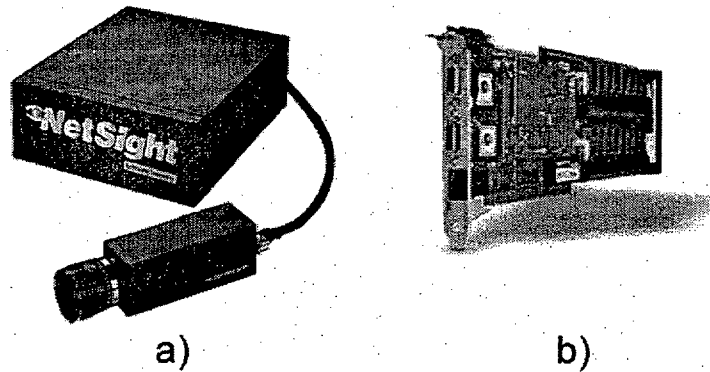


Figura A.4: NetSight, *Modular Vision Engine* desarrollado por Imaging Technology.

igual al desarrollo de compiladores que aprovechen estas mejoras. Los compiladores actuales no generan un código que aproveche estos nuevos conjuntos de instrucciones de estos procesadores. Para obtener altos rendimientos es necesario aún ayudar al compilador manualmente a organizar el código y, una vez hecho esto, puede ser esencial programar alguna sección del núcleo del algoritmo en ensamblador para sacar partido de los registros del procesador y sus extensiones multimedia. Si no se aplicara ninguna clase de ayuda manual a la optimización del código, el procesador podría funcionar por debajo de su capacidad real.

Entre todos los procesadores de propósito general actualmente disponibles en el mercado, La familia Pentium de Intel y por extensión Athlon de AMD parecen tener el futuro más prometedor. Se han desarrollado varios sistemas de visión basados en Pentium ofreciendo unos buenos resultados en tareas comunes de algoritmos de visión. *NetSight*, desarrollado por Imaging Technology es un ejemplo de estos, véase Figura A.4.

#### A.4.1. Familia Intel Pentium

A lo largo de la última década se han visto espectaculares crecimientos en la potencia de los económicos PCs domésticos. Estos ordenadores han seguido estrictamente la ley de Moore, la cual enuncia que:

*"La capacidad computacional y la complejidad (número de transistores por chip) de un microprocesador se dobla cada uno o dos años, y el coste por CPU se reduce a la mitad."*

Procesador Intel	Fecha de introducción del producto	Rendimiento (MIPs)	Número de transistores en el chip
8086	1978	0.8	29 K
Intel 286	1982	2.7	134 K
Intel386 DX	1985	6.0	275 K
Intel486 DX	1989	20	1.2 M
Pentium	1993	100	3.1 M
Pentium Pro	1995	440	5.5 M
Pentium II	1997	466	7 M
Pentium III	1999	1000	8.2 M
Pentium IV	2001	2000	24.2 M

Cuadro A.1: Progresión de la capacidad computacional de los procesadores de Intel

El interés general creciente por las aplicaciones multimedia en PCs ha hecho que esta clase de procesadores hayan aumentado enormemente sus capacidades multimedia y rendimiento, véase Tabla A.1. Desde 1997, con la introducción de la tecnología MMX (Multi-Media eXtensions), el número de mejoras al respecto han experimentado un crecimiento espectacular. Afortunadamente, esta clase de mejoras hardware son completamente aprovechables y directamente adaptables para tareas de visión.

Además de estas mejoras, el diseño de la familia de procesadores P6 (Pentium Pro, Pentium II, Pentium III y Pentium IV) se basa en una arquitectura superescalar de tres pistas. El término *superescalar de tres pistas* significa que usando técnicas de procesamiento paralelo el procesador es capaz, por término medio, de decodificar, ejecutar y finalizar tres instrucciones por ciclo de reloj. Para manejar este nivel de flujo de instrucciones, los procesadores de la familia P6 usan un arquitectura supersegmentada de doce estados desacoplados que soporta ejecuciones de instrucción fuera de orden. La figura A.6 muestra una vista conceptual de esta segmentación dividida en cuatro unidades de procesamiento (unidad de búsqueda/decodificación, unidad de ejecución, unidad de retiro y reserva de instrucciones). Instrucciones y datos son suministrados a estas unidades a través de su unidad de interface del bus. Para asegurar el flujo de instrucciones y datos hacia las unidades, la familia P6 incorpora caché de dos niveles. La caché L1 posee una memoria de instrucciones de 16 KBytes y de datos de otros 16 KBytes, ambos se encuentran estrechamente acoplados a la segmentación. La caché L2 posee 256 KBytes, 512 KBytes o 1 MByte de RAM estática que es conectada al procesador a través de un bus de caché de 64 bits.

Una extensa descripción de la arquitectura P6 de Intel se encuentra en (Intel Corporation, 1999a) y para un referencia corta e introductoria (Coelho y Hawash, 1998). A

continuación se comentan algunas de las características más destacables de la arquitectura de la familia P6 de Intel.

### MMX (Multi-Media eXtensions)

Con los procesadores Pentium, Intel introdujo una nueva aproximación a las aplicaciones multimedia, las cuales necesitaban una cantidad enorme de recursos computacionales. La introducción de un nuevo conjunto de ocho registros largos (nombrados como mm0 a mm7) e instrucciones que hacían uso de ellos representó un gran avance en el rendimiento de las aplicaciones multimedia. Básicamente, consiste en la paralelización de algunas operaciones que, de forma natural, se adaptan a un esquema SIMD (simple instrucción múltiples datos). Estos son ocho nuevos registros de 64 bits que, aunque su nombre sugiera que son de aplicación específica, son realmente de propósito general. Estos registros pueden ser divididos en ocho "subregistros" de ocho bytes, en cuatro palabras, en dos dobles palabras o tomarlo como un único registro de cuádruple palabra, y de esta forma pueden ser procesados en paralelo con otros registros mmx con la misma subdivisión lógica. Las nuevas instrucciones para manejar estos registros incluyen operaciones tales como sumas, restas, multiplicaciones, divisiones, operaciones lógicas, operaciones de empaquetamiento y desempaquetamiento, etc. Con la introducción del Pentium III se añadió una nueva y útil instrucción para su aplicación en este trabajo: la suma de diferencias absolutas, cuyo nemónico es "psadbw". Esta instrucción se encuentra entre el repertorio ya que es muy empleada en algoritmos de codificación de vídeo mpeg. La Figura A.5 muestra la operación aplicada sobre ocho bytes, que pueden corresponder a ocho pixels de una imagen en escala de grises y que son almacenados en un registro mmx, en este caso mm0. Otros ocho bytes están contenidos en el registro mm1. Se realiza una resta individual sobre cada par de bytes y posteriormente se toma su valor absoluto. Finalmente todos estos resultados parciales son sumados y almacenados de nuevo en mm0.

### Microarquitectura de ejecución dinámica de la familia P6

La familia P6 posee varias características de funcionamiento que optimizan el rendimiento del microprocesador intentando mantener todas sus etapas siempre ocupadas.

- Predicción múltiple de saltos: predice la ejecución del programa a través de múltiples saltos, acelerando el flujo de trabajo del microprocesador.



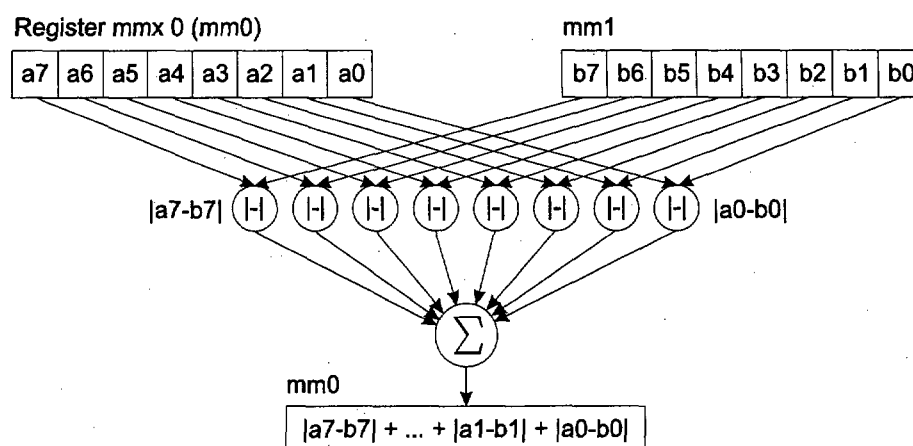


Figura A.5: El nemónimo "psadbw" corresponde a una nueva instrucción introducida con el Pentium III, la cual corresponde a la suma de diferencias absolutas.

- Análisis del flujo de datos: esquema de reordenación de instrucciones mediante el análisis de dependencias de datos entre instrucciones.
- Ejecución especulativa: lleva a cabo la ejecución especulativa de instrucciones y, basada en este esquema de optimización, asegura que la unidad de ejecución superescalar permanece siempre ocupada, incrementando el rendimiento.

### Bus independiente dual (DIB)

El procesador Pentium III soporta la arquitectura de bus independiente dual de alto rendimiento. La arquitectura DIB coloca la caché de nivel 2 en un bus dedicado de alta velocidad, liberando al bus del sistema del tráfico de la caché. Esto proporciona un significativo incremento en el ancho de banda global y permite grandes mejoras en el rendimiento global y escalabilidad.

### Caché de nivel 1 no bloqueante

El procesador Pentium® III incluye dos cachés de nivel 1 de 16 Kb. uno para instrucciones y otro para datos. La caché de nivel 1 proporciona accesos rápidos a datos usados recientemente, incrementando el rendimiento global.

### Caché de transferencia avanzada de nivel 2 de 256 KB

La caché de transferencia avanzada (ATC) consiste en mejoras en la microarquitectura para proporcionar un interface de datos de mayor ancho de banda entre la caché

de nivel 2 y el núcleo del microprocesador el cual es completamente escalable con la frecuencia del núcleo del procesador. Entre las características del ATC se incluyen:

- Caché de nivel 2 no bloqueante, de velocidad completa y contenida en el mismo chip.
- 8-way set associativity.
- Bus de datos de 256 bits hacia la caché de nivel 2.
- Interface de latencia reducida a los datos de caché.

### **Caché de nivel 2 no bloqueante**

Ciertas versiones del procesador Pentium® III incluyen una caché de nivel 2 (L2) fuera del chip. Esta caché L2 consta de 512 Kb. unificados y no bloqueantes que mejoran el rendimiento con respecto a las soluciones que implantan la caché directamente sobre la placa madre.

### **Sistema de buffering avanzado**

El sistema de buffering avanzado consiste en optimizaciones de los tamaños de buffer del bus del sistema que resultan en un incremento en la utilización del ancho de banda disponible en el bus del sistema de 100 y 133 MHz.

# Apéndice E

## Soporte Experimental: WinVision

El soporte experimental para la medición del rendimiento y precisión de los diferentes algoritmos estudiados ha sido llevado a cabo mediante la aplicación WinVision. WinVision es un software desarrollado específicamente para esta tesis, siendo altamente útil para el control de toda clase de experimentos on-line y off-line. Las aplicaciones que utilizan escenarios en interiores pueden ser analizadas en vivo, mientras que las aplicaciones que tienen lugar en exteriores pueden ser estudiadas off-line con vídeo previamente grabado. Está soportado por la tecnología DirectShow de Microsoft, la cual permite trabajar con flujos de vídeo de una forma cómoda y rápida. WinVision ha sido desarrollado utilizando Microsoft Visual C++ 6.0 y las librerías de DirectMedia 6.0, las cuales incluyen las librerías de DirectShow. La aplicación se ejecuta sobre PC bajo Microsoft Windows 2000 Professional. Actualmente, WinVision está formado por un filtro de transformación programable de DirectShow que ejecuta los algoritmos de seguimiento a estudiar.

### E.0.2. Microsoft DirectShow

Los servicios de DirectShow de Microsoft ofrecen la posibilidad de tratar flujos de información multimedia proveniente de dispositivos de captura, como cámaras digitales, así como de ficheros locales o servidores de internet. Específicamente, esto hace posible la reproducción de vídeo y audio comprimido en diferentes formatos, incluyendo AVI, MPEG, QuickTime, etc.

En el interior de los servicios de DirectShow se halla un sistema modular de componentes conectables llamados filtros, los cuales se ordenan en una configuración particular para cada necesidad llamada grafo de filtros. Un componente denominado

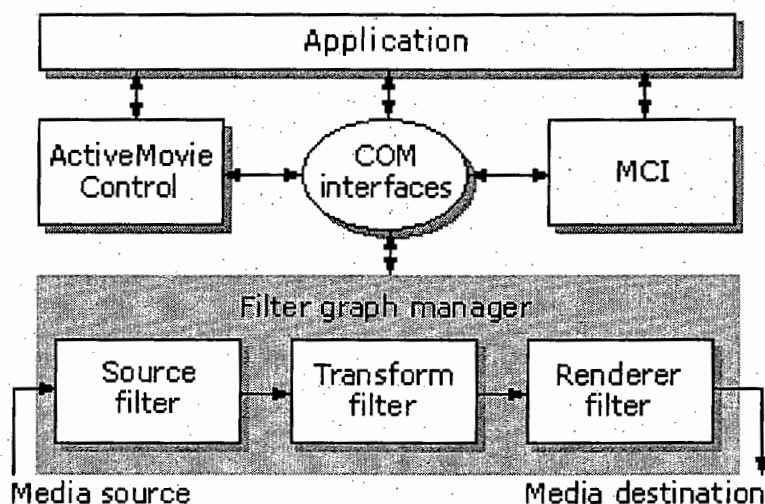


Figura E.1: El manejador del grafo de filtros controla el flujo de datos, lo cual significa comenzar, pausar o parar la reproducción del flujo. También permite controlar la reproducción con una duración determinada o la búsqueda de un determinado punto en el flujo. El manejador del grafo de filtros permite a la aplicación especificar estas acciones, y en consecuencia hacer las llamadas apropiadas a los métodos de los filtros. Así mismo, permite a los filtros mandar eventos hacia las aplicaciones. De esta forma, una aplicación puede, por ejemplo, obtener el estado de un filtro en concreto que ha instalado.

manejador del grafo de filtros configura la conexión entre estos filtros y controla el flujo de datos, véase Figura E.1. Ofrece también un conjunto de interfaces COMs (Component Object Model) para que las aplicaciones puedan acceder al grafo de filtros, controlar el flujo y capturar los eventos de los filtros.

Los dos componentes básicos utilizados en la arquitectura del DirectShow son los filtros y los pines. Un filtro realiza una tarea específica, tal como leer datos de un disco o de una cámara, transformar el flujo de datos o almacenar el flujo de datos en disco o visualizarlo en pantalla. Para cada flujo de datos que maneja se habilita al menos un pin. Un pin es un objeto creado por un filtro y que representa un punto de conexión unidireccional para el flujo de datos, Figura E.2.

Los pines de entrada aceptan datos hacia en filtro, y los filtros de salida proporcionan datos a otros filtros. Un filtro fuente proporciona un pin de salida para cada flujo de datos que contenga el fichero. Un filtro de transformación típico, tal como un filtro de compresión/descompresión (codec), proporciona un pin de entrada y otro de salida. Configuraciones más complejas también son posibles:



Figura E.2: La figura muestra la disposición de los diferentes filtros más comunes en un grafo de filtros. Un filtro fuente, el cual toma sus datos de ficheros en disco, de servidores de internet o de reproductores de vídeo. Un filtro de transformación, el cual toma datos de algún otro filtro, los procesa y los envía por su pin de salida. Y por último un filtro final, encargado normalmente de visualizar el contenido del flujo de datos en pantalla (en el caso de un filtro de render) o almacenarlos en un fichero local.

### E.0.3. Filtro WinVision

La Figura E.3 muestra el filtro WinVision ubicado casi al final de la cadena, normalmente antes del filtro de render o visualización. Esto es así debido a que el filtro opera sobre vídeo sin comprimir, es decir RGB (24 bits por pixel ó 1 byte por canal). En las secuencias de vídeo pregrabado es necesario incluir un filtro con el correspondiente codec asociado al tipo de fichero, en secuencias en vivo puede ser necesario algún filtro de conversión YUV a RGB. Una vez realizado el procesamiento dentro del filtro, la imagen resultante, la cual sigue estando en componentes RGB (24 bits por pixel), es enviada mediante el pin de salida del filtro hacia un filtro de render, almacenamiento en disco o algún filtro de multiplexación si se desean realizar varias operaciones simultáneas sobre la imagen.

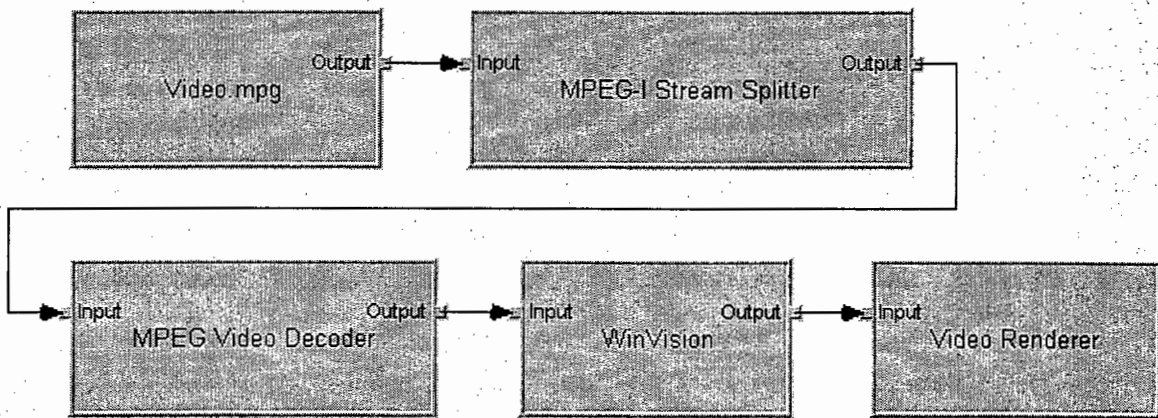


Figura E.3: La figura muestra un ejemplo de una cadena de filtros que toma un fichero del disco duro, realiza algún tipo de computación sobre la secuencia del fichero y posteriormente lo visualiza en pantalla. El primer bloque es la fuente (video.mpg), que es un fichero codificado en mpeg-1 almacenado en el disco duro. El segundo filtro divide vídeo y audio en dos flujos independientes, aunque en este caso el fichero de la secuencia no contenga ninguna banda de audio. El tercer filtro decodifica el mpeg-1 en sucesivos cuadros sin comprimir los cuales toma el cuarto filtro (WinVision) para llevar a cabo su algoritmo de búsqueda de objetos de interés en la imagen. Finalmente, un quinto filtro (Video Renderer) muestra el resultado en pantalla.

# Bibliography

- Abidi, M. A. y Gonzalez, R. C. (1992). *Data Fusion in Robotics and Machine Intelligence*. Academic Press, Inc.
- Allen, P., Yoshimi, B. y Timcenko, A. (1994). 'Hand-eye coordination for robotics tracking and grasping.' *Visual Servoing. World Scientific*, págs. 33-70.
- Aloimonos, J. y Weiss, I. (1988). 'Active vision.' *International Journal of Computer Vision*, págs. 333-356.
- Andersson, M. (1994). *Tracking Methods in Computer Vision*. Proyecto Fin de Carrera, Computational Vision and Active Perception Laboratory (CVAP).
- Arbeiter, J. y Bessler, R. (1986). 'A two-dimensional real-time video pyramid.' *RCA Review*, 1, págs. 3-31.
- Balkenius, C. y Kopp, L. (1997). 'Elastic template matching as a basis for visual landmark recognition and spatial navigation.' Informe Técnico UMCS-97-4-1. <http://www.cs.man.ac.uk/csonly/cstechrep/titles97.html>, Department of Computer Science, Manchester University.
- Barattin, M., Cucchiara, R. y Piccardi, M. (1998). 'A rule-based vehicular traffic tracking system.' En 'CVPRIP'98, First International Workshop on Computer Vision, Pattern Recognition and Image Processing,' .
- Barnea, D. I. y Silverman, H. F. (1972). 'A class of algorithms for fast digital image registration.' *IEEE Trans. Computers*.
- Bartlett, M. y Sejnowski, T. (1997). 'Independent component of face images: a representation for face recognition.' En 'Procs. of the Annual Joint Symposium on Neural Computation,' Pasadena, CA.
- Beauchemin, S. S. y Barron, J. L. (1995). 'The computation of optical flow.' *ACM Computing Surveys*, 27(3), págs. 433-467.

- Bernardino, A. y Santos-Victor, J. (1999). 'Binocular tracking: Integrating perception and control.' *IEEE Transactions on Robotics and Automation*, **15**.
- Biederman, I. (1987). 'Recognition by components: A theory of human image understanding.' *Psychol. Review*, **94**, págs. 115-147.
- Blakenius, C. y Kopp, L. (1996). 'The xt-1 vision architecture.' *Linde, P., Sparr, G (eds)*.
- Bolles, R. C. y Cain, R. A. (1982). 'Recognizing and locating partially visible objects: The local feature focus method.' *International Journal of Robotics Research*, **1**, págs. 57-82.
- Brooks, R. A. (1990). 'Elephants don't play chess.' *Robotics and Autonomous Systems*, **(6)**, págs. 3-15.
- Brown, L. G. (1992). 'A survey of image registration techniques.' *ACM Computing Surveys*, **24(4)**, págs. 325-376.
- Campani, M. y Verri, A. (1992). 'Motion analysis from first-order properties of optical flow.' págs. 56(1):90-107. CVGIP: Image understanding.
- Canny, J. F. (1986). 'A computational approach to edge detection.' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6(8)**, págs. 679-698.
- Chan-Hung, Y.-S. C. (2000). 'Video-based realtime eye tracking technique for autostereoscopic displays.' págs. 188-193. Fifth Conference Artificial Intelligence and Applications.
- Chen, L., Fujishiro, I. y Peng, Q. (2000). 'Fast LIC image generation based on significance map.' *Lecture Notes in Computer Science*, págs. 537-546.
- Choe, Y. (2002). 'Second order isomorphism: A reinterpretation and its implications in brain and cognitive sciences.' En 'Proceedings of the 24th Annual Conference of the Cognitive Science Society,' págs. 6 páginas, en imprenta. Erlbaum.
- Clark, J. C., Carlsson, S. y Zisserman, A. (1996). 'Detecting and tracking linear features efficiently.' En 'Proc. of the British Machine Vision Conference,'
- Clark, J. J. y Yuille, A. L. (1990). *Data Fusion for Sensory Information Processing Systems*. Kluwer Academic Publishers.



- Coelho, R. y Hawash, M. (1998). *DirectX, RDX, RSX, and MMX Technology: A Jumpstart Guide to High Performance APIs*. Addison Wesley Developers Press.
- Collins, R., Lipton, A., y Kanade, T. (1999). 'A system for video surveillance and monitoring.' In Proc. American Nuclear Society (ANS) Eighth International Topical Meeting on Robotics and Remote Systems.
- Coradeschi, S. y Saffiotti, A. (2000). 'Perceptual anchoring of symbols for action.' En 'Procs. of the 17th Int. Conf. on Artificial Intelligence (IJCAI-01).'
- Cox, G. (1995). 'Template matching and measures of match in image processing.'
- Crowley, J. L. y Bedrune, J. M. (1994). 'Integration and control of reactive processes.' págs. 47-58. Proc. ECCV'94, Springer-Verlag.
- Darrell, T., Moghaddam, B. y Pentland, A. (1996). 'Active face tracking and pose estimation in an interactive room.' págs. 67-72. Proc. IEEE Conf. on Computer Vision and Pattern Recognition.
- Deriche, R. y Giraudon, G. (1993). 'A computational approach for corner and vertex detection.' *IJCV*, 10(2), págs. 101-124.
- Doi, M. y Hashimoto, S. (2001). 'Recognition environment for a mobile robot using omni-directional image sensing and stereophonic distance measurement.' IEEE-RAS International Conference on Humanoid Robotics.
- Domínguez, A. C., Andersson, M. y Christensen, H. I. (2000a). 'A software architecture for programming robotic systems based on the discrete event systems paradigm.' Informe técnico, ISRN KTH/NA/P-00/13-SE. Centre for Autonomous Systems, KTH (Royal Institute of Technology).
- Domínguez, A. C., Hernández, F. M. y Cabrera, J. (1999). 'A control architecture for active vision systems.' págs. 395-404. Proc. of the VIII National Symposium on Pattern Recognition and Image Analysis.
- Domínguez, A. C., Hernández, F. M. y Cabrera, J. (2000b). 'A control architecture for active vision systems.' págs. 144-153. *Frontiers in Artificial Intelligence and Applications: Pattern Recognition and Applications*. M. I. Torres and A. Sanfeliu, IOS Press.
- Edelman, S. (1997). 'Computational theories of object recognition.' *Trends in Cognitive Sciences*.

- Edelman, S. (1998). 'Representation is representation of similarities.' *Behavioral and Brain Sciences*, **21**, págs. 449-498.
- Edelman, S. (1999). *Representation and Recognition in Vision*. The MIT Press.
- Edelman, S., Cutzu, F. y Duvdevani-Bar, S. (1996). 'Similarity to reference shapes as a basis for shape representation.' In Cottrell, G., editor, *Proceedings of COGSCI'96*, San Diego, CA.
- Edelman, S. y Weinshall, D. (1991). 'A self-organizing multipleview representation of 3d objects.' *Biological Cybernetics*, **64**, págs. 209-219.
- Faugeras, O. (1993). *Three-Dimensional Computer Vision : A Geometric Viewpoint*. MIT Press.
- Gil, S., Milanese, R. y Pun, T. (1994). 'Feature selection for object tracking in traffic scenes.' págs. 253-266. Conf. SPIE, Photonic Sensors and Controls for Commercial Applications - Intelligent Vehicle Highway Systems.
- Greco, A., Cangelosi, A. y Harnard, S. (1998). 'A connectionist model for categorical perception and symbol grounding.' En 'Proceedings of the Eighth International Conference on Artificial Neuronal Networks,' págs. 399-404.
- Guccione, S. A. (1995). *Programming Fine-Grained Reconfigurable Architectures*. Proyecto Fin de Carrera, University of Texas at Austin.
- Guccione, S. A. (1997). 'Software for reconfigurable computing.' *Embedded Systems Programming (Europe)*, **1**(2), págs. 39-41.
- Guerra, C. (1996). 'Sensores espacio variantes.' Informe Técnico GIAS Technical Report, GIASTR-0396, GIAS. Universidad de Las Palmas de Gran Canaria, <http://mozart.dis.ulpgc.es/Publications/publications.html>.
- Guerra, C., Castrillón, M., Cabrera, J. y Hernández, M. (1998a). 'An active vision system integrating fast and slow processes.' págs. 487-496. SPIE'98 Symposium on Intelligent Systems and Advanced Manufacturing.
- Guerra, C., Castrillón, M., Cabrera, J. y Hernández, M. (1998b). 'Real-time tracking system using c80 dsps and a binocular robotic head.' págs. 440-444. IASTED International Conference on Signal and Image Processing (SIP'98).
- Hagedoorn, M. (2000). *Pattern Matching using Similarity Measures*. Tesis Doctoral, University of Utrecht.

- Hager, G. D. y Belhumeur, P. (1996). 'Real-time tracking of image regions with changes in geometry and illumination.' págs. 403–410. IEEE Conf. on Computer Vision and Pattern Recognition.
- Hall, D. y Crowley, J. L. (1999). 'Tracking fingers and hands with a rigid contour model in an augmented reality.' 1st International Workshop on Managing Interactions in Smart Environments.
- Harnad, S. (1990). 'The symbol grounding problem.' *Physica D*, (42), págs. 335–346.
- Hernández, F. M. (1987). *Reconocimiento de Formas Tridimensionales Mediante Categorización Global de Vistas*. Tesis Doctoral, Universidad Politécnica de Canarias.
- Hernández, M., Cabrera, J. y Guerra, C. (1999). 'Deseo: An active vision system for detection, tracking and recognition.' págs. 376–391. International Conference on Vision Systems ICVS'99.
- Hummel, J. E. (2000). 'Where view-based theories break down: The role of structured in shape perception and object recognition.' Informe técnico, University of California, Los Angeles.
- Hutchinson, S., Hager, G. D. y Corke., P. I. (1996). 'A tutorial on visual servo control.' *IEEE Trans. Robot. Automat.*, 12(5), págs. 651–670.
- Illingworth, J. y Kittler, J. (1987). 'A survey of efficient hough transform methods.' En 'Third Alvey Vision Conference, Cambridge,' págs. 319–326.
- Inoue, H., Tachikawa, T. y Inaba, M. (1992). 'Robot vision system with a correlation chip for real-time tracking, optical flow and depth map generation.' págs. 1621–1629. IEEE International Conference on Robotics and Automation.
- Intel Corporation (1999a). *The Intel Architecture Software Developers Manual, Volume 1: Basic Architecture (Order Number 243190)*. Intel Corporation.
- Intel Corporation (1999b). *The Intel Architecture Software Developers Manual, Volume 2: Instruction Set (Order Number Mirar el numero)*. Intel Corporation.
- Isard, M. y Blake, A. (1998). 'Condensation - conditional density propagation for visual tracking.' *International Journal of Computer Vision*, 29(1), págs. 5–28.
- Jones, S. D. y Crowley, J. L. (1995). 'Robust task achievement.' Symposium for Intelligent Robotic Systems, Pisa, Italy.

- Joshi, R. y Sanderson, A. C. (1999). *Multisensor Fusion: A Minimal Representation Framework*. World Scientific Publishing.
- Jurie, F. y Dhome, M. (2001). 'A simple and efficient template matching algorithm.' págs. 544-549. Int. Conference on Computer Vision and Pattern Recognition.
- Jurie, F. y Dhome, M. (2002). 'Hyperplane approximation for template matching.' *IEEE Pattern Analysis and Machine Intelligence*, **24**(7), págs. 996-1000.
- Kahn, R. E., Swain, M. J., Prokopowicz, P.Ñ. y Firby, R. J. (1996). 'Gesture recognition using the perseus architecture.' Proc. CVPR'96.
- Kalman, R. E. (1960). 'A new approach to linear filtering and prediction problems.' *Transaction of the ASME-Journal of Basic Engineering*, págs. 35-45.
- Kass, M., Witkin, A. y Terzopoulos (1987). 'Snakes: Active contour models.' *International Journal of Computer Vision*, **1**(4), págs. 133-144.
- Kishore, M. S. y Rao, K. V. (2000). 'A study of correlation technique on pyramid processed images.' *Sadhana*, **25**, págs. 37-43.
- Kishore, M. S. y Rao, K. V. (2001). 'Robust correlation tracker.' *Sadhana*, **26**, págs. 227-236.
- Kragić, D. y Christensen, H. I. (2000). 'Tracking techniques for visual servoing tasks.' *IEEE International Conference on Robotics and Automation*, **2**, págs. 1663-1669.
- Krotkov, E. (1989). *Active Computer Vision by Cooperative Focus and Stereo*. Springer-Verlag.
- Loy, G., Holden, E. y Owens, R. (2000). 'A 3d head tracking for an automatic lipreading system.' Proceedings of Australian Conference on Robotics and Automation (ACRA2000).
- Loy, G., Holden, E. y Owens, R. (2001). 'A 3d head tracking for an automatic lipreading system.'
- Mae, Y., Yamamoto, S., Shirai, Y. y Miura, J. (1996). 'Optical flow based realtime object tracking by active vision system.' Proc. 2nd Japan-France Congress on Pattern Recognition.
- Marr, D. (1982). *Vision*. W.H.Freeman and Company.

- Martin, J. y Crowley, J. L. (1995). 'Comparison of correlation techniques.' Conference on Intelligent Autonomous Systems, IAS'95.
- Moghaddam, B. (1999). 'Principal manifolds and bayesian subspaces for visual recognition.' Informe Técnico TR-99-35, Mitsubishi Electric Research Laboratory, 201 Broadway, Cambridge MA 02139, USA.
- Moravec, H. P. (1977). 'Towards automatic visual obstacle avoidance.' *5th IJCAI, Cambridge*.
- Murase, H. y Nayar, S. (1993). 'Learning and recognition of 3d objects from appearance.' IEEE 2nd Qualitative Vision Workshop.
- Murase, H. y Nayar, S. (1995). 'Visual learning and recognition of 3d objects from appearance.' *IJCV*, **14**, págs. 5-24.
- Murase, H., Nayar, S. y Nene, S. (1996). *Parametric Appearance Representation*. Early Visual Learning.
- Murray, D. W., Bradshaw, K. J., McLauchlan, P. F., Reid, I. D. y Sharkey, M. (1995). 'Driving saccade to pursuit using image motion.' *International Journal of Computer Vision*, págs. 16(3):205-228.
- Murray, D. W., Reid, I. D. y Davison, A. J. (1996). 'Steering without representation using active fixation.' British Machine Vision Conference.
- Newell, A. (1980). 'Physical symbol systems.' *Cognitive Science*, (4), págs. 135-183.
- O'Hagan, R. y Zelinsky, A. (1997). 'Finger track - a robust and real-time gesture interface.'
- Panerai, F., Capurro, C. y Sandini, G. (1995). 'Space variant vision for an active camera mount.' Informe Técnico TR 1/95, LIRA-lab - DIST University of Genova, Italy.
- Parhami, B. (1994). 'Voting algorithms.' *IEEE Transactions on Reliability*, **43**(3), págs. 617-629.
- Peers, S. M. C. (1998). 'Blackboard system approach to electromagnetic sensor data interpretation.' *Expert Systems*, **15**(3), págs. 197-215.
- Pentland, S. y Choudhury, T. (2000). 'Face recognition for smart environments.' *IEEE Computer*, **33**(2), págs. 50-55.

- Pfeifer, R. y Scheier, C. (1999). *Understanding Intelligence*. MIT Press.
- Poggio, T. y Edelman, S. (1990). 'A network that learns to recognize three-dimensional objects.' *Nature*, **343**, págs. 263–266.
- Pratt, W. K. (1974). 'Correlation techniques of image registration.' *IEEE Transactions on Aerospace Electronic Systems*, *AES-10*, págs. 353–350.
- Pudipeddi, B., Abbott, A. L. y Athanas, P. M. (1997). 'Real-time hierarchical visual tracking using a configurable computing machine.' *IEEE*, págs. 153–157.
- Rao, V. y Rajes, P. (1997). 'Dynamic appearance-based recognition.' págs. 540–546. Proc. of CVPR'97.
- Ratha, N. K. y Jain, A. K. (1999). 'Computer vision algorithms on reconfigurable logic arrays.' *IEEE Transactions on Parallel and Distributed Systems*, **10**(1), págs. 29–43.
- Reynolds, J. (1998). *Autonomous Underwater Vehicle: Vision System*. Tesis Doctoral, Robotic Systems Lab. Department of Engineering. Australian National University Canberra. Australia.
- Russell, R. y Sinha, P. (2001). 'Perceptually-based comparison of image similarity metrics.' Informe Técnico 2001-014, MIT - Artificial Intelligence Laboratory.
- Sawasaki, N., Morita, T. y Uchiyama, T. (1996). 'Design and implementation of high-speed visual tracking system for real-time motion analysis.' págs. 478–483. IEEE ICPR.
- Scaroff, S. y Isidoro, J. (1998). 'Active blobs.' En 'Proc. International Conference on Computer Vision,' .
- Shepard, R.Ñ. y Chipman, S. (1970). 'Second-order isomorphism of internal representations: Shapes of states.' *Cognitive Psychology*, **1**, págs. 1–17.
- Shi, J. y Tomasi, C. (1994). 'Good features to track.' En 'IEEE Conference on Computer Vision and Pattern Recognition, (CVPR94),' Seattle.
- Sánchez-Nielsen, E. y Hernández, M. (2000). 'Tracking moving objects using the hausdorff distance. a method and experiments.' págs. 164–172. *Frontiers in Artificial Intelligence and Applications: Pattern Recognition and Applications*, IOS Press.
- Spath, H. (1980). *Cluster Analysis Algorithms: For Data Reduction and Classification of Objects*. Ellis Horwood Limited.

- Spengler, M. y Schiele, B. (2001). 'Towards robust multi-cue integration for visual tracking.' *Lecture Notes in Computer Science*, **2095**.
- Stiefelhagen, R., Meier, U. y Yang, J. (1997). 'Real-time lip-tracking for lipreading.' En 'Proc. Eurospeech '97,' págs. 2007-2010. Rhodes, Greece.
- Stillman, S., Tanawongsuwan, R. y Essa, I. (1998). 'A system for tracking and recognizing multiple people with multiple cameras.' Informe Técnico GIT-GVU-98-25. <http://www.gvu.gatech.edu/perception/projects/smartspaces>, Georgia Institute of Technology, Graphics, Visualization and Usability Center.
- Sun, R. (2000). 'Symbol grounding: A new look at an old idea.' *Philosophical Psychology*, **13**(3), págs. 149-172.
- Swain, M. J. y Ballard, D. H. (1991). 'Color indexing.' *International Journal of Computer Vision*, **7**(1), págs. 11-32.
- Tagare, D. H. (1997). 'Deformable 2-d template matching using orthogonal curves.' *IEEE Transactions on Medical Imaging*, **16**(1), págs. 108-117.
- Texas Instruments (1998a). *Master Processor User's Guide*. Texas Instruments.
- Texas Instruments (1998b). *Parallel Processor User's Guide*. Texas Instruments.
- Texas Instruments (1998c). *Transfer Controller User's Guide*. Texas Instruments.
- Texas Instruments (1998d). *Video Controller User's Guide*. Texas Instruments.
- Toyama, K. (1998). 'Prolegomena for robust face tracking.' Informe Técnico MSR-TR-98-65, Microsoft Research.
- Triesch, J. (2000). 'Democratic integration: A theory of adaptive sensory integration.' Informe Técnico NRL00.1, University of Rochester. Computer Science Department. New York.
- Triesch, J. y von der Malsburg, C. (2000). 'Self-organized integration of visual cues for face tracking.' En 'Proceedings of the Fourth International Conference on Automatic Face and Gesture Recognition,' págs. 102-107. Grenoble, France.
- Turk, M. y Pentland, A. (1991a). 'Eigenfaces for recognition.' *Journal of Cognitive Neuroscience*, **3**(1), págs. 71-86.
- Turk, M. y Pentland, A. (1991b). 'Face recognition using eigenfaces.' *IEEE Computer Vision and Pattern Recognition*, **3**(1), págs. 586-591.

- Ullman, S. (1984). 'Visual routines.' *Cognition*, 18, págs. 97-159.
- Ullman, S. (1987). 'Visual routines.' *Readings in Computer Vision*, M.A. Fischer and O. Fiirschein (Eds.), págs. 298-327.
- Ullman, S. (1989). 'Aligning pictorial descriptions: An approach to object recognition.' *Cognition*, (32), págs. 193-254.
- Ullman, S. y Basri, R. (1991). 'Recognition by linear combination of models.' *IEEE Trans. on Pattern Analysis and Machine Intelligence*, (13), págs. 992-1006.
- Venot, A., Lebruchec, J. F. y Roucayrol, J. C. (1984). 'A new class of similarity measures for robust image registration.' *Computer Vision, Graphics, and Image Processing*, 28(2), págs. 176-184.
- Vogt, P. (2000). *Lexicon Grounding on Mobile Robots*. Tesis Doctoral, Vrije Universiteit Brussel.
- Vogt, P. (2002). 'The physical symbol grounding problem.' *Cognitive Systems Research Journal*, 3(3), págs. 429-457.

