

TESIS DOCTORAL

**MODELOS DE INTERACCIÓN LATERAL
EN COMPUTACIÓN ACUMULATIVA PARA
LA OBTENCIÓN DE SILUETAS**

Antonio FERNÁNDEZ CABALLERO

Licenciado en Informática por la Universidad Politécnica de Madrid

DEPARTAMENTO DE INTELIGENCIA ARTIFICIAL

FACULTAD DE CIENCIAS

UNIVERSIDAD NACIONAL DE EDUCACION A DISTANCIA

Año 2.001

DEPARTAMENTO DE INTELIGENCIA ARTIFICIAL
FACULTAD DE CIENCIAS
UNIVERSIDAD NACIONAL DE EDUCACION A DISTANCIA

**MODELOS DE INTERACCIÓN LATERAL
EN COMPUTACIÓN ACUMULATIVA PARA
LA OBTENCIÓN DE SILUETAS**

AUTOR:

Antonio FERNÁNDEZ CABALLERO

Licenciado en Informática por la Universidad Politécnica de Madrid

DIRECTOR:

Dr. José MIRA MIRA

Universidad Nacional de Educación a Distancia

CO-DIRECTOR:

Dr. Miguel Angel FERNÁNDEZ GRACIANI

Universidad de Castilla-La Mancha

Agradecimientos

A mis tres mujeres, Ana, Alicia y Patricia, de las que he podido disfrutar un poco menos en estos años de tesis.

A mis codirectores José Mira Mira y Miguel Angel Fernández Graciani por su dedicación y apoyo constante y por la generosidad que me han demostrado al dejarme usar trabajos suyos como puntos de partida y complemento de los trabajos que yo he desarrollado en esta tesis.

A mis compañeros de trabajo del *Departamento de Informática* de la *Escuela Politécnica Superior de Albacete* de la *Universidad de Castilla-La Mancha* y en especial a mis amigos del *Grupo de Investigación en Computación Neuronal e Inteligencia Artificial*.

A Maite López Bonal por su siempre presente sonrisa y buen ánimo. A Francisco Javier Gómez Quesada por sus incipientes comentarios. Gracias también por su valiosísima herramienta de tratamiento de imágenes que tantas horas de trabajo nos han costado.

A Modesto Montoya Santos por su gran labor de programación en las herramientas anexas a la tesis. A Juan Diego Delicado Hernández por sus rápidas pulsaciones al ir pasando los sucesivos borradores hasta llegar a la presente versión escrita. A ambos igualmente el mérito de la filmación en interiores y exteriores de las secuencias de imágenes reales usadas en esta memoria. A Gabriel Sebastián por su valiosa y desinteresada ayuda en la obtención de las imágenes sintéticas, dada su gran experiencia en Director.

También a mis compañeros del *Departamento de Inteligencia Artificial* de la *Facultad de Ciencias* de la *Universidad Nacional de Educación a Distancia*.

Indice

Lista de símbolos	ix
Lista de figuras	xi
Lista de tablas	xix
Lista de ejemplos	xxi
Objetivos y resumen de la tesis	xxiii

Capítulo 1. Introducción y estado del arte

1.1. Planteamiento del problema: Extracción de las siluetas de los objetos deformables en secuencias de imágenes	1
1.1.1. Introducción al análisis del movimiento	1
1.1.2. Estimación del movimiento y seguimiento	5
1.1.3. Planteamiento del problema	6
1.2. Análisis del movimiento en secuencias de imágenes	9
1.2.1. Introducción	9
1.2.2. Modelos computacionales	14
1.2.2.1. Modelos de detección local de movimiento	15
1.2.2.1.1. Modelos basados en el análisis del gradiente	16
1.2.2.1.2. Modelos basados en la correlación	25
1.2.2.1.3. Otros modelos	34
1.2.2.2. Modelos basados en la atención selectiva visual	37
1.2.2.2.1. Modelos de obtención de clases (clustering)	37
1.2.2.2.2. Modelos de correspondencia y de correlación (matching)	38
1.2.2.2.3. Modelos de restricciones de uniformidad	44
1.2.3. Cuadro resumen de los modelos computacionales de análisis del movimiento en secuencias de imágenes	49

1.3. Análisis del movimiento de los objetos deformables	51
1.3.1. Introducción	51
1.3.2. Modelos computacionales	53
1.3.2.1. Aproximaciones basadas en modelos probabilísticos	53
1.3.2.1.1. Modelos basados en la distribución lineal de puntos	53
1.3.2.1.2. Modelos basados en el método del elemento finito	59
1.3.2.2. Aproximaciones basadas en los contornos 2D	61
1.3.2.2.1. Serpientes y serpientes Kalman	62
1.3.2.2.2. Esplines activos	63
1.3.2.3. Modelos de alto nivel	66
1.3.2.3.1. Descomposición en eigenimágenes	67
1.3.2.3.2. Modelos articulados	68
1.3.3. Cuadro resumen de los modelos computacionales de análisis del movimiento de los objetos deformables	69

Capítulo 2. Interacción lateral en computación acumulativa

2.1. Detección de movimiento en el sistema visual	73
2.1.1. Vías del sistema visual	73
2.1.2. Organización en columnas y capas	76
2.1.3. Movimiento en el campo visual	77
2.1.4. Representación computacional de la visión	80
2.2. Modelos de computación neuronal	83
2.2.1. Introducción	83
2.2.2. Arquitectura general de una red neuronal	83
2.2.3. Modelo de computación neuronal en una capa	85
2.2.4. Modelo de computación acumulativa local	90
2.2.5. Modelo de computación de doble temporización	93
2.2.6. Modelo de interacción lateral recurrente	94

2.3. Modelo de interacción lateral en computación acumulativa	101
2.3.1. Interacción lateral usando unidades de computación acumulativa	101
2.3.2. Aprendizaje en el modelo	104
2.3.3. Pasos generales asociados al modelo	106
2.3.4. Tareas de inferencia del modelo	108
2.3.4.1. Inferencia local sobre el centro	108
2.3.4.2. Inferencia sobre la realimentación de la periferia	110
2.3.4.3. Comparación y acción	111
2.3.4.4. Distribución	111
2.3.5. Lógica de un módulo de interacción lateral en computación acumulativa	112
2.3.5.1. Lógica de temporización	113
2.3.5.2. Lógica de carga	114
2.3.5.3. Lógica de descarga	116
2.3.5.4. Control de interacción lateral	117
2.3.5.5. Lógica de carga lateral	119
2.3.6. Tareas de control en el modelo	121

Capítulo 3. Obtención de las siluetas de los objetos deformables

3.1. Conocimiento del dominio del movimiento	123
3.1.1. Problemas generales para la obtención de siluetas	123
3.1.1.1. El problema de la apertura	123
3.1.1.2. Los límites del movimiento	125
3.1.1.3. Las oclusiones	125
3.1.1.4. El falso movimiento del fondo de la imagen	125
3.1.1.5. El contexto espacio-temporal	126
3.1.1.6. La velocidad del movimiento	126
3.1.1.7. El color en la detección de movimiento	127
3.1.1.8. La discriminación de más de un objeto en movimiento	128

3.1.2. Estudio de otras aproximaciones	128
3.1.2.1. Métodos basados en la correlación de características	128
3.1.2.2. Métodos basados en el gradiente	129
3.1.2.3. Métodos basados en regiones	130
3.1.2.4. Sustracción del fondo y detección de cambios	131
3.1.2.4.1. Detección de cambios	131
3.1.2.4.2. Sustracción del fondo	132
3.1.3. Aplicación del modelo de interacción lateral a partir del conocimiento del dominio del movimiento	133
3.1.3.1. Línea general adoptada	133
3.1.3.2. Solución al problema de la apertura	134
3.1.3.3. Solución a los límites del movimiento	135
3.1.3.4. Solución a las oclusiones y movimiento debido al fondo	135
3.1.3.5. Solución al contexto espacio-temporal	136
3.1.3.6. Solución a la velocidad del movimiento	137
3.1.3.7. Segmentación en bandas de nivel de gris	139
3.1.3.8. Clasificación de las siluetas detectadas	141
3.2. Arquitectura del método	143
3.2.1. Arquitectura multicapa del modelo de interacción lateral en computación acumulativa	143
3.2.2. Capa 0: Segmentación por bandas de nivel de gris	147
3.2.3. Capa 1: Interacción lateral para la computación acumulativa	149
3.2.4. Capa 2: Interacción lateral para el reparto de carga por bandas de nivel de gris	159
3.2.5. Capa 3: Interacción lateral para la fusión de objetos en movimiento	166
3.2.6. Capa 4: Aplicación	172
3.3. Influencia de los parámetros básicos del modelo de interacción lateral en computación acumulativa	173
3.3.1. Introducción	173
3.3.2. Oclusiones	174
3.3.3. Umbralización Global Simple vs. Umbralización Uniforme con	

Solapamiento	175
3.3.4. Influencia de los parámetros de la capa 0	178
3.3.4.1. Influencia del parámetro Número de Bandas de Nivel de Gris	178
3.3.5. Influencia de los parámetros de la capa 1	180
3.3.5.1. Influencia del parámetro Valor de Descarga por Detección de Movimiento	180
3.3.5.2. Influencia del parámetro Valor de Recarga por Vecindad	182
3.3.5.3. Influencia del parámetro Número de Vecinos para la Interacción Lateral	185
3.3.5.4. Influencia del parámetro Valor Mínimo de Mancha por Banda de Nivel de Gris	188
3.3.6. Influencia de los parámetros de la capa 2	190
3.3.6.1. Influencia del parámetro Número de Vecinos para el Reparto de Carga por Bandas de Nivel de Gris	190
3.3.6.2. Influencia del parámetro Valor Mínimo de Mancha para la Fusión de Objetos	193
3.3.7. Influencia de los parámetros de la capa 3	195
3.3.7.1. Influencia del parámetro Número de Vecinos para el Reparto de Carga por Objeto	195
3.4. Aprendizaje en el modelo de interacción lateral en computación	
Acumulativa	199
3.4.1. Introducción	199
3.4.2. Aprendizaje en el modelo	199
3.4.3. Arquitectura del aprendizaje	203
3.4.4. Algoritmo de aprendizaje	203
3.4.4.1. Inicialización de las capas	208
3.4.4.2. Aprendizaje en la capa 3	209
3.4.4.3. Aprendizaje en la capa 2	210
3.4.4.4. Aprendizaje en la capa 1	211
3.4.4.5. Aprendizaje en la capa 0	214

Capítulo 4. Implementación

4.1. Esquema general del entorno	215
4.1.1. Antecedentes	215
4.1.1.1. Sistema TIRANO	215
4.1.1.2. Sistema BRONTO	217
4.1.1.3. Sistema SkeleTTon y Base de Datos SkeleTTon	218
4.1.1.4. Sistema Intelligent SilhFINDER	220
4.1.2. Descripción general del entorno	221
4.1.2.1. Descripción hardware del entorno	221
4.1.2.2. Entorno de desarrollo y operación	222
4.1.2.3. Conceptos básicos	224
4.1.2.3.1. Unidades de dispensación de imágenes	224
4.1.2.3.2. Secuencias	225
4.1.2.3.3. Sesiones	227
4.1.2.3.4. Agentes	228
4.1.3. El Módulo de Dispensación de Imágenes	229
4.1.4. El módulo Vista	234
4.1.5. El módulo Núcleo de la aplicación	240
4.1.6. El módulo Capas de Procesamiento de Imágenes	240
4.1.6.1. El nivel de Interacción Lateral en Computación Acumulativa	243
4.1.6.2. El nivel de Escena	245
4.2. Vista del usuario: Aplicación Base de Datos SkeleTTon	247
4.2.1. Menús de la aplicación	247
4.2.2. Menú Control de secuencias	249
4.2.2.1. Opción Crear secuencias	250
4.2.2.1.1. Detalle de secuencias MAGNETO	251
4.2.2.1.2. Detalle de secuencias OTROS	251
4.2.2.1.3. Detalle de secuencias GENIMG	252
4.2.2.1.4. Detalle de secuencias CAMARA	252
4.2.2.2. Opción Modificar secuencias	253
4.2.2.3. Opción Ver secuencias	254

4.2.2.4. Opción Ver informes de secuencias	254
4.2.3. Menú Control de sesiones	255
4.2.3.1. Opción Crear sesiones	256
4.2.3.2. Opción Modificar sesiones	257
4.2.3.3. Opción Edición de sesiones	258
4.2.3.4. Opción Ver sesiones	261
4.2.4. Menú Control de agentes	262
4.2.4.1. Opción Eliminación de agentes	264
4.2.5. Menú Informes globales	265
4.3. Vista del usuario: Aplicación Intelligent SILHFinder	273
4.3.1. Control de acceso	273
4.3.2. Entorno del sistema	275
4.3.3. Menús de la aplicación	279
4.3.3.1. Menú Agente Intelligent SILHFinder	280
4.3.3.2. Menú Ambientes	291
4.3.3.3. Menú Modelos	308
4.3.3.4. Menú Escenas	313
4.3.3.5. Menú Monitorización de datos	329
4.3.3.6. Menú Entorno de trabajo	357
4.3.3.7. Menú Herramientas	367
4.3.3.8. Menú Ayuda sobre Intelligent SILHFinder	371
4.3.3.9. Menú Ver barras	374

Capítulo 5. Datos y resultados

5.1. Metodología empleada	345
5.2. Segmentación a bajo nivel	349
5.2.1. Segmentación a partir del movimiento	349
5.2.1.1. Segmentación a bajo nivel – Debido al movimiento - Siluetas	350
5.2.1.2. Segmentación a bajo nivel – Debido al movimiento	

– Contornos	356
5.2.2. Segmentación a partir de la luminosidad	357
5.2.2.1. Segmentación a bajo nivel – Debido a la luminosidad	
– Cambios de Luz	357
5.3. Aplicaciones de alto nivel	359
5.3.1. Imágenes Médicas	359
5.3.1.1. Angiografías	360
5.3.1.2. Ecografías fetales	362
5.3.2. Escenas de Tráfico	365
5.3.2.1. Autopistas	367
5.3.2.2. Cruces de Calles	369
5.3.3. Escenas de Vigilancia	372
5.3.3.1. Personas	373
5.3.4. Imágenes Satélite	375
5.3.4.1. Meteosat	376
Conclusiones y principales aportaciones	379
Bibliografía	383

Lista de símbolos

Símbolos del modelo general de Interacción Lateral en Computación Acumulativa

G ,	función global a nivel de conocimiento; a definir, en cada caso, en la utilización del modelo de interacción lateral en computación acumulativa
$E_i(x,y,t)$,	entrada desde la capa anterior, espacio de entrada i , al módulo (x,y) en el instante t
$S(x,y,t)$,	salida hacia el módulo (x,y) de la capa siguiente en el instante t , valor de descarga
$C(x,y,t)$,	valor de carga del módulo (x,y) en el instante t
R	frontera funcional en los espacios de entrada y salida
F_g ,	función de acumulación espacial en la escala de tiempos global t
F_b ,	función de acumulación espacio-temporal en la escala de tiempos local T
v_{des}	valor mínimo de carga; descarga
v_{sat}	valor máximo de carga; saturación
θ ,	valor del umbral de disparo

Símbolos de la Capa 0 del modelo, Segmentación por Bandas de Nivel de Gris

n ,	<i>Número de Bandas de Nivel de Gris</i>
$GL(x,y,t)$,	<i>Nivel de Gris del pixel (x,y) en el instante de tiempo t (entrada a la capa 0)</i>
$GLL_k(x,y,t)$,	<i>Pertenencia a la Banda de Nivel de Gris k del pixel (x,y) en el instante de tiempo t (salida hacia las capas 1 y 2, subcapas k)</i>

Símbolos de la Capa 1 del modelo, Interacción Lateral para la Computación Acumulativa

v_{des}	<i>Valor Mínimo de Permanencia (descarga)</i>
v_{sat}	<i>Valor Máximo de Permanencia (saturación)</i>
v_{dm}	<i>Valor de Descarga por Detección de Movimiento</i>
v_{rv}	<i>Valor de Recarga por Vecindad</i>
l_{pers}	<i>Número de Vecinos para la Interacción Lateral</i>
θ_{pers}	<i>Valor Mínimo de Mancha por Banda de Nivel de Gris (umbral)</i>
$PM_k(x,y,t)$	<i>Valor de Permanencia del pixel (x,y) en la banda de nivel de gris k en el instante de tiempo t (salida hacia la capa 2, subcapa k)</i>

Símbolos de la Capa 2 del modelo, Interacción Lateral para el Reparto de Carga

l_{car}	<i>Número de Vecinos para el Reparto de Carga por Bandas de Nivel de Gris</i>
θ_{car}	<i>Valor Mínimo de Mancha para la Fusión en Objetos (umbral)</i>
$C_k(x,y,t)$	<i>Valor de Carga del pixel (x,y) en la banda de nivel de gris k en el instante de tiempo t (salida hacia la capa 3)</i>

Símbolos de la Capa 3 del modelo, Interacción Lateral para la Fusión en Objetos

l_{obj}	<i>Número de Vecinos para el Reparto de Carga por Objeto</i>
θ_{obj}	<i>Valor Mínimo de Detección de Silueta (umbral)</i>
$S(x,y,t)$	<i>Valor de Silueta del pixel (x,y) en el instante de tiempo t (salida hacia la capa 4 de aplicación)</i>

Lista de figuras

Capítulo 1.

- Figura 1.1.** Objetos en movimiento
- Figura 1.2.** Sistema de control de tráfico
- Figura 1.3.** Imágenes médicas (1)
- Figura 1.4.** Imágenes médicas (2)
- Figura 1.5.** Imagen tomada del tráfico en una autovía
- Figura 1.6.** Siluetas de los objetos móviles presentes en la escena
- Figura 1.7.** Cinco heurísticas del movimiento en secuencias de imágenes
- Figura 1.8.** Los valores de v_G frente a θ_G forman una curva coseno de amplitud $|v|$ y fase $-\theta_v$
- Figura 1.9.** La velocidad (u,v) se encuentra en una recta perpendicular al vector (E_x, E_y)
- Figura 1.10.** Modelo de detección de movimiento de Hassenstein y Reichardt (1956)
- Figura 1.11.** Modelo de detección de movimiento de Barlow y Levick (1965)
- Figura 1.12.** A. Una representación de la barra vertical moviéndose hacia la derecha. B. Una representación espacio-temporal de la barra muestreada a lo largo del tiempo. El tiempo forma la tercera dimensión
- Figura 1.13.** Estructura matemática del sensor de movimiento escalar
- Figura 1.14.** La estructura del vector de sensores de movimiento
- Figura 1.15.** Secuencia de imágenes
- Figura 1.16.** Memoria de permanencia actualizada mediante una secuencia en la que un cuadrado se mueve con movimiento rectilíneo y uniforme
- Figura 1.17.** Gráfica espacio-tiempo (pixel/cuadro) obtenida analizando la memoria de permanencia de la figura 1.16
- Figura 1.18.** Los puntos a, b y c (indicados por círculos) se desplazan 0.6 unidades hacia la izquierda. Las nuevas posiciones de los puntos (renombrados A, B y C) se indican con cuadrados. Las correspondencias hipotéticas iniciales (vecinos más próximos) se indican con flechas
- Figura 1.19.** Organización de los vectores de datos
- Figura 1.20.** Matching de características o valores de intensidad entre dos tramas sucesivas
- Figura 1.21.** Paso de suma local (excitación)
- Figura 1.22.** Obtención de la velocidad que corresponde a la excitación máxima en cada posición (x,y)

Capítulo 2.

- Figura 2.1.** Canales de forma y movimiento en el sistema visual [Liv88]
- Figura 2.2.** Representación esquemática de los pasos en el análisis del movimiento en la corteza visual primaria
- Figura 2.3.** Modelo de computación neuronal en un nivel
- Figura 2.4.** Esquema de computación neuronal en una capa
- Figura 2.5.** Funciones umbrales más conocidas. (a) Abrupta. (b) Tangente hiperbólica. (c) Sigmoide.
- Figura 2.6.** La función de Gauss. (a) Representación tridimensional. (b) Gráfico de contornos
- Figura 2.7.** Modelo de computación acumulativa local
- Figura 2.8.** Comparación de los tiempos global y local
- Figura 2.9.** Modelo de computación de doble temporización
- Figura 2.10.** Obtención de las regiones del espacio de entrada. (a) La diferencia de gaussianas. (b) Las dos funciones gaussianas
- Figura 2.11.** (a) Función LOG. (b) Función DOG
- Figura 2.12.** Una función de obtención de la dirección. (a) $a=1, b=2, \alpha = 0$. (b) $a=1, b=2, \alpha = \frac{\pi}{4}$. (c) $a=1, b=2, \alpha = \frac{\pi}{2}$. (d) $a=1, b=2, \alpha = \frac{3\pi}{4}$
- Figura 2.13.** Interacción lateral (a) no recurrente (b) recurrente
- Figura 2.14.** Modelo de interacción lateral recurrente
- Figura 2.15.** El módulo (x,y) y sus conexiones con la capa anterior (E), la capa siguiente (S) y laterales con sus cuatro vecinos más próximos
- Figura 2.16.** Modelo de neurona de interacción lateral en computación acumulativa
- Figura 2.17.** Pasos generales del modelo computacional
- Figura 2.18.** Tareas genéricas
- Figura 2.19.** Lógica de un módulo de interacción lateral en computación acumulativa
- Figura 2.20.** Entradas / salidas de la LOGICA DE TEMPORIZACION
- Figura 2.21.** Obtención de los temporizadores clk para el tiempo local T y CLK para el tiempo global t
- Figura 2.22.** Descripción hardware de la LOGICA DE TEMPORIZACION
- Figura 2.23.** Entradas / salidas de la LOGICA DE CARGA
- Figura 2.24.** Descripción hardware de la LOGICA DE CARGA
- Figura 2.25.** Entradas / salidas de la LOGICA DE DESCARGA
- Figura 2.26.** Descripción hardware de la LOGICA DE DESCARGA
- Figura 2.27.** Entradas / salidas del CONTROL DE INTERACCION LATERAL
- Figura 2.28.** Descripción hardware del CONTROL DE INTERACCION LATERAL
- Figura 2.29.** Entradas / salidas de la LOGICA DE CARGA LATERAL
- Figura 2.30.** Descripción hardware de la LOGICA DE CARGA LATERAL

Capítulo 3.

- Figura 3.1.** Detección de movimiento del fondo de la imagen
- Figura 3.2.** Diferencia de imágenes (a) sin y (b) con intersección
- Figura 3.3.** Obtención de los valores de carga
- Figura 3.4.** Puntos con carga no máxima aislados y unidos a puntos con carga máxima
- Figura 3.5.** Activación, reactivación y desactivación
- Figura 3.6.** Descomposición de un elemento móvil en sus diferentes bandas de nivel de gris
- Figura 3.7.** Obtención de la silueta a partir de las sombras de las diferentes bandas de nivel de gris de un elemento
- Figura 3.8.** Discriminación de diferentes objetos
- Figura 3.9.** Representación esquemática del proceso de visión artificial
- Figura 3.10.** Configuración multicapa
- Figura 3.11.** Capa 0. Conexiones externas
- Figura 3.12.** Capa 0. Determinación de la banda de nivel de gris
- Figura 3.13.** Capa 1. Subcapa k. Conexiones laterales en forma de malla
- Figura 3.14.** Capa 1. Subcapa k. Conexiones externas
- Figura 3.15.** Esquema funcional del modelo en su capa 1
- Figura 3.16.** Lógica de carga asociada a la capa 1 del modelo
- Figura 3.17.** Lógica de carga lateral asociada a la capa 1 del modelo
- Figura 3.18.** Capa 1. Subcapa k. Estructura modular para cada pixel de la imagen
- Figura 3.19.** Capa 2. Subcapa k. Conexiones laterales en forma de malla
- Figura 3.20.** Capa 2. Subcapa k. Conexiones externas
- Figura 3.21.** Esquema funcional del modelo en su capa 2
- Figura 3.22.** Lógica de carga asociada a la capa 2 del modelo
- Figura 3.23.** Lógica de carga lateral asociada a la capa 2 del modelo
- Figura 3.24.** Capa 2. Subcapa k. Estructura modular para cada pixel de la imagen
- Figura 3.25.** Capa 3. Conexiones laterales en forma de malla
- Figura 3.26.** Capa 3. Conexiones externas
- Figura 3.27.** Lógica de carga asociada a la capa 3 del modelo
- Figura 3.28.** Esquema funcional del modelo en su capa 3
- Figura 3.29.** Capa 3. Estructura modular para cada pixel de la imagen
- Figura 3.30.** Entradas y salidas al modelo de interacción lateral durante la fase de aprendizaje
- Figura 3.31.** Arquitectura para el aprendizaje
- Figura 3.32.** Diagrama de flujo del aprendizaje

Capítulo 4.

- Figura 4.1.** Esquema del sistema BRONTO
- Figura 4.2.** Estructura modular del sistema SkeleTTon
- Figura 4.3.** Interconexión de los elementos integrantes del sistema SkeleTTon
- Figura 4.4.** Funcionamiento del procedimiento de ventana en un programa basado en Windows
- Figura 4.5.** Diagrama de clases básicas de toda aplicación MFC
- Figura 4.6.** Diagrama de uso del Módulo de Dispensación de Imágenes
- Figura 4.7.** Diagrama de clases del Módulo de Dispensación de Imágenes
- Figura 4.8.** Diagrama de secuencias del Módulo de Dispensación de Imágenes
- Figura 4.9.** Diagrama de uso del módulo Vista
- Figura 4.10.** Diagrama de uso del uso crear Agente
- Figura 4.11.** Diagrama de uso del uso establecer Agente
- Figura 4.12.** Diagrama de uso del uso establecer Escena
- Figura 4.13.** Diagrama de clases del módulo Vista (interfaz de usuario)
- Figura 4.14.** Diagrama de estados de la clase SkeletonView
- Figura 4.15.** Diagrama de secuencia del módulo Núcleo de la aplicación
- Figura 4.16.** Diagrama de uso del módulo Capas de Procesamiento de Imágenes
- Figura 4.17.** Diagrama de uso del uso aplicar Modelo
- Figura 4.18.** Diagrama de secuencias del Modelo de Interacción Lateral en Computación Acumulativa (ILCA)
- Figura 4.19.** Menú inicial
- Figura 4.20.** Menú de la aplicación
- Figura 4.21.** Unidades de dispensación de imágenes
- Figura 4.22.** Control de secuencias
- Figura 4.23.** Nuevos autores
- Figura 4.24.** Formulario de creación de secuencias
- Figura 4.25.** Detallar secuencias Magneto
- Figura 4.26.** Detallar secuencias Otros
- Figura 4.27.** Detallar secuencias Genimg
- Figura 4.28.** Modificar secuencias Genimg
- Figura 4.29.** Ver secuencias
- Figura 4.30.** Informes de secuencias
- Figura 4.31.** Control de sesiones
- Figura 4.32.** Crear sesión
- Figura 4.33.** Modificar sesiones
- Figura 4.34.** Editar sesiones

- Figura 4.35.** Insertar secuencia
- Figura 4.36.** Ver sesiones
- Figura 4.37.** Crear agente
- Figura 4.38.** Detallar agente
- Figura 4.39.** Menú de informes globales
- Figura 4.40.** Informes globales por sesiones
- Figura 4.41.** Informes globales por secuencia
- Figura 4.42.** ACCESO --- Control de acceso a Intelligent SILHFinder
- Figura 4.43.** ACCESO --- Control de acceso a Intelligent SILHFinder. Selección de un usuario
- Figura 4.44.** ACCESO --- Control de acceso a Intelligent SihFinder. despedida
- Figura 4.45.** Marco principal de la aplicación Intelligent SILHFinder
- Figura 4.46.** Menú Agente Intelligent SILHFinder
- Figura 4.47.** Marco de Intelligent SILHFinder. Agente abierto
- Figura 4.48.** Base de Datos SkeleTTon
- Figura 4.49.** AGENTES I.S. --- Abrir Agente
- Figura 4.50.** AGENTES I.S. --- Abrir Agente. Resultado de la selección
- Figura 4.51.** AGENTES I.S. --- Menú Agente Intelligent SILHFinder. Agente definido
- Figura 4.52.** AGENTES I.S. --- Guardar Agente
- Figura 4.53.** AGENTES I.S. --- Guardar Agente como
- Figura 4.54.** Marco de Intelligent SILHFinder. Aprendizaje del Agente arrancado
- Figura 4.55.** Mensaje de sesión detenida por el usuario durante el Aprendizaje del Agente
- Figura 4.56.** Mensaje de sesión detenida por Fin de Aprendizaje del Agente
- Figura 4.57.** Marco de Intelligent SILHFinder. Agente arrancado
- Figura 4.58.** Ejecutando Secuencia Cámara
- Figura 4.59.** Mensaje de sesión detenida por el usuario
- Figura 4.60.** Mensaje de sesión finalizada correctamente
- Figura 4.61.** Mensaje de confirmación de Salida de Intelligent SilhFinder
- Figura 4.62.** Menú Ambientes
- Figura 4.63.** AMBIENTES --- Parámetros Globales del Agente
- Figura 4.64.** AMBIENTES --- Restricciones de las Imágenes
- Figura 4.65.** AMBIENTES --- Preprocesado de las Imágenes del Agente
- Figura 4.66.** AMBIENTES --- Segmentación de las Imágenes del Agente
- Figura 4.67.** AMBIENTES --- Modelo de procesado asociado al Agente
- Figura 4.68.** AMBIENTES --- Postprocesado de las Imágenes del Agente
- Figura 4.69.** AMBIENTES --- Escena asociada al Agente
- Figura 4.70.** AMBIENTES --- Parámetros de Aprendizaje del Agente

- Figura 4.71.** Menú Modelos, Submenú Interacción Lateral en Computación Acumulativa
- Figura 4.72.** MODELOS --- ILCA -- Parámetros de la Interacción Lateral para la Computación Acumulativa
- Figura 4.73.** MODELOS --- ILCA -- Parámetros de la Interacción Lateral para el Reparto de Carga
- Figura 4.74.** MODELOS --- ILCA -- Parámetros de la Interacción Lateral para la Fusión de Objetos
- Figura 4.75.** Menú Escenas
- Figura 4.76.** Menú Escenas, Submenú Segmentación a bajo nivel
- Figura 4.77.** Menú Escenas, Submenú Segmentación a bajo nivel – Debida al Movimiento
- Figura 4.78.** Menú Escenas, Submenú Segmentación a bajo nivel – Debida a la Luminosidad
- Figura 4.79.** ESCENAS --- Segmentación a bajo nivel – Debida al Movimiento - Siluetas
- Figura 4.80.** ESCENAS --- Segmentación a bajo nivel – Debida al Movimiento - Contornos
- Figura 4.81.** ESCENAS --- Segmentación a bajo nivel – Debida a la Luminosidad - Cambios de Luz
- Figura 4.82.** Menú Escenas, Submenú Aplicaciones de alto nivel
- Figura 4.83.** Menú Escenas, Submenú Aplicaciones de alto nivel – Imágenes Médicas
- Figura 4.84.** Menú Escenas, Submenú Aplicaciones de alto nivel – Escenas de Tráfico
- Figura 4.85.** Menú Escenas, Submenú Aplicaciones de alto nivel – Escenas de Vigilancia
- Figura 4.86.** Menú Escenas, Submenú Aplicaciones de alto nivel – Imágenes Satélite
- Figura 4.87.** ESCENAS --- Aplicaciones de alto nivel – Imágenes Médicas - Angiografías
- Figura 4.88.** ESCENAS --- Aplicaciones de alto nivel – Escenas de Tráfico - Autopistas
- Figura 4.89.** ESCENAS --- Aplicaciones de alto nivel – Escenas de Tráfico - Cruces de Calles
- Figura 4.90.** ESCENAS --- Aplicaciones de alto nivel – Escenas de Vigilancia - Personas
- Figura 4.91.** ESCENAS --- Aplicaciones de alto nivel – Imágenes Satélite - Meteosat
- Figura 4.92.** Menú Monitorización de datos durante el Aprendizaje del Agente
- Figura 4.93.** MONITORIZACIÓN DE DATOS --- Imágenes del Aprendizaje
- Figura 4.94.** Menú Monitorización de datos
- Figura 4.95.** MONITORIZACIÓN DE DATOS --- Parámetros Globales
- Figura 4.96.** MONITORIZACIÓN DE DATOS --- Restricciones de las Imágenes
- Figura 4.97.** MONITORIZACIÓN DE DATOS --- Preprocesado de las Imágenes
- Figura 4.98.** MONITORIZACIÓN DE DATOS --- Segmentación de las Imágenes
- Figura 4.99.** Menú Monitorización de datos, Submenú Modelo – Interacción Lateral en Computación Acumulativa
- Figura 4.100.** MONITORIZACIÓN DE DATOS --- MODELOS – ILCA - Interacción Lateral para la Computación Acumulativa
- Figura 4.101.** MONITORIZACIÓN DE DATOS --- MODELOS – ILCA - Interacción Lateral para el Reparto de Carga

- Figura 4.102.** MONITORIZACIÓN DE DATOS --- MODELOS – ILCA - Interacción Lateral para la Fusión de Objetos
- Figura 4.103.** MONITORIZACIÓN DE DATOS --- Postprocesado de las Imágenes
- Figura 4.104.** MONITORIZACIÓN DE DATOS ---- ESCENAS --- Segmentación a bajo nivel – Debida al Movimiento - Siluetas
- Figura 4.105.** MONITORIZACIÓN DE DATOS ---- ESCENAS --- Segmentación a bajo nivel – Debida al Movimiento - Contornos
- Figura 4.106.** MONITORIZACIÓN DE DATOS ---- ESCENAS --- Segmentación a bajo nivel – Debida a la Luminosidad - Cambios de Luz
- Figura 4.107.** MONITORIZACIÓN DE DATOS ---- ESCENAS --- Aplicaciones de alto nivel – Imágenes Médicas - Angiografías
- Figura 4.108.** MONITORIZACIÓN DE DATOS ---- ESCENAS --- Aplicaciones de alto nivel – Escenas de Tráfico - Autopistas
- Figura 4.109.** MONITORIZACIÓN DE DATOS ---- ESCENAS --- Aplicaciones de alto nivel – Escenas de Tráfico - Cruces de Calles
- Figura 4.110.** MONITORIZACIÓN DE DATOS ---- ESCENAS --- Aplicaciones de alto nivel – Escenas de Vigilancia - Personas
- Figura 4.111.** MONITORIZACIÓN DE DATOS ---- ESCENAS --- Aplicaciones de alto nivel – Imágenes Satélite - Meteosat
- Figura 4.112.** MONITORIZACIÓN DE DATOS --- Imágenes
- Figura 4.113.** Menú Entorno de trabajo. Ningún Agente seleccionado
- Figura 4.114.** Menú Entorno de trabajo
- Figura 4.115.** ENTORNO DE TRABAJO --- Opción Configuración HW / SW
- Figura 4.116.** ENTORNO DE TRABAJO --- Directorios de Aplicaciones
- Figura 4.117.** ENTORNO DE TRABAJO --- Tratamiento de Imágenes
- Figura 4.118.** ENTORNO DE TRABAJO --- Configuración de la Traza
- Figura 4.119.** Menú Herramientas.
- Figura 4.120.** Acceso a la Base de Datos SkeleTTon
- Figura 4.121.** VTR Prototipo - Inicial
- Figura 4.122.** VTR Prototipo - Principal
- Figura 4.123.** VTR Prototipo - Conexión
- Figura 4.124.** VTR Prototipo – Principal. Puerto abierto
- Figura 4.125.** VTR Prototipo - Comandos
- Figura 4.126.** Menú Ayuda
- Figura 4.127.** AYUDA SOBRE I.S. --- Temas de Ayuda Intelligent SilhFinder
- Figura 4.128.** AYUDA SOBRE I.S. --- Ayuda de Windows

Figura 4.129. AYUDA SOBRE I.S. --- Acerca de Intelligent SILHFinder

Figura 4.130. Menú Ver barras

Figura 4.131. Marco principal de Intelligent SILHFinder sin barra de herramientas

Figura 4.132. Marco principal de Intelligent SILHFinder sin barra de estado

Capítulo 5.

Figura 5.1. Objetos rígidos

Figura 5.2. Objetos deformables

Figura 5.3. Sistema de vigilancia en cruces de calles

Figura 5.4. Imagen por satélite

Lista de tablas

Capítulo 3.

- Tabla 3.1.** Juego de ensayo para la explicación del parámetro Número de Bandas de Nivel de Gris
- Tabla 3.2.** Juego de ensayo para la explicación del parámetro Valor de Descarga por Detección de Movimiento
- Tabla 3.3.** Juego de ensayo para la explicación del parámetro Valor de Recarga por Vecindad
- Tabla 3.4.** Juego de ensayo para la explicación del parámetro Número de Vecinos para la Interacción Lateral
- Tabla 3.5.** Juego de ensayo para la explicación del parámetro Valor Mínimo de Mancha por Banda de Nivel de Gris
- Tabla 3.6.** Juego de ensayo para la explicación del parámetro Número de Vecinos para el Reparto de Carga por Bandas de Nivel de Gris
- Tabla 3.7.** Juego de ensayo para la explicación del parámetro Valor Mínimo de Mancha para la Fusión en Objetos
- Tabla 3.8.** Juego de ensayo para la explicación del parámetro Número de Vecinos para el Reparto de Carga por Objeto

Capítulo 5.

- Tabla 5.1.** Valores de los parámetros del modelo para la obtención de la silueta de una persona en movimiento
- Tabla 5.2.** Valores de los parámetros del modelo para la obtención de siluetas a partir del cambio de luminosidad
- Tabla 5.3.** Valores de los parámetros del modelo para la obtención del resultado de una angiografía
- Tabla 5.4.** Valores de los parámetros del modelo para la obtención de las siluetas de los vehículos en movimiento
- Tabla 5.5.** Valores de los parámetros del modelo para la obtención de las siluetas de los vehículos en cruces de calles
- Tabla 5.6.** Valores de los parámetros del modelo para la obtención de las siluetas de las personas en un sistema de vigilancia
- Tabla 5.7.** Valores de los parámetros del modelo para la obtención de las siluetas de las nubes en movimiento

Lista de ejemplos

Capítulo 3.

- Ejemplo 3.1.** Imágenes de ensayo para la explicación de las oclusiones
- Ejemplo 3.2.** Imágenes de ensayo para la explicación de los dos tipos de segmentación
- Ejemplo 3.3.** Imágenes de ensayo para la explicación de la influencia del parámetro Número de Bandas de Nivel de Gris
- Ejemplo 3.4.** Imágenes de ensayo para la explicación de la influencia del parámetro Valor de Descarga por Detección de Movimiento
- Ejemplo 3.5.** Imágenes de ensayo para la explicación de la influencia del parámetro Valor de Recarga por Vecindad
- Ejemplo 3.6.** Imágenes de ensayo para la explicación de la influencia del parámetro Número de Vecinos para la Interacción Lateral
- Ejemplo 3.7.** Imágenes de ensayo para la explicación de la influencia del parámetro Valor Mínimo de Mancha por Banda de Nivel de Gris
- Ejemplo 3.8.** Imágenes de ensayo para la explicación de la influencia del parámetro Número de Vecinos para el Reparto de Carga por Bandas de Nivel de Gris
- Ejemplo 3.9.** Imágenes de ensayo para la explicación de la influencia del parámetro Valor Mínimo de Mancha para la Fusión en Objetos
- Ejemplo 3.10.** Imágenes de ensayo para la explicación de la influencia del parámetro Número de Vecinos para el Reparto de Carga por Objeto

Capítulo 5.

- Ejemplo 5.1.** Obtención de la silueta de una persona en movimiento usando la umbralización global simple
- Ejemplo 5.2.** Obtención de la silueta de una persona en movimiento usando la umbralización uniforme con solapamiento
- Ejemplo 5.3.** Aplicación del modelo al movimiento de una fuente de frutos secos
- Ejemplo 5.4.** Aplicación del modelo a la rotación de una caja de cereales
- Ejemplo 5.5.** Aplicación del modelo al alejamiento de unas gafas de sol sobre una hoja escrita
- Ejemplo 5.6.** Obtención del contorno de una persona
- Ejemplo 5.7.** Obtención de siluetas a partir del cambio de la luminosidad
- Ejemplo 5.8.** Aplicación del modelo a angiografías
- Ejemplo 5.9.** Aplicación del modelo a ecografías fetales. (1) Cráneo. (2) Fémur. (3) Abdomen

Ejemplo 5.10. Aplicación del modelo a autopistas

Ejemplo 5.11. Aplicación del modelo a cruces de calles

Ejemplo 5.12. Aplicación del modelo a la vigilancia de personas

Ejemplo 5.13. Aplicación del modelo al seguimiento de nubes

Objetivos y resumen de la tesis

*Entonces, tomando la mano del ciego, le sacó de la aldea;
y escupiendo en sus ojos, le puso las manos encima, y le preguntó si veía algo.
El, mirando, dijo: Veo los hombres como árboles, pero los veo que andan.
Luego le puso otra vez las manos sobre los ojos, y le hizo que mirase;
y fue restablecido, y vio de lejos y claramente a todos.*

Marcos 8:21-25

La Santa Biblia (RVR 1960)

El objetivo del trabajo plasmado en esta memoria es abordar la problemática de la obtención de las siluetas de todos los elementos móviles presentes en una secuencia indefinida de imágenes. Se presenta un novedoso modelo general de computación basado en una arquitectura modular de inspiración biológica que da lugar a los denominados modelos de interacción lateral en computación acumulativa. Se muestra igualmente que la aplicación de estos modelos al problema objeto de esta tesis presenta unos resultados más que satisfactorios.

Veamos en extenso el contenido de los distintos capítulos de esta tesis.

El **Capítulo 1. Introducción y estado del arte** está formada por tres grandes bloques, enfocando el problema a afrontar y presentando algunas aproximaciones anteriores. El apartado **1.1. Planteamiento del problema: Extracción de siluetas de objetos deformables en secuencias de imágenes** introduce el planteamiento del problema. El apartado **1.2. Análisis del movimiento en secuencias de imágenes** introduce el estado del arte del análisis de movimiento en secuencias de imágenes. Por fin, el apartado **1.3. Análisis del movimiento de los objetos deformables** destaca los retos aportados por el análisis más específico del movimiento de los objetos deformables. También aquí se presentan algunos trabajos acometidos hasta la actualidad.

El **Capítulo 2. Interacción lateral en computación acumulativa** ofrece el modelo general de interacción lateral en computación acumulativa base para el enfoque de la solución al problema propuesto. El punto **2.1. Detección de movimiento en el sistema visual** aporta todas las consideraciones que han llevado a la elección de un modelo basado en la biología y en la computación neuronal. El apartado **2.2. Modelos de computación neuronal** introduce los conceptos básicos que van a llevar al modelo presentado en este trabajo. El apartado **2.3. Modelo**

de interacción lateral en computación acumulativa describe el modelo general. Esta parte describe en toda extensión las ideas principales que han llevado a la plena consecución de esta tesis doctoral. Las nociones a destacar desde este mismo momento son las de interacción lateral, computación acumulativa y doble temporización.

El **Capítulo 3. Obtención de las siluetas de los objetos deformables** entra de lleno en la aplicación del modelo presentado en el capítulo segundo a la problemática descrita en el capítulo 1. El punto **3.1. Conocimiento del dominio del movimiento** realiza una serie de reflexiones acerca de las características del problema y compara la solución adoptada con otras soluciones genéricas presentadas en el pasado. El apartado **3.2. Arquitectura del método** muestra la estructura multicapa del modelo general de interacción lateral en computación acumulativa y explica detenidamente el significado de todos los parámetros de interés en el modelo. De ello se deriva finalmente la arquitectura adoptada para la fase de aprendizaje en el modelo de interacción lateral. En **3.3. Influencia de los parámetros básicos del modelo de interacción lateral en computación acumulativa** se ofrece una visión de cómo influyen los diferentes parámetros del modelo en la obtención de las siluetas. Por último, en este capítulo, se introduce el **3.4. Aprendizaje en el modelo de interacción lateral en computación acumulativa** aplicado a la tarea de obtención de las siluetas de los objetos móviles.

El **Capítulo 4. Implementación** ofrece la visión del prototipo asociado al modelo aplicado a la extracción de las siluetas de los objetos deformables en secuencias de imágenes. En el apartado **4.1. Esquema general del entorno** se especifica la aplicación del modelo al problema. El grueso del trabajo de implementación viene descrito en forma de las dos aplicaciones principales *Intelligent SILHFinder* y *Base de Datos SkeleTton* en los dos siguientes apartados, a saber **4.2. Vista del usuario: Aplicación Base de Datos SkeleTton** y **4.3. Vista del usuario: Aplicación Intelligent SILHFinder**. El apartado 4.2. aporta una visión pormenorizada de la aplicación software que simula todas las ideas expuestas en esta tesis. El punto 4.3. presenta una aplicación complementaria a *Intelligent SILHFinder*, la *Base de Datos SkeleTton* que ofrece toda la potencia de una base de datos relacional para el manejo de los datos de la aplicación principal.

El **Capítulo 5. Datos y resultados** muestra la validez de los algoritmos a partir de un conjunto extenso de datos y resultados. Se incorporan ejemplos de algunas aplicaciones de interés relacionadas con la detección de movimiento y que, por tanto, han sido presentadas a la solución ofrecida en esta tesis.

En **Conclusiones y principales aportaciones** se destacan las que nosotros creemos que han sido las principales aportaciones de esta memoria y se mencionan las dos líneas de trabajo futuro en las que ya estamos trabajando.

CAPITULO 1. Introducción y estado del arte

1.1. Planteamiento del problema: Extracción de las siluetas de los objetos deformables en secuencias de imágenes

1.1.1. Introducción al análisis del movimiento

El movimiento desempeña un papel importante en la apreciación de nuestro entorno. A partir del movimiento visual es posible inferir conocimiento acerca de la estructura tridimensional de la escena observada [Mar82] [Fau93]. Puede, asimismo, ayudar en la detección de siluetas [Fau87], informar acerca de la profundidad relativa de los objetos en movimiento [Tek95] y darnos las claves de las propiedades materiales de los objetos en movimiento, tales como su rigidez y transparencia. El ejemplo de la figura 1.1. formado por dos elementos (una tetera y un juguete) podría servir para los fines anteriormente citados a partir del movimiento de los mismos.

La información de movimiento puede ser igualmente la base para las predicciones del tiempo restante para el impacto y de las trayectorias de los objetos que se mueven en una escena [Hor86]. Un gran número de estudios psico-visuales han demostrado que el movimiento es clave en el campo visual.

El análisis del movimiento visual a partir de imágenes cambiantes en el tiempo es un área importante en visión por ordenador [Bal82] y en procesamiento de imágenes [Jai89]. Es por ello que ha sido investigado de modo intenso y extenso a lo largo de muchos años [Hor81] [Jai81a] [Fau87] [Agg88] [Mit96]. En general, el problema del análisis del movimiento puede agruparse en dos grupos principales: la estimación de la velocidad instantánea de los objetos o parte de los mismos [Bar94] y el seguimiento, es decir, la determinación de la trayectoria en el tiempo de los objetos en movimiento [Der90] [Mey94]. Obviamente, estos grupos están estrechamente relacionados. Ambos se verán involucrados en muchas de las aplicaciones que usan imágenes dependientes del tiempo.



Figura 1.1. Objetos en movimiento¹

El análisis de movimiento en secuencias de imágenes es aplicable en multitud de campos. En general, en todos aquellos en los que el movimiento en la escena juegue un papel importante en la definición del problema a solventar.

En particular puede hablarse de casos de detección de elementos con características de velocidad determinadas. Casos concretos pueden encontrarse en control de tráfico, seguridad, vigilancia, y otros campos similares. El siguiente ejemplo sería típico de un sistema de control de tráfico:



Figura 1.2. Sistema de control de tráfico²

Posiblemente una de las aplicaciones más obvias de análisis del movimiento se encuentre en el campo de la robótica [Hor86]. Para los robots autónomos el movimiento visual es una fuente de información inagotable para el soporte en cuanto a ayuda a la navegación y la planificación de rutas [Nai98] [Wet97]. La navegación de un robot a partir de la mera información visual bidimensional es

¹ Data for this research were partially provided by the Calibrated Imaging Laboratory at Carnegie Mellon University, supported by ARPA, NSF, and NASA.

² Copyright © 1998 by H.-H. Nagel. Institut für Algorithmen und Kognitive Systeme. Fakultät für Informatik, Universität Karlsruhe (TH). Postfach 6980, D-76128 Karlsruhe, Germany

un problema de difícil resolución. Algunas de las razones, entre otras, serían las distorsiones inducidas por el propio movimiento del robot, la pérdida de información incurrida por la proyección de tres dimensiones en dos dimensiones, además del problema de análisis del movimiento en sí.

Sin embargo, las técnicas desarrolladas en este campo están demostrando su utilidad en entornos más específicos. Los brazos industriales, por ejemplo, pueden desarrollar un gran número de operaciones sobre objetos pasando sobre una cinta transportadora [Lew93] [Ste85]. Para ello, se basan en la información recogida por una cámara situada por encima de la cinta. Los vehículos autónomos, a su vez, son capaces de seguir el trazado de una carretera, siempre que el vehículo disponga de la información de ciertas marcas características de las carreteras como pueden ser las líneas centrales de la calzada [Kua88].

En otras aplicaciones de análisis del movimiento, la visión tridimensional no tiene por qué ser el objetivo principal, y, por ende, no tiene por qué interesar una descripción completa tridimensional de la escena en movimiento. Entre estas aplicaciones se encuentran la interpretación de las imágenes tomadas desde satélite o las imágenes astronómicas. Un ejemplo de este tipo de interpretación es el del seguimiento de la formación de nubes para la predicción del tiempo [Col92]. También es de interés la restauración y mejora de las imágenes [Ira93], teniendo como ejemplo más claro la eliminación de manchas en las películas antiguas de cine [Vla96].

Otra área en la que el análisis del movimiento es de gran importancia reside en las imágenes médicas, donde, por ejemplo, puede ser usada para monitorizar patrones de movimiento del corazón a partir de imágenes MR [Pri92] [Fun96] [Rat96], o también para mejorar e interpretar imágenes escaneadas de ultrasonido [Che95] [Qui97]. Está claro que este tipo de aplicaciones persigue unos objetivos muy diferentes a los de la clásica visión tridimensional.

Se muestran dos ejemplos de imágenes médicas:



Figura 1.3. Imágenes médicas (1)³

³ Copyright © 1995, University of Kentucky, College of Medicine.

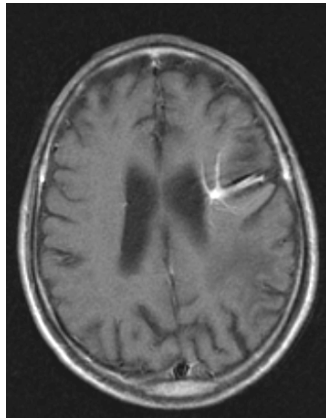


Figura 1.4. Imágenes médicas (2)⁴

El análisis del movimiento también está encontrando un uso creciente en los sistemas multimedia. Tales aplicaciones incluyen el seguimiento del movimiento de una cámara para la construcción y actualización de conjuntos virtuales [Lar97]. Se utiliza el movimiento para la indexación automáticamente de los videoclips, para la recuperación en grandes bases de datos [Idr97], así como para la codificación eficiente de los datos de vídeo para su futura transmisión [Tek95].

En el campo de la compresión de datos de vídeo, la información de movimiento se usa explotando las redundancias temporales en los datos. La idea básica es la de hacer uso de la compensación del movimiento dentro de un entorno de codificación predictiva. Dicho de otro modo, se aprovecha que la diferencia espacial entre dos tramas de vídeo adyacentes es posiblemente pequeña debido a la alta tasa de muestreo temporal. Esta puede reducirse aún más si el movimiento de las regiones es conocido. Entonces se puede desplazar cada región antes de tomar las diferencias entre las tramas. La diferencia de trama desplazada puede codificarse entonces eficientemente utilizando técnicas de compresión de datos estándar [Jai81b]. Al transmitir una trama clave, y transmitiendo a continuación solamente el campo movimiento codificado junto con la diferencia de trama desplazada codificada, se consigue una considerable reducción de la tasa de bits [LeG91] [Li94].

Una aproximación de alto nivel relacionada con lo expuesto anteriormente, y que incorpora algunas de las técnicas de visión tridimensionales mencionadas con anterioridad, es la de la codificación basada en modelos, en la que se construye un modelo geométrico tridimensional en una escena limitada. El modelo puede constar, por ejemplo, de la cabeza y los hombros de una persona, y usarse para videoconferencias. Una vez que el modelo es conocido en los nodos de recepción y de

⁴ Courtesy of University of Palermo, Institute of Radiology "P. Cignolini".

transmisión, el modelo puede ser animado transmitiendo los datos de movimiento codificados [Ryd87] [Li93] [Pea95].

Como hemos visto, las aplicaciones que se benefician de la estimación, del análisis y del seguimiento a partir de la detección de movimiento son muy diversas. Los requerimientos que imponen en la estimación y el análisis del movimiento también son diversos. Esto se refleja en la gran cantidad de técnicas de estimación y análisis que han sido propuestas en los años recientes.

1.1.2. Estimación del movimiento y seguimiento

El problema más típico del movimiento es el análisis del movimiento de los datos en las imágenes digitales bidimensionales correspondientes a una secuencia de muestras espacio-temporales de una escena tridimensional. Estas secuencias espacio-temporales, conocidas como tramas, son capturadas por medio de una cámara a una frecuencia de trama dada, o sea, a intervalos de tiempo regularmente espaciados. Cada una de las tramas puede verse como la codificación de la intensidad de luz que entra por la cámara en cada instante de tiempo. Las secuencias se capturan normalmente a una frecuencia de trama de 25-30 Hz, ya que, para el ojo humano, una frecuencia temporal de este tipo da una apariencia de movimiento continuo y realista al proyectarse la secuencia como película o videoclip [Tek95].

Ya que la secuencia capturada representa una proyección tridimensional en un espacio bidimensional, la relación entre los cambios observados en las intensidades de las imágenes de la secuencia, denominado flujo óptico, y el movimiento de los objetos tridimensionales en la escena, es compleja. Una de las razones principales para esta complejidad es el hecho de que una proyección 3D en 2D implica la pérdida de información relacionada con la profundidad de los puntos de la escena. Esto se denomina ambigüedad de la proyección [Tek95]. Viene a decir que, en proyección perspectiva, algunos puntos de la escena tridimensional con velocidades potencialmente diferentes se proyectan sobre el mismo punto bidimensional de la imagen [Tru98]. La relación entre el flujo óptico y la escena depende igualmente de las propiedades del propio movimiento. La escena puede ser estática y la cámara dinámica, o la cámara estar estática, capturando una escena dinámica constituida por varios objetos en movimiento, o también tenemos que tanto la cámara como la escena pueden ser dinámicas. Otro factor a tener en cuenta es si los objetos en movimiento son rígidos o cambian su silueta debido al movimiento, es decir, que son no rígidos o deformables.

Una complicación adicional reside en el hecho que el flujo óptico no equivale necesariamente a la proyección del movimiento tridimensional, ya que puede también depender de otros factores tales como la iluminación global de la escena. Tomemos el ejemplo clásico es el de una esfera en rotación. En un ambiente de iluminación difusa el flujo óptico observable valdrá cero a pesar del movimiento 3D. No obstante, si la esfera está parada pero existe una fuente de luz en movimiento, el flujo óptico no valdrá cero, llevando erróneamente a la idea de que existe movimiento en la escena. Una estimación de movimiento no ambigua solamente puede obtenerse si las proyecciones de los objetos en movimiento exhiben un grado suficiente de variación de intensidad.

Debemos considerar dos puntos claves antes de poder afrontar los problemas de la estimación de movimiento y del seguimiento. El primer punto es qué entidades espaciales se han de tomar para estimar el movimiento o el seguimiento. Podemos hacer formulaciones en términos de objetos de la escena, características o puntos individuales. La elección muchas veces viene dictada por el tipo de aplicación para la que se está realizando el análisis del movimiento. Una aproximación centrada en los píxeles puede verse como una aproximación a bajo nivel al problema de visión, mientras que el utilizar un modelo de movimiento orientado al objeto es una tarea de visión de alto nivel. El uso de regiones o características cae en algún lugar entre estos dos extremos. Cuando el propósito es el seguimiento parece pertinente acercarse a la escala de alto nivel, aunque en teoría sería posible efectuar un seguimiento basado en los píxeles individuales.

El segundo punto es si debe usarse un modelo tridimensional o un modelo bidimensional. Toda escena es inherentemente tridimensional, mientras que una secuencia de imágenes es un conjunto de proyecciones bidimensionales. La aplicación puede demandar del análisis del movimiento 3D completo, o podría ser suficiente analizar el movimiento en el plano de proyección 2D. La estimación del movimiento tridimensional a partir de las proyecciones bidimensionales es un problema difícil de resolver. Este puede considerarse no resuelto mientras no se efectúen las restricciones pertinentes al entorno. Por otra parte, la estimación del campo de movimiento bidimensional proyectado sería un problema más restringido, pero su solución también está lejos de ser evidente.

1.1.3. Planteamiento del problema

El problema que se pretende abordar es el de la discriminación de ciertos objetos deformables que capten nuestra atención en una escena determinada. Estos objetos de la vida real serán detectados a partir del movimiento de alguna de las partes que lo componen. Dicho

movimiento, que se podrá captar en una secuencia indefinida de fotogramas, permitirá ir obteniendo las siluetas de los elementos en movimiento.

El ejemplo de la figura 1.5. muestra una imagen de una secuencia de imágenes tomadas por una cámara en un puente elevado sobre una autovía.



Figura 1.5. Imagen tomada del tráfico en una autovía

A partir de la secuencia de imágenes, tomando en consideración el movimiento que aparece en la escena, se pretende obtener las siluetas de todos los objetos deformables móviles presentes en la escena. En nuestro caso, la solución vendría dada por la siguiente imagen resultante (figura 1.6.), donde se detectan seis elementos a los que se han dado colores diferentes.



Figura 1.6. Siluetas de los objetos móviles presentes en la escena

Cuando los elementos dejen de moverse, dejarán de pertenecer a los elementos de interés, con lo que dicha silueta dejará de interesarnos, y por tanto, a partir de ese instante, no formará más

parte de los elementos discriminados. Las secuencias de imágenes estarán formadas por imágenes reales tomadas por una cámara.

Como ya se ha comentado, la forma de obtener dichos elementos va a partir únicamente de la información de movimiento existente en la escena. Se trata pues de un problema típico de análisis de movimiento. De algún modo, el método puede englobarse genéricamente en el apartado de los modelos basados en la diferencia de imágenes, tomando en cuenta, por tanto, la función de intensidad. Más concretamente, está ligado al comportamiento genérico de las memorias de permanencia [Fer92], como se verá más adelante.

Específicamente, diremos que el observador es incapaz de discernir ningún tipo de elemento u objeto, salvo que éste se ponga en movimiento. Por lo tanto, una imagen, donde todos los elementos son estáticos, no aporta información alguna. Dicho de otro modo, el sistema va a atender únicamente a aquellos píxeles de una imagen de 512x512 (véase la imagen completa) donde sea capaz de detectar algún cambio en el nivel de gris entre dos fotogramas consecutivos.

En escenas reales, como las que vivimos en todo momento, no tienen por qué moverse la totalidad de las partes que componen un objeto. Por ejemplo, si tomamos el caso del cuerpo humano (el objeto en este caso), sabemos que éste está formado por un gran número de miembros que no tienen por qué estar moviéndose simultáneamente. Así, una persona, sentada frente a su ordenador, y escribiendo una carta, moverá las manos, parte de los brazos, la cabeza, ..., pero podrá tener las piernas y los pies perfectamente quietos. El sistema será capaz de detectar aquellas partes del objeto en movimiento. Es decir, que la silueta del objeto estará formada únicamente por las manos, los brazos y la cabeza.

No obstante, si otras partes del mismo objeto se ponen en movimiento, será fundamental poder asociarlas a las partes anteriormente detectadas, obteniendo el objeto como un todo a través de los mecanismos de asociación pertinentes.

El problema no se limita a la observación, detección y seguimiento de un solo objeto deformable en la escena, sino que será capaz de discriminar (diferenciar) todos los objetos que ofrezcan algún tipo de movimiento.

1.2. Análisis del movimiento en secuencias de imágenes

1.2.1. Introducción

El análisis de movimiento es de una importancia crucial. Aunque ciertos animales puedan estar faltos de alguna capacidad visual tal como la percepción del color, todos ellos tienen capacidades de proceso del movimiento. Las lesiones en este sistema tienen consecuencias devastadoras en un gran número de funciones vitales, entre las que se incluyen la automotricidad, la percepción del movimiento de los objetos y la orientación.

El sueño de la automatización inteligente en el campo de la visión artificial consiste, desde los años 50, en dotar a los ordenadores de capacidades comparables a las de los organismos biológicos. En los últimos años ha habido un crecimiento rápido en las ideas aportadas en el campo de la visión por ordenador y, por ende, en el del análisis de movimiento.

Se pretende dar en este primer capítulo una panorámica general de la situación actual de la automatización del análisis de movimiento en secuencias de imágenes.

Se ofrece, en esta breve introducción, una panorámica de la problemática que conlleva el análisis de movimiento en secuencias de imágenes. Una sencilla introducción a este tema puede encontrarse en [Fer93].

Para comenzar, se dirá que el movimiento se detecta siempre a partir de una secuencia de imágenes. Se define una secuencia de imágenes como un conjunto ordenado de imágenes. Las secuencias de imágenes de interés son tomas de cuatro dimensiones espacio-temporales. En la mayoría de los casos, tal como ocurre en las películas, las imágenes son proyecciones bidimensionales de un mundo físico tridimensional, superpuestas secuencialmente a lo largo del tiempo. En un gran número de aplicaciones prácticas, la entrada ha de ser una secuencia de esta naturaleza, y la continuidad del movimiento ha de ser inferido de discretas diferencias locales en los puntos de las imágenes.

A continuación, se pasa a hacer un poco de historia para apreciar cuál ha sido la evolución del análisis de movimiento en secuencias de imágenes, así como las causas de esta evolución. Durante los años 60 y 70, se produce un fuerte desarrollo de técnicas de tratamiento de imagen en general, pero enfocadas principalmente al análisis de imágenes estáticas. Esto es debido principalmente a problemas técnicos, ya que el volumen de información que impone el tratamiento

de secuencias de imágenes es mucho mayor que el necesario para un análisis de imagen estático. En los últimos años, el aumento, tanto de capacidad computacional como de almacenamiento, ha permitido abordar el estudio y la realización de algoritmos y de sistemas para el análisis de movimiento en secuencias de imágenes.

También es posible que se haya asumido desde el principio que era más fácil entender el movimiento a partir de un número elevado de imágenes analizadas a gran velocidad. Esta caracterización de la captación del movimiento puede ser muy ilustrativa. Presupone que la visión implica el proceso de escenas estáticas. Otra aproximación viene a decir que el movimiento de un objeto es una excelente fuente para la segmentación de imágenes. El desarrollo de técnicas que van en esta dirección permite, mediante un análisis de la evolución dinámica de la secuencia de imágenes, extraer parámetros estáticos de los elementos que conforman la escena.

Existen pues dos opciones básicas a resaltar en cuanto al proceso a realizar y resultados obtenidos. Resumiendo, una primera opción consiste en realizar un análisis basado en características estáticas de imágenes sucesivas, y obtener, a partir de este análisis, características dinámicas de los elementos de la escena. También podría procederse analizando en la secuencia de imágenes las características dinámicas para obtener características estáticas y/o dinámicas de los elementos de la escena. Esto quiere decir que los análisis estáticos y dinámicos pueden compatibilizarse y apoyarse mutuamente.

Ha de tenerse en cuenta que normalmente se tiende a determinar y describir cómo se relacionan las imágenes entre sí. Esta información se deriva de las imágenes estáticas, y se presentan inmediatamente dos aproximaciones: la de buscar las diferencias entre las imágenes, o bien la de buscar las similitudes entre las imágenes. Estas dos aproximaciones son complementarias y se usan en numerosas ocasiones conjuntamente. Un paradigma general para el análisis de movimiento es el siguiente:

1. Segmentar (describir) las imágenes individuales. Un caso especial importante es aquél en el cual el proceso de descripción (segmentación) es nulo y la descripción es la propia imagen.
2. Describir las diferencias o similitudes entre las descripciones.
3. Construir una descripción de la secuencia como un todo a partir de las primitivas y de las descripciones de diferencias o similitudes de relevancia para el propósito perseguido.

Algunos ejemplos de qué se pretende mediante el análisis de movimiento en secuencias de

imágenes se explican a continuación:

- Un objetivo básico de este tipo de análisis es la detección de movimiento en la escena. Como se va a ver más adelante, algunos algoritmos de los que se exponen en este trabajo reducen su capacidad a este término.
- La obtención de parámetros de movimiento tales como trayectoria, velocidad, etc., es un objetivo algo más ambicioso, que añade una problemática extensa y conlleva una importante capacidad de cálculo.
- Pueden también abordarse objetivos referentes a la segmentación de elementos de movimiento característico o la definición de contornos basada en características dinámicas de la escena.

La problemática que envuelve el análisis de movimiento en secuencias de imágenes es amplia y plural. Amplia como se va a ver a continuación, y plural porque emerge de multitud de campos de naturaleza física, geométrica o electrónica.

Un problema básico para la realización de este tipo de análisis es la cantidad y frecuencia de información. Una imagen generalmente contiene gran cantidad de información. Para un caso típico de una imagen de señal de televisión, podemos hablar de 512×512 (262.144) palabras de 8 bits. Si se pasa de hablar de una sola imagen a hablar de una secuencia, y se continúa con el caso típico de una señal de televisión, en el que aparece una nueva imagen cada 40 milisegundos, habrán de procesarse más de 6 Mbytes de información en cada segundo si se desea trabajar en tiempo real.

El problema del análisis de movimiento tiene una naturaleza claramente geométrica. Debido a las características de los sensores y de los sistemas de computación con que es abordado, el problema geométrico se convierte en un efecto sobre el nivel de gris. Esto añade problemática, puesto que al modificar la naturaleza del problema, muchos conceptos se difuminan, dificultando su percepción.

Al problema fundamental que conlleva el análisis de secuencias de imágenes se le ha llamado el problema de la correspondencia [Dud73]. Conceptualmente consiste en asociar los puntos que en una imagen de la secuencia corresponden a un elemento concreto de la escena física, con los que este mismo elemento genera en la imagen siguiente de la secuencia. Esto es una tarea elemental para cualquier sistema biológico de visión, pero es muy difícil de abordar computacionalmente desde una máquina de flujo de instrucciones, que guarda en su memoria, de acceso aleatorio, las dos imágenes de una secuencia.

Para enfrentarse a este problema se pueden plantear varias estrategias entre las que cabe comentar las siguientes:

- a) Análisis de la imagen estática y extrapolación de cuál será el movimiento de los elementos de imagen mediante un estudio global de los elementos de la escena y su historia.
- b) Análisis de la secuencia de imágenes y realización de una función de correspondencia entre los elementos, basada en un estudio local a cada punto de imagen, sin conocimiento de cuál es el elemento de escena al que pertenece.

El problema de la correspondencia está fuertemente asociado a la relación entre la velocidad y el muestreo. Dependiendo del periodo de muestreo, según la velocidad de los elementos de la escena, éstos se situarán en puntos más o menos distantes en imágenes posteriores de la secuencia. La relación velocidad-muestreo, jugará pues un papel de importancia capital para la validez del sistema que se proponga.

Físicamente, el sensor, mediante el que se muestrea la escena, es plano. En algunos casos, se puede hablar de movimientos planos, cuando la escena es plana y los movimientos se realizan en este plano. Se ve entonces reducida la problemática. Pero éste no es el caso habitual, lo normal es que la escena forme parte del mundo tridimensional. Aparecerá por esta causa una amplia problemática, sobre la que a continuación se realizan algunos comentarios. Un mundo en tres dimensiones, muestreado sobre un sensor plano, introduce una serie de efectos procedentes de la relación forma-colocación-situación, entre los que pueden comentarse, por ejemplo, la relación entre la distancia y el tamaño, o la existente entre la forma geométrica y el punto de vista desde el que se percibe el objeto.

Otra problemática, más conectada al entorno físico de trabajo, es la que aparece por motivo de variaciones del nivel de gris no debidas al movimiento, sino a ruidos eléctricos en el sistema de captación, o a alteraciones en la iluminación o en la naturaleza de los elementos de la escena.

Con todas estas premisas, los investigadores se han visto abocados al uso de modelos con mayores restricciones que el propio mundo. La coherencia de la mayoría de los objetos tridimensionales y su continuidad a través del tiempo son dos principios generales que, aunque violados en algunas ocasiones, guían muchas heurísticas de segmentación y de correspondencia de puntos. Un ejemplo de ello es la correspondencia asumida de regiones en imágenes con objetos. Las imágenes en movimiento son otro ejemplo: la coherencia de los objetos implica que habrá

parecidas condiciones de "continuidad" en las posiciones y las velocidades de los puntos vecinos de la imagen.

A continuación se describen cinco heurísticas utilizables en puntos coincidentes de imágenes separadas por un corto espacio de tiempo [Pra79].

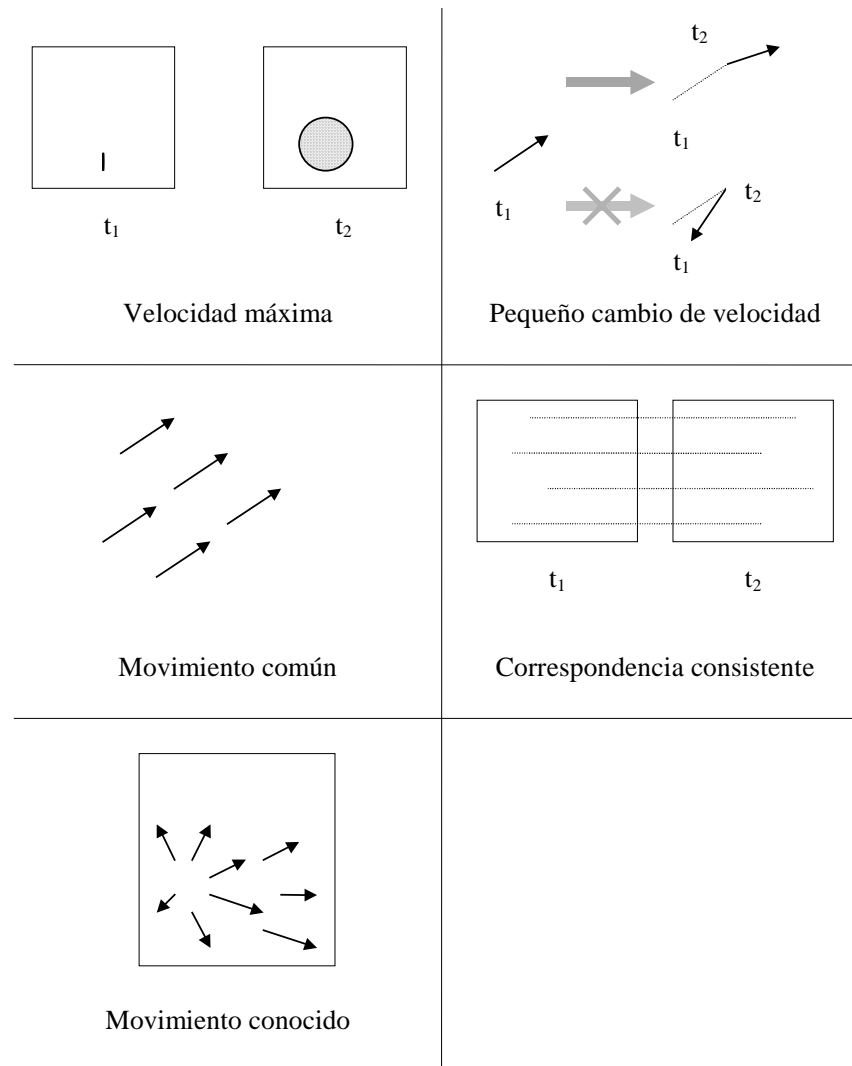


Figura 1.7. Cinco heurísticas del movimiento en secuencias de imágenes

1. *Velocidad máxima.* Si V es la velocidad máxima de un punto respecto de un dispositivo estático de toma de imágenes, entonces puede moverse como mucho en Vdt entre dos imágenes separadas en un tiempo dt . De este modo, dada la posición de un punto en una imagen, esta restricción limita la zona en la que puede aparecer el punto en la segunda imagen.
2. *Pequeño cambio de velocidad.* Ya que la mayoría de los objetos físicos tienen una

masa finita, esta heurística es una consecuencia de las leyes físicas y de la restricción de un "intervalo pequeño" entre las imágenes.

3. *Movimiento común.* Los objetos espacialmente coherentes aparecen, normalmente, en las imágenes sucesivas como regiones de puntos compartiendo un "movimiento común".
4. *Correspondencia consistente.* Dos puntos de una imagen generalmente no se corresponden con un solo punto de otra imagen (salvo en el caso de oclusiones).
5. *Movimiento conocido.*

La figura 1.7. muestra gráficamente cada una de las heurísticas descritas.

1.2.2. Modelos computacionales

El estudio pretende centrarse en los más conocidos modelos computacionales de detección inicial de movimiento y de análisis del movimiento que van a permitir un posterior acercamiento al apartado de procesado de bajo nivel, y concretamente en los aspectos de extracción de características y de segmentación.

La segmentación en el movimiento es un problema ya clásico en la visión artificial por ordenador. Los dos métodos más populares de extracción de objetos en movimiento se basan en:

- a) el cálculo del flujo óptico
- b) la diferencia de imágenes

El primer conjunto de técnicas de segmentación del movimiento se basa en el cálculo del flujo óptico. El campo velocidad puede ser segmentado para identificar los diferentes objetos en movimiento en la imagen. Existen básicamente dos aproximaciones para calcular el mapa de disparidad entre dos tramas.

La aproximación de continuidad (o aproximación basada en el gradiente) utiliza la variación espacio-temporal según la famosa ecuación de restricción del movimiento [Hor81]. Esta aproximación se verá en detalle más adelante en este estudio, y será completada con las tres técnicas complementarias, a saber una técnica de optimización local [Tho81], una técnica de optimización global [Hor81] y una aproximación de obtención de clases [Fen79].

La aproximación discreta (llamada también aproximación basada en características) para

calcular el flujo óptico consiste en extraer aquellas características que correlacionen ambas tramas.

El segundo conjunto de técnicas para la segmentación del movimiento, basado en la diferencia entre imágenes, es muy sencillo. De nuevo se dispone de dos categorías de técnicas de diferenciación. La diferencia entre dos imágenes se puede obtener entre una trama tomada de una secuencia de imágenes y otra trama de referencia de la que se sabe que es estática. También se puede hacer la diferencia entre dos tramas cualesquiera de una secuencia de imágenes.

Se pasa a recoger, pues, en este estudio, algunos de los más conocidos modelos computacionales de detección inicial de movimiento y de análisis del movimiento en secuencias de imágenes. Estos modelos son, a su vez, evaluados de acuerdo a su verosimilitud fisiológica y perceptual. Básicamente podemos partir de dos grandes enfoques al problema que nos dan las dos orientaciones genéricas más conocidas:

- a) los modelos basados en la detección local de movimiento
- b) los modelos basados en la atención selectiva visual

Antes de discutir estos modelos, se da una lista parcial de criterios utilizados para evaluar la bondad de los mismos. Algunos temas importantes a tener en cuenta son:

1. el tipo de movimiento que el modelo soporta (por ejemplo, traslacional plano y/o movimiento general 3-D)
2. cómo el modelo se enfrenta a varios objetos en movimiento
3. la habilidad del modelo para producir varios fenómenos perceptuales, incluyendo, por ejemplo, el movimiento no rígido percibido de algunas figuras rígidas en movimiento, efectos coherentes (por ejemplo, captura de movimiento y transparencia)
4. la validez fisiológica y perceptual de las presunciones tomadas, los algoritmos usados y los resultados obtenidos

1.2.2.1. Modelos de detección local de movimiento

Se han propuesto algunos modelos para la detección inicial del movimiento. Estos modelos se dividen, a grandes rasgos, en dos clases:

- (1) los modelos basados en el análisis del gradiente espacio-temporal, que calculan la

velocidad a partir de la variación de la intensidad en el espacio y en el tiempo producida por el movimiento de una imagen

- (2) los modelos basados en la correlación, cuya estructura básica está inspirada en los primeros trabajos acerca del sistema visual del escarabajo y de la mosca [Has56] y de la retina del conejo [Bar65]

1.2.2.1.1. Modelos basados en el análisis del gradiente

Los modelos basados en el análisis del gradiente calculan la velocidad tomando la derivada temporal de la intensidad en un punto dividida por la derivada espacial de la intensidad en ese mismo punto

$$V_x = -\frac{dI}{dt} / \frac{dx}{dt}$$

donde V_x representa la velocidad en la dirección x [Fen79] [Hor81] [Mar81]. Este modelo permite la determinación punto a punto de la disparidad basándose en criterios puramente locales sin la necesidad de examinar largas secuencias de imágenes.

El modelo de determinación de la velocidad de Fennema y Thompson [Fen79].

Fennema y Thompson describen un modelo que cuantifica la velocidad y la dirección de varios objetos en movimiento en una secuencia de imágenes digitales. A partir del desarrollo de una relación entre la variación de la intensidad en el tiempo, el gradiente espacial y la velocidad, se determina el movimiento utilizando técnicas de segregación por clases.

Se parte de la relación, observada por Limb y Murphy [Lim75], que existe entre los cambios temporales en la intensidad en un punto y el gradiente de intensidad espacial en la vecindad del mismo punto. Se trata de forma independiente el movimiento a lo largo de los ejes x e y . Si ΔI representa el cambio de intensidad en un punto debido al movimiento Δx a lo largo del eje x , se tiene

$$\Delta x = \frac{\Delta x}{\Delta I} \approx -\frac{\Delta I}{G_x}$$

donde G_x es el gradiente espacial de la intensidad a lo largo del eje x . Una relación similar se obtiene para el eje y . Desafortunadamente, ΔI depende del movimiento a lo largo de ambos ejes, de modo que el movimiento en x afecta a la estimación de Δy , y el movimiento en y afecta al término

Δx . Por ello los valores de Δx y de Δy pueden ser inadecuados para un punto de imagen aislado. Sin embargo, si la dirección del gradiente está distribuida aleatoriamente, los valores medios de Δx y Δy dan una buena estimación de la velocidad verdadera. Al hablar de valores medios, se restringe el modelo al análisis del movimiento de un único elemento.

Fennema y Thompson amplían los trabajos de Limb y Murphy para permitir el análisis de varios elementos móviles. Su técnica se denomina *Gradient Intensity Transform Method* (GITM).

El movimiento puede representarse mediante un vector velocidad

$$v = \frac{ds}{dt} \quad (1)$$

De la propia definición del gradiente, un cambio incremental en la intensidad debido a un desplazamiento espacial del punto evaluado dp , satisface la relación

$$di = G \cdot dp$$

Al ser la superficie la que se mueve, y no el punto, se tiene

$$di = -G \cdot ds \quad (2)$$

Las coordenadas polares, tanto del gradiente, como de la velocidad son:

$$G = (|G|, \theta_G) \quad \text{y} \quad v = (|v|, \theta_v)$$

De (1) y (2) obtenemos

$$di = -G \cdot v dt = -|G||v| \cos(\theta_G - \theta_v) dt$$

Sea v_G la magnitud del vector velocidad proyectado sobre el vector gradiente:

$$v_G = -|v| \cos(\theta_G - \theta_v) = \frac{di/dt}{|G|} \quad (3)$$

Si la velocidad permanece constante en la imagen, los valores de v_G tomados contra θ_G se encontrarán a lo largo de una curva coseno de amplitud $|v|$ y fase $-\theta_v$. Si tenemos una variación suficiente de θ_G la relación entre v_G y θ_G definirá v de forma única.

Ya que el análisis anterior solamente es válido si el gradiente en un punto permanece constante a lo largo del tiempo entre muestras, solamente se consideran aquellos puntos en los que el gradiente en los tiempos t_1 y t_2 satisfaga las restricciones

$$\left[|G(t_1)| - |G(t_2)| \right] < a_1 \quad \text{y}$$

$$\theta_G(t_1) - \theta_G(t_2) < a_2 \quad \text{o}$$

$$2\pi - [\theta_G(t_1) - \theta_G(t_2)] < a_2$$

Además, con el fin de eliminar el ruido, únicamente se procesan los puntos para los cuales el cambio de intensidad entre tramas es suficientemente grande:

$$|\Delta I| > a_3$$

Para incrementar considerablemente el número de puntos que satisfagan las restricciones anteriores, la imagen puede ser filtrada con filtros paso bajo, lo que resulta en una operación de uniformizado.

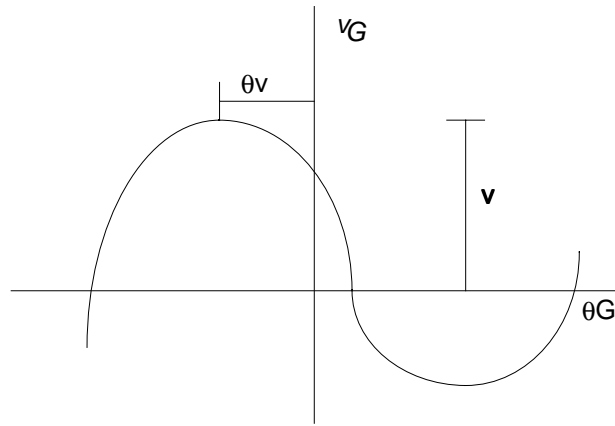


Figura 1.8. Los valores de v_G frente a θ_G forman una curva coseno de amplitud $|v|$ y fase $-\theta$

La velocidad verdadera $(|v|, \theta_v)$ se determina con la utilización de una variante de la transformada de Hough. En cada uno de los puntos, se usa ΔI como una estimación de di/dt . De (3), obtenemos que los valores posibles de v que pueden dar lugar a ΔI en ese punto están restringidos a pertenecer a la curva

$$|v| = \frac{v_G}{\cos(\theta_G - \theta_v)} = -\frac{\Delta I}{|G| \cos(\theta_G - \theta_v)} \quad (4)$$

Si la dirección del gradiente cambia en puntos diferentes de un objeto en movimiento, entonces los valores en estos puntos pertenecerán a curvas diferentes del espacio $(|v|, \theta_v)$. La intersección de estas curvas determinará v de forma única.

La resolución de la intersección de forma analítica es una tarea compleja, pero la aproximación mediante la transformada de Hough permite una simplificación computacional importante. El espacio $(|v|, \theta_v)$ es tratado como un vector discreto de acumuladores. En cada punto de imagen, ΔI y G se usan en la ecuación (4) para trazar una curva a través del espacio $(|v|, \theta_v)$ que corresponda a las posibles velocidades. Esta curva se digitaliza en una malla acumuladora y se incrementa cada uno de los acumuladores correspondientes a un punto de la curva. Cuando se han introducido las curvas correspondientes a todos los puntos de la imagen, tendremos que un pico en el vector acumulador se corresponde con una velocidad.

Allí donde tengamos más de un objeto en movimiento, las diferentes superficies en movimiento se harán manifiestas por medio de los diferentes picos en los acumuladores. En primer lugar se toma el pico más prominente de todos en el vector acumulador. La velocidad correspondiente se denomina v_1 y a todos los puntos que satisfagan

$$|v_1| \cos(\theta_G - \theta_{v_1}) + \left| \frac{\Delta I}{G} \right| < t$$

para un umbral t apropiado se les asignará la velocidad v_1 . Se repite el análisis utilizando la imagen original menos los puntos que ya tienen asignada una velocidad. Este procedimiento se repite hasta que no queden más picos suficientemente definidos en el vector acumulador.

El modelo del flujo óptico de Horn y Schunck [Hor81].

A partir del artículo "Determining Optical Flow" de Horn y Schunck se comienza a tomar muy en serio el concepto de flujo óptico, al describirse por primera vez un esquema para la determinación del flujo óptico de una imagen.

Antes de proseguir, queda evidente la necesidad de explicar qué se entiende por flujo óptico. El flujo óptico es la distribución de las velocidades aparentes del movimiento de formas brillantes en una imagen. El flujo óptico, al provenir del movimiento relativo de los objetos y del observador, pueda revelar una información importante de los objetos observados y el ratio de cambios en la orientación. Las discontinuidades en el flujo óptico pueden igualmente ayudar a segmentar las imágenes en regiones que se corresponden con objetos diferentes.

Horn y Schunck parten de la idea inicial de que el flujo óptico no puede ser calculado en un punto de la imagen con independencia de sus puntos vecinos, a menos que se introduzcan unas restricciones adicionales. Y ello porque el campo velocidad tiene dos componentes en cada punto de la imagen, mientras que el cambio en el brillo de la imagen debido al movimiento lleva asociada una sola restricción en un punto en el plano de la imagen. El movimiento de una forma en una dirección altera el brillo en un punto determinado, pero el movimiento en la otra dirección no

conlleva cambio alguno. Por ello, no pueden determinarse localmente los componentes del movimiento en esta última dirección.

En su disertación, los autores restringen el modelo a un mundo particular sencillo en el que la velocidad aparente de las formas brillantes puede ser identificada directamente con el movimiento de las superficies en la escena. Se toma como premisa igualmente que las superficies estudiadas son planas, evitando de este modo las variaciones de brillo debidas a las sombras. Otra suposición es que la iluminación incidente es uniforme en toda la superficie. El brillo en un punto de la imagen será proporcional a la reflexión de la superficie en el punto correspondiente del objeto. Se considera igualmente que la variación en la reflexión es uniforme y que no existen discontinuidades espaciales. Esta última condición asegura que el brillo de la imagen es una función diferenciable.

En primer lugar se busca una ecuación que relacione el cambio en el brillo de la imagen en un punto con el movimiento del brillo de la forma. Llamando $E(x,y,t)$ al brillo de la imagen en un punto (x,y) del plano de la imagen en el tiempo t , y considerando que el brillo en un punto dado de la forma es constante, se tiene

$$\frac{dE}{dt} = 0$$

Utilizando la regla de la cadena para la diferenciación

$$\frac{\partial E}{\partial x} \frac{dx}{dt} + \frac{\partial E}{\partial y} \frac{dy}{dt} + \frac{\partial E}{\partial t} = 0$$

Llamando

$$u = \frac{dx}{dt} \quad y \quad v = \frac{dy}{dt}$$

es posible ver que nos encontramos ante una ecuación lineal con dos incógnitas u y v

$$E_x u + E_y v + E_t = 0$$

donde se han introducido las abreviaturas adicionales E_x, E_y y E_t para las derivadas parciales del brillo de la imagen respecto de x, y y t , respectivamente. La restricción en la velocidad de flujo local expresada en esta ecuación se ilustra en la siguiente figura 2.3.

Puede escribirse la ecuación en la siguiente forma:

$$(E_x, E_y) \cdot (u, v) = -E_t$$

Así la componente del movimiento en la dirección del gradiente de brillo (E_x, E_y) es igual a

$$-\frac{E_t}{(E_x^2 + E_y^2)^{1/2}}$$

Sin embargo, no puede calcularse la componente del movimiento en los ángulos rectos respecto del gradiente del brillo. Por ello, la velocidad del flujo (u, v) no puede ser calculado localmente sin introducir restricciones adicionales.

De este modo, los autores introducen el concepto de restricción de uniformidad. Si cada uno de los puntos de la forma de brillo pudiera moverse independientemente, no quedarían muchas posibilidades para recuperar la velocidad. Sin embargo, puede afirmarse que los puntos vecinos de un mismo objeto tienen velocidades similares y que el campo velocidad de las formas de brillo de la imagen varía suavemente casi en todas partes. Las discontinuidades en el flujo se asocian a aquellos puntos en los que un objeto esconde a otro.

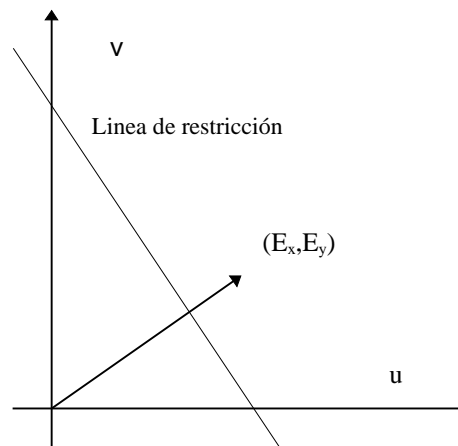


Figura 1.9. La velocidad (u, v) se encuentra en una recta perpendicular al vector (E_x, E_y) .

Una manera de expresar esta restricción adicional es minimizando el cuadrado de la magnitud del gradiente de la velocidad del flujo óptico:

$$\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 \quad \text{y} \quad \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2$$

La estimación de las derivadas parciales del brillo E_x, E_y y E_t se debe hacer de un modo consistente, y para ello se usa un conjunto que da la estimación de E_x, E_y y E_t en un punto en el centro de un cubo formado por ocho medidas. Cada una de las estimaciones es la media de cuatro primeras diferencias tomadas en medidas adyacentes en el cubo.

$$E_x \approx \frac{1}{4} (E_{i,j+1,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i+1,j,k} + E_{i,j+1,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i+1,j,k+1})$$

$$E_y \approx \frac{1}{4} (E_{i+1,j,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i,j+1,k} + E_{i+1,j,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i,j+1,k+1})$$

$$E_t \approx \frac{1}{4} (E_{i,j,k+1} - E_{i,j,k} + E_{i+1,j,k+1} - E_{i+1,j,k} + E_{i,j+1,k+1} - E_{i,j+1,k} + E_{i+1,j+1,k+1} - E_{i+1,j+1,k})$$

donde $E_{i,j,k}$ es la medida de brillo en la intersección de la línea i y columna j en la secuencia de imagen k . La unidad de longitud es el intervalo de espaciado de la malla en cada secuencia de imagen y la unidad de tiempo es el periodo de muestreo de la secuencia de imagen.

Del mismo modo se procede a aproximar las Laplacianas de u y v . Una aproximación válida toma la forma

$$\nabla^2 u \approx \kappa (u^m_{i,j,k} - u_{i,j,k}) \quad \text{y} \quad \nabla^2 v \approx \kappa (v^m_{i,j,k} - v_{i,j,k})$$

Tomando la siguiente asignación de pesos a los puntos vecinos

1/12	1/6	1/12
1/6	-1	1/6
1/12	1/6	1/12

las medias locales u^m y v^m se definen del siguiente modo:

$$u^m_{i,j,k} = \frac{1}{6} \{u_{i-1,j,k} + u_{i,j+1,k} + u_{i+1,j,k} + u_{i,j-1,k}\} +$$

$$\frac{1}{12} \{u_{i-1,j-1,k} + u_{i-1,j+1,k} + u_{i+1,j+1,k} + u_{i+1,j-1,k}\}$$

$$v^m_{i,j,k} = \frac{1}{6} \{v_{i-1,j,k} + v_{i,j+1,k} + v_{i+1,j,k} + v_{i,j-1,k}\} +$$

$$\frac{1}{12} \{v_{i-1,j-1,k} + v_{i-1,j+1,k} + v_{i+1,j+1,k} + v_{i+1,j-1,k}\}$$

El factor de proporcionalidad κ es igual a 3 cuando se calcula del modo anterior y de nuevo se supone que la unidad de longitud es igual al intervalo de espaciado de la malla.

El problema queda pues reducido a minimizar la suma de los errores en la ecuación del ratio de cambio en el brillo de la imagen

$$\xi b = E_x u + E_y v + E_t$$

y de la medida de la separación de la uniformidad en la velocidad del flujo

$$\xi^2 = (\partial u / \partial x)^2 + (\partial u / \partial y)^2 + (\partial v / \partial x)^2 + (\partial v / \partial y)^2$$

El error total a minimizar es

$$\xi^2 = \int (\alpha^2 \xi_c^2 + \xi_b^2) dx dy$$

siendo α^2 un factor de peso que debiera ser aproximadamente igual al ruido esperado en la estimación de $E_x^2 + E_y^2$. La minimización tiene como objetivo encontrar unos valores aceptables para la velocidad (u, v) del flujo óptico. Se obtiene

$$E_z^2 u + E_z E_y v = \alpha^2 \nabla^2 u - E_z E_t$$

$$E_x E_y u + E_y^2 v = \alpha^2 \nabla^2 v - E_y E_t$$

Al resolver estas ecuaciones se obtiene

$$(\alpha^2 + E_x^2 + E_y^2)(u - u^m) = -E_x [E_x u^m + E_y v^m + E_t]$$

$$(\alpha^2 + E_x^2 + E_y^2)(v - v^m) = -E_y [E_x u^m + E_y v^m + E_t]$$

habiéndose obtenido una pareja de ecuaciones para cada punto de la imagen. El método de resolución del sistema escogido por Horn y Schunck es un método iterativo de tipo Gauss-Seidel. Se calcula un nuevo conjunto de estimaciones de velocidad (u^{n+1}, v^{n+1}) a partir de las derivadas parciales y de la media de las estimaciones de velocidad previas (u^n, v^n)

$$u^{n+1} = u^{mn} - E_x [E_x u^{mn} + E_y v^{mn} + E_t] / (\alpha^2 + E_x^2 + E_y^2)$$

$$v^{n+1} = v^{mn} - E_y [E_x u^{mn} + E_y v^{mn} + E_t] / (\alpha^2 + E_x^2 + E_y^2)$$

El modelo de determinación de la dirección de Marr y Ullman [Mar81].

En contraste con otros esquemas basados en el análisis del gradiente, el modelo de Marr y Ullman calcula solamente la dirección del movimiento (indicado por el signo de la derivada temporal) y no la velocidad (indicada por la magnitud de los cambios en la intensidad espacio-temporal), y restringe los cálculos a los bordes de la imagen.

Este modelo parte de la construcción de unidades de selección de la dirección. La imagen es filtrada por campos receptivos para la detección de pasos por cero de la función

$$\nabla^2 G(x, y) * I(x, y)$$

La idea básica es que los cambios en la intensidad pueden detectarse buscando los pasos por cero de la convolución del operado $\nabla^2 G$ con la imagen, siendo ∇^2 la Laplaciana y G una distribución

bidimensional, radial y simétrica de Gauss.

Además se mide

$$\frac{\partial[\nabla^2 G(x, y) * I(x, y)]}{\partial t}$$

en los pasos por cero, permitiendo restringir la dirección local del movimiento a 180°.

El modelo de predicción de la dirección del movimiento de Lawton [Law89].

Este otro modelo basado en la obtención del gradiente, calcula la dirección, tomando la diferencia de la salida de la suma y de la diferencia de parejas multiescaladas de filtros Gabor simétricos pares e impares en dos intervalos diferentes de tiempo.

Las expresiones matemáticas que describen una función Gabor simétrica par F_{ES} y una función Gabor simétrica impar F_{OS} son

$$F_{ES}(f, x, \sigma^2) = \cos(2\pi fx) \cdot e^{-(x-x_0)^2/2\sigma^2}$$

siendo f la frecuencia espacial de la forma

x la posición espacial horizontal examinada

x_0 la posición de fijación del observador

σ^2 la variabilidad del periodo espacial para localizar la posición x_0 de la diferencia más acuciada del contraste

La dirección del movimiento de una forma respecto del fondo de la imagen en los intervalos diferentes de tiempo se obtiene cuando la salida de los mecanismos que calculan la diferencia y la suma de parejas de canales paso banda simétricos pares e impares escalados a través del fondo de referencia exceden el umbral.

Sin lugar a dudas, los modelos basados en la obtención del gradiente espacio-temporal, que miden el movimiento a partir de la distribución de la luminiscencia pura [Fen79] [Hor81] son modelos biológicamente defendibles. Al menos más que aquellos que procesan en primer lugar la imagen con filtros paso banda de frecuencia espacial orientada o no orientada [Mar81] [Law89].

Los modelos que incluyen filtros de frecuencias espaciales miden la velocidad en lugares donde los cambios de intensidad son significativos (por ejemplo, en los bordes). Al contrario, los modelos sin filtros también pueden calcular la velocidad a partir de gradientes lineales de

intensidad. Las evidencias psicofísicas y fisiológicas, no obstante, indican que el sistema visual es bastante insensible a los gradientes lineales de intensidad - los humanos no perciben bien los gradientes de intensidad, ya sean estacionarios [McC74] o móviles [Nak83]. Así pues, las neuronas en el sistema visual (véase LGN y el área V1) no responden bien a gradientes lineales de intensidad o a niveles absolutos de luminiscencia.

Además, un sistema de movimiento que opera en valores de intensidad pura es considerablemente más susceptible de confundir factores tales como los cambios globales en el nivel de iluminación, que son, a su vez, causantes de los cambios locales en la intensidad. De este modo, los cambios en la iluminación ambiental de las superficies pueden ser incorrectamente interpretados como movimientos de las propias superficies. Contrariamente a esto, los gradientes de intensidad pronunciados en los bordes de los objetos son mucho más estables bajo estas condiciones, así como bajo condiciones de ruido en general.

1.2.2.1.2. Modelos basados en la correlación

El segundo tipo de modelos calcula la dirección a partir de las correlaciones en las actividades de los campos receptivos de entrada vecinos. Los cambios en un campo receptivo de entrada son correlacionados con los cambios en los campos receptivos vecinos después de un intervalo de tiempo, claro está, siempre que el movimiento sea en la dirección de esos vecinos. Estos modelos comparten algunas propiedades organizativas comunes:

- (1) un par de campos receptivos de entrada de selección de la posición espacial, y muchas veces también la frecuencia espacial de un estímulo
- (2) una combinación de señales de los dos campos receptivos, de modo que se obtiene una respuesta direccional selectiva
- (3) una o varias etapas de integración espacial y/o temporal

Todos estos modelos calculan la dirección, pero no calculan explícitamente la velocidad.

El modelo de selectividad relacional de Reichardt y Hassenstein [Has56].

En el modelo de Hassenstein y Reichardt, las señales de un par de fotorreceptores (uno de ellos retardado por un filtro temporal paso baja) se multiplican para determinar la similitud en la

correlación de las dos entradas, y a continuación se comparan con la salida de una subunidad complementaria para determinar la direccionalidad.

La respuesta a este detector analógico, después de la etapa de multiplicación, viene especificada por

$$R(t) = \iint W(t_1, t_2) I(t - t_1) I(t - t_2) dt_1 dt_2$$

donde $W(t_1, t_2)$ representa la función de transferencia amplificada para los diferentes filtros.

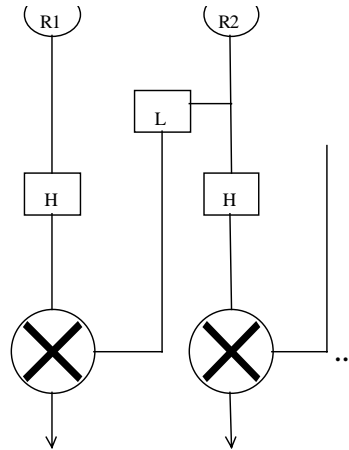


Figura 1.10. Modelo de detección de movimiento de Hassenstein y Reichardt (1956)

El modelo de selectividad de la dirección de Barlow y Levick [Bar65].

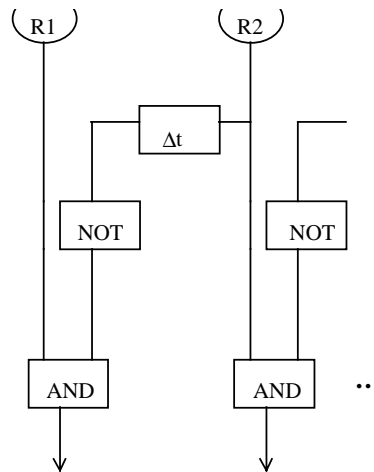


Figura 1.11. Modelo de detección de movimiento de Barlow y Levick (1965)

El modelo de Barlow y Levick propone que una función discreta AND-NOT puede aplicarse a las señales de entrada de modo que el detector de movimiento responda solamente cuando la primera entrada esté en ON y la segunda entrada, desfasada, esté en OFF.

Otros dos modelos son lineales en sus etapas iniciales en que las salidas de los dos canales de entrada son sumadas y no multiplicadas [Ade85] [Wat85]. Se adquiere la selectividad direccional a través de la suma de las señales de entrada en vez de vía un esquema de retardo y de correlación. Estos modelos tienen filtros de entrada sincronizados a diferentes frecuencias espaciales y temporales.

Las representaciones del dominio de frecuencia espacio-temporal y del dominio espacio-tiempo son intercambiables para un sistema lineal. Por ello, el movimiento visual de una imagen, representado normalmente como una función de luminiscencia del espacio y del tiempo - $L(x,y,t)$ - puede ser expresada en términos de frecuencia tomando la transformada de Fourier de $L(x,y,t)$ para obtener $L(w_x, w_y, w_t)$. De modo similar, la respuesta a un filtro de entrada puede ser definida como una función de respuesta a un impulso en el dominio espacio-tiempo (lo que es una descripción de la respuesta de un filtro a pequeños manchas de luz a través del espacio y del tiempo) o como una función de transferencia de un sistema en el dominio de frecuencia espacio-temporal (lo que es una descripción a la respuesta de un filtro a sinusoides de todas las frecuencias). Estas representaciones son equivalentes en un sistema lineal: la convolución de la imagen con una función de respuesta a impulso es igual a la transformada inversa de Fourier del producto de la función de transferencia del sistema con la transformada de Fourier de la imagen.

Se estudia a continuación la representación de la velocidad de la imagen en el dominio de Fourier, así como una descripción en el dominio espacial de los modelos. Como ejemplo de esta representación, considérese un caso simplificado en el que el movimiento de un estímulo simple (una onda senoide reticular) está restringido a una dimensión espacial (por ejemplo, cuando se mueve horizontalmente). Una representación de frecuencia espacio-temporal de una retícula moviéndose hacia la derecha consiste en un sistema de coordenadas de dos ejes - un eje x de frecuencia espacial horizontal y un eje y de frecuencia temporal. La representación de la frecuencia de una retícula yendo hacia la derecha consiste en dos puntos posicionados simétricamente respecto del origen - un punto en el cuadrante derecho superior y el otro en el cuadrante izquierdo inferior. Se asigna un valor al punto (su energía) que es proporcional a la amplitud de la senoide. Los dos puntos se encuentran en una diagonal que pasa por el origen. La velocidad de una senoide en movimiento es simplemente la frecuencia temporal dividida por la frecuencia espacial:

$$V_x = FT/FE$$

ya que la pendiente de la diagonal que pasa a por el origen es igual a FT/FE , éste representa la velocidad de la retícula.

Cualquier estímulo complejo moviéndose a una velocidad particular en una dimensión, cuando viene representado en el dominio de frecuencia, tendrá valores de energía de frecuencia espacio-temporal situados por debajo de una línea recta que pasa por el origen. La pendiente de la línea representa la velocidad y el signo de la pendiente representa la dirección. Cuando se representan ambas dimensiones espaciales (x e y), el espacio de Fourier es tridimensional, y los estímulos coherentes de movimiento tienen valores de energía espacio-temporal por debajo de un plano que pasa por el origen. Los detectores representados en el espacio de Fourier responden a regiones limitadas de este espacio.

Dos modelos han sido construidos con detectores de movimientos sincronizados a diferentes rangos de frecuencias espaciales y temporales [Ade85] [Wat85]. Un único filtro tendrá una activación máxima para una retícula que se mueva en la dirección apropiada y que posea la frecuencia espacial y temporal apropiada. La construcción de filtros de este tipo requieren de varios pasos. En primer lugar, se definen dos impulsos espaciales respuesta y dos impulsos respuesta temporales. Estas funciones describen simplemente cómo un campo receptivo suma sus entradas en varias posiciones y tiempos para determinar la salida, véase offsets en fases espaciales y temporales (generalmente, por cuadraturas, o sea, fases de 90 grados). Los dos campos receptivos espaciales son, por ejemplo, dos funciones Gabor - esto es, las funciones seno y coseno ponderadas por una función de Gauss. Estos perfiles de campos receptivos se parecen a aquellos de las células simples y son espacialmente simétricos (siempre que estén generados por una función coseno) o antisimétricos (cuando estén generados por una función seno).

El modelo de energía espacio-temporal de Adelson y Bergen [Ade85].

El modelo de Adelson y Bergen utiliza una representación del dominio espacio-tiempo [Fah81]. Los estímulos de movimiento se representan en un espacio tridimensional. x e y son las dos dimensiones espaciales y t es la dimensión temporal.

El movimiento puede ser percibido en imágenes continuas o muestreadas cuando hay energía de la orientación espacio-temporal apropiada. El problema de la detección de movimiento es pues el problema de la detección de la orientación espacio-temporal.

En este modelo, todo impulso respuesta espacial se multiplica por todo impulso respuesta temporal.

$$H_{st}(x,t) = H_s(x) \cdot H_t(t)$$

siendo $H_s(x)$ el impulso respuesta espacial, $H_t(t)$ el impulso respuesta temporal

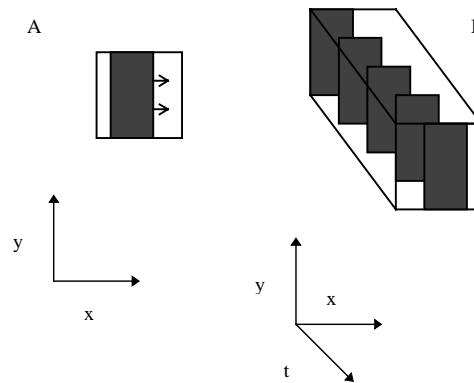


Figura 1.12. A. Una representación de la barra vertical moviéndose hacia la derecha. B. Una representación espacio-temporal de la barra muestreada a lo largo del tiempo. El tiempo forma la tercera dimensión.

A continuación, los filtros resultantes se suman y se restan para producir cuatro filtros selectivos direccionales - un par de filtros selectivos hacia la izquierda y un par de filtros selectivos hacia la derecha, estando los miembros de cada pareja en cuadratura. Se toma la raíz cuadrada de las respuestas a cada par y se suman, obteniendo la energía asociada con una dirección particular de movimiento. Esto se hace de este modo para que la respuesta de la unidad selectiva direccional sea independiente de la fase, o sea, no oscilará y no cambiará el signo a través del tiempo dependiendo de la correlación entre los estímulos y el perfil del campo receptivo. Finalmente, los detectores de selección de las direcciones opuestas de movimiento se combinan, formando un canal oponente. El modelo real consta de tres detectores usados para medir la energía de movimiento hacia la izquierda, hacia la derecha y estacionaria.

El modelo de la visión humana del movimiento de Watson y Ahumada [Wat85].

El modelo de Watson y Ahumada usa una representación del dominio de frecuencia espacio-temporal del movimiento de la imagen y de los campos receptivos de entrada. El modelo se construye en dos pasos.

El primer paso consta de un vector de sensores lineales que seleccionan la localización 2-D, la frecuencia espacial 2-D y la dirección. A cada uno de estos sensores se les denomina sensor escalar de movimiento.

El filtro temporal básico viene determinado por la respuesta al impulso $f(t)$ y la función de transferencia. La respuesta del impulso es:

$$f(t) = \xi[f_1(t) - \xi f_2(t)]$$

donde

$$f_i(t) = \frac{u(t)}{\tau(n_i - 1)!} \left(\frac{t}{\tau}\right)^{n_i - 1} e^{-t/\tau}$$

siendo $u(t)$ la función de paso unidad. La función de transferencia viene dada por la transformada de Fourier de la respuesta del impulso.

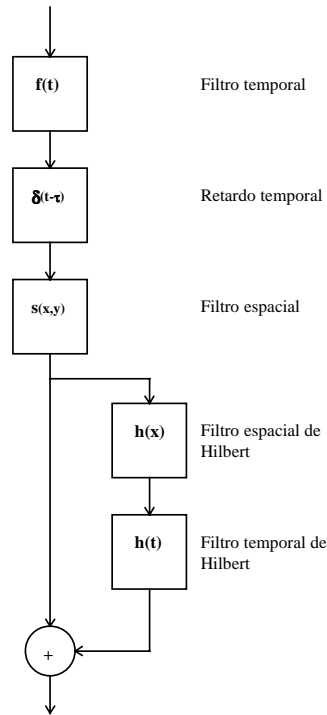


Figura 1.13. Estructura matemática del sensor de movimiento escalar

El filtro espacial está asociado a una respuesta de impulso $g(x, y)$ separable en sus componentes horizontal y vertical

$$g(x, y) = a(x) \cdot b(y)$$

$$a(x) = \exp\left[-(x/\lambda)^2\right] \cos(2\pi u, x)$$

$$b(x) = \exp\left[-(y/\lambda)^2\right]$$

La transformada de Hilbert para una función $g(x)$ viene dada por $h(x) g(x)$, donde

$$h(x) = -1/\pi x$$

El segundo paso del modelo trata de la obtención de la velocidad y de la dirección de una forma a partir de la cooperación entre un número de sensores preparados para la detección de diferentes direcciones.

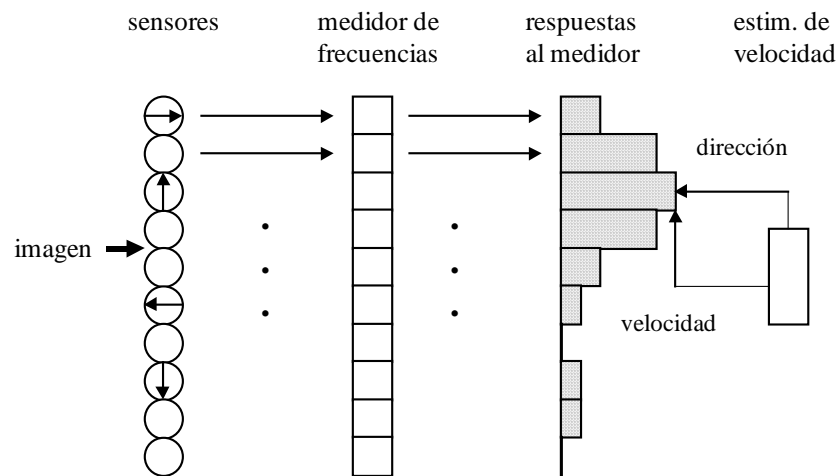


Figura 1.14. La estructura del vector de sensores de movimiento

Los modelos recientes de detección de movimiento son notables en su intento de construir detectores de movimiento a partir de las propiedades básicas de filtrado espacial y temporal de las neuronas. Sin embargo, queda aún un largo camino por recorrer hasta relacionar estos modelos más íntimamente con los componentes de la circuitería neuronal real. Muchos de estos modelos necesitan de cinco a seis pasos jerárquicos, multiplicando, sumando, restando, elevando al cuadrado, y sumando las entradas iniciales.

Una implementación neurofisiológica aceptable de estos pasos no es inmediatamente evidente. En la corteza primaria se encuentra un alto porcentaje de las células de selección direccional en la capa 4B, que es la que provee de la mayor proyección al área MT. Esta es un importante área para el análisis del movimiento. Las neuronas en la capa 4B, a su vez, reciben su entrada de las neuronas de la capa 4C α . Así, los cinco o seis pasos propuestos deberían ser implementados entre estas dos capas. Además, estos modelos proponen que un par de campos receptivos de células simples proporcionan la entrada al detector de movimiento. Esta suposición es cuestionable, al existir informes conflictivos de si la entrada de la capa 4C α a las neuronas de selección de movimiento de la capa 4B consiste en neuronas de selección de orientación [Bul80] [Haw84] o en neuronas que no seleccionan la orientación [Bla84] [Hub77].

Una segunda consecuencia de tener células simples de selección no direccional alimentando a un detector de movimiento de célula compleja de selección direccional está en la presunción subyacente que las células simples de selección direccional no juegan ningún papel en la percepción

y la detección del movimiento. Muchas de las neuronas de la capa 4B podrían ser células simples de selección de la dirección.

En tercer lugar, los modelos postulan que solamente dos campos receptivos (y no un número mayor) sirven como entrada al detector de movimiento, sugiriendo que la activación óptima de un detector puede ser producida por un par de estímulos idénticos, en los cuales la presentación del segundo estímulo se desplaza ligeramente en el espacio y en el tiempo respecto de la presentación del primer estímulo. La evidencia psicofísica contradice esta predicción. Algunos estudios, utilizando una variedad de paradigmas, indican que la presentación de múltiples tramas de un estímulo de movimiento aparente puede:

1. incrementar en gran medida la probabilidad de detectar el movimiento en manifestaciones de puntos ocasionales [Lap82]
2. incrementar la distancia máxima por desplazamiento para la percepción correcta de la dirección del movimiento en un estímulo de movimiento aparente de rango corto y de puntos ocasionales - una secuencia de dos desplazamientos incrementaron grandemente (50 %) el desplazamiento máximo aceptable por paso comparado con un desplazamiento [Nak84]
3. incrementar el rendimiento de la discriminación de la velocidad - una secuencia de cinco a ocho presentaciones de un estímulo de movimiento aparente (un corte de luz) fue necesario para igualar el rendimiento de discriminación de la velocidad obtenido con blancos en movimiento continuo [McK85]

Otra suposición crucial de los modelos está en que los filtros son lineales. Para activar las unidades de entrada, algunos modelos toman una transformada de Fourier del modelo, que la convierte del dominio espacio-tiempo al dominio de frecuencias espacio-temporales. Las neuronas poseen propiedades no lineales intrínsecas como la saturación y la rectificación. Así, al contrario que los modelos propuestos, las neuronas reales no serían capaces de descomponer un estímulo de movimiento complejo en sus componentes de frecuencias espacio-temporales.

Finalmente, una característica que falta en estos modelos es la inhibición. Esta sección concluye con la discusión de la evidencia de un mecanismo inhibitorio en la detección del movimiento, así como con un modelo que incorpora esta característica. Se ha demostrado por medios fisiológicos [Bar65] [Eme81] y farmacológicos [Sil77] que la inhibición desempeña un papel crucial en la obtención de la selección direccional.

El modelo de energía espacio-temporal de Heeger [Hee87].

Algunos modelos de energía de detección del movimiento, como se ha visto al hablar de los modelos de detección local de movimiento, tienen filtros 1-D, y por ello, solamente pueden detectar el movimiento en una dimensión espacial [Ade85]. Otros modelos emplean filtros selectivos de orientación 2-D [Wat85] [Hee87]. En estos modelos, cada uno de los canales de selección de la orientación mide el componente de movimiento 1-D perpendicular a su orientación.

Heeger, por ejemplo, elabora los filtros de Gabor del dominio espacio-tiempo 1-D a filtros Gabor 2-D. Cada uno de los filtros orientados tiene definidas sus esferas de sensibilidad en el dominio de frecuencias espacio-temporales 3-D. La velocidad de una forma compleja en movimiento define un plano en el espacio de frecuencias espacio-temporales. Si se presenta una forma con una sola orientación (por ejemplo, una retícula de onda sinusoidal) el problema de la apertura se presenta a sí mismo como que la energía de una imagen de este tipo se restringe a una línea que puede estar contenida en muchos planos (puede estar representada por muchas velocidades). La solución al problema de la apertura es posible solamente si se presenta más de una orientación en la imagen. El modelo resuelve el problema de la apertura en varios pasos:

- (1) un conjunto de filtros de energía Gabor es asociado a diferentes porciones del espacio de frecuencias espacio-temporales 3-D
- (2) una imagen es convertida del dominio espacio-tiempo al dominio de frecuencias espacio-temporales
- (3) el contenido de la frecuencia espacio-temporal de la imagen causa una actividad distribuida a través de estos filtros
- (4) se hace un estimación de mínimos cuadrados para determinar el plano que mejor se acopla a la forma distribuida de actividad
- (5) este plano define la velocidad de la imagen (o porción de la imagen)

Los modelos energéticos de movimiento son atractivos en su uso de filtros espacio-temporales que les permiten detectar el movimiento directamente a partir de las secuencias de imágenes. Mientras que está claro que los detectores de movimiento en los diversos niveles del sistema visual son sensibles a los aspectos espaciales y temporales de las imágenes en movimiento, no está tan claro que efectúen una descomposición exacta en frecuencias espacio-temporales, tal como hacen estos modelos. La falta de un conjunto base adecuado de filtros espaciales y temporales, así como la falta de linealidad inherente en la respuesta neuronal puede estar en el origen de esta falta de descomposición [Val88].

1.2.2.1.3. Otros modelos

Otros modelos son difíciles de catalogar en los vistos hasta el momento [Fer92].

El modelo de memorias de permanencia de Fernández Graciani y Mira [Fer97].

Se plantea un método aplicable en un amplio número de casos reales, y con posibilidad de una realización hardware para su funcionamiento en tiempo real: los sistemas basados en memorias de permanencia. Estos sistemas están próximos a las técnicas de diferenciación, pero incorporan las suficientes novedades como para hacerlos mucho más funcionales y útiles.

Esta solución parte de secuencias de imágenes binarias, segmentadas mediante métodos de umbralización sobre nivel de gris o sobre otro tipo de parámetros. Sobre ello se procede a comentar su concepto, funcionamiento y limitaciones.

Se denomina memoria de permanencia a un sistema formado por una memoria y una lógica con las siguientes propiedades:

- (i) La memoria debe tener una capacidad de almacenamiento, tal que se almacenen tantos datos como puntos de imagen sea capaz de detectar el sensor en un cuadro de imagen, o al menos tantos como la zona de imagen que se desee procesar.
- (ii) La lógica realizará sobre los datos de la memoria las operaciones siguientes (los valores 0 o 1 corresponden a la imagen ya binarizada):
 - si el punto de imagen (x,y) tiene, para este cuadro, el valor uno, incrementará el valor del dato almacenado en la posición (x,y) de la memoria de permanencia en una cantidad S hasta un máximo MAX
 - si el punto de imagen (x,y) correspondiente, tiene un valor de cero, restará al dato almacenado en la posición (x,y) de la memoria de permanencia una constante R hasta un mínimo MIN

Tendremos pues, que el valor de cada dato de esta memoria de permanencia dependerá del intervalo de tiempo que haya transcurrido desde que la imagen binarizada tomó el valor uno en ese mismo punto. Se tiene como resultado que un punto de coordenadas (x,y) se "activa bruscamente" cuando la imagen toma el valor uno en ese punto, y se "desactiva lentamente" mientras la imagen no vuelva a tomar valor uno en ese punto. Elijiendo convenientemente los valores de S, R, MAX y MIN, se obtiene distinta información acerca de la historia más inmediata de cada punto del sensor.

En la figura 1.15., puede verse gráficamente una secuencia, representando mediante barras el valor de cada dato, en una zona de la memoria de permanencia.

En el primer cuadro de la secuencia de imágenes (ya binarizada) aparece como activado (1) el

punto E5. En la memoria de permanencia, el dato cuya dirección coincide con la posición activada, sumará S a su valor. Este viene indicado elevando en S la altura de la columna correspondiente.

En el segundo cuadro de la secuencia, el punto que aparece activado es D4, debido al movimiento del elemento de imagen. Su posición correspondiente en la memoria de permanencia sumará S. Y la del punto E5 restará R. Con la afluencia de nuevas imágenes de la secuencia, el punto móvil irá creando una estela en la memoria de permanencia. El análisis de esta estela proporcionará información acerca del movimiento del punto.

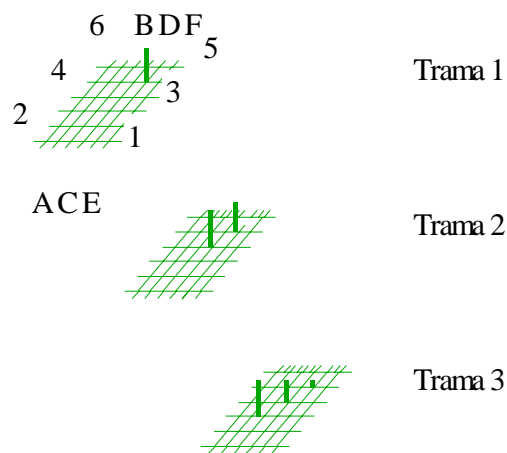


Figura 1.15. Secuencia de imágenes

Se realiza ahora un análisis de la memoria de permanencia para la obtención de parámetros de movimiento. Las memorias de permanencia pueden aplicarse en diversos tipos de análisis de secuencias de imágenes. Se intenta, a continuación, clasificar los tipos de análisis que pueden realizarse y hacer ver cómo puede obtenerse a partir de este análisis la información deseada.

Se diferencian dos tipos de procesos: locales y globales. En el caso de procesos locales no se somete a proceso a todos los puntos de la memoria de permanencia, sino a un subconjunto de ellos. Un caso de este tipo es el acceso aleatorio de un procesador a la memoria de permanencia. El procesador, mediante el análisis de estelas concretas, puede realizar el cálculo del módulo de la velocidad y de la trayectoria seguida por el elemento en su historia más inmediata. Otra amplia familia de posibilidades la forman los procesos globales. Mediante hardware específico, puede procesarse la memoria de permanencia por métodos de gradiente para obtener información acerca de qué elementos están asociados a velocidades concretas, qué elementos difieren en su trayectoria con respecto al resto o cuál es la cantidad de movimiento en la escena. Es conveniente hacer notar que es posible concebir sistemas que realicen sobre la misma memoria de permanencia análisis de tipo global y local simultáneamente.

Lo anteriormente expuesto puede verse gráficamente en el siguiente ejemplo. A partir de la figura

1.16., en la que puede verse el estado de la memoria de permanencia transformada en niveles de gris, se muestra, en la correspondiente gráfica 1.17., los resultados de algún caso intencionadamente sencillo, que permite ilustrar el comportamiento de las memorias de permanencia.

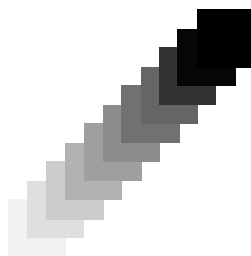


Figura 1.16. Memoria de permanencia actualizada mediante una secuencia en la que un cuadrado se mueve con movimiento rectilíneo y uniforme

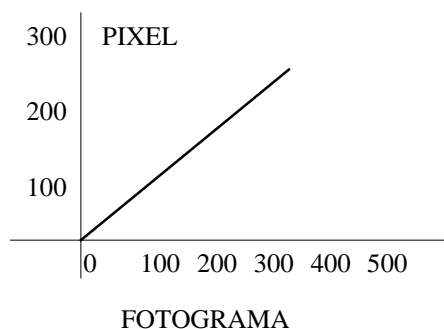


Figura 1.17. Gráfica espacio-tiempo (pixel/cuadro) obtenida analizando la memoria de permanencia de la figura 1.16

Como ventaja de este método, se puede destacar la posibilidad del sistema de ser realizado en hardware para trabajar en tiempo real sobre secuencias de imágenes. En concreto, el método permite obtener información acerca del movimiento de elementos de imagen. Lo que permite analizar vibraciones, segmentar la imagen en zonas de flujo de direcciones distintas, detectar elementos de movimiento característico, etc.

Entre las limitaciones de este tipo de sistemas se encuentran las variaciones en la etapa de binarización, no referentes al movimiento de los elementos de imagen. También aparecen limitaciones en cuanto a la densidad de los puntos en la escena, ya que una densidad alta de puntos conlleva una superposición de estelas. Otra limitación obvia es la que introduce el muestreo de la escena, que limita claramente la variación de los movimientos que pueden analizarse.

1.2.2.2. Modelos basados en la atención selectiva visual

Estos modelos pueden ser agrupados naturalmente en varias clases principales. Las clases de modelos discutidos incluyen

- (1) modelos que incorporan líneas de restricciones, también denominados de obtención de clases
- (2) modelos de correspondencia o "matching", que seleccionan y correlacionan características resaltadas de las imágenes en secuencias de imágenes estudiadas en intervalos cortos de tiempo
- (3) modelos que utilizan una restricción de uniformidad, que, cuando se aplica a las imágenes producidas por el movimiento 3-D, produce un campo velocidad que varía suavemente de punto a punto
- (4) otros modelos que no caen en los anteriores tipos

1.2.2.2.1. Modelos de obtención de clases (clustering)

Para la primera clase de modelos, se utilizan restricciones múltiples para el movimiento en el espacio (extraídas a partir de estimaciones locales de la velocidad) para obtener la velocidad de la forma [Fen79] [Ser89].

El modelo de la transformada de Hough de Fennema y Thompson [Fen79].

Fennema y Thompson discuten cómo múltiples líneas de restricciones, cada una de las líneas definiendo una familia de posibles movimientos de objeto para cada vector de movimiento de componente, intersecciona en el espacio de velocidad para determinar el movimiento de un objeto en traslación. Este modelo ha direccionado primeramente el movimiento rígido planar traslacional. Fennema y Thompson suponen que la imagen está formada por distintas regiones de movimiento uniforme. Un algoritmo modificado de transformada de Hough se utiliza para determinar las velocidades traslacionales de estas diferentes regiones.

Para más información, véase el modelo de Fennema y Thompson en el apartado 1.2.1.1.

El modelo de aprendizaje mediante una regla de Hebb de Sereno [Ser89].

El modelo ha demostrado que el problema de la apertura para el movimiento de formas puede ser

solucionado, utilizando un aprendizaje no supervisado (al estilo del aprendizaje simple de Hebb) en una red de capas hacia adelante.

La capa de entrada está formada por unidades asociadas a un rango de direcciones y velocidades de componente. Un conjunto completo de unidades se establece en un número de (x,y) localizaciones retinotópicas. Existe un conjunto de capas de salida con funciones de asociación no especificadas a priori. La primera capa de curvas de asociación son lineales, solapándose a media altura. La activación puede variar en el rango de -1.0 a 1.0 (véase, que la inhibición está presente) y los efectos de la dirección y de la velocidad interactúan linealmente. La probabilidad de que una unidad de la primera capa conecte con una unidad de la segunda capa es una campana de Gauss con su centro en la posición retinotópica similar en la segunda capa. La activación para cada una unidad de salida es una suma lineal ponderada de sus entradas (con los valores máximo y mínimo recortados) y los pesos cambiados como una fracción del producto de las actividades presinápticas y postsinápticas (de nuevo son recortados los valores de los pesos extremos). Un estímulo consiste en un conjunto de contornos orientados al azar, un contorno por cada localización (x,y) . La forma completa se mueve con una velocidad determinada. El aprendizaje supone la presentación de alrededor de 1000 formas.

Después del entrenamiento, las unidades de la capa de salida son probadas con un número de tipos de formas diferentes. Al ser probadas con estos estímulos, las curvas de asociación de algunas unidades se parecen a las unidades de selección de formas del área MT. Otras unidades muestran el comportamiento de selección del componente (de modo parecido a muchas neuronas MT). Además, hay un número continuo de unidades variando su asociación primitiva a la dirección hacia unidades primeramente asociadas a la velocidad. Antes del aprendizaje, las mismas unidades no mostraban una selectividad direccional importante.

1.2.2.2. Modelos de correspondencia y de correlación (matching)

En la segunda clase de modelos, o sea, en los esquemas de correlación y de correspondencia, se mide el movimiento a partir de secuencias discretas de imágenes, identificando e igualando las características en las imágenes correspondientes a lo largo de tramas sucesivas [Tho81] [Pra83].

La siguiente figura ilustra el problema - las estimaciones locales de la correspondencia basada en los vecinos más próximos no tienen por qué ser coherentes.

El problema de la apertura no aparece en estos esquemas de imágenes. Las dos clases principales de técnicas de correlación incluyen las técnicas basadas en la correlación cruzada de las intensidades y las técnicas que correlacionan rasgos o características simbólicas.

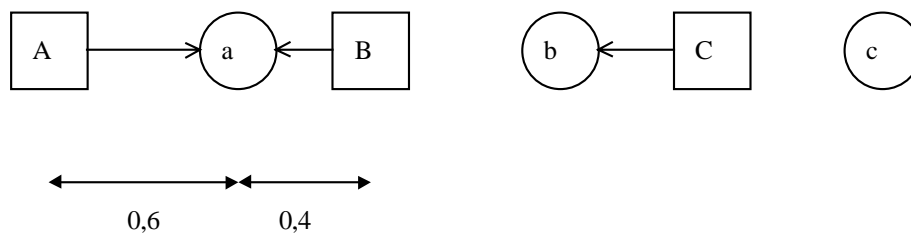


Figura 1.18. Los puntos a, b y c (indicados por círculos) se desplazan 0.6 unidades hacia la izquierda. Las nuevas posiciones de los puntos (renombrados A, B y C) se indican con cuadrados. Las correspondencias hipotéticas iniciales (vecinos más próximos) se indican con flechas.

Las técnicas de correlación simbólica definen estructuras geométricas a ser correlacionadas en la imagen. Al restringir los candidatos a la correlación para las estructuras de imágenes distintas, más que la intensidad de los píxeles, se obtiene la ventaja de que estas estructuras muestran una mayor estabilidad a lo largo de las tramas de imágenes y que necesitan menos correlaciones (computacionalmente son menos caros).

Los esquemas de correlación de rasgos correlacionan puntos característicos (por ejemplo, terminaciones o puntos de alta curvatura), informaciones de los bordes o combinaciones más complejas de estructuras geométricas de bordes o áreas de la imagen.

El modelo de estimación e interpretación a bajo nivel del movimiento visual de Thompson y Barnard [Tho81].

El modelo apuesta por la correlación de puntos característicos que son elegidos, buscando en cada imagen (de una secuencia) aquellas áreas de alta variabilidad en varias direcciones. Estas áreas suelen corresponder a esquinas, laterales, etc.

El "matching" se realiza por aproximaciones sucesivas bajo un esquema de relajación probabilística. El parecido de los candidatos a "matching" se estima inicialmente por medio de una medida de correlación cruzada. Estas estimaciones se ven actualizadas iterativamente considerando las posibles correlaciones en la vecindad espacial de cada punto.

En primer lugar se aplica un operador de variabilidad a pequeñas ventanas (de 3x3 o 5x5). Se calcula la diferencia cuadrática media entre los píxeles adyacentes en las cuatro direcciones posibles. El operador devuelve el valor mínimo de entre los calculados. A continuación, todos los valores que no representen un máximo local se ponen a cero. Se especifica un número deseado de puntos característicos y se crean los puntos que se correspondan con los valores restantes distintos de cero.

El proceso de correlación se inicia con la determinación de todas las posibles correspondencias entre los puntos característicos en dos imágenes sucesivas. Se crea una lista de posibles correlaciones en la segunda imagen de los puntos característicos de la primera imagen, en forma de un conjunto de etiquetas para cada punto. Se añade a la lista otra etiqueta única L^* , que indica que no hay una correlación exacta en la segunda imagen.

De este modo, cada punto característico f de la primera imagen lleva asociada una lista de etiquetas de la forma

$$L^i = \{l_1^i, l_2^i, \dots, l_m^i, l^*\}$$

donde $l_j^i = (ds_x, ds_y)_j^i$

La tarea de "matching" se convierte en la obtención de la etiqueta correcta para todo f .

El modelo MATCH de Prager y Arbib [Pra83].

Prager y Arbib exploran un algoritmo cooperativo, en el que los datos están en forma de vectores retinotópicos o capas que son procesadas en paralelo por vectores de elementos computacionales. Estos elementos están formados por ciertos campos receptivos, es decir, que obtienen sus entradas de vecindades localizadas en las capas de datos. Dado el solapamiento de estas vecindades, estos elementos se comunican entre sí. Al ser la computación iterativa y proseguir hasta la consecución de ciertos criterios de consistencia, el proceso puede verse como un ejemplo de proceso de relajación.

El trabajo describe los procesos para calcular el flujo óptico a partir del movimiento de los puntos de las escenas. Se reserva el término flujo óptico para designar el flujo verdadero asociado al movimiento y se usa el término vector de campo velocidad para hablar de las estimaciones del campo del flujo.

La tarea genérica es la de generar y mantener la capa D de vectores de campo velocidad que representan el movimiento de los puntos.

P y Q son dos tramas consecutivas de puntos de características extraídas, que en la figura están superpuestas. Todos los puntos dibujados caen dentro del campo receptivo de un único elemento de la capa D . La capa C contiene correcciones a la capa D , consistentes en intentar relacionar los puntos de P con los puntos de Q , utilizando los vectores de D como estimaciones. Estas correcciones obtienen uno de los dos componentes de los elementos del vector D . El otro componente (no dibujado) se obtiene haciendo una media ponderada de los elementos de alrededor, utilizados para dar uniformidad al flujo.

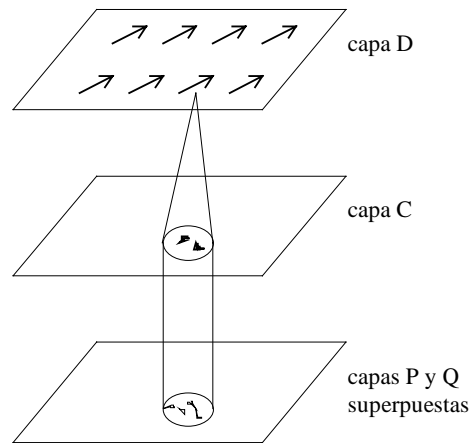


Figura 1.19. Organización de los vectores de datos

La entrada a este sistema está compuesta por vectores registrados de valores de intensidad. Estos datos en bruto sufren una serie de transformaciones hasta extraer aquellas características de interés para los autores. Estas características son lados, esquinas y otros puntos de alto contraste, y se extraen con algún método de tipo convolución. El resultado, después de una posible umbralización, estará compuesto por vectores de puntos característicos.

Los autores hacen las siguientes consideraciones en su trabajo: movimiento uniforme, cuerpos rígidos y distribución "razonable" de puntos característicos en las superficies.

Se define una imagen preprocesada P como un conjunto de parejas $\{(PX(i), PT(i)), i = 1, \dots, n\}$, significando el par $(PX(i), PT(i))$ que la característica $PT(i)$ se encuentra en la localización $PX(i)$. Se usan igualmente las notaciones (X_i, T_i) si se conoce P , o bien P_i cuando no se necesita hacer referencia individual a los atributos de localización o de característica. Se define también una distancia en el espacio W de todos los pares (posición, característica), de tal modo que $|(X_i, T_i) - (X_j, T_j)|$ se incrementa tanto con la distancia $|X_i - X_j|$ entre dos posiciones, como con la diferencia entre los tipos de características T_i y T_j de los dos puntos. Explícitamente, se introduce una métrica de características F y a continuación se define una función f que decrementa rápidamente con la distancia entre dos puntos de W :

$$a_{ij} = f((X_i, T_i), (X_j, T_j)) = e^{-|X_i - X_j|^2 - kF(T_i, T_j)^2}$$

donde el parámetro k determina la importancia relativa de los tipos de características en el proceso de correlación.

Supongamos que P y Q son dos tramas preprocesadas sucesivas en una secuencia de imágenes, siendo Q la más reciente. El objetivo del algoritmo es el de obtener el vector campo velocidad asociado a P y Q . Siempre que se asuma que todo punto de Q tiene un predecesor en P , es posible

asociar con todo punto Q_j en la localización $QX(i)$ un vector $DX(i)$, así como encontrar algún par $(PX(j), PT(j))$ en P tal que

$$\begin{aligned} PX(j) + DX(i) &= QX(i) \\ PT(i) &\approx QT(i) \end{aligned}$$

Estas condiciones, aunque necesarias para una correcta correlación, no son suficientes. No existe una garantía real que $PX(j)$ hizo realmente el movimiento $DX(i)$ para alcanzar el punto $QX(i)$. Por ello, se añade una nueva restricción, que viene a decir que los puntos de la superficie de un objeto en movimiento se mueven con una velocidad muy próxima a la de los puntos vecinos. Dicho de otro modo, $DX(i)$ es aproximadamente igual a la media de los valores DX de los puntos vecinos.

El algoritmo MATCH depende de una función llamada función atracción. Sea X o X_j un punto típico de W . Para un subconjunto $S = \{ S_j \}$ de W , se define la función discontinua de atracción $A[X;S]$ para X en W esencialmente como una función de máxima proximidad vecinal. Dicho de otro modo, $A[X;S]$ es la función que para X_j en S maximiza la función $f(X, X_j)$.

Se define igualmente la función continua de atracción $\hat{A}(X;S)$ como

$$\frac{\sum_{X_j \in S} X_j f(X, X_j)}{\sum_{X_j \in S} f(X, X_j)}$$

\hat{A} es pues una media ponderada de los X_j que favorece a aquellos X_j más próximos a X . De ese modo \hat{A} se aproxima a A cuando el vecino más cercano se ha definido correctamente.

Con ayuda de la función de atracción, y a través de un proceso de relajación, se efectúan correcciones al valor de DX en cada punto. Supongamos que $DX(i)$ para algún Q_i se ve perturbado del estado de equilibrio, mientras que todos los demás puntos no se ven afectados. Sea P_j el punto correcto correlacionado con Q_i . A menos que la perturbación sea demasiado grande, la cantidad $QX(i) - DX(i)$ seguirá estando más cerca de $PX(j)$ que de $PX(k)$, para cualquier $k \neq j$.

En general, es más fácil trabajar con rasgos más simples, pero hay más probabilidad de falsas correlaciones. Las características más complejas son más difíciles de procesar, pero es más fácil encontrar la correspondencia correcta. Sin embargo, ya que la forma de los rasgos complejos cambia en los movimientos complejos, estos esquemas funcionan mejor en movimientos rígidos planos. La mayoría de las evidencias psicofísicas y fisiológicas sugieren que el movimiento no se mide a través de procesos simples de correlación.

El modelo de algoritmo paralelo de cálculo del flujo óptico en tiempo real de Bülthoff, Little y Poggio [Bül89].

El modelo de Bülthoff, Little y Poggio propone un algoritmo simple paralelo que calcula el campo de velocidad, partiendo de una secuencia de imágenes. En primer lugar, se realiza un procedimiento de correlación en las características de la imagen (por ejemplo, los bordes o los valores de intensidad de la imagen) para diferentes desplazamientos de una imagen de un paso temporal al siguiente paso. Los desplazamientos son de diferentes tamaños y direcciones, correspondiendo a diferentes velocidades. Varias capas de procesadores retienen los resultados del procedimiento de correspondencia. Cada una de las capas contiene los resultados de los diferentes tamaños de desplazamientos de imagen con los procesadores de cada capa ordenados en un mapa espacial. Una buena correlación viene determinada por un alto nivel de activación.

Después, cada procesador suma la entrada excitadora de los procesadores vecinos. Un valor alto de excitación se produce por parte de la velocidad "correcta", si el campo de movimiento es localmente uniforme.

Finalmente, las activaciones de los procesadores en las diferentes capas (representando velocidades diferentes) son comparadas en cada una de las posiciones (x,y) y se selecciona la velocidad con la máxima excitación.

Algunas características destacables de este modelo son:

- (1) utiliza un algoritmo rápido paralelo
- (2) reproduce algunas ilusiones percibidas por los humanos
- (3) puede ser aplicado con éxito a campos de velocidad producidos por movimientos 3-D, siempre que se asuma que el campo de velocidad es localmente constante

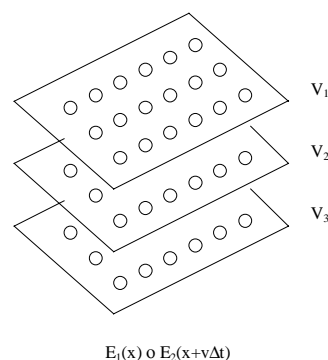


Figura 1.20. Matching de características o valores de intensidad entre dos tramas sucesivas

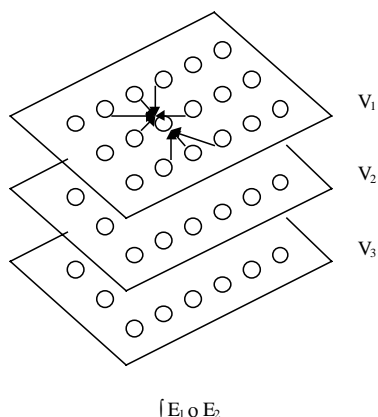


Figura 1.21. Paso de suma local (excitación)

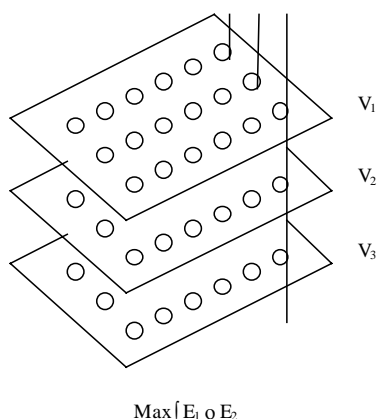


Figura 1.22. Obtención de la velocidad que corresponde a la excitación máxima en cada posición (x,y)

1.2.2.2.3. Modelos de restricciones de uniformidad

El tercer tipo de modelos utiliza una restricción de uniformidad para calcular el campo velocidad producido por un movimiento general 3-D - la proyección del movimiento 2-D de objetos expuestos a rotación o traslación en profundidad [Hor81] [Hil84] [Koc86] [Yui88]. El argumento para la restricción de uniformidad es que, ya que los objetos normalmente tienen superficies uniformes (superficies que cambian gradualmente en su profundidad), el campo velocidad producido por un objeto en movimiento también variará de modo suave.

Puede calcularse el campo velocidad más uniforme obteniendo el campo velocidad de menor variación. La uniformidad del campo velocidad puede medirse tomando las primeras derivadas o derivadas de orden superior de la velocidad a lo largo de las áreas de la imagen o a lo

largo de los contornos. Horn y Schunck [Hor81] midieron la variación en las componentes x e y de la velocidad (u, v) en el espacio bidimensional (x, y) .

Un problema aparece en este algoritmo de uniformidad cuando una imagen contiene más de un objeto en movimiento, lo que causa discontinuidades abruptas en el campo de flujo. El algoritmo de uniformidad en tipos heterogéneos de movimientos conduce a errores sustanciales en la estimación de la velocidad.

El modelo de medida del movimiento visual de Hildreth [Hil84].

Hildreth propuso un algoritmo de uniformidad que remedia parcialmente el problema expuesto anteriormente, al restringir la uniformidad a los contornos conectados, que normalmente definen los límites de los objetos singulares. Ella utilizó la derivada primera de la velocidad a lo largo del contorno como una medida de la variación para definir el primer término en la siguiente función de coste, que será minimizada:

$$\theta(\mathbf{V}) = \int \left| \frac{\delta \mathbf{V}}{\delta s} \right|^2 ds + \beta + \int (\mathbf{V} \cdot \mathbf{u}^\perp - v^\perp)^2 ds$$

\mathbf{V} representa el conjunto de todos los posibles campos vectoriales a lo largo del contorno. El primer término representa una medida del cambio total en la velocidad a lo largo de todo el contorno. Se trata del cuadrado de la longitud de la derivada primera de la velocidad a lo largo del contorno, donde s denota la longitud del arco. El segundo término representa la restricción obtenida de la imagen. Es la diferencia al cuadrado entre el valor calculado $\mathbf{V} \cdot \mathbf{u}^\perp$ (siendo \mathbf{u}^\perp el vector unidad apuntando hacia la dirección perpendicular al contorno) y los valores medidos de los componentes perpendiculares de la velocidad (v^\perp). β es el factor de peso que expresa la confianza en la exactitud de los datos medidos.

El algoritmo trabaja bien para objetos simples bajo un movimiento 3-D general y produce campos de velocidad que están en acuerdo cualitativo con varias ilusiones perceptuales. Además, el algoritmo produce siempre una solución única aunque los datos no especifiquen una única solución. Por ejemplo, si un contorno lleva a cabo movimientos múltiples, incluyendo traslación, rotación y deformación, puede que el movimiento de un punto dado no pueda ser especificado de forma única. Mientras que el algoritmo reduce el problema a unificar la velocidad en las discontinuidades del movimiento, las superficies de las imágenes reales frecuentemente contienen contornos discontinuos y segmentos de línea debido a cambios de textura y cambios en la reflectancia de la superficie.

El modelo de redes “neuronales” análogas a la visión primaria de Koch, Marroquin y Yuille [Koc86].

Este modelo introduce procesos de línea en la función de coste para codificar explícitamente las discontinuidades del movimiento. Un proceso de línea se pone en las localizaciones donde el gradiente del flujo óptico (véase, la rapidez en el cambio de la velocidad) entre dos puntos vecinos es alto. Se refuerzan configuraciones específicas de procesos de línea - por ejemplo, si ocurren a lo largo de un contorno continuo.

El modelo descrito en este apartado parte de la formulación del problema de la reconstrucción de una superficie en términos de la minimización de una función cuadrática de la energía según los estudios efectuados por Grimson [Gri81]. La función de energía o de coste $E(f)$ a minimizar, derivada del problema inverso

$$Bf = d + n$$

donde se conocen el dato d y el operador lineal B , n representa el proceso del ruido y f , que ha de ser calculada, viene dada por

$$E(f) = \|Bf - d\|^2 + \alpha \|Sf\|^2$$

El primer término da la medida de la distancia de la solución al dato real, mientras que el segundo término corresponde al valor de regulación requerido para un planteamiento correcto del problema. α es el parámetro de regulación y S es un operador lineal). En el caso de interpolación en superficies, B es una matriz diagonal con sus elementos iguales a 1 en aquellas posiciones donde la profundidad es conocida y 0 en las demás posiciones. El estabilizador S corresponde al operador asociado a la membrana o a la lámina, dependiendo del tipo de uniformidad deseado.

Sin embargo, este proceso tiene sus limitaciones, falla en los contornos y por tanto no se comporta correctamente en la detección de las discontinuidades.

La aportación de Koch, Marroquin y Yuille comienza en este punto. Se introduce un "proceso de línea" no observable que indica la presencia o ausencia de discontinuidades entre elementos adyacentes. La función a minimizar en este caso se torna en

$$E(f, h) = \sum (f_{i+1} - f_i)^2 (I - h_i) + c_D \sum (f_i - d_i)^2 + c_L \sum h_i$$

Aquí, h_i se corresponde con el proceso de línea. Si el gradiente de f se hace excesivamente grande, véase pues que $(f_{i+1} - f_i)^2 > c_L$, será más barato partir la superficie e insertar una línea, pagando el precio c_L , que interpolar uniformemente. El proceso de línea h_i introduce un mínimo local en la función de energía, presentándose ahora un problema no cuadrático.

El término $(f_i - d_i)^2$, que describe la diferencia entre el dato medido d_i y el valor aproximado de la superficie f_i , es ponderado por c_D , que, a su vez, depende de la relación señal - ruido.

El modelo de la teoría computacional para la percepción del movimiento visual coherente de Yuille y Grzywacz [Yui88].

Yuille y Grzywacz usan derivadas de orden superior en un modelo de restricción de uniformidad basado en áreas que reproduce algunos fenómenos perceptuales relacionados con la coherencia del movimiento - en manifestaciones que contenían una cantidad apreciable de movimiento incoherente.

La teoría divide el cálculo del movimiento en dos etapas, denominadas de medida y de uniformidad, respectivamente.

La etapa de medida estima o restringe el campo velocidad a partir de la información de la imagen. Esta primera etapa, no obstante, se descompone, a su vez, en la medida del movimiento de rango corto y en la medida del movimiento de rango largo.

En la etapa de uniformidad se construye un campo velocidad en la totalidad del campo visual, incluso allí donde no se haya detectado estimación alguna de movimiento. Este campo velocidad se restringe en base a las restricciones encontradas en la etapa de medida y simultáneamente se uniformiza tanto como sea posible.

Para el caso del movimiento de rango corto, se define la medida de la velocidad $M(V_i)$ en el punto r_i , siendo V_i la velocidad verdadera de la imagen. La etapa de uniformidad construye un campo velocidad $v(r)$ que minimiza la siguiente función para las dos componentes de v

$$E(v(r), V_i) = \sum (M(v(r_i)) - M(V_i))^2 + \lambda \int \sum c_m / \nabla_m v / 2$$

donde $\lambda \geq 0$ y $c_m \geq 0$ son constantes, y ∇_m es igual al operador laplaciano elevado a $(m/2)$, si m es par, o ∇_m es igual al operador laplaciano elevado a $(m-1/2)$, si m es impar. La suma se toma para todos los puntos en los cuales tenemos información disponible. La minimización del primer término fuerza a que el campo velocidad calculado v respete el paso de medida. El segundo término introduce la uniformidad del campo.

En el caso del movimiento de rango largo, la función a minimizar será

$$E(v(r), V_{ai}) = \sum V_{ai} \left(v(r_a) - \frac{r - r_a}{\delta t} \right)^2 + \lambda \int \sum c_m / \nabla_m v / 2$$

siendo r_a un número de puntos en el tiempo t , r_i los puntos en el tiempo $t + \delta t$. Se define una

matriz V_{ai} , tal que $V_{ai} = 1$ si el punto a se correlaciona con el punto i ; en caso contrario, $V_{ai} = 0$. La minimización del primer término fuerza la compatibilidad con los datos, mientras que el segundo término fuerza la uniformidad.

Los algoritmos basados en restricciones de uniformidad son problemáticos debido a su falta de viabilidad fisiológica. Por ser estos algoritmos iterativos y cooperativos, parece improbable que pudieran procesar lo suficientemente rápido como para ser implementados en un sistema neuronal real. Además, estos algoritmos calculan derivadas espaciales de la velocidad que requieren de la comparación explícita punto a punto de las velocidades adyacentes. Los datos fisiológicos y psicofísicos, sin embargo, sugieren que la información de la velocidad de los puntos adyacentes en el espacio no pueden ser comparados directamente, por causa de la relativamente mala resolución espacial del sistema de movimiento. Al contrario, parece que las señales de velocidad son integradas en el espacio antes de ser ejecutadas las comparaciones.

1.2.3. Cuadro resumen de los modelos computacionales de análisis del movimiento en secuencias de imágenes

Método	Características	Autor(es)	Referencia	
Detección local de movimiento	<ul style="list-style-type: none"> - Cálculo de la velocidad tomando la derivada temporal de la intensidad en un punto dividida por la derivada espacial de la intensidad en ese punto. - Modelos biológicamente defendibles. - Los cambios en la iluminación ambiental de las superficies pueden ser incorrectamente interpretados como movimientos de las superficies. - Los gradientes de intensidad pronunciados en los bordes de los objetos son estables bajo condiciones de ruido en general. 	Fennema y Thompson	[Fen79]	
		Horn y Schunck	[Hor81]	
		Marr y Ullman	[Mar81]	
		Lawton	[Law89]	
	- Correlación	<ul style="list-style-type: none"> - Cálculo de la dirección a partir de la correlación de las actividades de los campos receptivos de entrada vecinos. - Modelos alejados de la circuitería neuronal real. - No contemplan ningún mecanismo inhibitorio en la detección del movimiento. 	Reichardt y Hassenstein	[Has56]
			Barlow y Levick	[Bar65]
			Adelson y Bergen	[Ade85]
			Watson y Ahumada	[Wat85]
		Heeger	[Hee87]	

Método	Características	Autor(es)	Referencia	
Atención selectiva visual	- Clustering	- Utilización de restricciones múltiples, extraídas a partir de estimaciones locales de movimiento, para la obtención de la velocidad de la forma.	Fennema y Thompson	[Fen79]
		- Permiten incorporar la inhibición.	Sereno	[Ser89]
	- Matching	- Medición del movimiento a partir de secuencias discretas de imágenes, identificando e igualando las características en las imágenes.	Thompson y Barnard	[Tho81]
		- No existe evidencia biológica de que el movimiento se mida a través de procesos simples de correlación.	Prager y Arbib	[Pra83]
			Bülthoff, Little y Poggio	[Bül89]
	- Restricciones de uniformidad	- Obtención del campo velocidad a partir de la restricción de uniformidad de las superficies.	Hildreth	[Hil84]
		- Son modelos faltos de toda viabilidad fisiológica.	Koch, Marroquin y Yuille	[Koc86]
		- Son modelos problemáticos cuando aparece más de un objeto en escena.	Yuille y Grzywacz	[Yui88]

1.3. Análisis del movimiento de los objetos deformables

1.3.1. Introducción

El estudio del movimiento de los objetos deformables es uno de los aspectos del análisis de movimiento con mayor proyección de futuro. El creciente interés viene motivado por innumerables aplicaciones en áreas tan dispares como puede ser la medicina o la vigilancia del entorno. El estudio del movimiento de objetos deformables cubre un amplio rango de condiciones dependientes tanto del contenido de la imagen como de la propia naturaleza del movimiento. Podemos hablar de tres grandes clases en el movimiento de objetos deformables. Las clases se refieren a (1) los cuerpos articulados, (2) la deformación elástica, y, (3) los fluidos.

El procesamiento de la deformación es muy costoso debido a que el movimiento puede ser altamente complejo y la forma del objeto es incierta. Ya ha habido diferentes aproximaciones al problema. Se diferencian básicamente en el modo en que modelan la forma y el movimiento, así como en el método de optimización adoptado. Las formulaciones pueden ser continuas o discretas, deterministas o estadísticas, paramétricas o no. Evidentemente, lo importante en cualquier caso sigue siendo la representación de la forma y la de su movimiento deformable. Es posible, igualmente, considerar la representación de la forma y del movimiento en su conjunto o por separado.

En general, podemos afirmar que se ha prestado gran atención a las técnicas basadas en los contornos activos (serpientes) o superficies deformables [Kas88] [Coh91] [Bla93] [Ter88] [Ter91] [Pen90] [Pen91a]. Estas técnicas son conceptualmente atractivas y permiten integrar distintos tipos de requerimientos, tales como la adecuación a los datos y la conformidad a propiedades físicas globales (principalmente propiedades de regularidad). Estos métodos, no obstante, adolecen de numerosas limitaciones. El mayor inconveniente reside en el hecho que la inicialización debe elegirse cercana a la solución deseada. Por ello, la aproximación inicial se suele hacer a ojo o explotando todo el conocimiento relacionado con la aplicación concreta considerada. Estos métodos también se ven afectados por los demás contornos de la imagen y la presencia de diferentes regiones y texturas en la imagen. La utilización de técnicas basadas en las serpientes en el contexto de la estimación de movimiento deformable es, sobre todo, interesante en el seguimiento de objetivos, siempre que se

disponga de un paso de predicción preciso [Bas94]. El manejo simultáneo de múltiples serpientes sigue siendo un problema abierto.

Un método alternativo consiste en la utilización de patrones paramétricos 2D, envolviendo una parametrización compacta global para representar las formas de interés [Yui92]. Las soluciones de este tipo son excesivamente específicas del problema concreto tratado, al utilizarse información a priori de la forma. La información de movimiento se busca a través de las variaciones temporales de los valores de los parámetros del modelo. También se han probado modelos paramétricos de tipo B-spline bajo deformaciones forzadas [Bas94]. Estos modelos aparecen como más generales y robustos que los anteriormente mencionados.

En el estudio del movimiento no rígido es importante tener en cuenta tanto las deformaciones globales como las locales. Los modelos estadísticos tales como los modelos MRF (campos aleatorios de Markov) se ajustan muy bien a este propósito [Gre91] [Ami91] [Ker94a]. Si se adopta un esquema estocástico de relajación para minimizar la función de energía, la inicialización puede ser aleatoria, como se demuestra en [Ker94a].

El movimiento articulado es de un interés especial en el análisis del movimiento humano. El estudio cuantitativo del movimiento humano (movimiento facial, gestos, etc.) lleva a multitud de aplicaciones, incluyendo la rehabilitación clínica, la biomecánica deportiva, el diseño de nuevas interfaces hombre-máquina en sistemas de realidad virtual, la vigilancia de ciertos lugares, etc. Algunas técnicas, que extraen características simples de la imagen o marcas inscritas en el cuerpo, ya han sido exploradas. Algunos desarrollos explotan directamente la función de intensidad y consideran un modelo geométrico y cinemático articulado 3D para dirigir el análisis de imágenes [Hor91] [Roh94] [Guo94].

Sin lugar a dudas, la mayor fuente de datos representante de los movimientos deformables complejos reside en las imágenes biomédicas. El trabajo de investigación es realmente importante en este campo, sobre todo en modelos elásticos 3D [Hor91] [Mis91]. También se consideran modelos 2D, por ejemplo, en el proceso de imágenes de rayos X o de ultrasonido para el estudio del movimiento del corazón [Mai89] [Her92] [Coo94a].

El proceso por ordenador del movimiento de fluidos en secuencias de imágenes sigue siendo un tema reciente, aunque ya se puede hablar de unos primeros intentos pioneros [Rao92] [For94]. Puede

quedar en una herramienta complementaria de análisis cuantitativo y cualitativo, pero puede enfocarse igualmente hacia la explotación de imágenes de satélite en meteorología u oceanografía [Mau95].

En general, el análisis de modelos deformables 2D o 3D (articulados, elásticos, fluidos) es importante y tiene frente a sí un gran futuro investigador.

El seguimiento automático del movimiento de un objeto no rígido como puede ser una persona en una secuencia de imágenes es un problema que necesita generalmente de algún tipo de conocimiento para poder ser. En este capítulo se discuten algunas técnicas actuales en análisis de movimiento de objetos deformables.

1.3.2. Modelos computacionales

1.3.2.1. Aproximaciones basadas en modelos probabilísticos

Puede derivarse algún tipo de información estadística a priori a partir de información de entrenamiento utilizando el análisis de la componente principal. Esto sería el caso en el modelo de distribución de puntos. Otra alternativa es la de utilizar restricciones debidas a las leyes físicas que limitan las formas de los objetos a deformaciones elásticas. Un ejemplo es el de la aproximación del método del elemento finito. Estas aproximaciones se clasifican bajo el término "basadas en modelos", ya que incorporan información a priori de una forma aproximada. Estas dos aproximaciones clave tienen muchas similitudes y pueden ser incluso combinadas [Coo94b]. De algún modo, ambas son modelos lineales, ya que las características de los modelos se relacionan a través de una transformación lineal para modelar los parámetros de la forma. En ambos casos se obtienen modelos de forma lineales usando métodos de eigenanálisis, y ambos métodos producen modelos altamente compactos a partir de un pequeño conjunto de parámetros.

1.3.2.1.1. Modelos basados en la distribución lineal de puntos

El análisis estadístico de datos señalados 2D ha llegado a ser una herramienta reconocida en visión por ordenador, véase, por ejemplo, los métodos morfológicos [Boo91]. Un avance reciente en esta área está en el modelo de distribución de puntos presentado por Cootes et al. [Coo92a] [Coo92b]

[Coo93]. En general, un modelo de distribución de puntos es un modelo estadístico de un conjunto de puntos 2D o 3D. El modelo estadístico descrito por Cootes et al. es un modelo lineal (siempre que ignoremos las componentes rotacionales del modelo), al que denominaremos modelo de distribución lineal de puntos. El modelo de distribución lineal de puntos ha sido usado con éxito para la interpretación de imágenes médicas [Hil94] [Coo93], el reconocimiento automático de caras [Lan94] y el análisis de secuencias de imágenes (por ejemplo, usando un modelo deformable estocástico [Ker94b]).

El modelo de distribución lineal de puntos de Cootes et al. [Coo92a].

En un modelo de distribución de puntos, una forma se representa por medio de un conjunto de n puntos característicos etiquetados. Cada uno de los puntos se corresponde con una característica particular, muchas veces de tipo biológico, del objeto.

El modelo de distribución lineal de puntos se basa en un análisis estadístico de las coordenadas de estos puntos en un conjunto de entrenamiento. Cada una de las formas de entrenamiento se representa por un vector forma de X , cuyas coordenadas son las coordenadas de los puntos señalados. Modelando en dos dimensiones, tenemos

$$X = (x_0, y_0, x_1, y_1, \dots, x_{n-1}, y_{n-1})^T$$

siendo (x_i, y_i) la posición del punto i -ésimo señalado en el conjunto de formas de entrenamiento.

Las formas de entrenamiento se alinean mediante una técnica generalizada de análisis de "procrustes" [Gow75]. Su utiliza un método de mínimos cuadrados ponderados para alinear cada una de las formas con la forma media. Los pesos se eligen de modo que se da más importancia a los puntos señalados más estables. Este proceso resulta en un vector forma medio \bar{x} y un conjunto de vectores forma de entrenamiento x_j alineados. El siguiente paso del análisis consiste en restar el vector forma medio de cada uno de los vectores forma de entrenamiento. Así,

$$dx_i = x_i - \bar{x}$$

Entonces, la matriz de covarianza S de dimensión $2n \times 2n$ se obtiene calculando

$$S = E(dx dx^T)$$

donde $E(\dots)$ es el operador de cálculo de la media sobre el conjunto de entrenamiento.

Las variaciones de los puntos señalados se representan por medio de los $2n$ eigenvectores de S que resuelven

$$S e_i = \lambda_i e_i$$

donde $\lambda_0 \geq \lambda_1 \geq \dots \lambda_{2n-1} \geq 0$. Los eigenvectores forman una base ortonormal en el espacio de formas. De este modo los vectores forma dx pueden reescribirse como

$$dx = \sum_{i=0}^{2n-1} b_i e_i$$

siendo $b_i = dx \cdot e_i$.

Puede demostrarse que los parámetros b_i son linealmente independientes y que la varianza total de cada uno de los eigenvectores es igual al del eigenvalor asociado.

$$E(b_i b_j) = \begin{cases} 0 & : i \neq j \\ \lambda_i & : i = j \end{cases}$$

Así, los eigenvectores correspondientes a los eigenvalores mayores representan los modos más significativos de variación. Un subconjunto que contenga los m eigenvectores más significativos se retiene como base del espacio de formas. Una forma en el espacio de modelos x puede describirse como una suma de la forma media y una suma ponderada de eigenvectores.

$$x = \bar{x} + P b$$

siendo P una matriz de dimensión $2n \times m$, cuyas columnas son los m eigenvectores más significativos y $b = (b_0, \dots, b_{m-1})^T$ es un vector de m coeficientes. Dado un vector de forma alineado x' , la aproximación por mínimos cuadrados a la forma en el espacio de modelado viene dada por una proyección lineal

$$b = P^T (x' - \bar{x})$$

Este análisis de eigenvectores no es más que una aplicación del análisis de la componente principal o transformada de Karhunen-Loeve.

El modelo de forma activa de Cootes et al. [Coo92b].

Cootes et al. describen un método, el modelo de forma activa, que optimiza localmente los parámetros de forma del modelo de distribución lineal de puntos, con el fin de localizar las características en una imagen [Coo92b]. El modelo de forma activa puede verse como una aplicación particular 2D de la técnica de refinamiento de Lowe [Low91]. El modelo de distribución lineal de puntos se ajusta perfectamente a este tipo de aproximación iterativa dada la simplicidad en derivar una matriz jacobiana apropiada de características para actualizar los parámetros de la forma.

El modelo de forma activa asume una estimación inicial en cuanto a orientación, escala y posición del modelo, así como los parámetros lineales de la forma. Dados estos parámetros, la forma modelo puede ser proyectada en la imagen utilizando

$$X = Q(s, \theta)[\bar{x} + Pb] + X_c$$

donde $Q(s, \theta)$ es una rotación en θ y un escalado en s y X_c es una traslación en (X_c, Y_c) . El vector de forma X representa la posición de los n puntos señalados en coordenadas de imagen.

En cada iteración del proceso de refinamiento, los movimientos sugeridos para cada uno de los puntos señalados dX_i se calculan a partir de las características de la imagen. La aproximación usual es la de buscar el borde más fuerte a lo largo de la normal al contorno del modelo en cada punto señalado. El vector dX_i se iguala al valor del desplazamiento de la posición del punto señalado estimado respecto de la característica del borde, y se escala proporcionalmente a la fuerza del borde.

Dado este conjunto de desplazamientos representados por el vector desplazamiento de forma dX , se calculan las estimaciones para los cambios en los parámetros de pose $dX_c, dY_c, d\theta$ y el cambio relativo en la escala ds .

Cootes et al. proyectan los desplazamientos de puntos de la trama de imagen en desplazamientos en la trama de coordenadas del modelo dx utilizando la ecuación

$$dx = Q\left(\left(s(1 + ds)\right)^{-1}, -(\theta + d\theta)\right)\left[Q(s, \theta)x + dX - dX_c\right] - x$$

que puede reescribirse en la forma

$$Q(s', \theta')dx = dX - \underbrace{(Q(s', \theta') - Q(s, \theta))}_{\text{term1}}[x] - \underbrace{dX_c}_{\text{term2}}$$

donde $s' = s(1 + ds)$ y $\theta' = \theta + d\theta$. Esta ecuación puede interpretarse como una corrección de los desplazamientos dX , tomando en cuenta la pose actualizada. El término 1 elimina los cambios en escala y rotación y el término 2 elimina el cambio de origen.

Los desplazamientos de los puntos del modelo dx se proyectan en ajustes a los parámetros del vector forma b , usando

$$db = P^T dx$$

que no es más que la solución de mínimos cuadrados al problema

$$Jdb = dX'$$

donde dX' es el vector de desplazamientos corregidos de puntos en la imagen y J es la matriz jacobiana con respecto a los m parámetros de forma. Así,

$$J_{ij} = \frac{\delta(dX'_i)}{\delta b_j}$$

Los parámetros de forma y pose se actualizan utilizando un esquema de actualización ponderado como se muestra a continuación:

$$X_c \rightarrow X_c + w_t dX_c$$

$$Y_c \rightarrow Y_c + w_t dY_c$$

$$\theta \rightarrow \theta + w_\theta d\theta$$

$$s \rightarrow s(1 + w_s ds)$$

$$b \rightarrow b + W_b db$$

donde w_s, w_θ, w_t son pesos escalares y W_b es una matriz diagonal de pesos para cada uno de los parámetros de la forma. En el modelo de forma activa convencional, W_b se iguala, o bien a la matriz identidad, o, preferiblemente, a cada peso se le asigna un valor proporcional a la desviación estándar del parámetro de forma correspondiente sobre el conjunto de entrenamiento. Esto permite que los parámetros más significativos de la forma varíen con una mayor libertad.

Cada uno de los pasos iterativos refina los parámetros de forma y pose, reduciendo el error entre las características del borde de la imagen y el modelo proyectado. Después de cada

iteración los parámetros de la forma son restringidos para asegurar que la forma está lo suficientemente cercana a la forma media en términos de la métrica de distancia de Mahalanobis. Explícitamente,

$$s^2 = \sum \frac{b_i^2}{\lambda_i}$$

$$b'_i = \begin{cases} \left(\frac{s_{max}}{s} \right) b_i & s > s_{max} \\ b_i & \text{en caso contrario} \end{cases}$$

donde s_{max} es la distancia máxima permitida a la media. La restricción asegura que el vector b cae dentro de una hiperelipsoide centrada en el origen. Los puntos dentro de esta hiperelipsoide tienen una densidad de probabilidad razonablemente alta a priori, si se asume que el conjunto de formas de entrenamiento se muestrearon según una distribución de Gauss de la forma media [Has94].

Con el fin de mejorar la velocidad y robustez del modelo de forma activa, se puede practicar un mecanismo de búsqueda multiescala [Coo94d].

El modelo iterativo de Lowe [Low91].

Lowe describe un esquema iterativo para la adecuación de modelos parametrizados 3D a imágenes [Low91]. Este esquema se basa en el método de Newton, estabilizándose usando restricciones a priori. Dado un vector de parámetros no lineales p , se calcula una secuencia de estimadores por medio de

$$p^{(i+1)} = p^{(i)} - q$$

A cada iteración se calcula q , minimizando

$$\|Jq - e\|^2 + \alpha^2 \|W(q - d)\|^2$$

donde e es el error entre las posiciones estimada y observada de las características del modelo en la imagen, d es un vector de constantes de los parámetros dados a priori, W es una matriz diagonal en la que cada peso es inversamente proporcional a la desviación estándar σ_i para el parámetro i , J es la matriz jacobiana, α es un peso de corrección ajustado dinámicamente y que afecta a la estabilidad y al grado de convergencia.

El primer término de la ecuación dirige la solución hacia los datos de la imagen y el segundo término estabiliza la solución atrayéndola hacia la posición de inicio d . Para asegurar que la solución final se acerca a los datos de la imagen, el punto de partida d en el modelo anterior se respeta al resultado de la iteración previa.

Al aplicar este esquema a los parámetros en un modelo de forma activa, se minimiza el siguiente error en cada iteración

$$\|(b' - b) - db\|^2 + \alpha^2 \|W(b' - b)\|^2$$

donde b' es el vector de parámetros de forma actualizado y W es una matriz diagonal de pesos donde

$$W_{ii} = \frac{1}{\sqrt{\lambda_i}}$$

Ello nos lleva a la ecuación de actualización

$$b'_i = b_i + \left(\frac{\lambda_i}{\alpha^2 + \lambda_i} \right) db_i$$

similar, pero no idéntica, a la ecuación de actualización del modelo de forma activa.

1.3.2.1.2. Modelos basados en el método del elemento finito

El método del elemento finito es una técnica de ingeniería usada para la simulación computacional eficiente de sistemas físicos [Bat82]. Pentland y Sclaroff describen la aplicación de estas técnicas a problemas en visión por ordenador [Pen91a] [Pen91b] [Scl93] [Scl94]. La aproximación tomada es la de construir un modelo elástico físico de un objeto deformable y utilizar el análisis del elemento finito para producir un conjunto ortogonal compacto de parámetros de la forma, válido para las tareas de seguimiento y de reconocimiento. Nastar y Ayache han aplicado con éxito estas técnicas al análisis de secuencias temporales de conjuntos de datos médicos 3D [Nas94a] [Nas94b].

En contraste con la aproximación basada en el entrenamiento del modelo de distribución de puntos, el método del elemento finito usa un modelo físico generado a partir de un ejemplo único de la forma del objeto, junto con algunas suposiciones acerca de las propiedades materiales físicas del objeto.

Usando ciertas técnicas de análisis modal es posible reducir la dimensionalidad de la representación de la forma en el método del elemento finito sin una pérdida significativa de precisión.

El concepto básico del método del elemento finito es el de representar un cuerpo en términos de un conjunto de regiones o elementos descritos por un conjunto de nodos etiquetados. La cantidad de desplazamiento se aproxima por medio de un conjunto de funciones parciales continuas sobre el cuerpo, definido sobre un número finito de subdominios, denominados elementos. La función de interpolación usada es continua y, generalmente, polinómica de bajo orden.

De este modo, la forma de un objeto se representa mediante un conjunto de desplazamientos nodales U a partir de una forma inicial con representación nodal X . Una forma ha de ser vista como el resultado de la deformación de un material inicial elástico. Al contrario que el modelo de distribución de puntos, el método del elemento finito aporta una caracterización analítica de la superficie del objeto entre nodos.

El modelo de análisis modal de Sclaroff y Pentland [Scl94].

Utilizando propiedades conocidas o asumidas del objeto, la masa global y la matriz de rigidez se obtienen formulando las integrales apropiadas sobre cada elemento y sumando en todo el dominio.

La ecuación gobernante resultante describe la evolución del sistema a lo largo del tiempo bajo la influencia de cargas externas actuando en los nodos. Para un sistema de n nodos en d dimensiones, la ecuación viene dada por

$$M\ddot{U} + C\dot{U} + KU = R(t)$$

donde U es el vector de dimensión $dn \times 1$ de desplazamientos nodales, M , C y K son matrices simétricas de dimensión $dn \times dn$, descriptivas de la masa, la humedad y la rigidez entre cada punto dentro del objeto y R es un vector de dimensión $dn \times 1$ de fuerzas externas actuando sobre los nodos.

La aproximación basada en el análisis modal cambia el sistema anterior, al llevarlo a una base de modos de vibración libres M -ortogonales, obtenidos por resolución del problema de eigenvalores

$$K\phi_i = \omega_i^2 M\phi_i$$

Si se asume la fórmula de Rayleigh ($C = a_0M + a_1K$), el sistema de ecuaciones se transforma en dn ecuaciones diferenciales independientes de segundo orden. Esto se consigue definiendo una matriz de transformación Φ cuyos vectores son los eigenvectores ϕ_i .

$$\Phi = [\phi_1, \phi_2, \dots, \phi_{dn}]$$

Entonces, poniendo $U = \Phi \tilde{U}$, la ecuación gobernante se torna en

$$\ddot{\tilde{U}} + \tilde{C}\dot{\tilde{U}} + \Omega^2\tilde{U} = \Phi^T R(t)$$

donde Ω^2 es una matriz diagonal de eigenvalores

$$\Omega^2 = \begin{pmatrix} \omega_1^2 & & & \\ & \omega_2^2 & & \\ & & \ddots & \\ & & & \omega_{dn}^2 \end{pmatrix}$$

y $\tilde{C} = a_0I + a_1\Omega^2$ también es diagonal. Cada modo de vibración lleva asociada una frecuencia ω_i . Los modos de vibración con frecuencias más altas pueden ignorarse ya que, teóricamente, tienen poca amplitud y suelen ser difíciles de medir con cierto grado de precisión. Los modos con frecuencias más bajas tienden a corresponderse con deformaciones intuitivas tales como el doblado y el corte.

Las amplitudes y velocidades modales pueden estimarse dinámicamente integrando a lo largo del tiempo la ecuación gobernante transformada o utilizando un mecanismo de filtrado Kalman [Gel74]. La aproximación mediante análisis modal nos asegura varios beneficios. La ecuación gobernante de segundo orden es simplificada y existe una reducción en la dimensionalidad conseguida al ignorar los modos de altas frecuencias. Ello redundará en un seguimiento más rápido y más eficiente de las formas.

1.3.2.2. Aproximaciones basadas en los contornos 2D

Otras aproximaciones como las serpientes, las serpientes de Kalman y los esplines activos efectúan un menor número de suposiciones sobre las formas. Estos métodos son 2D, aproximaciones basadas en los contornos, donde la forma del objeto es restringida a ser continua y suave, y con una deformación igualmente suave.

Estas aproximaciones más generales no se describen generalmente como basadas en modelos. En la interpretación de imágenes reales, o sea imágenes capturadas por una cámara en un entorno exterior, se requiere normalmente de un conocimiento previo más detallado. Podemos encontrarnos con los problemas de auto-oclusión. Igualmente, las imágenes reales muchas veces son de poca calidad debido a una iluminación pobre y a una resolución baja. Otros problemas a tener en cuenta son las sombras, las reflexiones, debidas, por ejemplo, a superficies mojadas en la carretera, y las condiciones climáticas pobres (lluvia, nubes, etc.). Estos problemas pueden resolverse únicamente incorporando información a priori al modelo.

1.3.2.2.1. Serpientes y serpientes Kalman

La serpiente (o modelo de contorno activo) [Kas87] ofrece un poderoso mecanismo para la interpretación de las imágenes a bajo nivel, especialmente para el seguimiento de objetos deformables en el plano [Ley93]. Una serpiente es un esplín de energía mínima que es atraído hacia ciertas características de la imagen, como pueden ser los bordes. Una función de energía interna regula el problema, modelando el esplín como una membrana elástica mediante restricciones de suavizado. Una técnica de minimización de energía local, como un método Euler, se emplea de modo que los contornos discretizados se deslizan por la pared más próxima en la superficie de energía. El sistema dinámico puede considerarse en términos de fuerzas de imagen que atraen el contorno hacia las características de borde y de fuerzas internas elásticas que mantienen el suavizado.

El modelo de contorno activo de Kass et al. [Kas87].

Una serpiente sencilla minimiza la energía

$$E_{snake} = \int_0^1 E_{int}[v(s)] + E_{image}[v(s)] ds$$

donde las funciones de coordenadas de contorno se denominan $v(s) = (x(s), y(s))$.

La energía de deformación interna viene dada por

$$E_{int} = \alpha(s) |v_s(s)|^2 + \beta(s) |v_{ss}(s)|^2$$

Los dos parámetros físicos $\alpha(s)$ y $\beta(s)$ controlan la tensión y rigidez del contorno en un punto determinado. Las fuerzas externas de imagen se obtienen a partir del potencial de energía E_{image} , que puede tomar el valor

$$E_{image}(x, y) = -c \|\nabla(G_\sigma \circ I(x, y))\|$$

donde $G_\sigma \circ I$ denota la convolución de un filtro gaussiano sobre la imagen y σ controla la escala espacial. El difuminado gaussiano efectivamente incrementa el tamaño del muro de energía alrededor de un mínimo local. Según la serpiente va alcanzando el equilibrio, la escala espacial del filtro gaussiano se va reduciendo hasta obtener detalles más finos.

Con el fin de alcanzar la minimización, la serpiente es discretizada a puntos de muestreo regulares $v_i = (x_i, y_i)$, y se aplica un procedimiento de optimización local iterativo.

Terzopoulos y Szeliski han mostrado que el sistema de la serpiente elástica es equivalente a un filtro Kalman en estado de equilibrio con matriz de covarianza unitaria constante [Ter92]. Una ventaja de esta aproximación reside en el hecho que los parámetros del modelo pueden obtenerse de un modelo de sensor estadístico variable en el tiempo.

Terzopoulos et al. han extendido el modelo de serpiente 2D a modelos 3D elásticamente deformables [Ter87a] [Ter87b].

1.3.2.2.2. Esplines activos

Blake et al. describen otro entorno estadístico para el seguimiento eficaz de contornos usando un mecanismo de filtrado de Kalman [Bla93]. Estos denominados esplines activos evolucionan a partir del principio de la serpiente. Con el fin de lograr una mayor eficiencia computacional, se representa un contorno como una curva paramétrica, por ejemplo, un B-esplín cúbico. La continuidad y elasticidad explícitas del B-esplín permite usar un modelo estocástico sencillo para el seguimiento del contorno, sin la necesidad de incorporar una función de regularización de energía interna explícita. Puede incorporarse algún conocimiento anterior al seguidor por medio de un acoplador elástico con una muestra B-esplín (contornos acoplados [Cur92]). Este mecanismo de muestreo persistente mejora la estabilidad, incorporando memoria de forma y restringiendo la distribución anterior de la forma del

contorno. Una muestra de forma invaginare afín extendida permite que el contorno alcance las transformaciones afines más rápidamente [Bla93].

El modelo de B-splines de Blake et al. [Bla93].

Una curva (cerrada) B-esplín $v(s) = (X(s), Y(s))$ se define paramétricamente para $0 \leq s \leq N$ en términos de N puntos de control variables en el tiempo $Q_k = (X_k(t), Y_k(t))$ por medio de

$$v(s) = \sum_{k=1}^N B_k(s) Q_k = H(s) Q_k$$

donde B_k es una función de interpolación cúbica para el i-ésimo punto de control y

$$H(s) = (B_1(s), B_2(s), \dots, B_N(s))$$

El espacio de estados viene representado por los vectores de estado $X = (X_1, \dots, X_N)$ e $Y = (Y_1, \dots, Y_N)$.

Blake et al. introducen una métrica de distancia asociada al espacio de estados dada por

$$d(X, X') = |X - X'|$$

donde la norma $|\dots|$ se define como

$$|X^2| = \int_0^N X(s)^2 ds = X^T H X$$

y la matriz H viene dada por

$$H_{i,j} = \int_0^N B_i(s) B_j(s) ds$$

Un contorno observado $(X_f(s), Y_f(s))$ se define buscando a lo largo de las normales (o líneas paralelas) a partir de la estimación actual (\hat{X}, \hat{Y}) dentro de una ventana de búsqueda. Se obtiene analíticamente una ventana de búsqueda elíptica a partir de la covarianza de la estimación actual. Buscando mayor velocidad, el contraste se examina en tres puntos: en la curva estimada y en los dos extremos de la ventana de búsqueda. El punto que ofrece el contraste mayor (por ejemplo, en cuanto a gradiente de intensidad) se retiene como el valor

observado de $(X_f(s), Y_f(s))$. Si no existen medidas significativas en ninguno de los tres puntos, la ventana de búsqueda se hace la mitad de grande y el proceso se repite. Cuando no se encuentran características significativas en la ventana, no se anota ninguna observación.

En realidad, las medidas $(X_f(s), Y_f(s))$ se realizan en puntos discretos de la curva. Sin embargo, puede mostrarse que un modelo de sensores teórico continuo es equivalente a las observaciones en el espacio de estados $(X(s), Y(s))$, la aproximación de mínimos cuadrados a los puntos de la curva continua observada, con una matriz de covarianza asociada R , para cada X_f e Y_f dada por

$$R = rH^{-1}$$

siendo r la constante de medida de varianza.

La varianza de la medida de un punto se pone proporcional al tamaño de la ventana de búsqueda, reflejando que los errores de medida son mayores cuanto mayor es la escala de búsqueda. Como toda medida de un punto se realiza buscando a lo largo de una línea recta, las medidas X e Y se acoplan.

Las posiciones de los puntos de control se modelan usando un modelo de velocidades constantes con aceleraciones aleatorias expresadas por la ecuación

$$\frac{d}{dt} \begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} = \begin{pmatrix} \ddot{X} \\ \ddot{Y} \end{pmatrix} + \begin{pmatrix} 0 \\ w \end{pmatrix}$$

donde $w(t)$ es una media no nula, un proceso gaussiano de ruido temporalmente no correlacionado. Se aplica una ecuación independiente similar a Y . Si suponemos una distribución de ruido de Gauss isotrópica homogénea, la matriz de covarianza para w es proporcional a H^{-1} .

Entre tramas de imagen sucesivas no se realizan observaciones y la matriz de covarianza P , asociada a la estimación aumentada de estado (\hat{X}, \hat{X}) , se actualiza apropiadamente. Las observaciones de las características en los puntos $(X_f(s), Y_f(s))$ en el instante $t = t_k$ se aplican secuencialmente, usando la ecuación de actualización del filtro de Kalman

$$\begin{pmatrix} \hat{X} \\ \hat{X} \end{pmatrix} \rightarrow \begin{pmatrix} \hat{X} \\ \hat{X} \end{pmatrix} + K(s)(X_f(s, t_k) - H(s)\hat{X})$$

donde la ganancia de Kalman viene dada por

$$K(s) = P \begin{pmatrix} H(s)^T \\ 0 \end{pmatrix} \left[(H(s)|0)P \begin{pmatrix} H(s)^T \\ 0 \end{pmatrix} + \sigma^2 \right]^{-1}$$

y σ es la desviación estándar de la medida individual del punto.

Se puede aplicar un mecanismo de muestreo persistente usando una entrada virtual de O , aplicado al filtro pero acoplado fuera del subespacio V de transformaciones afines de la muestra. La muestra estabiliza el sistema, previniendo el contorno de volverse enmarañado e incrementando su robustez.

Una ventaja de usar un entorno de filtro estadístico Kalman es que la covarianza de la estimación actual modela la varianza posiciones de cada uno de los puntos del contorno. Si asumimos isotropía, se construye una ventana circular alrededor de cada punto del contorno de radio $2\rho(s,t)$, donde ρ^2 es la varianza posicional en S dada por

$$\rho(s)^2 = (H(s)|0)P(H(s)|0)^T$$

En ausencia de medidas de imagen, cuando se ha perdido toda noción del contorno, la escala de búsqueda se incrementa según la incertidumbre de las estimaciones de estado se van incrementando con el tiempo. De modo similar, la ganancia de Kalman se incrementará de modo que, si se aplican nuevas medidas, el contorno reaccionará rápidamente y se adaptará a las características de la imagen.

Una vez que el contorno estimado está razonablemente cercano al contorno del objeto, la ventana de búsqueda y la ganancia de Kalman se decrementan, permitiendo explotar la coherencia del movimiento y recuperar el contorno de un modo más eficaz.

1.3.2.3. Modelos de alto nivel

Los modelos de alto nivel permiten incorporar una gran cantidad de información acerca de la forma del objeto, e incluso del movimiento esperado en el tiempo. Algunos ejemplos de modelos 3D representativos para el caso del seguimiento de personas caminando son el modelo basado en cilindros WALKER [Hog83] y un modelo similar utilizado por Rohr [Roh93]. Estos modelos complejos consisten en una representación 3D explícita del objeto generado por un experto humano. El modelo trabaja con pocos parámetros. En el caso del modelo de Rohr existe un solo parámetro de pose. Ello redundante en un

seguimiento rápido y robusto, pero falla cuando la entrada no se adecua al entrenamiento previo. Estas aproximaciones utilizan modelos basados en la conceptualización teórica. Consecuentemente, una aproximación de este tipo sufre cuando la realidad difiere del modelo preconcebido, aunque es posible permitir algún grado de tolerancia de error. La aproximación alternativa, aquella orientada al dato, construye un modelo a partir de un conjunto representativo de datos reales de entrenamiento.

Otras aproximaciones asumen que las uniones del cuerpo humano han sido marcadas (por ejemplo, véase Chen y Lee [Che92]). El modelo de eigenimágenes no representacional de Murphy et al. [Mur93] está estrechamente relacionado con la aproximación de eigencaras de Turk y Pentland [Tur91] y las extensiones a niveles de gris del modelo de distribución de puntos [Coo93] [Coo94c]. Sin embargo, uno de los mayores problemas de las representaciones basadas en imágenes es el coste computacional al operar con ventanas relativamente grandes de píxeles de imagen (por ejemplo, cálculo del flujo óptico). Además, la dimensionalidad del modelo resultante sigue siendo alta (típicamente se usan 30 parámetros en un modelo) y el método generalmente se basa en una cámara fija.

1.3.2.3.1. Descomposición en eigenimágenes

Murphy et al. [Mur93] describen una aproximación novedosa al análisis del movimiento humano basado en la descomposición en eigenimágenes. Su aproximación está basada en tareas, en oposición al paradigma convencional representacional adoptado en visión por computadora. La base del método es la utilización de la transformada de Karhunen-Loeve sobre un conjunto estadísticamente representativo de imágenes de entrenamiento.

Se utiliza un procedimiento de transformada de Karhunen-Loeve modificada por eficiencia computacional. Las imágenes de tamaño $n \times m$ se consideran nm vectores de elementos. n y m son generalmente grandes (> 64), llevando a vectores de imágenes con más de 8000 elementos. De una manera similar a como se hacía en el modelo lineal de distribución de puntos, el vector medio imagen se elimina y se calcula una eigenbase linealmente independiente denominada eigenimágenes. Dadas N imágenes de entrenamiento, siendo N típicamente igual a 100, las eigenimágenes pueden calcularse a partir de los eigenvectores de una matriz pseudocovariante de dimensión $N \times N$.

Una imagen que sea similar a las imágenes contenidas en el conjunto de entrenamiento puede representarse mediante una combinación lineal de un subconjunto de las eigenimágenes. Generalmente serán suficientes 30 coeficientes para representar las imágenes para el reconocimiento de la pose.

En los experimentos de Murphy et al. no se usan las imágenes tal cual. Es la magnitud del flujo óptico en cada pixel el que se ofrece como entrada a la transformada de Karhunen-Loeve. Las secuencias de imágenes se representan por medio de los 30 coeficientes más significativos de Karhunen-Loeve. La información resultante nutre a una red neuronal clasificadora.

Aunque esta aproximación se ajusta bien a tareas de reconocimiento de alto nivel, sigue siendo computacionalmente costosa, necesitando de muchas operaciones sobre los píxeles. El método tampoco funciona excesivamente bien en un entorno ruidoso. Una aproximación similar ha sido tomada por Turk y Pentland para el caso de la representación de caras [Tur91]. Cootes et al. han combinado una aproximación de eigenimágenes con el modelo lineal de distribución de puntos [Coo93] [Coo94c].

1.3.2.3.2. Modelos articulados

Los modelos articulados basados en primitivas se han aplicado con éxito en una gran variedad de aplicaciones (por ejemplo, en el refinamiento de Lowe [Low91]). El modelo WALKER [Hog83], se usa un modelo representacional de una persona caminando, basado en el modelo del cuerpo de Marr y Nishihara [Mar78]. El modelo WALKER de Hogg representa la forma de un objeto en términos de cilindros representantes de las partes rígidas del cuerpo y conectadas apropiadamente en las juntas.

Una aproximación similar, basada en los trabajos de Hogg, es la de Rohr [Roh93]. En su trabajo, Rohr reduce el espacio de búsqueda de parámetros, siguiendo solamente un parámetro de postura, basándose la posición dentro de un ciclo de caminata genérico. En concreto, el modelo genérico está basado en un conjunto de caminatas de 60 hombres. El movimiento está limitado a ser paralelo al plano de la imagen.

1.3.3. Cuadro resumen de los modelos computacionales de análisis del movimiento de los objetos deformables

Método	Características	Autor(es)	Referencia
Modelos probabilísticos			
- Distribución lineal de puntos	<ul style="list-style-type: none"> - Se basa en el análisis de la componente principal. - Se basa en un estudio estadístico de las coordenadas de un conjunto de puntos característicos durante el entrenamiento. - Funciona con éxito para la interpretación de imágenes médicas, el reconocimiento automático de caras y el análisis de secuencias de imágenes. 	<p>Cootes et al.</p> <p>Lowe</p>	<p>[Coo92a]</p> <p>[Coo92b]</p> <p>[Low91]</p>
- Método del elemento finito	<ul style="list-style-type: none"> - Se basa en el uso de restricciones debidas a las leyes físicas que limitan las formas de los objetos a deformaciones elásticas. - Se construye un modelo elástico físico de un objeto deformable a partir de un ejemplo único de la forma del objeto. - Es muy valioso en tareas de seguimiento y reconocimiento. - Se ha aplicado con éxito a datos médicos 3D. 	Sclaroff y Pentland	[ScI94]

Método	Características	Autor(es)	Referencia
<p>Contornos 2D</p> <p>- Serpientes</p>	<ul style="list-style-type: none"> - Una serpiente es un esplín de energía mínima que es atraído hacia ciertas características de la imagen. - Interesante en el seguimiento de objetivos deformables. - Es un poderoso mecanismo para la interpretación de las imágenes a bajo nivel. - El mayor inconveniente está en que la inicialización debe elegirse cercana a la solución deseada. La aproximación inicial suele hacerse a ojo. - Se ve afectado por los demás contornos de la imagen y la presencia de diferentes regiones y texturas en la imagen. 	<p>Kass et al.</p>	<p>[Kas87]</p>
<p>- Esplines activos</p>	<ul style="list-style-type: none"> - Efectúa un seguimiento eficaz de contornos usando un mecanismo de filtrado de Kalman. - La información de movimiento se busca a través de las variaciones temporales de los valores de los parámetros del modelo. - Son muy generales y robustos. 	<p>Blake et al.</p>	<p>[Bla93]</p>

Método	Características	Autor(es)	Referencia
Modelos de alto nivel	<ul style="list-style-type: none"> - Aproximación basada en tareas. - Utilización de la transformada de Karhunen-Loeve sobre un conjunto estadísticamente representativo de imágenes de entrenamiento. - Se ajusta bien a tareas de reconocimiento de alto nivel. - Es computacionalmente costosa. - No funciona excesivamente bien en un entorno ruidoso. 	Murphy et al.	[Mur93]
		Turk y Pentland	[Tur91]
		Cootes et al.	[Coo93]
			[Coo94c]
	<ul style="list-style-type: none"> - Modelos articulados - Basados en primitivas. - Basados en una representación 3D explícita generada por un experto humano. - Usan pocos parámetros, con lo que se obtiene un seguimiento rápido y robusto. - Fallan cuando la entrada no se adecúa al entrenamiento previo. 	Hogg	[Hog83]
		Horowitz y Pentland	[Hor91]
		Rohr	[Roh93]

CAPITULO 2. Interacción lateral en computación acumulativa

2.1. Detección de movimiento en el sistema visual

2.1.1. Vías del sistema visual

Todos los estudios en inteligencia artificial y análisis de reconocimiento de patrones por ordenador llevados a cabo hasta el momento han permitido apreciar que el cerebro reconoce la forma y el movimiento utilizando estrategias a las que no puede aproximarse ningún ordenador actual. La distancia se agrava aún más cuando se sabe que la información neuronal relacionada con la forma, el movimiento y el color no es transportada por una vía jerarquizada única o un canal de procesamiento único, sino al menos por tres vías de procesamiento cerebrales paralelas que interactúan.

La absorción de la luz y su transducción en señales eléctricas es llevada a cabo por los fotorreceptores. La información visual se transfiere entonces desde los receptores a las células ganglionares a través de las células bipolares, segregada en dos vías paralelas, una de centro ON y otra de centro OFF. Pasamos a explicar estos conceptos.

Las células ganglionares individuales nunca están inactivas, incluso en la oscuridad, pero su actividad espontánea está modulada por las entradas de interneuronas retinianas. Las entradas a cada célula ganglionar siempre tienen su origen en los mismos fotorreceptores de un área circunscrita de la retina, el campo receptivo de esa célula. Los campos receptivos de las células ganglionares tienen dos rasgos importantes.

Primero, los campos receptivos de las células ganglionares son circulares y varían en tamaño a lo largo de la retina. En la región foveal de la retina, en donde la agudeza visual es máxima, los campos receptivos son pequeños. En la periferia de la retina, donde la agudeza es baja, los campos son más extensos. Segundo, en la mayoría de las células ganglionares, el campo receptivo está dividido en dos partes: una zona circular central, denominada centro del campo

receptivo, y el resto del área del campo denominada periferia. Las células ganglionares presentan una respuesta óptima cuando el centro del campo recibe una iluminación distinta que la periferia. Una célula ganglionar de centro ON se excita cuando la luz estimula el centro de su campo receptivo y se inhibe cuando la luz estimula su periferia. Una célula ganglionar de centro OFF tiene las respuestas opuestas.

Estas transformaciones de la señal visual ayudan a los centros superiores a detectar los contrastes débiles y los cambios rápidos de la intensidad de la luz. Algunas células se ocupan de los rasgos generales del estímulo y su movimiento, otros transmiten la información acerca de los detalles espaciales finos y el color de la imagen visual.

El sistema visual organiza las tareas perceptuales siguiendo ciertas leyes innatas que gobiernan el patrón, la forma, el color, la distancia y el movimiento de los objetos en el campo visual. Este proceso de organización perceptual es continuo y dinámico. Como ejemplo clásico, tenemos la dicotomía figura-fondo que ilustra uno de los principios de la percepción visual: solo se selecciona como foco de atención una parte de la imagen mientras que el resto queda sumergido en el fondo. Las partes de una imagen que son centro de atención son sus límites y sus bordes.

En primer lugar, necesitamos considerar brevemente algunas características anatómicas de las vías visuales. Los axones de las células ganglionares de la retina forman el nervio óptico que proyecta ordenadamente al núcleo geniculado lateral (NGL) del tálamo. El NGL a su vez proyecta al córtex visual primario ipsilateral (V1) en el córtex estriado. Como resultado de esta proyección ordenada, el córtex estriado contiene un mapa completo de la retina.

Más allá del córtex estriado se encuentran las áreas extraestriadas, un conjunto de áreas visuales de alto orden que también contienen representaciones de la retina. Estas regiones difieren en la selectividad de sus células para distintas características de los estímulos. Por ejemplo, el área V5 (TM) está implicada fundamentalmente en el movimiento del campo visual, mientras que V4 está mucho más implicada en el color y en la orientación de los bordes. De este modo, el sistema visual tiene varias representaciones distintas en su superficie receptora, la retina, y cada una de ellas está implicada en el procesamiento de distintos aspectos de la información visual.

La segregación de la información visual comienza en la retina. La retina contiene células ganglionares de dos tamaños: células grandes (magnocelulares o de tipo M) y células pequeñas (parvocelulares o de tipo P). Cada uno de estos tipos transporta información distinta que llega a

distintas capas en el NGL y de ahí a distintas regiones corticales. En el encéfalo humano el NGL contiene seis capas de cuerpos celulares separados por las correspondientes capas de axones y dendritas. Los axones de las células M proyectan a las capas magnocelulares del NGL, mientras que los axones de las células P proyectan a las capas parvocelulares. Estos dos grupos de capas dan lugar a tres vías principales, dos desde las capas parvocelulares y una desde la capa magnocelular del NGL. Las tres vías irradian desde el NGL hasta V1, y de ahí a V2 y a otras áreas corticales extraestriadas (ver figura 2.2).

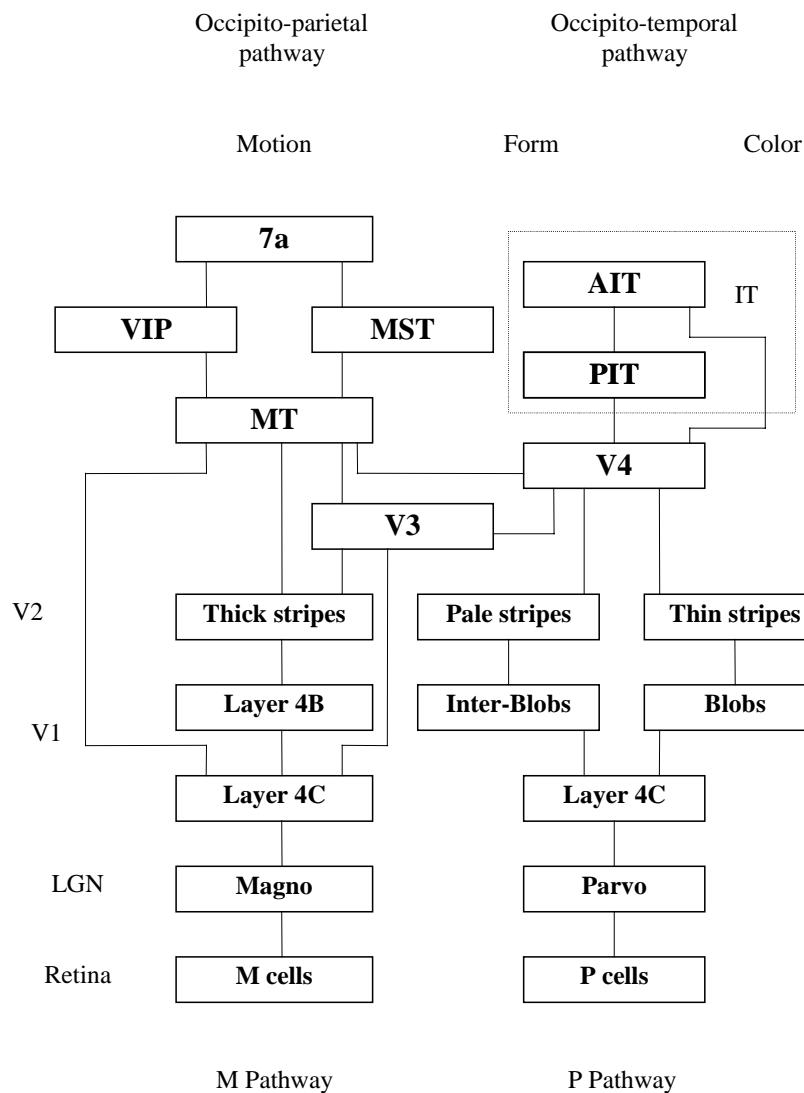


Figura 2.1. Canales de forma, color y movimiento en el sistema visual [Liv88]

Tanto V1 como V2 presentan subdivisiones importantes. En V1 tenemos unas regiones de blobs que forman un patrón repetitivo de puntos a modo de estructuras de aproximadamente

0,2 mm de diámetro, separadas por regiones denominadas regiones interblobs. En V2 tenemos dos tipos de bandas (anchas y finas) separadas por regiones interbandas.

La primera vía que surge de neuronas intercaladas entre las capas parvocelulares del NGL, la vía parvocelular-blob, está implicada en la percepción del color. Esta vía establece sinapsis en los blobs de las capas superficiales de V1. Desde V1 la vía llega a V2 y de ahí proyecta al área V4, un área en la que se encuentran muchas células con respuesta al color. Esta vía termina en el córtex inferotemporal, un área implicada en la percepción del color y la forma.

La segunda vía que surge en la vía parvocelular, la vía parvocelular-interblob, está implicada en la percepción de las formas. Esta vía termina en las capas profundas de V1 y recibe también una pequeña contribución de la vía magnocelular. Al igual que la vía implicada en el color, esta vía proyecta al córtex inferotemporal.

La tercera vía, la vía magnocelular-bandas anchas, está especializada en la detección del movimiento y en las relaciones espaciales, y contribuye a la percepción de la profundidad. Se extiende desde las capas magnocelulares del NGL, a través de las regiones interblobs de V1, hasta las bandas gruesas de V2 y de ahí al córtex TM (V5), el área implicada en la profundidad y el movimiento. El TM proyecta a otras áreas en el córtex parietal implicadas en la función visuoespacial.

2.1.2. Organización en columnas y capas

Las células del córtex visual primario presentan respuestas selectivas a estímulos con un eje de orientación y una posición retiniana específicos. Las neuronas del córtex visual primario están organizadas en columnas para responder a ejes de orientación únicos. En el córtex extraestriado las células selectivas al movimiento o a formas complejas también están organizadas en columnas funcionales. Las columnas de orientación están interrumpidas regularmente por los blobs.

Además del sistema de columnas destinado a los ejes de orientación y al sistema de blobs implicado en el color, el córtex estriado tiene un tercer componente: un sistema de columnas de dominancia ocular destinado a la visión binocular, que es importante para la percepción de la profundidad.

Las columnas de orientación del córtex visual primario se comunican entre sí por medio de conexiones horizontales entre células de una misma capa. Así, el córtex visual primario está

organizado funcionalmente en dos dimensiones perpendiculares: las columnas funcionales están formadas por conexiones locales entre distintas capas corticales mientras que columnas funcionales con propiedades de respuesta iguales están conectadas entre sí.

2.1.3. Movimiento en el campo visual

El sistema visual es capaz de calcular rápidamente el movimiento y la forma tridimensional de los objetos en un entorno a partir de las intensidades de luz variables a lo largo del tiempo que alcanzan el ojo. Una de las soluciones que el sistema visual ha tomado para el cálculo del movimiento es la extracción espacial de las señales de movimiento locales, combinando estas señales espacialmente en capas de proceso superiores para calcular tipos de movimiento más complejos.

Entre las capacidades de procesamiento de movimiento del sistema de visión animal se pueden incluir:

- la detección de las discontinuidades del movimiento que permite la segmentación de la imagen en objetos
- la detección de movimiento brusco periférico para la llamada de atención
- el control del movimiento del ojo, de la locomoción y de la posición
- la provisión de información acerca de la egomoción
- la estimación del movimiento de un objeto
- la estimación de la estructura tridimensional (3-D) de los objetos y de su entorno

El sistema visual tiene dos modos de detectar el movimiento: uno basado en el movimiento de la imagen y otro basado en el movimiento de la cabeza y los ojos. Aquí se va a considerar únicamente el movimiento de la imagen en el campo visual.

Según se van moviendo las superficies 3-D en el espacio, éstas van proyectando luz en el ojo, formando una imagen 2-D del mundo que va cambiando con el tiempo. El sistema visual debe reconstruir un sistema 3-D a partir de estas imágenes 2-D. Esta reconstrucción puede hacerse utilizando la información acerca de la organización del movimiento en la imagen cambiante. Sin embargo, el movimiento de los elementos en la imagen 2-D (véase su velocidad y dirección) no es

una propiedad inherente a la imagen, sino que debe ser inferida de las intensidades variables de la imagen.

Los datos neurofisiológicos y psicofísicos disponibles hasta la fecha sugieren que existen varios niveles en el análisis de movimiento en el sistema visual. La figura 2.2. presenta una descripción simplificada de estos niveles de proceso [Ser93].

En el primer nivel las medidas de movimiento local extraen los componentes del movimiento que se encuentran en la dirección perpendicular a los bordes de los elementos en la imagen. El segundo nivel combina las medidas de movimiento local de porciones de la imagen con el fin de calcular un número menor de estimaciones locales de traslación del móvil. Finalmente, un tercer nivel integra las estimaciones locales de movimiento traslacional para calcular movimientos no locales más complejos (por ejemplo, rotaciones globales). De este modo, en cada nivel, la información de movimiento localizado espacialmente parece estar combinado en un área para calcular menos movimientos locales, pero más movimientos complejos.

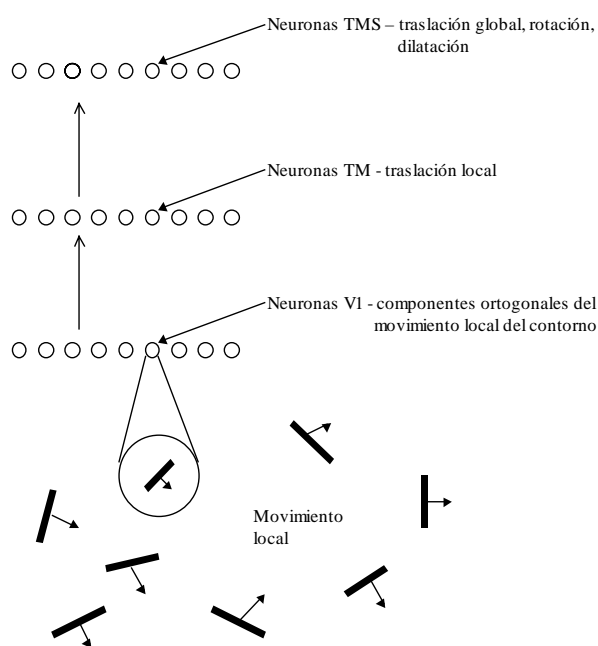


Figura 2.2. Representación esquemática de los pasos en el análisis del movimiento en la corteza visual primaria

Una característica fundamental de la arquitectura cortical de los sistemas sensoriales es la presencia de múltiples mapas topográficos proyectados unos sobre los otros. En el sistema visual, aproximadamente veinticinco de estos mapas están ordenados en una jerarquía de cinco a seis bandas definidas por el modelo de interconexiones hacia adelante y hacia atrás.

Al ir pasando de estaciones más bajas a estaciones más altas, los tamaños de los campos receptivos se van haciendo mayores debido a las proyecciones divergentes de las áreas más bajas a las más altas juntamente con el tamaño físico generalmente más grande de las áreas más bajas frente a las más altas. El incremento en el tamaño del campo receptivo implica que una neurona de nivel más alto integra señales de muchas neuronas de nivel más bajo con campos receptivos adyacentes y espacialmente restringidos, transformando información de entrada espacialmente localizada en representaciones más complejas y no locales. Estas transformaciones son fundamentales para el proceso de importantes características no locales de la visión, como pueden ser la traslación de las formas, la rotación, la dilatación, la obtención de la figura a partir de las sombras, la obtención de la estructura a partir del movimiento. La evidencia neurofisiológica de los estudios sugieren que los tres pasos del análisis de movimiento descritos anteriormente están representadas en al menos tres áreas corticales diferentes que están jerárquicamente unidas unas a otras a través de vías directas e indirectas. Así han sido descubiertas algunas células en las tres grandes áreas (V1, TM y TMS) que responden al movimiento del componente (área V1), al movimiento de la forma (área TM, temporal medial) y a los movimientos globales (área TMS, temporal superior medial).

Concretamente, la información sobre el movimiento nace en las grandes células retinianas de tipo M. Las células no tienen una sensibilidad especial al movimiento per se, pero responden mejor a dianas circulares en el campo visual cuyo contraste varía en el tiempo. Las señales generadas por las células de tipo M se transmiten a través de las capas magnocelulares del NGL a varias capas del córtex visual V1. En V1 las señales son procesadas por neuronas simples y complejas selectivas a la dirección del movimiento. Cada célula responde a la dirección del movimiento perpendicular a su eje de orientación. Estas señales de tipo M se elaboran posteriormente en el área TM o V5 y el área TMS. Después se remiten al área visomotora del lóbulo parietal donde los patrones de neuronas reflejan la velocidad y la dirección del movimiento de los objetos en el campo visual.

Los datos neurofisiológicos disponibles [Ser93] sugieren que la resolución del problema del movimiento ocurre en dos pasos: en primer lugar se extrae el movimiento del componente, y, entonces, éste se integra espacialmente para calcular el movimiento de la forma. El área de neuronas V1 seleccionan la velocidad, la dirección y la orientación de los contornos en movimiento. Estas neuronas sensibles a la dirección se concentran principalmente en la capa 4B de V1, proyectándose hacia el área TM, un área altamente especializada en el proceso del movimiento. El área TMS contiene tanto neuronas selectivas de movimiento de componentes como de formas.

La información disponible acerca de la arquitectura funcional de las áreas V1 y TM nos dice que están organizadas retinotópicamente (las porciones adyacentes del córtex contienen neuronas con campos receptivos que muestrean zonas espaciales adyacentes. En comparación con las neuronas V1, las neuronas TM poseen campos receptivos mucho mayores, lo que indica la existencia de la suma espacial en sus entradas de unidades V1 con campos receptivos adyacentes. Vemos pues que las neuronas selectivas de formas TM integran las velocidades 1D de áreas de espacio visuales.

Por lo tanto, las investigaciones fisiológicas y anatómicas indican que el sistema visual resuelve la detección de movimiento mediante una operación rápida de tipo hacia delante que transforma la información de movimiento 1D de regiones adyacentes espaciales en información de movimiento de formas 2D.

2.1.4. Representación computacional de la visión

Como ya hemos visto, la información importante para la visión viaja desde la retina al núcleo geniculado lateral. Las células de estas estructuras tienen campos receptivos circulares pequeños. Las células de las diferentes vías visuales tienen una respuesta selectiva a diferentes rasgos en el campo visual.

El éxito de un procesamiento en paralelo tan extenso en el sistema visual depende de conexiones tremendamente específicas. Regiones separadas de la retina proyectan al núcleo geniculado lateral talámico, de modo que el campo visual completo de cada ojo está representado en el núcleo. Las células de esta capa se comunican con células de las otras capas para formar una rica circuitería que transforma en el córtex visual primario la información visual.

Además de estar organizadas en seis capas, las células del córtex visual también están organizadas funcionalmente en sistemas de columnas: columnas de orientación específica, columnas de dominancia ocular y regiones blob. Las neuronas dentro de cada sistema columnar que tienen propiedades de respuesta similares están unidas por conexiones horizontales. Así la información fluye tanto vertical como lateralmente en el córtex visual de una manera altamente específica. Las unidades columnares parecen funcionar como módulos computacionales elementales, que reciben entradas variadas, las transforman y envían sus salidas a un número de diferentes regiones encefálicas.

Para describir las neuronas a nivel cortical se requiere de algunas construcciones a un nivel simbólico intermedio, entre el lenguaje natural y los niveles de implementación neurofisiológicos [Mir96]. En efecto, toda computación es conexionista (neuronal) en el dominio propio de sus procesadores. La distinción aparece en el dominio del observador externo que ve la computación simbólica como una computación de “grano grueso”.

Ya que todos los procesos descritos en el apartado anterior terminan en métodos conexionistas, simbólicos o híbridos, en este trabajo se ha optado por una solución conexionista avanzada, en la que se mantiene el grafo paralelo (la conectividad y el grano pequeño, la organización por capas o grupos funcionales, ...) como modelo formal subyacente, pero ampliando la capacidad computacional de los nodos. En efecto, al modelar neuronas, se están construyendo representaciones computacionales de datos experimentales [Mor96]. Esto significa que las propiedades y las limitaciones de estas representaciones del conocimiento y de las herramientas inferenciales se imponen en el modelo. Para ser más precisos, las conclusiones obtenidas de un modelo están siempre implícitas en la naturaleza de los operadores utilizados en su formulación.

En visión el paso de la computación analítica a la computación simbólica tiene lugar entre las células ganglionares y el córtex primario, al pasar del bajo nivel a los niveles altos donde se identifican los estímulos. Al modelar el proceso de bajo nivel, las correlaciones estructura-función pueden especificarse utilizando filtros adaptativos (recurrentes y no recurrentes) con campos receptivos centro-periferia y otros núcleos no lineales dependientes del tiempo. Hasta las células ganglionares no es necesaria la inyección de conocimiento adicional. Por el contrario, en el córtex, donde la información ha sido codificada y recodificada una y otra vez, e integrada con otras modalidades sensoriales, las herramientas analíticas no son adecuadas [Del78] [Mir80] [Mir81] [Mir82] [Mir96].

La mayoría de las descripciones del comportamiento del córtex se realizan mejor a nivel simbólico, usando símbolos neurofisiológicos, estructuras de datos, herramientas de representación del conocimiento y mecanismos de inferencia cooperativos [Del78] [Mir95] [Mir96]. La idea consiste en que todo el conocimiento que tenemos acerca de la función neuronal (procesos bioquímicos en los canales, computación sináptica, bifurcaciones en neurotransmisores específicos, actividad oscilatoria espontánea, ...) puede ser representada por una red de condicionales con ciclos iterativos de selección (campo dendrítico) y ejecución (generación de potencial de acción). Las redes neuronales a nivel inferencial pueden ser

modeladas por redes de condicionales sobre campos de datos que cubran sus campos receptivos (máscaras de muestreo sobre memorias FIFO), que mantienen los esquemas de conectividad y realimentación, así como las organizaciones en capas y columnas de la anatomía del córtex.

Ahora bien, cuando la complejidad del nivel de implementación aumenta, es más adecuado representar el conocimiento estático mediante objetos estructurados [Arb92] [Mir70] [Mir92]. Esta forma de representación del conocimiento asume que nuestro sistema nervioso organiza la experiencia perceptiva y la memoria en términos de un conjunto de situaciones reales tales que cuando se ven activadas en la percepción o evocadas en la memoria, estamos aceptando una gran cantidad de conocimiento implícito que es capaz de llenar todos los campos que aún quedan por especificar.

Crear un modelo es pues representar a nivel simbólico el problema que se desea describir. A este nivel todo proceso neuronal puede representarse mediante un conjunto de atributos en una estructura ilustrada en la siguiente tabla:

S1. Elemento de proceso neuronal.	Elemento de proceso neuronal correspondiente a la posición (x,y) en la capa n de proceso cooperativo.
S2. Campo receptivo.	Area de muestreo en el espacio de entrada y área de diálogo con los vecinos en el espacio de salida.
S3. Tarea de inferencia local.	Algoritmo de computación local. Incluye las formulaciones análoga y lógica como casos específicos. El paso natural de lo analítico a lo algorítmico se realiza sustituyendo las no linealidades de la función de umbralización con un condicional.
S4. Tarea de aprendizaje.	Mecanismo de aprendizaje supervisado o no supervisado que permite realizar modificaciones sobre S2 y S3.
S5. Tarea de control.	Permite la autoprogramación de los procesos neuronales. Requiere de al menos tres modos de funcionamiento: la inicialización, la computación y el modo de aprendizaje.

El resto del capítulo está enfocado a presentar el modelo de interacción lateral en computación acumulativa. Para ello, se presenta en primer lugar su inspiración en los modelos de computación neuronal para, a continuación, y basándose en la estructura de la tabla anterior, entrar de lleno en él.

2.2. Modelos de computación neuronal

2.2.1. Introducción

Se entiende por computación neuronal al cálculo realizado por arquitecturas modulares, organizadas en redes multicapa de gran número de procesadores elementales con alto grado de interconectividad y realizando localmente una función analógica no lineal. En las redes neuronales, la computación se organiza normalmente en una arquitectura multicapa, de modo que la primera capa recibe las entradas del medio y la última capa cierra el círculo ofreciendo el resultado de la computación. Las capas intermedias capturan la información del espacio de salida de la capa anterior y descargan en el espacio de entrada de la capa siguiente.

En los últimos años han ido apareciendo nuevos resultados acerca de la estructura, el desarrollo y el papel funcional de la conectividad lateral en el cortex. Estos resultados están ayudando a entender el cortex como un sistema dinámico en adaptación continua a partir de mecanismos de interacción lateral competitivos y cooperativos [Mii96]. Se han propuesto a la fecha algunas posibles funciones y mecanismos computacionales para estas conexiones laterales. Basándonos en los resultados conocidos y aprovechando todo el conocimiento actual de estas conexiones laterales, se va a establecer en este capítulo el denominado modelo de interacción lateral en computación acumulativa. Este modelo, como se podrá comprobar, va a incorporar las siguientes áreas motivo de debate [Mii96]: (1) el efecto de las interacciones laterales sobre los campos receptivos de las neuronas, (2) su papel en el establecimiento de efectos globales tales como la inhibición lateral, (3) el desarrollo de conexiones laterales, y, (4) las interacciones laterales en distintos niveles de computación.

2.2.2. Arquitectura general de una red artificial

Una red artificial está formada por una serie de unidades simples de proceso que se comunican enviándose señales a través de un gran número de conexiones ponderadas. Podemos distinguir entre un conjunto de aspectos predominantes en un modelo distribuido de estas características [McC86] [Rum86]:

- un conjunto de unidades de proceso (neuronas)

- un conjunto de estados de activación a_i para cada unidad que también determina la salida de la unidad
- conexiones entre las unidades. Generalmente, cada conexión se define por un peso w_{ij} que determina el efecto que tiene la señal de la unidad j sobre la unidad i
- una regla de propagación, que determina la entrada efectiva i_i a partir de sus entradas externas
- una función de activación F_i , que determina el nuevo nivel de activación a partir de la entrada efectiva $i_i(t)$ y del estado actual $a_i(t)$
- una función externa u offset θ_i para cada unidad
- un entorno en el que debe operar el sistema, proveyendo señales de entrada y, si fuese necesario, señales de error

Cada unidad realiza un trabajo relativamente simple: recibe entradas desde unidades vecinas y las utiliza para calcular una salida de salida que se propaga a otras unidades. Además de este proceso, existe una segunda tarea que es el ajuste de los pesos. El sistema es inherentemente paralelo, ya que muchas unidades pueden realizar sus cálculos al mismo tiempo.

Durante la operación, las unidades pueden ser actualizadas bien síncronamente bien asíncronamente. En la actualización síncrona todas las unidades actualizan su activación simultáneamente; en la actualización asíncrona, toda unidad tiene una probabilidad de actualizar su activación en el instante t , y, generalmente, solamente una unidad va a ser capaz de hacer esto en un instante determinado.

En la mayoría de los casos, se asume que cada unidad añade su contribución a la entrada de la unidad a la que está conectada. La entrada total a la unidad i es simplemente la suma ponderada de las salidas separadas de cada una de las unidades conectadas más un offset θ_i :

$$i_i(t) = \sum_j w_{ij}(t) a_j(t) + \theta_i(t)$$

La contribución de un peso w_{ij} positivo se considera una excitación, mientras que la de un peso negativo se considera una inhibición. Estas unidades que incluyen una regla de propagación, se denominan unidades sigma.

Al hablar de patrones de conexiones, la mayor distinción se puede hacer entre:

- Redes de propagación hacia adelante: el flujo de datos desde las unidades de entrada a las unidades de salida es estrictamente hacia adelante. El proceso de datos puede

extenderse sobre múltiples (capas de) unidades, pero no existen conexiones hacia atrás, o sea, las conexiones siempre se extienden desde las salidas de las unidades a las entradas de las unidades de la misma capa o de capas previas. Algunos ejemplos clásicos de redes de propagación hacia adelante son el *Perceptron* [Ros58] y el *Adaline* [Wid60].

- Redes recurrentes: contienen conexiones hacia atrás. Al contrario que en el caso anterior, aquí las propiedades dinámicas de la red son importantes. Algunos ejemplos de redes recurrentes han sido presentados por Anderson [And77], Kohonen [Koh77] y Hopfield [Hop82].

Una red neuronal debe ser configurada de tal modo que un conjunto de valores de entrada produzca el conjunto deseado de salidas. Existen varios métodos potenciar las conexiones existentes. Una forma es poner los pesos explícitamente utilizando el conocimiento previo. Otra manera es la de entrenar la red neuronal nutriéndola con unos patrones de aprendizaje y dejando que cambie sus pesos de acuerdo a alguna regla de aprendizaje.

Podemos catalogar las situaciones de aprendizaje en dos tipos diferentes:

- aprendizaje supervisado o asociativo, en el cual la red es entrenada proveyendo los patrones asociados entre entrada y salida
- aprendizaje no supervisado o autoorganizativo, donde una unidad (de salida) es entrenada para responder a clases de patrones en la entrada

Virtualmente todas las reglas de aprendizaje pueden considerarse como variantes de la regla de aprendizaje de Hebb sugerida en el libro clásico de Hebb “*Organization of Behavior*” [Heb49]. La idea básica es que si dos unidades i y j están activadas simultáneamente, sus interconexiones deben ser potenciadas. Si i recibe entrada de j , la versión más sencilla del aprendizaje hebbiano modifica el peso w_{ij} en

$$\Delta w_{ij} = \gamma a_i a_j$$

siendo γ una constante positiva de proporcionalidad que representa el ratio de aprendizaje.

2.2.3. Modelo de computación neuronal en una capa

El paradigma usual en cómo interpretar los procesos neuronales es el paradigma computacional [Kuh70] [Mir97a]. Dicho paradigma comienza diferenciando entre entorno y sistema. El comportamiento del sistema se describe a continuación en términos de un conjunto

de entradas (estímulos), un conjunto de salidas (respuestas) y un conjunto de procesos computables que, aplicados a estas entradas y a algunas variables internas (estados), modifican dichos estados y producen un conjunto de salidas que replican y predicen las respuestas neuronales.

Dentro de una capa, el modelo de computación neuronal sigue el esquema de la figura 2.3. Dicho modelo proviene de generalización de la propuesta de computación por capas [McC65] [Mir83] con módulos cooperativos (reglas de inferencia local) que muestrean información en los espacios de entrada y salida de acuerdo al tamaño y la forma de sus campos receptivos funcionales y descargan sus resultados en los correspondientes posiciones del espacio de salida.

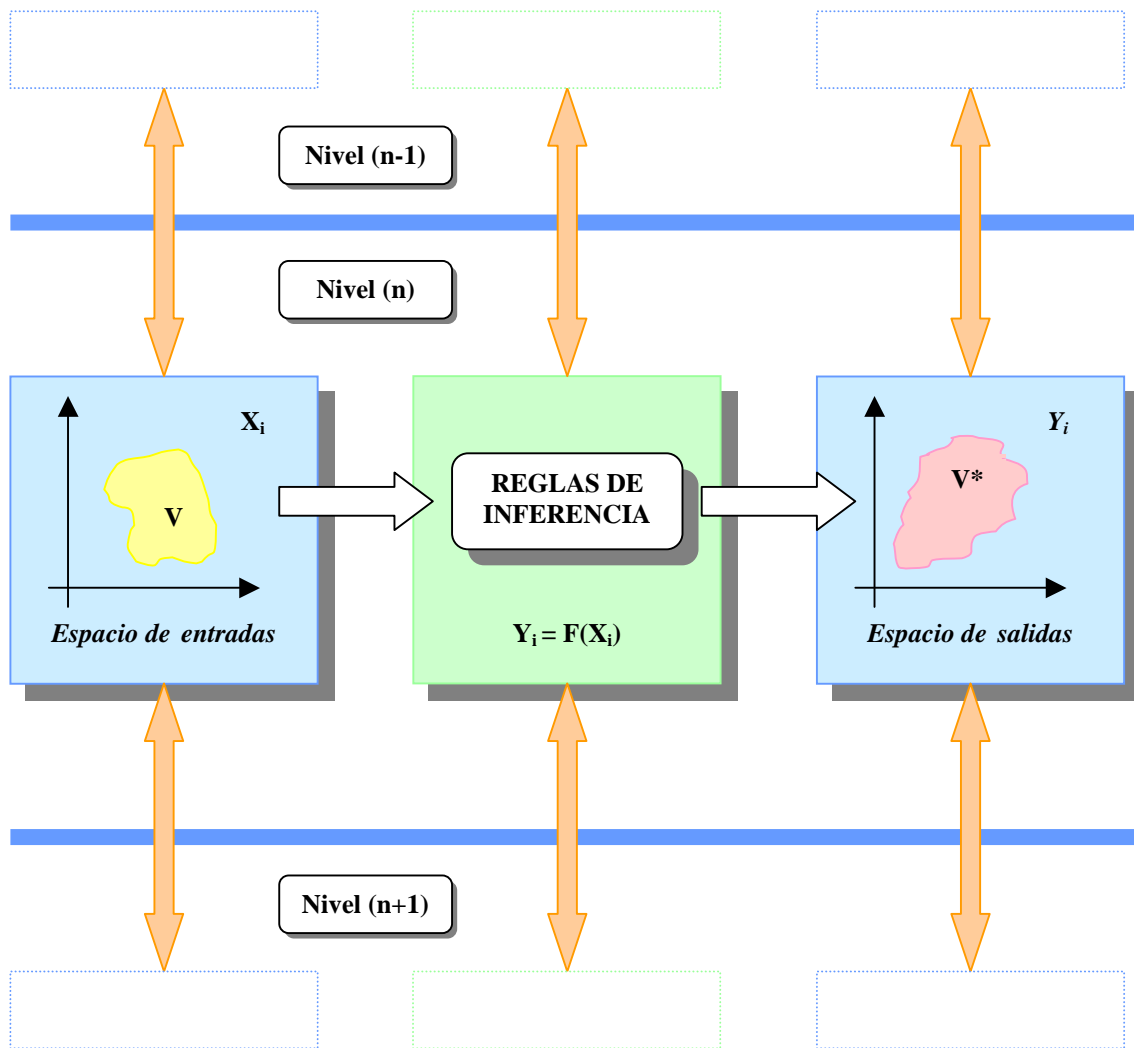


Figura 2.3. Modelo de computación neuronal en un nivel

En el modelado neuronal la utilización de reglas inferenciales para modelizar la función neuronal ha sido presentada por Mira, Delgado y Moreno-Díaz [Del89] [Mir92]. Existe evidencia experimental suficiente para formular un modelo inferencial de la neurona, ya que la neurona es demasiado compleja como para ser considerada un “procesador elemental”.

Siguiendo el paradigma general, en este modelo, todas las neuronas de una misma capa realizan el mismo tipo de computación de naturaleza paramétrica, muestreando los datos de un área de su espacio de entrada (organizado en forma de memoria FIFO) llamado campo receptivo, realizando una operación local sobre estos datos y descargando el resultado en su espacio de salida (memoria FIFO igualmente). No tenemos realimentación, sino más bien un cálculo hacia adelante. Si se permite que el campo receptivo incluya también el muestreo de una región del espacio de salida, tenemos una capa de neuronas recurrentes con realimentación.

La forma del campo receptivo en el espacio de salida define la conectividad local entre las neuronas vecinas en la capa. Así, también, la forma y el solapamiento entre los campos receptivos en el espacio de entrada determina parcialmente la función global de la red. La figura 2.4. muestra el esquema funcional general de una capa de neuronas [Mir95].

En los modelos analógicos la función neuronal se interpreta como una suma ponderada de las entradas seguida de una función de decisión. Es evidente que la suma ponderada es lineal. Para calcular la respuesta de una neurona en el modelo analógico, se descompone ésta en dos partes: excitación y umbral.

$$\text{Excitación: } y_j^*(t + \Delta t) = F_j(x_j) = \sum_{\forall i, k \in V_j} \omega_{ij}(t) \cdot x_i(t - k \cdot \Delta t)$$

$$\text{Umbral: } y_j(t + \Delta t) = G_j(y_j^*)$$

Las funciones umbral más conocidas son la abrupta y la continua no decreciente en sus dos versiones, es decir la sigmoide y la tangente hiperbólica.

$$\text{Abrupta: } y_j(t + \Delta t) = \begin{cases} 1, & \text{si } y_j^*(t) \geq \theta \\ -1, & \text{si } y_j^*(t) < \theta \end{cases}$$

$$\text{Sigmoide: } y_j(t + \Delta t) = \left[1 + e^{-\frac{y_j^*}{T}} \right]^{-1}$$

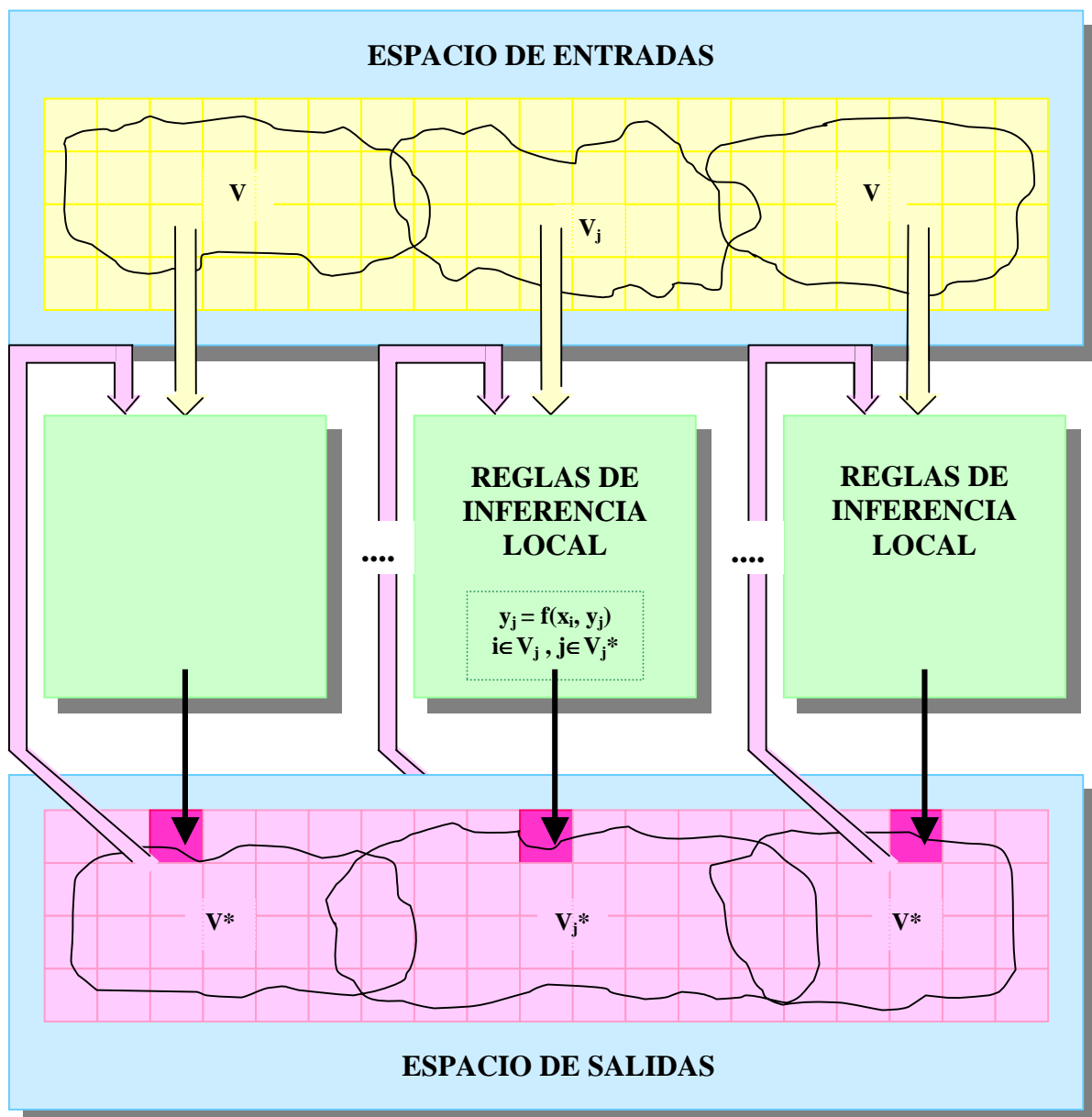


Figura 2.4. Esquema de computación neuronal en una capa

Tangente hiperbólica:

$$y_j(t + \Delta t) = \frac{1 - e^{-\frac{y_j^*}{T}}}{1 + e^{-\frac{y_j^*}{T}}}$$

Estas funciones umbral más conocidas, se muestran en la figura 2.5. Hay otras muchas variaciones posibles sobre la forma de la función umbral, desde dejar pasar la excitación sin acotarla o acotándola con dos zonas de saturación, hasta seleccionar el valor máximo.

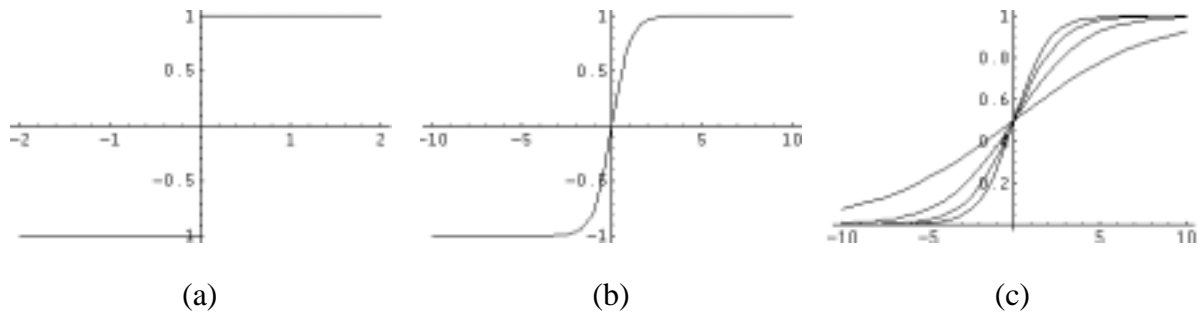


Figura 2.5. Funciones umbrales más conocidas. (a) Abrupta. (b) Tangente hiperbólica. (c) Sigmoide.

La función excitación viene definida en gran medida por la forma de los campos receptivos que definen tanto la conectividad hacia delante como la recurrencia del modelo. Es decir, la conectividad entre las neuronas de una misma capa hace depender cada respuesta de las respuestas de la misma unidad y de sus vecinas. La función excitación más conocida es, sin lugar a dudas, la función de Gauss, de la que se da una representación tridimensional, así como un gráfico de contornos en la figura 2.6.

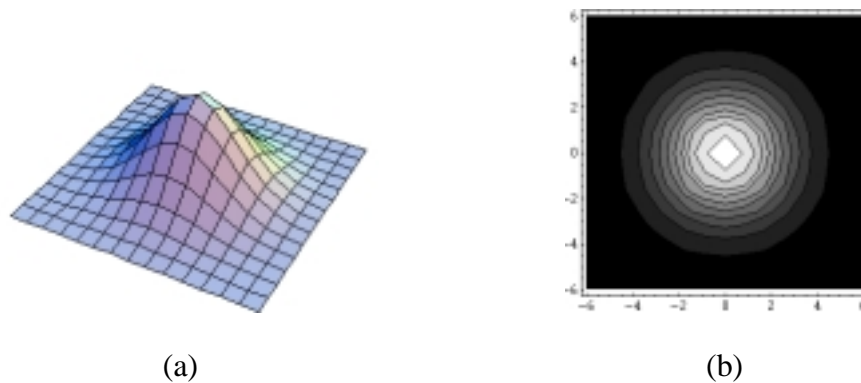


Figura 2.6. La función de Gauss. (a) Representación tridimensional. (b) Gráfico de contornos

Al pasar de redes analógicas a redes lógicas, debe modificarse simplemente el tipo de las variables de entrada y el conjunto de operadores usados para rellenar el mismo modelo. Ahora, las entradas son variables lógicas, las sumas son sumas lógicas y los productos son productos lógicos. Los parámetros son también de naturaleza booleana.

Por último, puede extenderse la capacidad computacional de las redes neuronales mediante el mismo modelo, sustituyendo el operador lógico por un condicional múltiple. Estas redes inferenciales (redes de condicionales) son ahora programas paralelos elementales con capacidad de aprendizaje. De algún modo, se ha sustituido la red de neuronas artificial por una

red de condicionales hardware que evalúan en paralelo y en tiempo real la satisfacción o no de una condición genérica espacio-temporal sobre los datos que cubre su campo receptivo.

2.2.4. Modelo de computación acumulativa local

La computación local en el córtex es de muy diferentes estilos. Si la computación ha de ser un paradigma apropiado para describir los sistemas nerviosos, debe ser distribuida a lo largo de todo el tejido con multiplicidad funcional de cada una de las neuronas. El estado actual del conocimiento incluye, entre otros, las siguientes propiedades computacionales en las neuronas reales [Mir95]:

- Hay sinapsis rápidas y lentas, lo que permite la coexistencia de dos niveles de intercomunicación, respuesta y aprendizaje.
- Pueden ser excitadoras o inhibitoras, y sobre esta distinción (sumar, restar; acumular, decrementar,...) se basa toda la computación analítica.
- Las sinapsis de naturaleza química son selectivas a distintos tipos de mensajeros, lo que permite interpretar la computación neuronal como “paso de mensajes” y “condicionales específicos”.
- Actúan de forma cooperativa y competitiva.
- La formulación más usual de la competitividad es la inhibición lateral que disminuye la respuesta de una unidad cuando están activas sus vecinas.
- Los contactos de las neuronas son plásticos y constituyen uno de los soportes estructurales de la autoprogramación (aprendizaje) y de la memoria.

El modelo más usual para decidir si se dispara o no el axon, como ya hemos visto, es el de suma espacio-temporal seguida de una función umbral. Sin embargo, la biología ofrece evidencia de modos de computación local mucho más complejos [Mir93a]:

- Traducción logarítmica en sensores y posterior cálculo sobre esa representación logarítmica.
- Inhibición y facilitación pre y postsináptica multiplicativas.
- Función de umbral adaptativa y con periodos refractorios absoluto y relativo.

- Computación no analítica que exige el uso de condicionales y otras estructuras de control.
- Existencia de actividad espontánea y distintos modos de respuesta.

Alguna de las opciones vistas han dado lugar al denominado modelo de computación acumulativa local [Fer92]. Este modelo responde básicamente a un módulo secuencial representado por su valor de carga. El valor de carga se denomina también valor de permanencia y generalmente se almacena en una memoria de permanencia. El proceso de computación acumulativa trabaja sobre un parámetro de entrada y responde con una salida denominada el valor de descarga del módulo. El valor de salida del elemento (x,y) , $Salida(x,y,t)$, es función del valor de carga $Permanencia(x,y,t)$, como recoge la siguiente ecuación:

$$Salida(x, y, t) = \eta Permanencia(x, y, t)$$

siendo η una función de umbralización dependiente del valor de umbral θ , tal que

$$\eta = \begin{cases} 0, & \text{si } Permanencia(x, y, t) < \theta \\ 1, & \text{en caso contrario} \end{cases}$$

Por lo tanto, vemos que la descarga solamente toma efecto si el valor de carga del elemento (x,y) supera un umbral mínimo θ .

El módulo calcula en cada instante de tiempo t el valor de carga $Permanencia(x,y,t)$ en función del valor de permanencia en el instante de tiempo anterior $t-\Delta t$, es decir $Permanencia(x,y,t-\Delta t)$, y en función del valor de la señal de entrada $Entrada(x,y,t)$. La función de permanencia se denomina F_p . Véase que en el modelo de computación acumulativa, el resultado anterior debe siempre caer entre dos límites establecidos v_{des} y v_{sat} correspondientes a los estados de descargado y saturado del elemento (x,y) en el instante de tiempo t .

$$Permanencia(x, y, t) = (1 - \lambda^2) \cdot F_p[Permanencia(x, y, t - \Delta t), Entrada(x, y, t)] + \frac{\lambda(\lambda + 1)}{2} v_{des} + \frac{\lambda(\lambda - 1)}{2} v_{sat}$$

donde λ es una función de acotación que lleva todo valor de carga al intervalo $[v_{des}, v_{sat}]$ y viene definida como

$$\lambda = \begin{cases} 1, & \text{si } F_p[\dots] \geq v_{sat} \\ 0, & \text{si } v_{des} < F_p[\dots] < v_{sat} \\ -1, & \text{si } v_{des} \geq F_p[\dots] \end{cases}$$

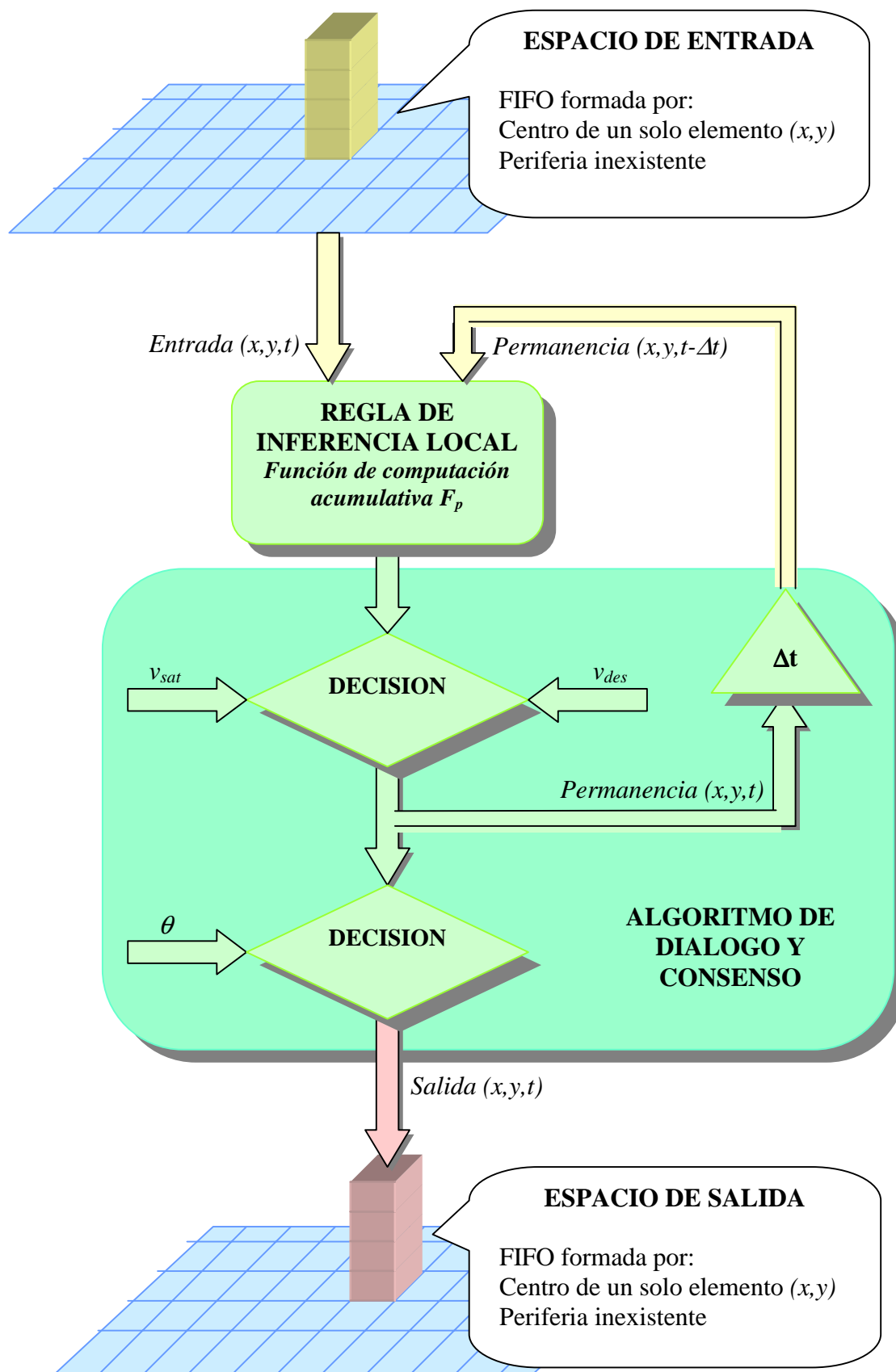


Figura 2.7. Modelo de computación acumulativa local

Gráficamente, tenemos el modelo de computación acumulativa local de la figura 2.7.

La estructura del espacio de entradas y salidas es la de las memorias FIFO para incluir el tiempo como una variable de cálculo. En el espacio de salida de una capa tenemos una pila de respuestas de las unidades de esta capa. Así, funciona como una memoria local y transitoria que salva las salidas de todas las neuronas de la capa durante algún intervalo de tiempo.

La naturaleza de la función de permanencia o computación acumulativa F_p puede ser analógica, lógica o inferencial, de modo que puede cubrir todos los mecanismos computacionales.

2.2.5. Modelo de computación de doble temporización

Las neuronas adultas son células especializadas en la integración y distribución de estados globales de excitación en los campos receptivos que se manifiestan como potenciales de acción. Sin embargo, la diversidad de contactos (axon-soma, dendrita-dendrita, dendrita-soma, ...), junto con la especificidad y la complejidad de la computación efectuada por estos contactos, hacen pensar en una microcomputación subcelular [Mir93a]. Es decir, antes de llegar al nivel de la neurona, y considerando solamente el proceso local existente en su campo dendrítico, ya existe evidencia suficiente experimental para formular un modelo de computación sináptica con las siguientes propiedades [Fer95a]:

1. Procesamiento local convergente alrededor de cada sinapsis.
2. Funcionamiento semiautónomo, donde cada sinapsis es capaz de acumular espacio-temporalmente las entradas locales (escala de tiempos t_n) y de descargar condicionalmente.
3. Transmisión atenuada de estas acumulaciones de coincidencias persistentes en las diferentes sinapsis del campo dendrítico hacia la zona “spike trigger” de la neurona que integra los resultados de estas memorias locales de carga / descarga en una escala de tiempos global (T_k).

Por lo tanto, vemos que uno de los aspectos específicos de la microcomputación subcelular es la existencia de dos escalas de tiempo:

(a) el tiempo local dendro-dendrítico $t_n = n \Delta t$

(b) el tiempo global axonal $T_k = k \Delta T$

con $\Delta T \gg \Delta t$.

Las siguientes propiedades son aplicables a este nivel: (a) un proceso de convergencia local alrededor de cada elemento, (b) un funcionamiento semiautónomo, donde cada elemento es capaz de acumular espacio-temporalmente las entradas locales en la escala de tiempo local T , y una descarga condicional, y, (c) una transmisión atenuada de estas acumulaciones de coincidencias persistentes hacia la periferia que se integra en la escala de tiempo global t . Vemos que estamos ante dos escalas de tiempos diferenciadas: (a) el tiempo local T , y, (b) el tiempo global t , $T \ll t$.

La figura 2.9. inspirada en Mira et al. [Mir96b] muestra un entorno para el modelado de procesos cooperativos a nivel simbólico, recogiendo las dos escalas de tiempo vistas con anterioridad. La figura 2.8., a su vez, muestra más claramente los intervalos de tiempo global y local.

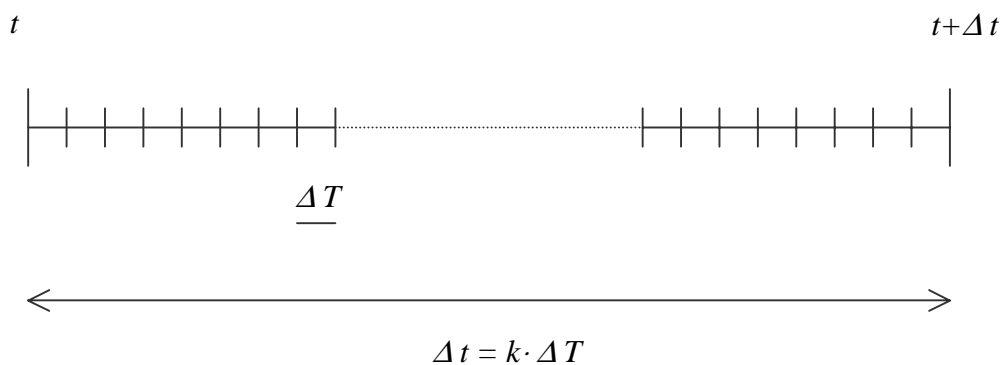


Figura 2.8. Comparación de los tiempos global y local

2.2.6. Modelo de interacción lateral recurrente

En los modelos de interacción lateral [Mir93b] [Mir96b] tenemos una capa de módulos del mismo tipo con conectividad local, de modo que la respuesta de un elemento dado no solamente depende de sus propias entradas, sino también de las entradas y salidas de los elementos vecinos. Desde un punto de vista computacional las redes de interacción lateral descomponen el espacio de entrada en tres regiones: centro, periferia y excluido.

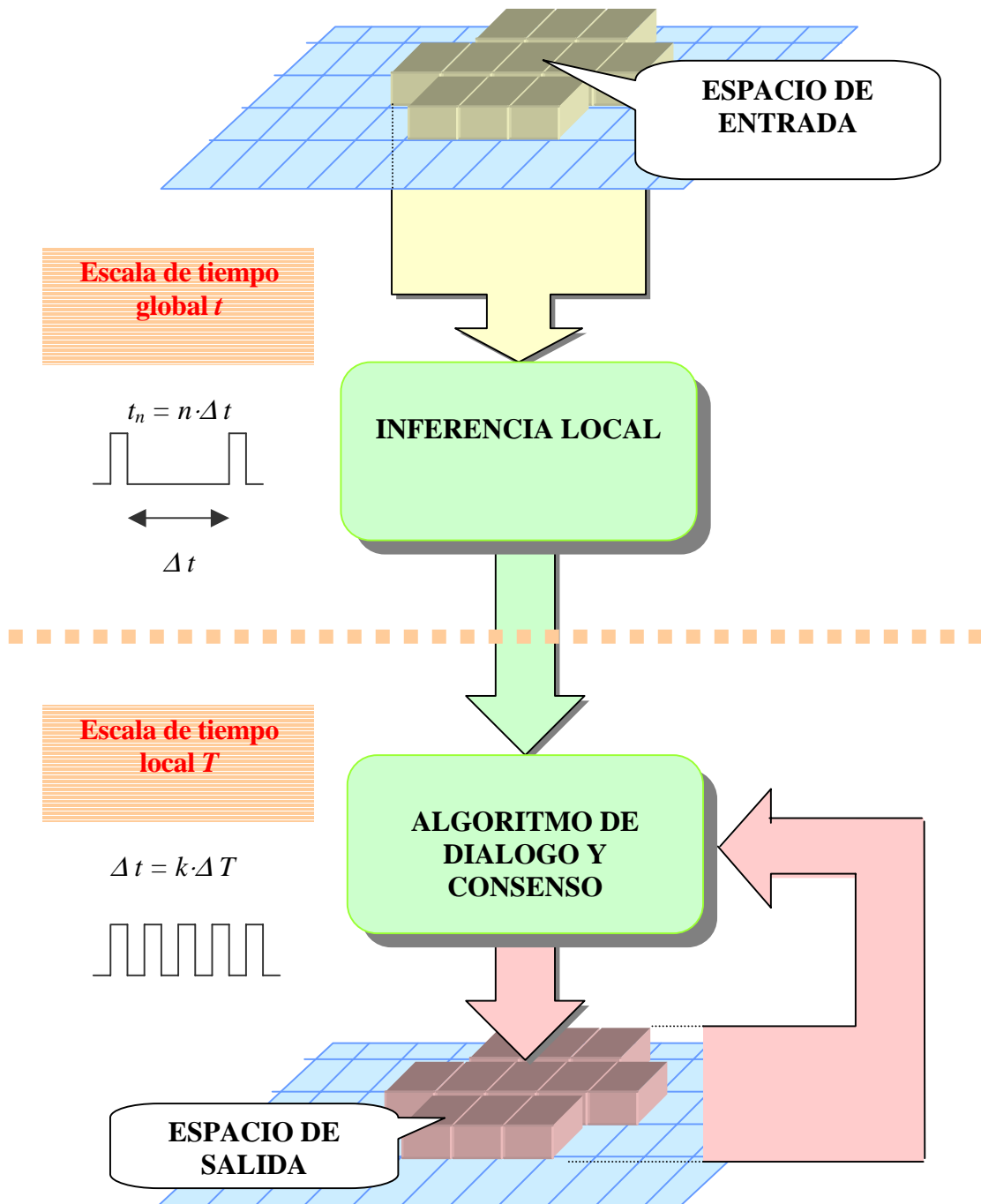


Figura 2.9. Modelo de computación de doble temporización

La figura 2.10. muestra la obtención de las tres regiones de un espacio de entrada a partir de la diferencia de dos gaussianas con diferentes desviaciones típicas σ_1 y σ_2 . Las dos funciones gaussianas se ofrecen superpuestas en la columna (b), mientras que el espacio de entrada resultante, con las tres regiones obtenidas, se ofrece en la columna (a) de la figura.

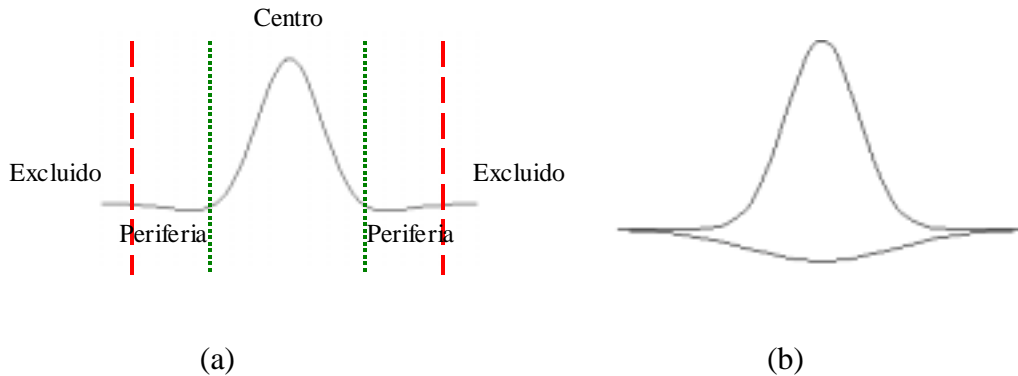


Figura 2.10. Obtención de las regiones del espacio de entrada. (a) La diferencia de gaussianas. (b) Las dos funciones gaussianas

La figura obtenida se conoce con las iniciales de DOG (Difference Of Gaussians) y recibe el nombre común de sombrero mexicano. Su ecuación general para dos coordenadas es:

$$DOG_{\sigma_1\sigma_2}(x, y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}}$$

Una excelente aproximación a la función DOG es la LOG (Laplacian Of Gaussian), cuya ecuación es:

$$LOG_{\sigma}(x, y) = \frac{1}{\pi\sigma^4} \left[\frac{x^2 + y^2}{2\sigma^2} - 1 \right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

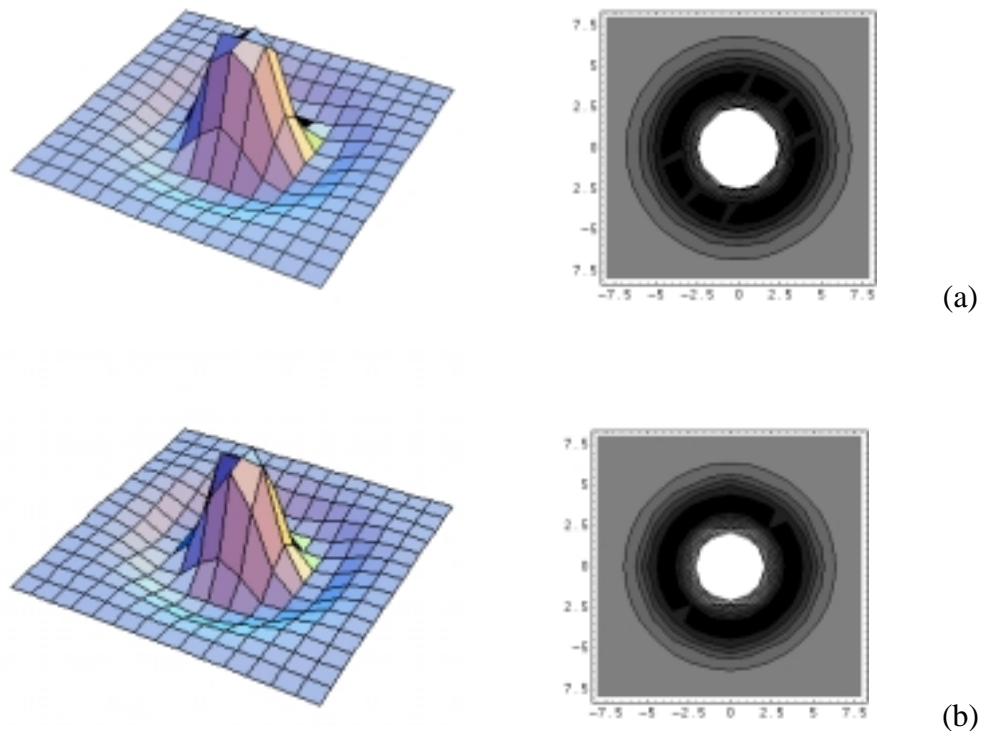


Figura 2.11. (a) Función LOG. (b) Función DOG

La literatura científica ha provisto de gran multitud de otras funciones que permiten descomponer el espacio de entrada en las tres regiones de centro, periferia y excluido.

Veamos ahora la importancia del carácter inhibitor o excitador de las sinapsis. En la neurología, encontramos que los campos receptivos en las neuronas reales están organizados en estructuras “centro”-“periferia” o “centro”-“periferia próxima”-“periferia lejana”, en las que se alternan zonas excitadoras con zonas inhibitoras. Si el centro es excitador y la periferia inhibitora, la red actúa como filtro pasa-alta (derivador espacial, temporal o espacio-temporal. Si se extiende la zona excitadora o inhibitora, entonces la función de la capa es de filtro pasa-baja (integrador espacio-temporal) que calcula valores medios.

De gran importancia son también las funciones u operadores basados en máscaras direccionales, capaces de detectar los bordes a partir de la correlación de la entrada con patrones en todas las orientaciones posibles. En la práctica, el número de orientaciones se limita a 8 (giradas 45 grados). El valor del gradiente se obtiene como el máximo de las respuestas a cada una de las máscaras, mientras que la dirección será la indicada por la orientación de dicha máscara. Se han propuesto muchas máscaras direccionales [Pra91].

Son también numerosas las funciones continuas utilizadas para la obtención de la direccionalidad. Aquí ofrecemos un ejemplo concreto en la figura 2.12. [Sir96], donde la función tiene la forma:

$$F(x, y) = e^{-\left[\frac{(x \cos \alpha - y \sin \alpha)^2}{a^2} + \frac{(x \sin \alpha + y \cos \alpha)^2}{b^2} \right]}$$

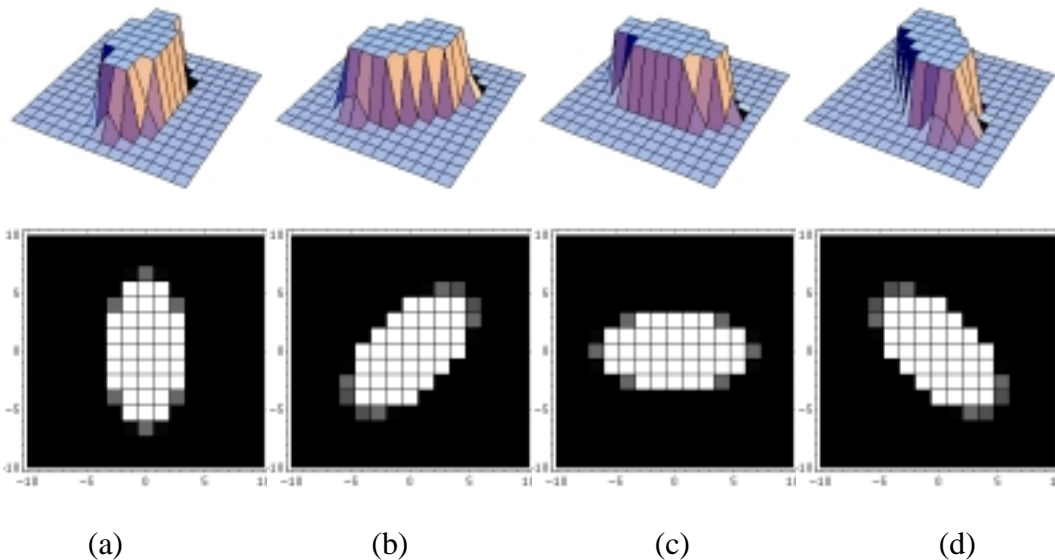


Figura 2.12. Una función de obtención de la dirección. (a) $a=1, b=2, \alpha=0$. (b) $a=1, b=2, \alpha = \frac{\pi}{4}$. (c) $a=1, b=2, \alpha = \frac{\pi}{2}$. (d) $a=1, b=2, \alpha = \frac{3\pi}{4}$

Concretando un poco más, en los modelos de interacción lateral se realizan los siguientes pasos: (a) un proceso sobre la zona central, (b) un proceso sobre las señales procedentes de la zona periférica, (c) una comparación de los resultados de estas operaciones y la generación de una decisión local, y, (d) una distribución sobre el espacio de salida. Existen dos expresiones generales para formular la interacción lateral [Mor69]: una para la interacción no recurrente y otra para la interacción recurrente. En la figura 2.13., *Entrada* (α, β) es la señal de entrada a un elemento situado en las coordenadas (α, β) , *Salida* (x, y) es la señal de salida de un elemento en la posición (x, y) , y $\kappa(\alpha, \beta, x, y)$ es el factor de peso que traslada el efecto del elemento (α, β) sobre el elemento (x, y) .

La salida del elemento (x, y) para el caso común de interacción recurrente puede escribirse a partir de Moreno-Díaz et al. [Mor69] como:

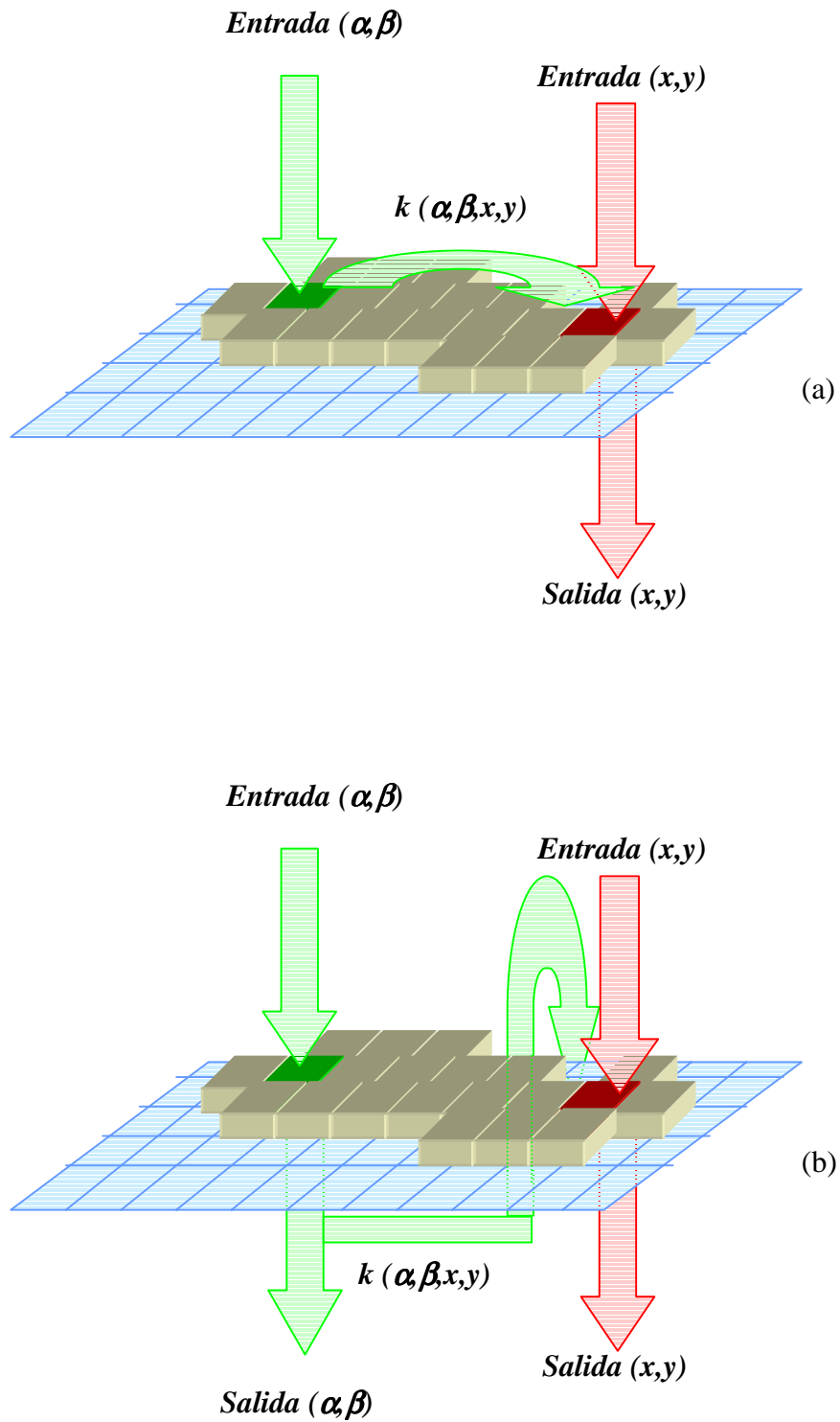


Figura 2.13. Interacción lateral. (a) No recurrente. (b) Recurrente.

$$Salida(x, y) = Entrada(x, y) + \iint_R \kappa(\alpha, \beta, x, y) Salida(\alpha, \beta) d\alpha d\beta$$

Lo más interesante de estos esquemas de conectividad con inhibición recurrente es que existe en prácticamente cualquier nivel de vertebrados e invertebrados, desde las primeras capas del camino sensorial (retina) hasta las capas más centrales del córtex cerebral. El otro aspecto interesante es que las redes de interacción lateral definen familias computacionales que pueden ser proyectadas según su naturaleza (analógica, lógica o inferencial) al seleccionar el “operador” sináptico.

Gráficamente podríamos tener el modelo de interacción lateral recurrente de la figura 2.14.

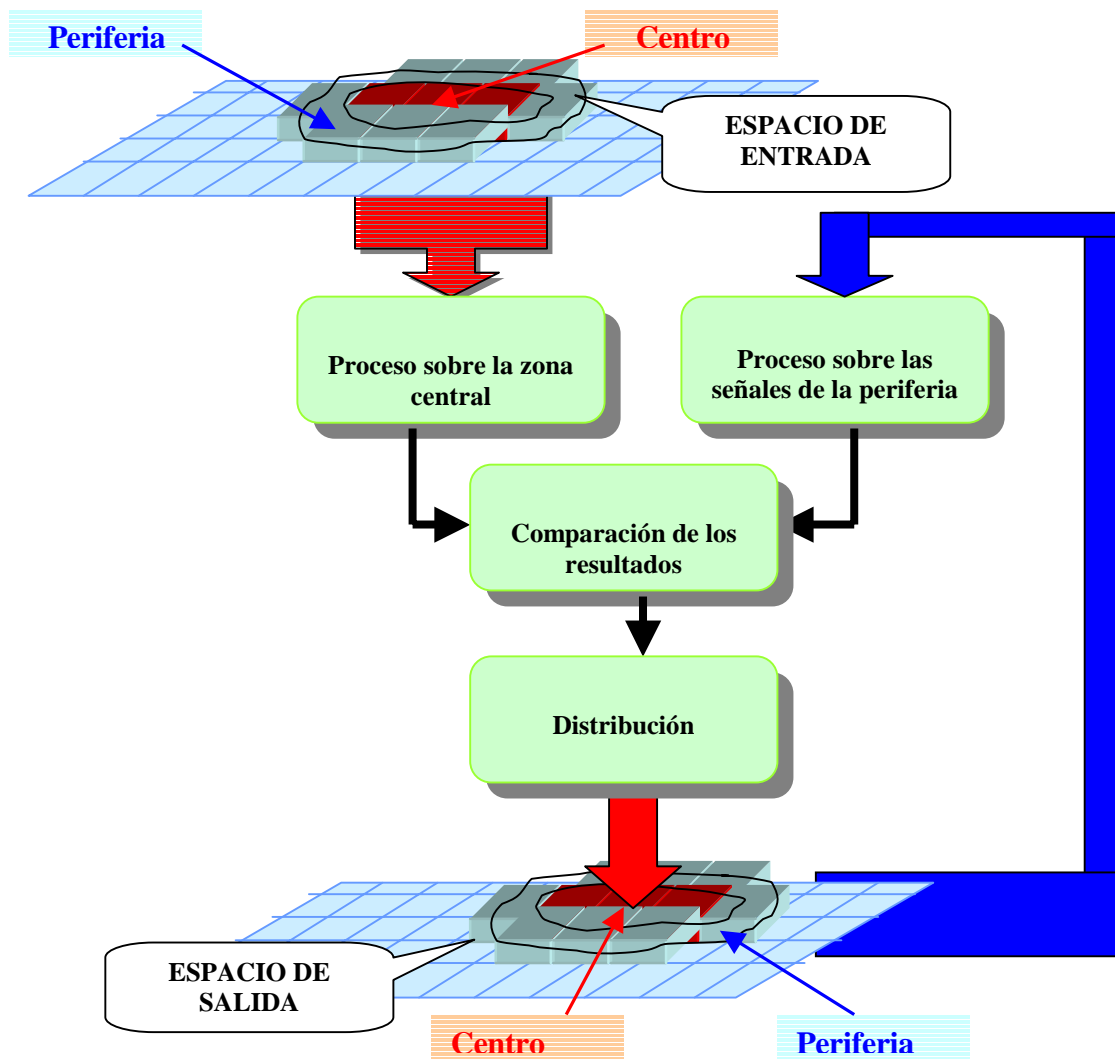


Figura 2.14. Modelo de interacción lateral recurrente

2.3. Modelo de interacción lateral en computación acumulativa

2.3.1. Interacción lateral usando unidades de computación acumulativa

El modelo propuesto incorpora todas las nociones de carácter general vistas anteriormente, agrupando nociones tan habituales en la biología como la computación acumulativa, la interacción lateral y la doble temporización. Por ello, se incorporan, a grandes rasgos, las ideas que siguen.

A un nivel simbólico tenemos espacios de entrada y salida y un conjunto de procesos cooperativos que realizan funciones de inferencia locales sobre el campo receptivo organizado como centro-periferia e intercambian la información básica dada y recibida por cada proceso en el área de diálogo. Tras la inferencia individual y local, se usan algoritmos de diálogo y consenso recurrentes y sucesivos, comparando las opiniones individuales con las de los vecinos de la periferia.

Todo proceso inscrito en el modelo de interacción lateral en computación acumulativa debe verse por medio de las siguientes características:

- C.1 Aplicación de la computación acumulativa sobre cada elemento central a partir del estado del propio elemento y de las entradas procedentes de las capas anteriores.
- C.2 Aplicación de los mecanismos de interacción lateral (inhibición lateral) de la periferia formada por los cuatro vecinos más próximos sobre el centro (x,y) , es decir, los elementos $(x-I,y)$, $(x+I,y)$, $(x,y-I)$, $(x,y+I)$, usando canales específicos que incorporen la capacidad de abrirse y cerrarse.
 - C.2.1 La interacción es de tipo recurrente.
 - C.2.2 La interacción lateral entre los distintos elementos tiene lugar a través de factores de peso, donde todos los vecinos tienen el mismo peso.
 - C.2.3 El efecto total sobre un elemento sujeto a interacción lateral es lineal, es decir, que los distintos efectos particulares sobre el mismo se suman.
 - C.2.4 En cuanto a la dimensionalidad del modelo, cabe decir que la interacción lateral toma lugar en las coordenadas espacio-temporales.

C.3 La escala de tiempos global t se usa (a) para la entrada de la capa anterior, (b) durante el proceso de computación acumulativa, y (c) para la salida hacia la capa posterior.

C.4 La escala de tiempos local $T \ll t$ se usa en todos los mecanismos cooperativos de interacción lateral.

Estas características llevan a un modelo donde cada módulo (x,y) en una capa dada puede representarse mediante la figura 2.15.

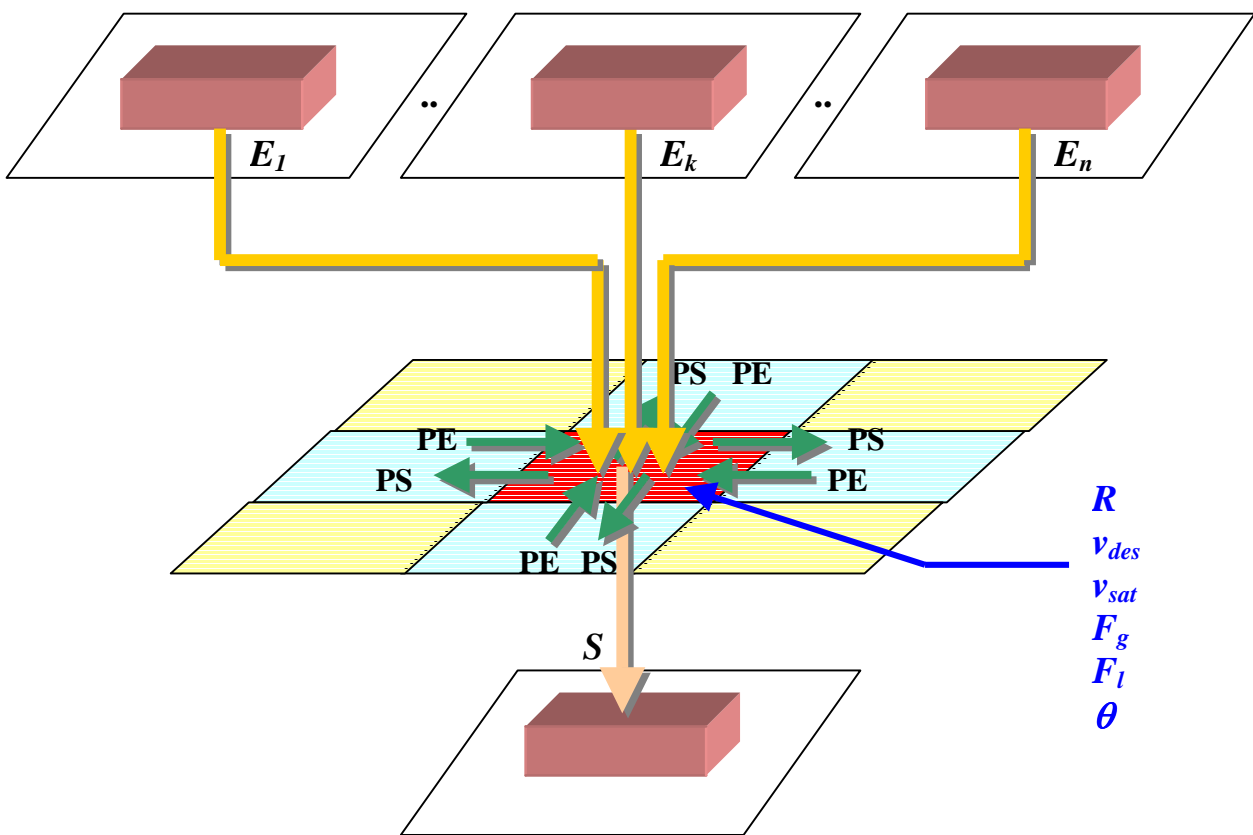


Figura 2.15. El módulo (x,y) y sus conexiones con la capa anterior (E), la capa siguiente (S) y laterales con sus cuatro vecinos más próximos

Esta serie de características específicas nos llevan igualmente a unir las funciones computacionales del modelo de interacción lateral y del modelo de computación acumulativa, a añadir la noción de doble temporización, obteniéndose el funcionamiento del modelo de interacción lateral en computación acumulativa en una capa. Gráficamente, podemos representar el modelo de interacción lateral en computación acumulativa según la figura 2.16.

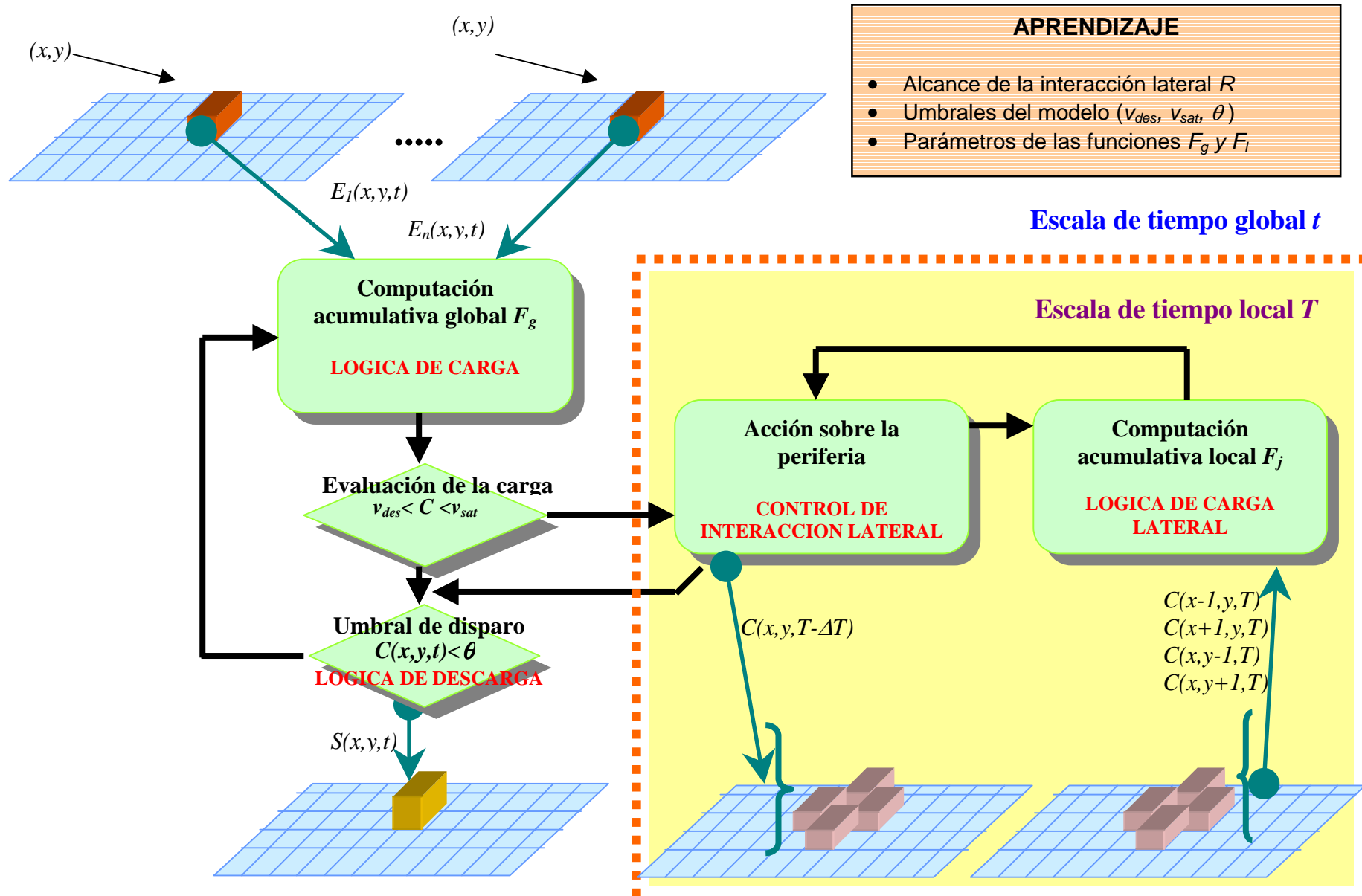


Figura 2.16. Modelo de neurona de interacción lateral en computación acumulativa

El proceso cooperativo realizado por un grupo funcional de neuronas se caracteriza a nivel simbólico por los siguientes parámetros:

$$\langle G, E_i(x, y, t), S(x, y, t), C(x, y, t), R, F_g, F_l, v_{des}, v_{sat}, \theta \rangle$$

donde,

G ,	función global a nivel de conocimiento; a definir, en cada caso, en la utilización del modelo de interacción lateral en computación acumulativa
$E_i(x, y, t)$,	entrada desde la capa anterior, espacio de entrada i , al módulo (x, y) en el instante t
$S(x, y, t)$,	salida hacia el módulo (x, y) de la capa siguiente en el instante t , valor de descarga
$C(x, y, t)$,	valor de carga del módulo (x, y) en el instante t
R	frontera funcional en los espacios de entrada y salida
F_g ,	función de acumulación espacial en la escala de tiempos global t
F_l ,	función de acumulación espacio-temporal en la escala de tiempos local T
v_{des} ,	valor mínimo de carga; descarga
v_{sat} ,	valor máximo de carga; saturación
θ ,	valor del umbral de disparo

La función global G puede entenderse fácilmente si pensamos en la idea de red neuronal genérica, con funciones de clasificación, organizada en distintos grupos funcionales que calculan de forma concurrente y/o secuencial y que incluye, al menos, tantos tipos de neuronas como capas. Las distintas capas estarían asociadas a la descomposición del problema en distintas tareas, como, por ejemplo, la extracción de características.

2.3.2. Aprendizaje en el modelo

Una destacada mención merece el aprendizaje en este modelo cooperativo ya que permite redefinir, dependiendo de la función global G elegida, todos los parámetros, incluyendo los

parámetros relacionados con los campos receptivos de entrada y salida, así como las fronteras funcionales de los mismos.

Está claro que no hay computación neuronal sin aprendizaje y las reglas del aprendizaje son los algoritmos de modificación de los parámetros de computación local de acuerdo con criterios globales de evaluación del comportamiento de la red. Independientemente del tipo de aprendizaje, todo aprendizaje en redes de neuronas artificiales se formula como un procedimiento para modificar el valor de un conjunto de parámetros de acuerdo con la historia de las señales que se reciben del medio durante la etapa de entrenamiento y de las respuestas de la red ante esas entradas [Mir95]. La función que realiza una neurona siempre se puede escribir de la forma:

$$y_j(t + \Delta t) = f_i \left[x_i(t - K_1 \cdot \Delta t); y_j(t - K_2 \cdot \Delta t); \bar{p}_j \right]$$

siendo \bar{p}_j un vector de parámetros $(p_j^1, p_j^2, \dots, p_j^q)$ y K_1 y K_2 el número de retardos en las FIFO de entrada y salida, respectivamente, que cubren los campos receptivos de entrada y salida.

Esta asociación se evalúa por distintos procedimientos y como consecuencia de esa evaluación se propone un nuevo valor para el vector de parámetros

$$\bar{p}_j(t + \Delta t) = A_j \left[\bar{p}_j(t); x_i(t - K_1 \cdot \Delta t); y_j(t - K_2 \cdot \Delta t); \text{criterio de aprendizaje} \right]$$

Estas funciones A_j son las responsables del aprendizaje. Su naturaleza, y la de los parámetros \bar{p}_j que modifican, depende del modelo de neurona y del algoritmo de aprendizaje. En el caso del modelo de interacción lateral en computación acumulativa, los parámetros que van modificando su valor durante el aprendizaje son v_{des} , v_{sat} y θ . Otros parámetros susceptibles de aprendizaje irán apareciendo como consecuencia de la utilización de funciones concretas F_g y F_l .

El concepto de vecindad adaptativa que se usa en visión artificial para seleccionar una escala de trabajo en cada punto de la imagen y en cada nivel de computación, es de gran interés en el modelo de interacción lateral en computación acumulativa en cuanto al aprendizaje se refiere. La idea es que tanto el tamaño como la forma de la periferia dependen de las características de los datos de la imagen a través de parámetros que definen la medida de la homogeneidad de cada uno de los píxeles. La adaptación dinámica (aprendizaje) de diferentes periferias permite estimar diferentes propiedades. La periferia puede tratarse como una función de ciertas medidas de homogeneidad sobre los campos receptivos. Evidentemente, cualquier

concepto de similitud debe ser inyectado como conocimiento externo en el diseño de la red. El parámetro que admite aprendizaje en este contexto es la frontera funcional R .

2.3.3. Pasos generales asociados al modelo

Son tres los pasos generales asociados al modelo computacional de interacción lateral en computación acumulativa, a saber una acumulación espacial, una acumulación espacio-temporal y una distribución. El paso segundo se ejecuta únicamente si la carga $C(x,y,t)$ se evalúa dentro de unos límites preestablecidos $[v_{des}, v_{sat}]$.

Estos pasos se pasan a detallar a continuación en la figura 2.17.

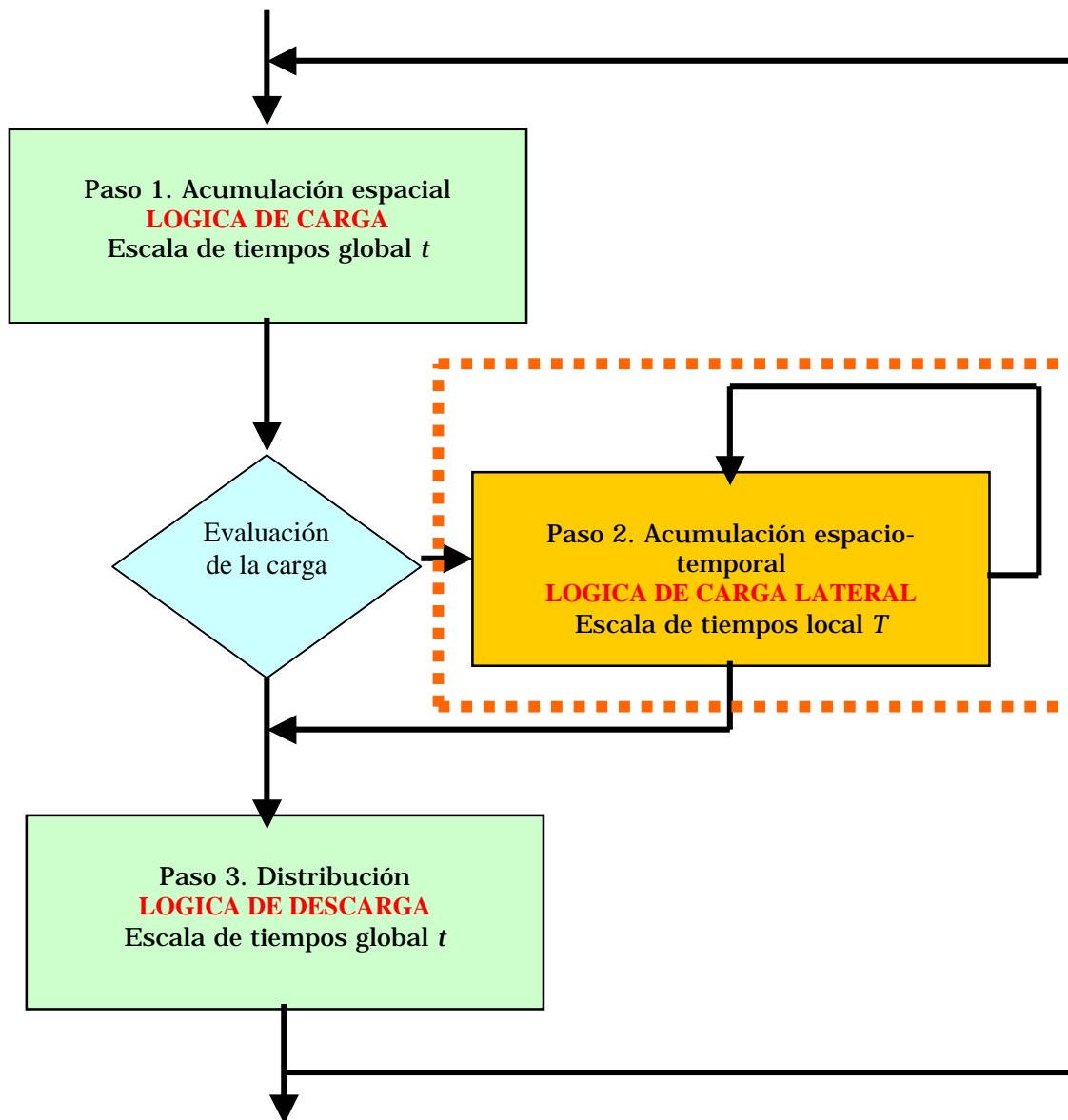


Figura 2.17. Pasos generales del modelo computacional

Paso 1. Acumulación espacial (escala de tiempos global t)

En primer lugar, la función de computación acumulativa global, dependiente de los valores de entrada de la capa anterior, actúa del siguiente modo sobre el valor de carga de cada uno de los elementos centrales (x,y) en el instante global de tiempo t :

$$C(x, y, t) = (1 - \lambda_1^2) \cdot F_g [C(x, y, t - \Delta T), \max(E_1(x, y, t), \dots, E_n(x, y, t))] + \frac{\lambda_1(\lambda_1 + 1)}{2} v_{des} + \frac{\lambda_1(\lambda_1 - 1)}{2} v_{sat}$$

$$\text{donde } \lambda_1 = \begin{cases} 1, & \text{si } F_g[\dots] \geq v_{sat} \\ 0, & \text{si } v_{des} < F_g[\dots] < v_{sat} \\ -1, & \text{si } v_{des} \geq F_g[\dots] \end{cases}$$

define una función de acotado que limita el valor de carga al intervalo $[v_{des}, v_{sat}]$. De este modo, se ha conseguido el valor de la opinión individual del elemento (x,y) a partir del valor de entrada al módulo y la historia del estado del propio módulo.

Paso 2. Acumulación espacio-temporal (escala de tiempos local T)

En la escala de tiempos local T tenemos una fase iterativa de actualización del valor de carga derivado de la interacción lateral de cada módulo (x,y) con sus cuatro vecinos más próximos, siempre que el elemento central (x,y) posea una carga suficiente comprendida entre los límites v_{des} y v_{sat} . Véase que el tiempo global $t = n \cdot \Delta t$ se congela durante el este diálogo iterativo. No así el tiempo local $T = k \cdot \Delta T$.

En primer lugar se da una escritura sobre el espacio de salida de la periferia siempre que los canales de salida del módulo (x,y) estén abiertos.

$$C(\alpha, \beta, T) = C(x, y, T), \quad \forall (\alpha, \beta) \in [x \pm 1, y \pm 1]$$

En segundo lugar, se realiza el recálculo de la carga del módulo (x,y) a partir de la información disponible en la periferia del módulo.

$$C(x, y, T) = (1 - \lambda_2^2) \cdot F_l \left[C(x, y, T - \Delta T), \sum_{\alpha=x-1}^{x+1} \sum_{\beta=y-1}^{y+1} C(\alpha, \beta, T - \Delta T) \right] + \frac{\lambda_2(\lambda_2 + 1)}{2} v_{des} + \frac{\lambda_2(\lambda_2 - 1)}{2} v_{sat}, \quad \forall (\alpha, \beta) \neq (x, y), (\alpha, \beta) \in R, (a, b) \in R$$

donde $\lambda_2 = \begin{cases} 1, & \text{si } F_l[...]\geq v_{sat} \\ 0, & \text{si } v_{des} < F_l[...]< v_{sat} \\ -1, & \text{si } v_{des} \geq F_l[...]\end{cases}$ define otra función de acotado que limita el valor de

carga al intervalo $[v_{des}, v_{sat}]$. Este es el modo mediante el cual cada una de las opiniones individuales es contrastada con la de sus vecinos próximos.

Paso 3. Distribución (escala de tiempos global t)

Por último, en la escala de tiempos global t , tiene lugar la salida hacia el siguiente nivel. La salida viene dada por:

$$S(x, y, t) = \eta C(x, y, t)$$

siendo $\eta = \begin{cases} 0, & \text{si } C(x, y, t) < \theta \\ 1, & \text{en caso contrario}\end{cases}$ una función de umbralización. Esta salida será reflejo del

algoritmo de consenso que ha producido un nuevo valor a partir de la combinación de los valores de todos los módulos cooperativos afectados.

2.3.4. Tareas de inferencia del modelo

Las tareas de inferencia y su descripción en lenguaje natural se dan en la figura 2.18. Véase que T.1 y T.4 ocurren en la escala de tiempos global t , correspondiendo a los pasos 1 y 3 del modelo computacional, respectivamente, mientras que T.2 y T.3 se dan en la escala de tiempos local $T \ll t$, correspondiendo al paso 2 del modelo.

Las tareas inferenciales precedentes se explican a continuación con todo detalle.

2.3.4.1. Inferencia local sobre el centro

T.1 INFERENCIA LOCAL SOBRE EL CENTRO (escala de tiempos global t)

T.1.1 Lectura de la entrada desde la capa anterior

Lectura de la información $E_1(x, y, t), \dots, E_n(x, y, t)$ en cada elemento (x, y) en el instante t desde la capa anterior.

Input $E_1(x, y, t), \dots, E_n(x, y, t)$

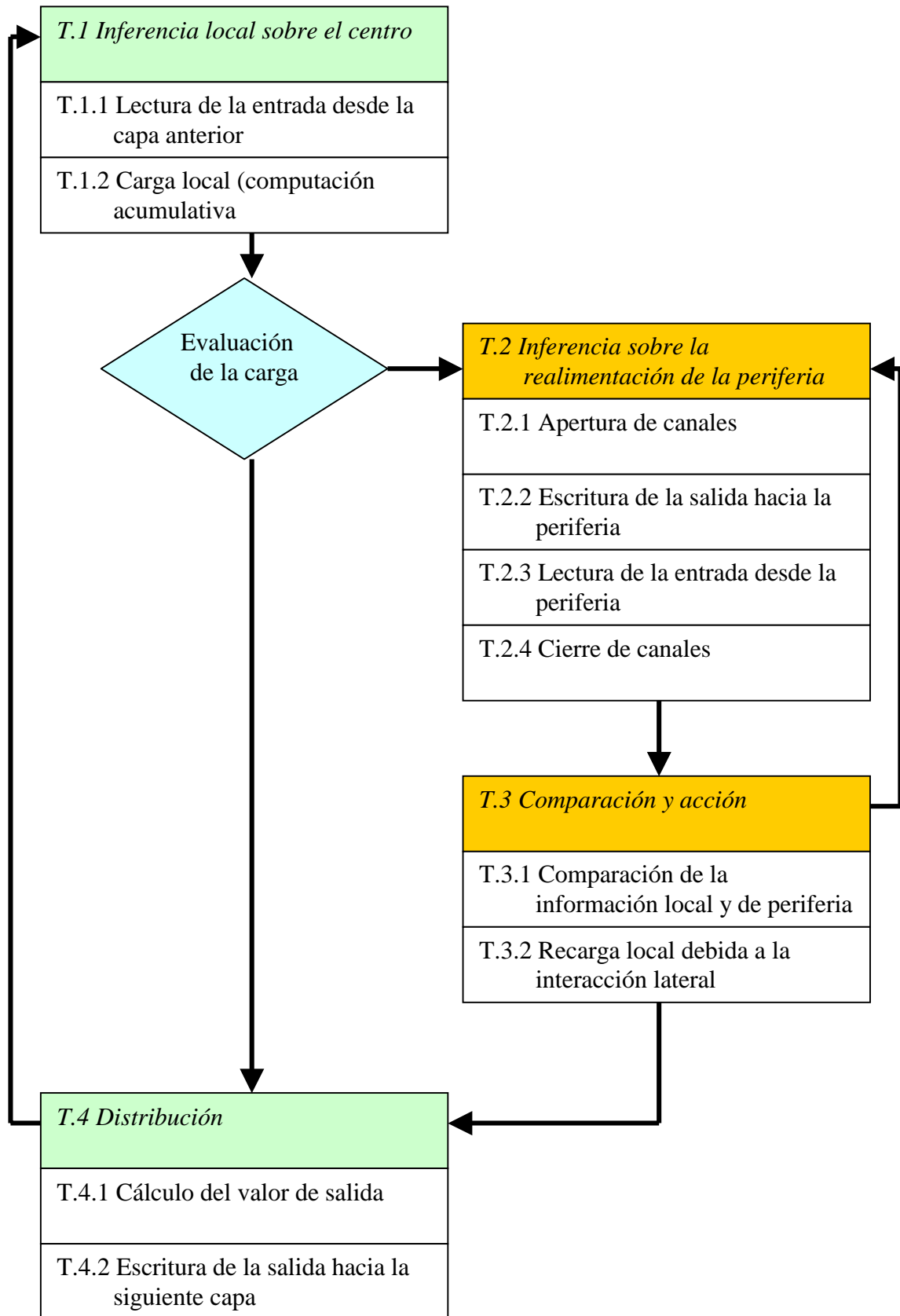


Figura 2.18. Tareas genéricas

T.1.2 Carga local (computación acumulativa)

Obtención del valor de carga $C(x,y,t)$, característico de la historia del elemento (x,y) en el instante t . Dependiendo del valor de carga anterior $C(x,y,t-\Delta t)$ y de la historia de los valores de entrada de la capa anterior, el valor de carga puede descargarse gradualmente hasta un valor mínimo de descarga completa v_{des} o cargarse del mismo modo hasta un valor máximo de saturación v_{sat} .

$$C(x,y,t) = F_g [C(x,y,t), \max(E_1(x,y,t), \dots, E_n(x,y,t))]$$

If $C(x,y,t) \leq v_{des}$ Then $C(x,y,t) = v_{des}$

Else If $C(x,y,t) \geq v_{sat}$ Then $C(x,y,t) = v_{sat}$

2.3.4.2. Inferencia sobre la realimentación de la periferia

T.2 INFERENCIA SOBRE LA REALIMENTACION DE LA PERIFERIA (escala de tiempos local T)

T.2.1 Apertura de canales

El control de los canales de recepción lateral tiene el siguiente significado: (a) si se han cerrado los canales de entrada de interacción lateral en el elemento, éste no va a aceptar ningún estímulo que proceda de la periferia, y, (b) si la estructura ha abierto sus canales de entrada de interacción lateral, sí se va a aceptar la entrada lateral. Dependiendo de la función genérica G , apertura controlada de todos los canales.

T.2.2 Escritura de la salida hacia la periferia

El elemento informa a sus vecinos de su valor de carga si los canales de salida están abiertos.

Output $C(x,y,T)$

T.2.3 Lectura de la entrada desde la periferia

Recepción lateral de la información de salida de los vecinos, siempre que los canales de entrada dedicados estén abiertos.

Input $C(\alpha,\beta,T-\Delta T)$

T.2.4 Cierre de canales

Cierre controlado (dependiente de la función genérica G) de los canales, tanto de entrada como de salida.

2.3.4.3. Comparación y acciónT.3 COMPARACION Y ACCION (escala de tiempos local T)T.3.1 Comparación de la información local y de periferia

La comparación de información se torna en evaluar el efecto de los pesos de cada una de las conexiones laterales sobre el elemento central.

T.3.2 Recarga local debida a la interacción lateral

Una recarga del valor de carga debida a la interacción lateral (recarga debida a la vecindad) puede realizarse en este instante. Aquí tenemos el último paso de la fase iterativa en la escala de tiempos local.

$$C(x,y,T) = F_l [C(x,y,T-\Delta), \sum \sum C(\alpha, \beta, T - \Delta T)]$$

$$\text{If } C(x,y,T) \leq v_{des} \text{ Then } C(x,y,T) = v_{des}$$

$$\text{Else}$$

$$\text{If } C(x,y,T) \geq v_{sat} \text{ Then } C(x,y,T) = v_{sat}$$

2.3.4.4. DistribuciónT.4 DISTRIBUCION (escala de tiempos global t)T.4.1 Cálculo del valor de salida

Verificación de si el valor de carga alcanza el umbral mínimo θ y obtención del valor de salida.

$$\text{If } C(x,y,t) \geq \theta \text{ Then } \eta = 0$$

$$\text{Else } \eta = 1$$

$$S(x,y,t) = \eta C(x,y,t)$$

T.4.2 Escritura de la salida hacia la siguiente capa

Escritura del valor de salida $S(x,y,t)$ hacia la capa siguiente.



2.3.5. Lógica de un módulo de interacción lateral en computación acumulativa

Para recoger el modelo de interacción lateral se propone una lógica modular autoprogramable para cada uno de los módulos (x,y) como la mostrada en la figura 2.19.

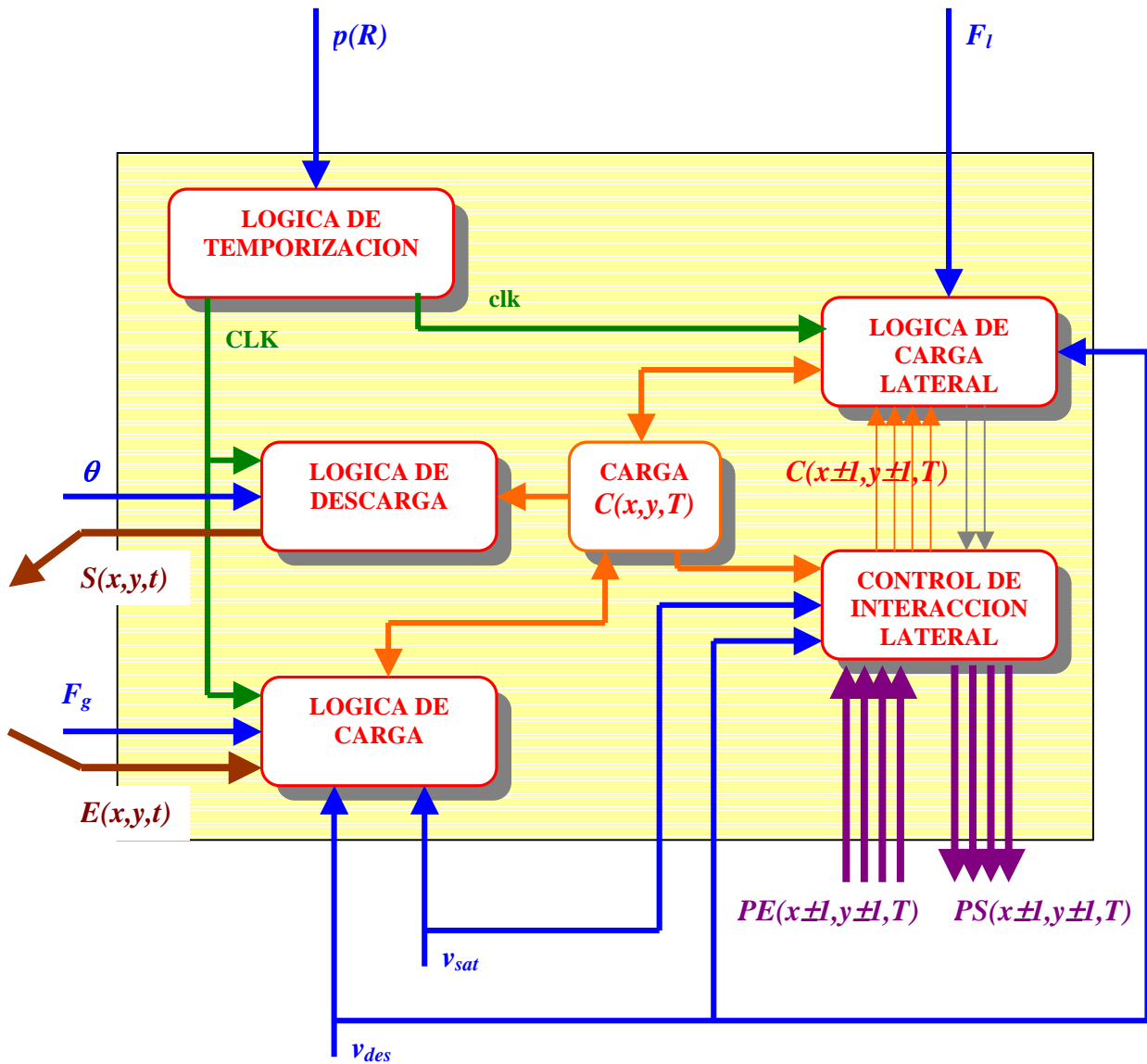


Figura 2.19. Lógica de un módulo de interacción lateral en computación acumulativa

A continuación se detalla cada una de las lógicas del esquema general anterior.

2.3.5.1. Lógica de temporización

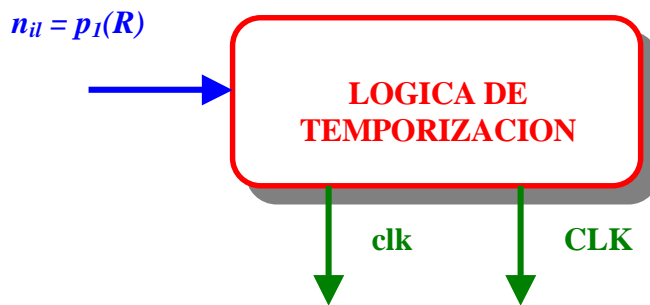


Figura 2.20. Entradas / salidas de la LOGICA DE TEMPORIZACION

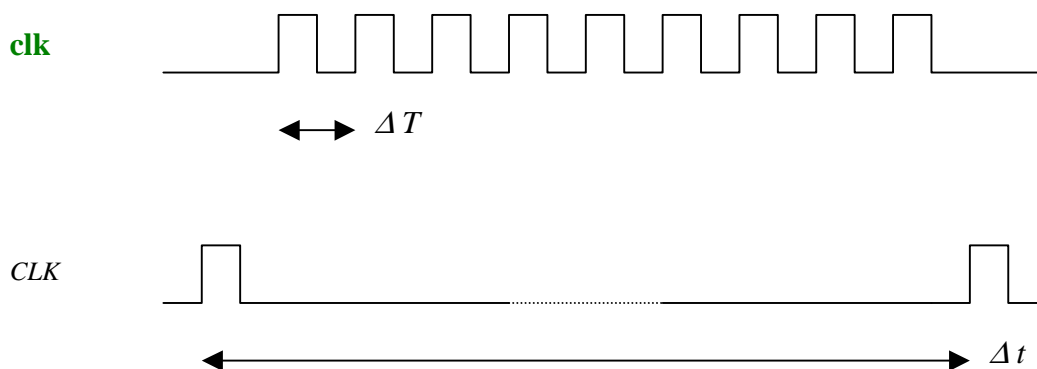


Figura 2.21. Obtención de los temporizadores clk para el tiempo local T y CLK para el tiempo global t

Descripción funcional:

La LOGICA DE TEMPORIZACION tiene como finalidad el obtener dos señales como salida de un mismo reloj, emulando las dos escalas de tiempo t y T . Estas señales son:

- clk: temporizador para la escala de tiempos local T
- CLK: temporizador para la escala de tiempos global t

El temporizador CLK (tiempo largo) se pone a '1' cada n_{il} ticks del temporizador clk (tiempo corto), dependiendo de los parámetros asociados al alcance de la interacción lateral R ,

pudiendo obtenerse una salida como la ofrecida en la figura 2.21. La figura 2.22. muestra el hardware que implementa dicha lógica.

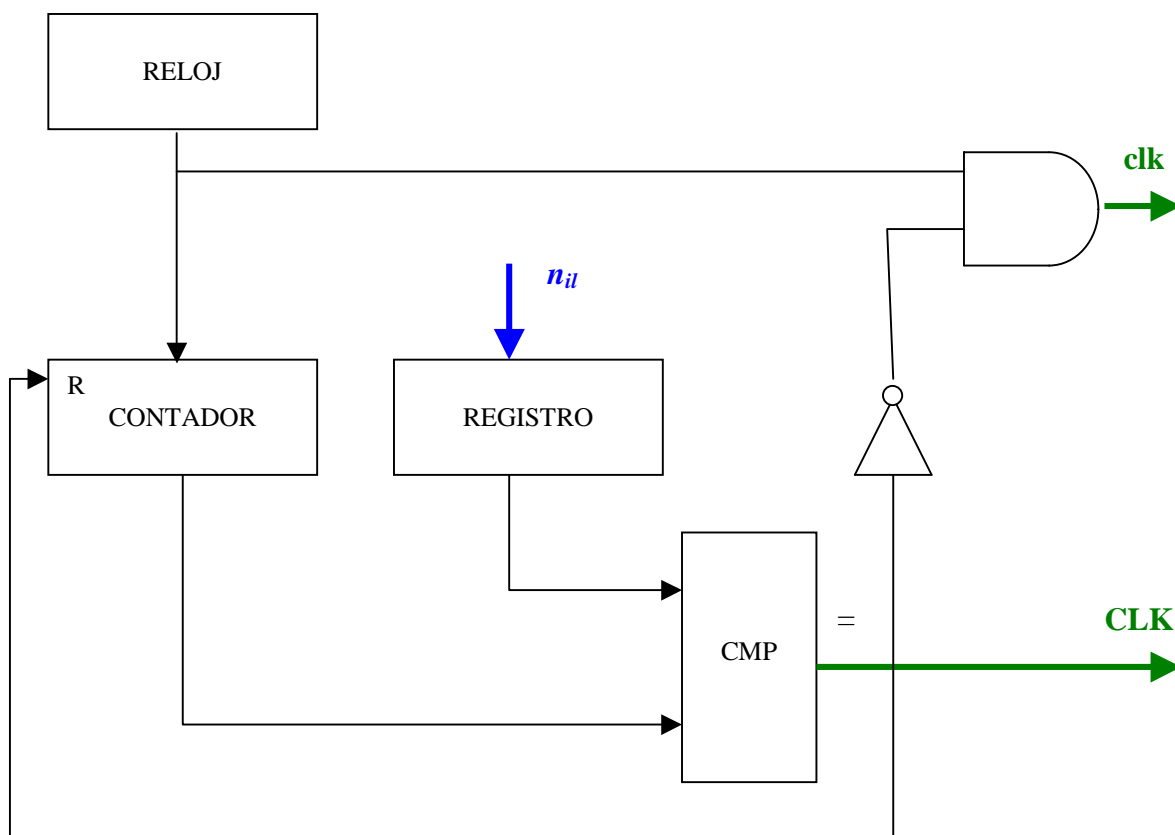


Figura 2.22. Descripción hardware de la LOGICA DE TEMPORIZACION

2.3.5.2. Lógica de carga

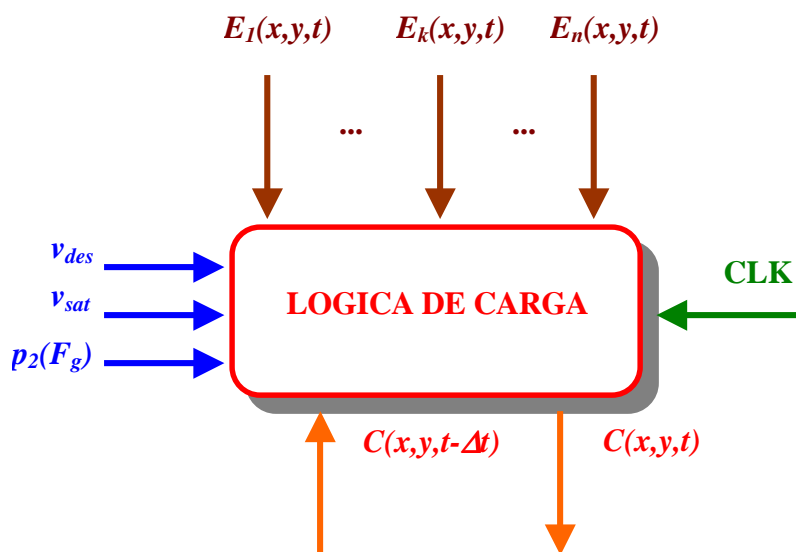


Figura 2.23. Entradas / salidas de la LOGICA DE CARGA

Descripción funcional:

La LOGICA DE CARGA recoge las operaciones de computación acumulativa a nivel global (escala de tiempos global t). Ofrece básicamente un nuevo valor de carga $C(x,y,t)$ a partir de la entrada desde la capa anterior en el mismo módulo (x,y) y del valor de carga en el instante anterior $C(x,y,t-\Delta t)$. La lectura de la entrada se efectúa a cada tick del temporizador CLK. Aparecen aquí los parámetros del modelo v_{des} , v_{sat} y los parámetros de la función de computación acumulativa global F_g que van a gobernar el modo de carga del módulo.

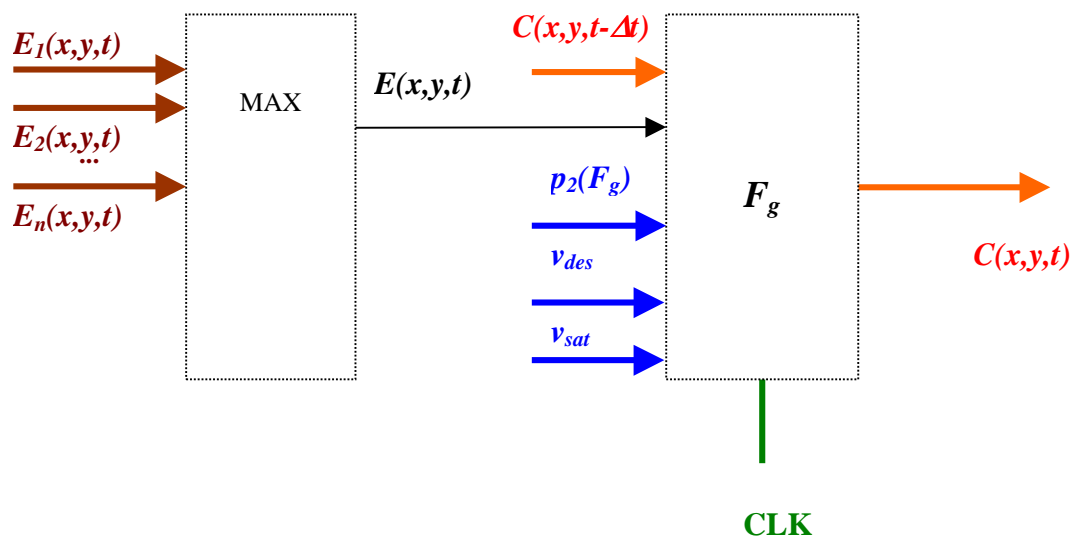


Figura 2.24. Descripción hardware de la LOGICA DE CARGA

Descripción algorítmica:

$$C(x,y,t) = F_g[C(x,y,t), \max(E_1(x,y,t), \dots, E_n(x,y,t))]$$

if $C(x,y,t) \leq v_{des}$

then $C(x,y,t) = v_{des}$

else if $C(x,y,t) \geq v_{sat}$ then $C(x,y,t) = v_{sat}$

2.3.5.3. Lógica de descarga

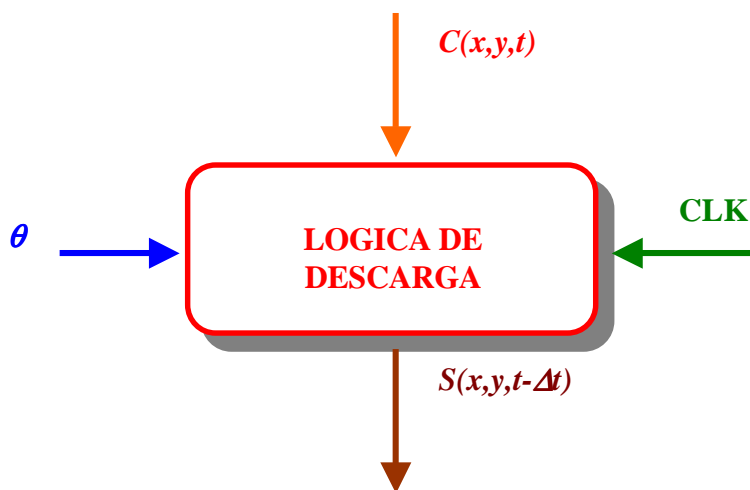


Figura 2.25. Entradas / salidas de la LOGICA DE DESCARGA

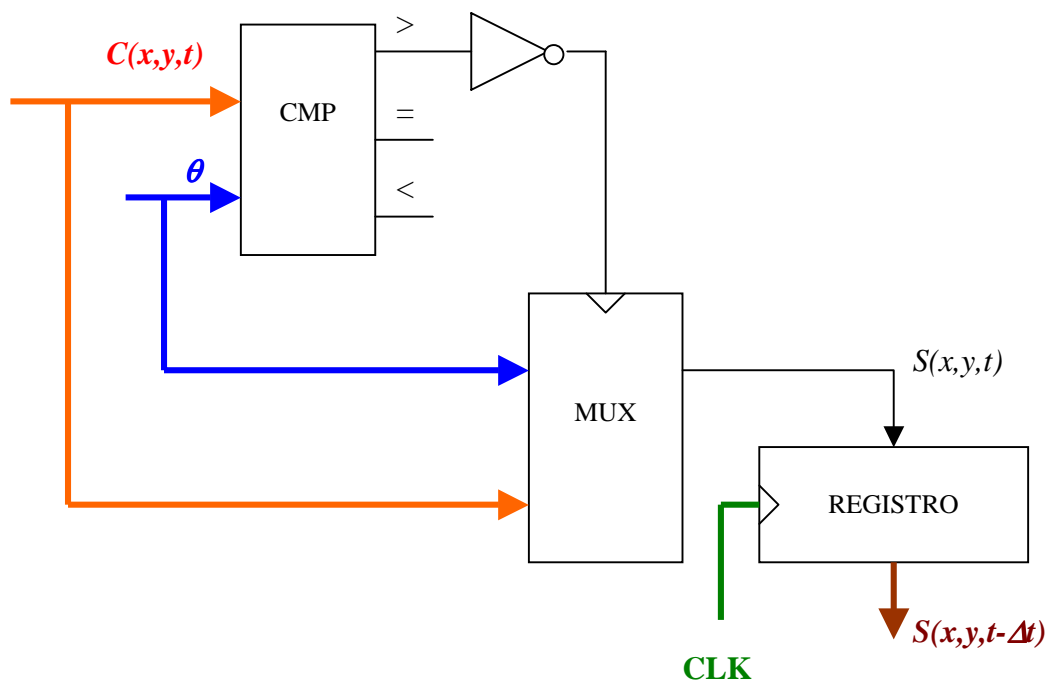


Figura 2.26. Descripción hardware de la LOGICA DE DESCARGA

Descripción funcional:

La LOGICA DE DESCARGA permite escribir hacia la siguiente capa el valor de salida del módulo (x,y) en su misma posición espacial. Este valor de salida $S(x,y,t-\Delta t)$ será el valor de carga del módulo $C(x,y,t)$ siempre que éste valga más que el umbral de disparo θ . En caso contrario, tenderemos como salida el propio umbral.

Descripción algorítmica:

```

if (C(x,y,t) >=  $\theta$ ) then S(x,y,t) = C(x,y,t)
else S(x,y,t) =  $\theta$ 

```

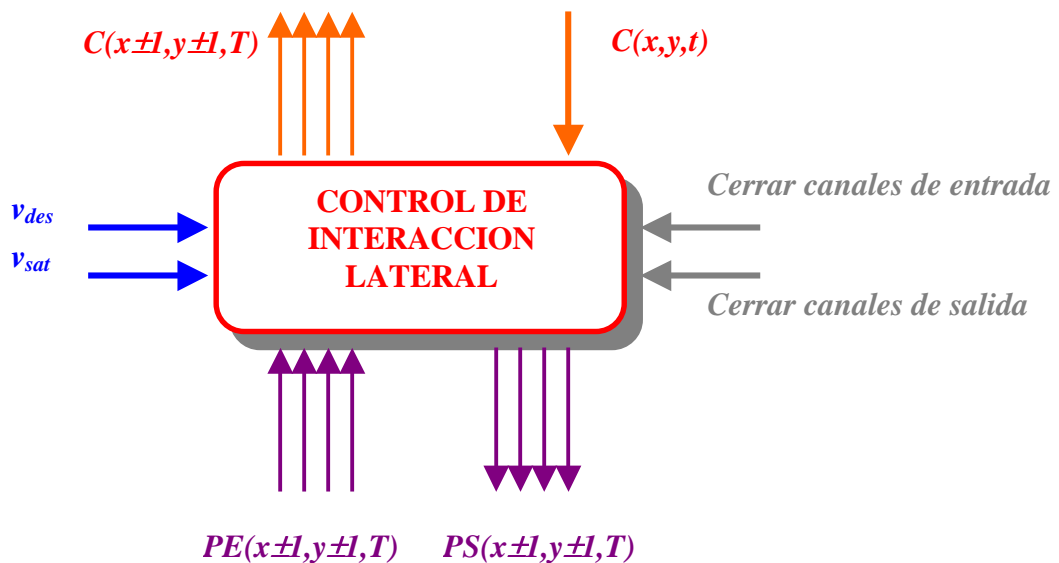
2.3.5.4. Control de interacción lateral

Figura 2.27. Entradas / salidas del CONTROL DE INTERACCION LATERAL

Descripción funcional:

El CONTROL DE INTERACCION LATERAL tiene como finalidad habilitar / deshabilitar tanto la salida lateral como hacia los cuatro vecinos del módulo (x,y) , a saber $(x-1,y)$, $(x+1,y)$, $(x,y-1)$ y $(x,y+1)$, como la entrada desde dichos vecinos. Las entradas laterales se denominan PE , mientras que las salidas son llamadas PS .

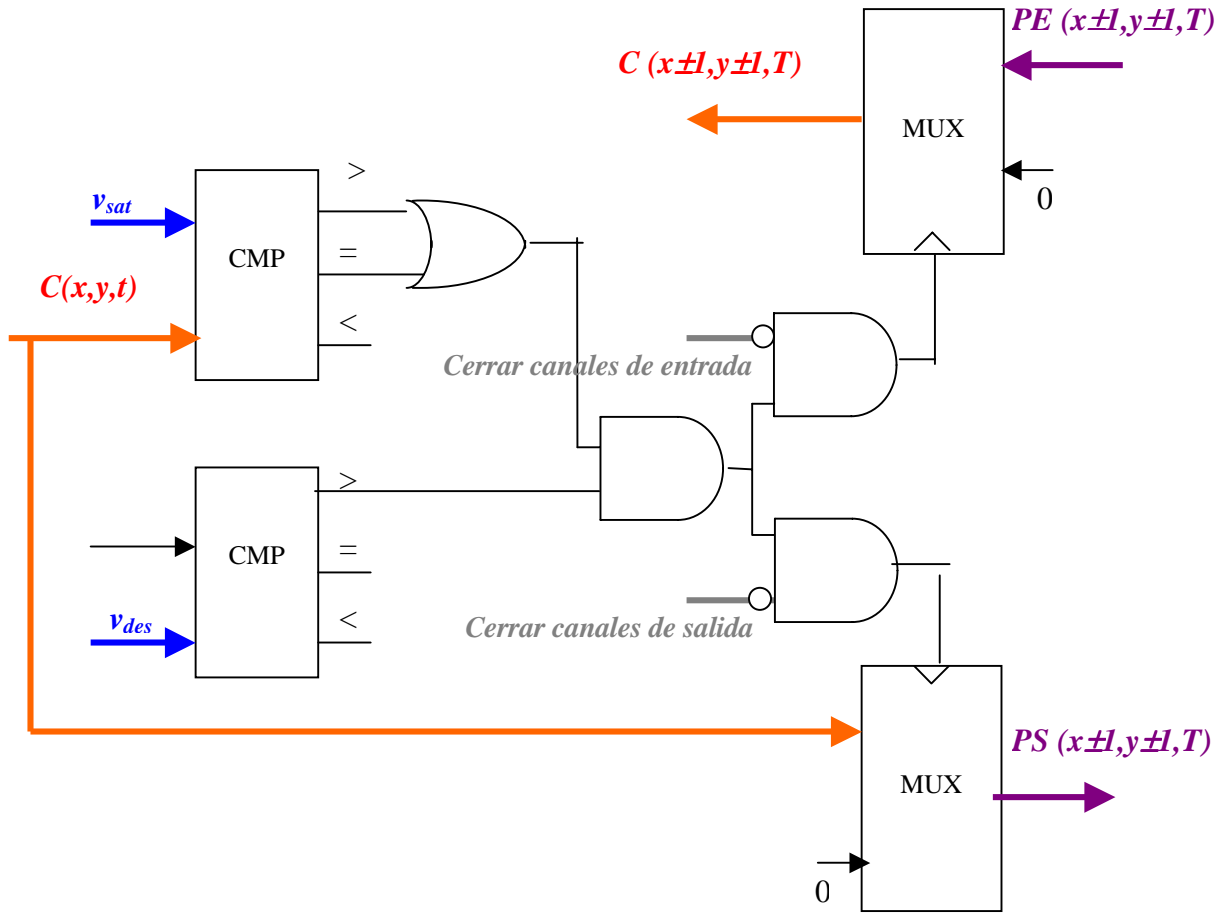


Figura 2.28. Descripción hardware del CONTROL DE INTERACCION LATERAL

La condición general para habilitar / deshabilitar la comunicación (interacción) lateral, es que el módulo tenga un valor de carga “aceptable”, es decir, que se cumpla $v_{des} < C(x,y,t) \leq v_{sat}$.

El modelo incorpora igualmente la posibilidad de deshabilitar (nunca habilitar) la interacción lateral desde la LOGICA DE CARGA LATERAL por medio de las señales *Cerrar canales de entrada* y *Cerrar canales de salida*.

Descripción algorítmica:

```

if (  $v_{des} < C(x,y,t) \leq v_{sat}$  ) then {
    if ( "Cerrar canales de entrada" == FALSE )
        then "Habilitar canales de entrada"
    else "Deshabilitar canales de entrada";
}
    
```



```

if ("Cerrar canales de salida" == FALSE)
  then "Habilitar canales de salida"
  else "Deshabilitar canales de salida";
}
else {
  Deshabilitar canales de entrada;
  Deshabilitar canales de salida;
}

```

2.3.5.5. Lógica de carga lateral

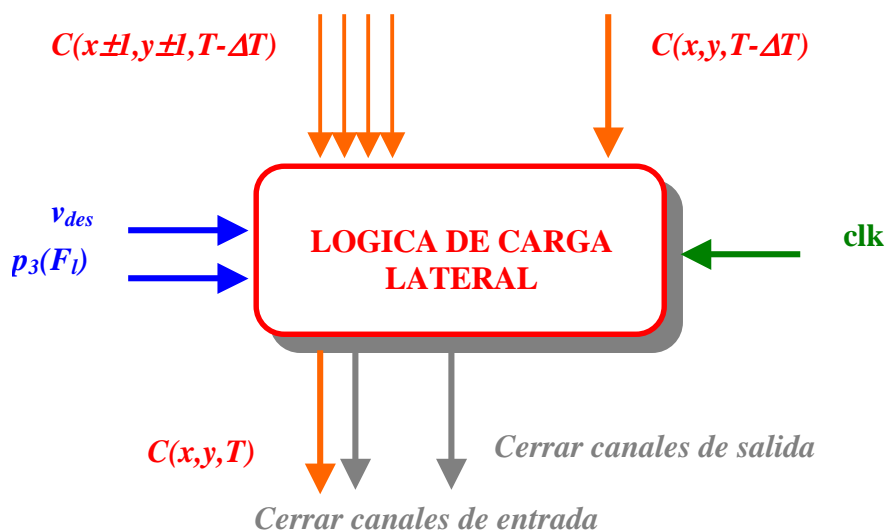


Figura 2.29. Entradas / salidas de la LOGICA DE CARGA LATERAL

Descripción funcional:

La LOGICA DE CARGA LATERAL recoge las operaciones de computación acumulativa a nivel local (escala de tiempos local T). Recalcula a cada tick del temporizador clk el valor de carga del módulo (x,y) , a saber $C(x,y,T)$ a partir del valor anterior de carga $C(x,y,T-\Delta T)$ y de la información de entrada proveniente de los cuatro vecinos más próximos (ya tratada

en el CONTROL DE INTERACCION LATERAL). Estas cargas se denominan $C(x-1,y,T-\Delta T)$, $C(x+1,y,T-\Delta T)$, $C(x,y-1,T-\Delta T)$ y $C(x,y+1,T-\Delta T)$, respectivamente.

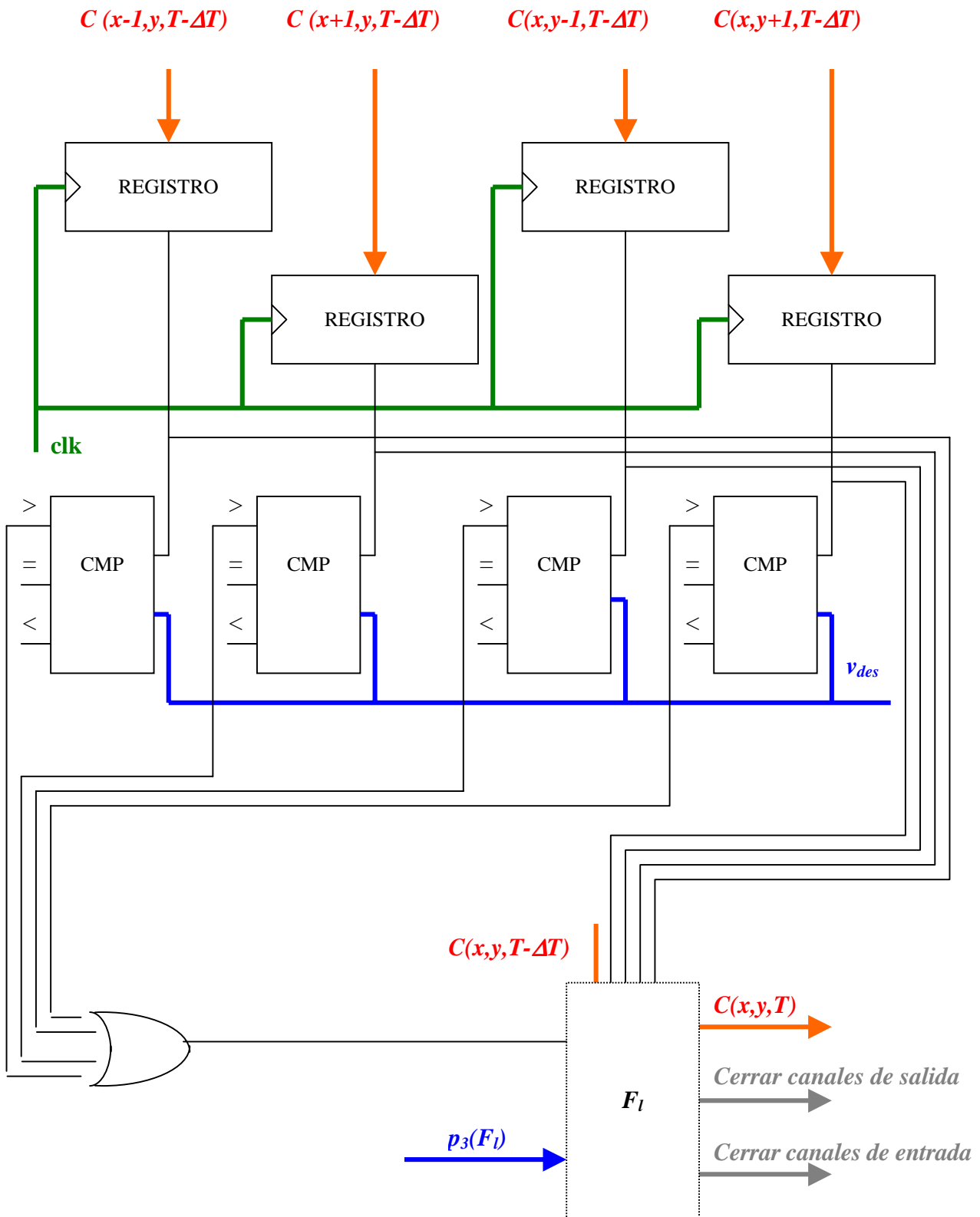


Figura 2.30. Descripción hardware de la LOGICA DE CARGA LATERAL

En general, se produce una comparación del valor de carga del módulo con los valores de los vecinos, y, a partir de dicha información se aplicará la función acumulativa a nivel local a partir de los parámetros $p_3(F_l)$. Esto se hará así solamente si el módulo está unido al menos a un módulo con cierta carga válida ($> v_{des}$).

Véase, por tanto, que la modificación de la carga se debe a una señal que se propaga linealmente en cuatro direcciones desde los módulos no descargados, mientras que la señal muere al llegar a los módulos descargados.

La lógica de cada uno de los módulos permite que la recarga de un módulo se pueda realizar una sola vez, tras lo cual se cierran los canales de entrada, o que la carga se vaya modificando de modo iterativo. De igual modo, cuando la información que llega desde los cuatro vecinos es no válida (todos los vecinos están descargados), se cierran los canales de salida.

Descripción algorítmica:

```

if ((C(x-1,y,T-ΔT)<>vdes) || (C(x+1,y,T-ΔT)<>vdes) ||
    (C(x,y-1,T-ΔT)<>vdes) || (C(x,y+1,T-ΔT)<>vdes)) {
    C(x,y,T) = F1(C(x,y,T-ΔT), C(x-1,y,T-ΔT), C(x+1,y,T-ΔT),
                  C(x,y-1,T-ΔT), C(x,y+1,T-ΔT));
    if ("Una sola recarga")
        "Cerrar canales de entrada" = TRUE;
}

if ((C(x-1,y,T-ΔT)==vdes) && (C(x+1,y,T-ΔT)==vdes) &&
    (C(x,y-1,T-ΔT)==vdes) && (C(x,y+1,T-ΔT)==vdes))
    "Cerrar canales de salida" = TRUE;

```

2.3.6. Tareas de control en el modelo

Todo modelo de computación debe incorporar las tareas de control necesarias que permitan decidir cuál es el modo local de funcionamiento de la capa neuronal. En este modelo se consideran los siguientes modos funcionales básicos:

- Inicialización con valores aleatorios ante una problemática global desconocida.
- Inicialización con valores conocidos ante una problemática global identificada.
- Aprendizaje supervisado, utilizando una función de error, modificando los parámetros del modelo.
- Proceso normal, sin modificación de parámetros.

CAPITULO 3. Obtención de las siluetas de los objetos deformables

3.1. Conocimiento del dominio del movimiento

Una vez visto en extenso el modelo de interacción lateral en computación acumulativa, y volviendo a la intención de este trabajo de aplicar dicho modelo genérico a la obtención de las siluetas de los objetos deformables en movimiento en una secuencia indefinida de imágenes, este capítulo está dedicado a exponer cómo se puede aplicar este modelo neuronal y biológicamente defendible a la tarea propuesta. Por ello, es imprescindible una aproximación al conocimiento del dominio del movimiento.

3.1.1. Problemas generales para la obtención de siluetas

En mayor o menor medida, existe un número significativo de problemas en todos los esquemas de análisis del movimiento entre los que se encuadra la búsqueda de las siluetas presentes en una escena. Y ello debido a la propia naturaleza de la proyección del movimiento. Existen ocho problemas de importancia que se describen a continuación y son: (1) el problema de la apertura, (2) los límites del movimiento, (3) las oclusiones, (4) el falso movimiento del fondo de la imagen, (5) el contexto espacio-temporal, (6) la velocidad del movimiento, (7) el color en el movimiento, y, (8) la discriminación de más de un objeto en movimiento.

3.1.1.1. El problema de la apertura

Una consecuencia de la no equivalencia entre el movimiento proyectado y el flujo óptico es que la información del movimiento es intrínseca a la estructura de la imagen, véase a la variación de los niveles de gris. El problema de la apertura se da cuando no hay suficiente variación en el nivel de gris en una región estudiada de la imagen que pueda llevar a la resolución unívoca del problema. Dicho de forma más clara, más de un candidato de movimiento sería igualmente válido en los datos

observados de la imagen. Específicamente, esto significa que solamente puede ser estimada la componente de velocidad en una dirección perpendicular a un gradiente de imagen significativo con algún grado de certeza.

Ocurre algo parecido en el sistema visual. En efecto, los detectores del sistema visual miran solo a una pequeña parte del campo visual. Surge un problema al asignar la velocidad verdadera de un objeto a partir de medidas locales, ya que el movimiento de un único segmento extendido de línea no determina el movimiento de un objeto que contenga ese segmento de línea [Fen79] [Hor81] [Mar81] [Ade82] [Hil84] [Wal76]. El movimiento paralelo a la línea es invisible. De este modo, una familia de posibles movimientos puede ser el resultado del movimiento detectado. El problema de la apertura se reduce a la asignación de la velocidad verdadera de todo el modelo a los elementos de una imagen en movimiento en aquellos puntos locales con posibles ambigüedades. Diremos, pues, que el movimiento del componente local o 1D es el movimiento aparente de líneas extendidas 1D, mientras que el movimiento de la forma o 2D es el movimiento real de los elementos.

La solución al problema de la apertura se resuelve si existen al menos dos medidas de movimiento de componentes locales que lleven a la estimación de la velocidad de la forma en un punto. En un movimiento simple como la traslación en un plano, el problema está ampliamente resuelto. En efecto, al ser la velocidad 2D la misma en todo el móvil, hay, en la gran mayoría de los casos, más de dos medidas de componentes locales para estimar la velocidad 2D. No es éste el caso, sin embargo, para el movimiento 3D y el movimiento rotatorio, en el que la velocidad real 2D varía de punto a punto. Por ello, la medida del movimiento 3D es ambigua [Eme81] [Tik77] y son necesarias unas restricciones adicionales para encontrar una solución única. Se entiende igualmente que el problema se agrava aún más, si nos dedicamos a objetos deformables, dada su problemática añadida.

Esto se puede conseguir por varias vías. Por un lado, se puede ir hacia un algoritmo de aprendizaje conexionista como herramienta para encontrar el mapeo entre los datos de la imagen (información del movimiento de los componentes) y las propiedades de interés del objeto (movimiento de la forma) [Ser93]. Otra aproximación (teoría de la regularización) explícitamente formula las restricciones que representen las propiedades ideales del mundo real (por ejemplo, la velocidad cambia suavemente), incorporándolas en una función de coste que ha de ser minimizada [Pog85].

3.1.1.2. Los límites del movimiento

Aparece una complicación adicional, también debida a la ambigüedad inherente de la proyección 3D en 2D. Donde existe una discontinuidad en la profundidad de la escena, por ejemplo, allí donde los objetos en movimiento se superponen, algunos puntos del espacio tridimensional con diferentes movimientos pueden proyectarse sobre un mismo punto 2D en el plano de la imagen. Esto desemboca en una discontinuidad en el campo del movimiento espacial. Ya que la región contiene realmente una serie de movimientos distintos, un esquema que asume la muy común suposición que toda región conlleva un único movimiento fallará al modelar estas discontinuidades intra-regionales y estimará un movimiento unitario para esta región. Este movimiento único se hará corresponder bien al movimiento dominante o el movimiento medio. Más aún, si las discontinuidades en el movimiento fueran solamente temporales, esto afectaría de gran modo a los algoritmos de obtención de siluetas y su posterior seguimiento.

3.1.1.3. Las oclusiones

Una consecuencia adicional de la ambigüedad de la proyección es que un objeto en movimiento llevará a que un fondo previamente cubierto se descubra y viceversa. Este efecto se conoce como oclusión. En estas regiones no existe correspondencia entre los datos de diferentes tramas y, por ello, el campo movimiento no estará definido. Ocurren situaciones similares cuando hay un cambio de escena o cuando entran y salen nuevos objetos de la escena. Ello hace que el análisis del movimiento sea una tarea compleja, ya que el movimiento debe obtenerse por otros medios en estas áreas.

3.1.1.4. El falso movimiento del fondo de la imagen

Queda aún prestar atención a la problemática que conlleva la captación de movimiento del propio fondo de la imagen. En efecto, donde hay detección motivada por el movimiento de un objeto, hay detección de movimiento debido al fondo de la imagen. El movimiento del objeto se torna en la invasión de una zona anteriormente perteneciente al fondo de la imagen, mientras que el falso movimiento del fondo de la imagen viene determinado por la zona que el objeto acaba de liberar. Puede verse mediante el ejemplo de la figura 3.1.

En dicho ejemplo podemos apreciar el movimiento de un autobús a través de dos imágenes de una misma secuencia, así como la imagen diferencia de ambas. Etiquetado bajo (A) tenemos el movimiento detectado debido al objeto, mientras que con (B) se ha etiquetado el falso movimiento debido al fondo de la imagen.

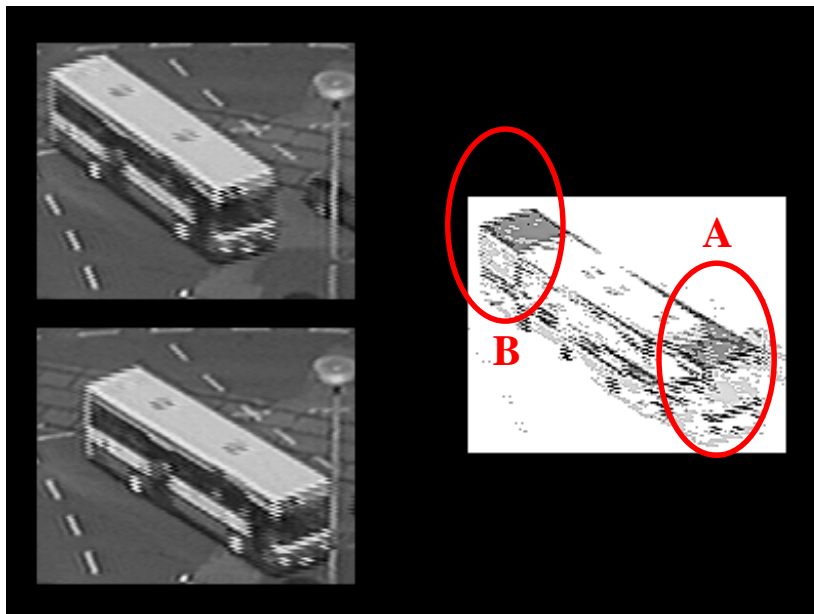


Figura 3.1. Detección de movimiento del fondo de la imagen

3.1.1.5. El contexto espacio-temporal

Está claro que los humanos se apoyan tanto en el contexto espacio-temporal como en un conocimiento a priori cuando interpretan la información del movimiento visual. Un conocimiento a priori es difícil de plasmar de manera significativa en un esquema de estimación general. Sin embargo, al considerar el contexto espacio-temporal, o sea, tratando de establecer qué regiones corresponden a la superficie del mismo objeto en la escena 3D, y cómo estas regiones se mueven en el tiempo, puede obtenerse alguna pista importante hacia la interpretación de la escena y, por tanto, para la obtención de las siluetas y su seguimiento.

3.1.1.6. La velocidad del movimiento

Cuando un objeto se mueve a lo largo de una imagen, tenemos dos casos posibles:

1. que el elemento se mueva con la suficiente velocidad, o que el muestreo sea lo suficientemente lento, como para que ninguno de los puntos del objeto ocupe lugar donde otro de los puntos del objeto estaba pisando en el instante anterior

2. que dicha intersección no sea nula

El caso en el que la intersección de un elemento de una imagen a otra es vacía no representa problema, ya que queda recogida en su totalidad la silueta del elemento (tanto en el instante anterior $t-\Delta t$ como en el instante actual t). No ocurre lo mismo cuando dicha intersección no es vacía.

Gráficamente podemos ver el siguiente ejemplo¹:

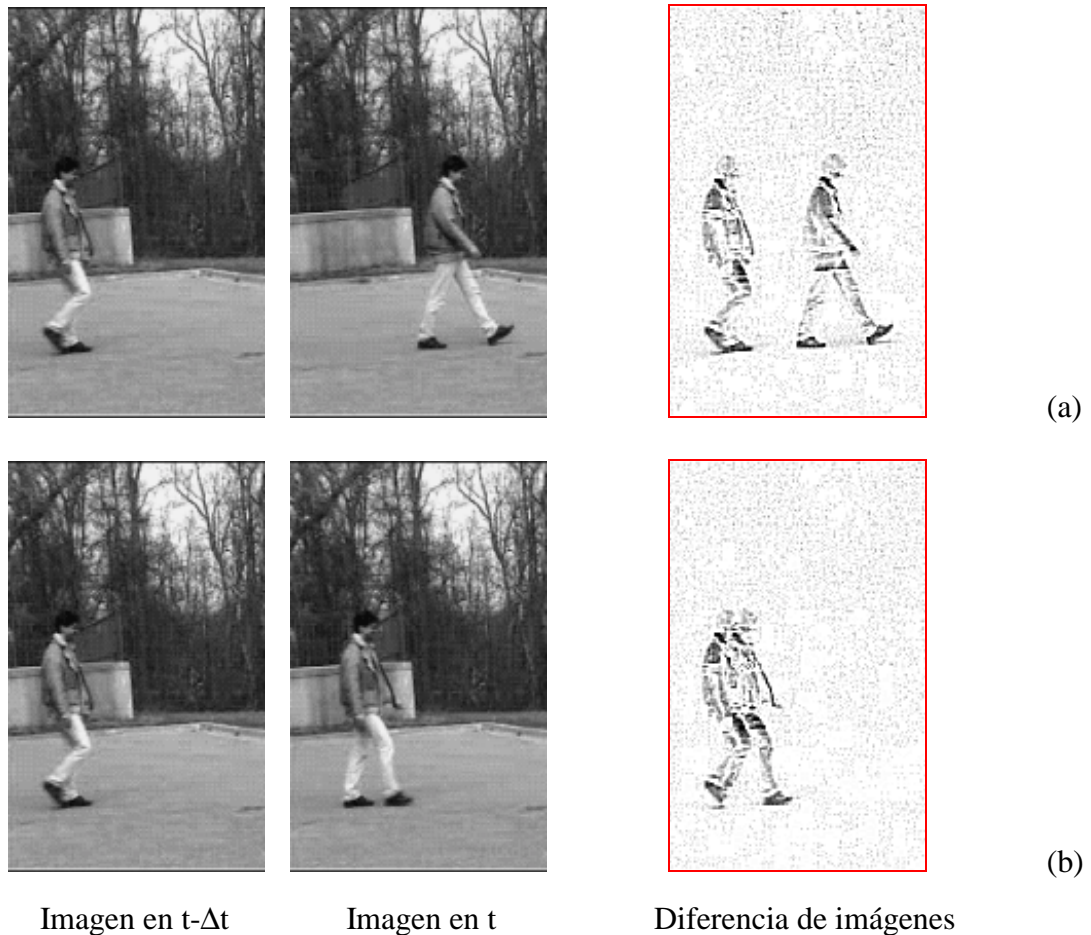


Figura 3.2. Diferencia de imágenes (a) sin y (b) con intersección

3.1.1.7. El color en la detección de movimiento

Comenzamos presentando en este apartado un método sencillo ya clásico para la detección de movimiento, que permite saber en qué puntos de una imagen en una secuencia de imágenes ha variado el color (caso más general) o nivel de gris (caso más específico). El algoritmo empleado en numerosos casos es:

¹ Imágenes cortesía de University of Maryland Institute for Advanced Computer Studies.

$$MOV(x, y, t) = \begin{cases} 1, & \text{si } NG(x, y, t) \neq NG(x, y, t - \Delta t) \\ 0, & \text{en caso contrario} \end{cases}$$

Se entiende que hay detección de movimiento $MOV(x,y,t)$ en un punto de la imagen (x,y) en el instante de tiempo t , si el nivel de gris NG en un punto de la imagen (x,y) ha variado entre los instantes $t-\Delta t$ y t .

El mecanismo descrito hasta el momento es robusto cuando los elementos que se pasean por la imagen tienen un mismo nivel de gris, lo que permite binarizar de un modo trivial la imagen entre puntos que cambian y no cambian el nivel de gris. El hecho constatado es que los cambios de uno de los niveles de gris va estorbando en la obtención de la silueta del otro nivel de gris.

3.1.1.8. La discriminación de más de un objeto en movimiento

Otro problema importante a la hora de acercarnos al movimiento en secuencias de imágenes reales es el de la alta complejidad que supone la obtención de todos los elementos (objetos) móviles en la escena. Debe añadirse a esta problemática, ya sería de por sí, la de la discriminación o clasificación de los objetos obtenidos. Este es otro problema no trivial con el que todo buscador de siluetas debe enfrentarse.

3.1.2. Estudio de otras aproximaciones

Siempre es instructivo considerar cómo han sido afrontados en el pasado los problemas descritos anteriormente. En términos generales, serían tres las aproximaciones distintas a la estimación del movimiento y con posibilidad de obtener las siluetas que podemos identificar. Estas tres aproximaciones se describen brevemente a continuación.

3.1.2.1. Métodos basados en la correlación de características

La primera aproximación, la estimación basada en la correlación de características, se basa en la extracción de un conjunto limitado de primitivas de las imágenes y su correlación en el tiempo utilizando algún tipo de estrategia de búsqueda [Hua94b]. Estas primitivas corresponden a características distintivas en la escena. Dadas las posiciones relativas de las características que se corresponden a lo largo de las tramas y a partir de un amplio conocimiento de la geometría proyectiva,

la posición relativa tridimensional y el movimiento pueden ser estimados bajo ciertas condiciones [Tru98]. La complejidad del problema de la correspondencia entre características puede decrementarse utilizando técnicas de seguimiento bajo la suposición que el movimiento de cada una de las características es pequeño entre tramas consecutivas [Der90]. Esto permite ofrecer predicciones acerca de las posiciones futuras de las características basadas en la historia de sus movimientos [Mey94].

El tipo de características utilizadas puede ser, por ejemplo, puntos [Fau87], líneas y bordes [Liu88] [Wen92] o esquinas [Liu89]. Ya que la complejidad de la correlación crece exponencialmente con el número de características, estos métodos utilizan pocas características, aunque altamente discriminatorias.

Los métodos basados en la correlación de características se usan típicamente en aplicaciones donde se requiere que las estimaciones resultantes se correspondan con el movimiento 3D de los objetos en la escena, tal como es el caso en la navegación de los robots autónomos.

3.1.2.2. Métodos basados en el gradiente

Los métodos de estimación basados en el gradiente están enfocados en estimar el movimiento aparente de patrones de intensidad en la secuencia, véase el flujo óptico. Para ello se apoyan en dos suposiciones. La primera suposición se basa en la observación que, aunque las posiciones de los objetos pueden cambiar en la escena, su brillo permanece constante, siempre que la iluminación global permanezca constante en el tiempo. Esta suposición se denomina algunas veces restricción de constancia de los datos [Bar94]. La segunda suposición es la de la coherencia espacial, basada en la observación que las superficies tienen una extensión espacial y, por tanto, los píxeles en una vecindad de imagen pequeña son susceptibles de corresponderse con puntos de la misma superficie tridimensional. Ya que el movimiento proyectado de los puntos de una superficie 3D variará gradualmente, una consecuencia de esta suposición es imponer una restricción de suavizado en el campo del flujo óptico.

La restricción de constancia de los datos asume que cualquier cambio observado en la intensidad de la imagen a lo largo del tiempo es causado por el movimiento relativo entre la cámara y los objetos en la escena. Por ello, ya que el movimiento es una cantidad diferencial, una aproximación a su estimación es la de medir el grado de cambio espacio-temporal en la función de intensidad de la imagen. La restricción de constancia de los datos puede formularse

como una ecuación diferencial en términos de las derivadas parciales de la función de intensidad de la imagen, conocida como la ecuación del flujo óptico. Esta puede ser estimada utilizando técnicas de aproximación numéricas basadas en diferencias finitas [Hor81]. Ya que el flujo óptico es bidimensional en cada punto de la imagen, contiene dos incógnitas. Por ello, para resolver la ecuación, se usa la suposición de coherencia espacial, bien localmente, obligando a que el flujo óptico sea constante sobre alguna pequeña región arbitraria, o bien globalmente, minimizando la variación total pixel a pixel de los vectores de movimiento estimados [Hor81]. La estimación basada en el gradiente podría parecer computacionalmente eficiente, pero debido al uso de ventanas de análisis pequeñas sufre necesariamente del problema de la apertura y de desviaciones naturales de la suposición de constancia de los datos, causadas, por ejemplo, por la presencia de ruido debido a la observación o por cambios en las condiciones globales de iluminación.

3.1.2.3. Métodos basados en regiones

Aparentemente, la aproximación más simple a la estimación del movimiento, y por ende a la obtención de las siluetas de los objetos en movimiento, es la de relacionar regiones entre las diferentes tramas por medio de la búsqueda de las mismas [Jai81b]. Esto representa, de algún modo, la filosofía opuesta a las técnicas de estimación basadas en el gradiente, vistas con anterioridad. En efecto, se usan ventanas de análisis mayores para prevenir los problemas del ruido y los efectos de la apertura. La idea básica es que la secuencia de imágenes está formada por un conjunto de regiones en las que cada una comporta un movimiento singular entre tramas. Cada trama se divide en bloques o triángulos y el movimiento de cada región se encuentra buscando la misma región en la siguiente trama, minimizando alguna medida de error, como puede ser la suma de las diferencias cuadradas entre los niveles de gris.

Este tipo de estimación forma la base de muchas técnicas de compresión de vídeo, donde se necesita que el campo movimiento pueda ser eficientemente codificado y transmitido. Debido a su naturaleza basada en bloques, cae de forma natural en un esquema de codificación basado en bloques como el MPEG [LeG91]. Igualmente, debido a sus grandes regiones de análisis, es robusto en presencia de ruido. Sin embargo, las desventajas de este tipo de aproximaciones residen en el alto coste computacional del procedimiento de búsqueda y su falta de resolución. En efecto, al tener un tamaño fijo de región, la aproximación no puede adaptarse a la multitud de tipos de datos tratados. No es siempre posible describir adecuadamente una escena general como un conjunto de bloques

cuadrados en movimiento de igual tamaño. El precio pagado para la robustez, conseguida por la adopción de regiones de análisis más grandes, provoca que los datos en cada región puedan violar la suposición de un sólo movimiento por región [Bla96].

3.1.2.4. Sustracción del fondo y detección de cambios

Los sistemas de vigilancia comerciales utilizan una sencilla técnica de procesamiento de imágenes de sustracción del fondo para recuperar los objetos en movimiento en una escena. Esta técnica necesita que la cámara esté fija (véase, estacionaria, con el zoom y la apertura fijos). Además, la sustracción del fondo conlleva muchas limitaciones. Incluso con una cámara fija, los métodos de sustracción son sensibles a los cambios de iluminación, al pobre contraste, a las reflexiones, a las oclusiones y al ruido en las imágenes.

3.1.2.4.1. Detección de cambios

La detección de cambios es uno de los métodos para la detección de objetos en movimiento en una secuencia de imágenes tomadas con una cámara fija. Dadas dos tramas de imagen sucesivas en escala de grises $I_{k+1}(x, y)$ e $I_k(x, y)$, se calcula una imagen diferencia restando las intensidades de las imágenes en cada pixel. Así,

$$D_k(x, y) = |I_{k+1}(x, y) - I_k(x, y)|$$

La imagen diferencia D_k suele umbralizarse para obtener una imagen binaria, resaltando aquellos pixeles donde ha ocurrido un cambio significativo en su intensidad.

Bajo la suposición de una cámara fija, con apertura fija y condiciones de iluminación constantes, los pixeles resaltados corresponden a partes de un objeto en movimiento. Si el objeto en movimiento tiene una textura uniforme, los pixeles resaltados se corresponderán con los bordes de cabeza y de cola del objeto. Si el objeto tiene una textura heterogénea, algunos pixeles internos también serán resaltados.

Los pixeles resaltados pueden agruparse por medio de técnicas de clasificación para obtener un conjunto de regiones. El proceso de una escena con uno o más objetos en movimiento separados en la imagen, resultará en regiones que se corresponderán con cada uno de los objetos en movimiento. Una aproximación más robusta, diferenciando características de la imagen

(como los bordes) es descrita por Jain et al. [Jai79]. Estas técnicas se han empleado en una gran variedad de aplicaciones [Hog83] [LiQ92] [Roh93].

3.1.2.4.2. Sustracción del fondo

Otra técnica poderosa, la sustracción del fondo, se basa en la disponibilidad de una imagen de referencia del fondo $I_{ref}(x, y)$. Esta imagen puede obtenerse adquiriendo la imagen desde una cámara fija cuando no hay objetos en movimiento en la misma. De un modo alternativo, una imagen de fondo puede obtenerse de una secuencia de imágenes $I_k(x, y)$ por obtención de la mediana a lo largo del tiempo.

$$I_{ref}(x, y) = \text{mediana} (I_0(x, y), I_1(x, y), \dots, I_n(x, y))$$

La mediana puede reemplazarse por una media robusta apropiada, actualizándose periódicamente para tomar en cuenta las condiciones cambiantes en la iluminación.

La sustracción de las imágenes (y la umbralización) se realiza del mismo modo que en la técnica de detección de cambios y los pixeles resaltados resultantes se corresponden con los objetos de interés. Suponiendo una cámara estacionaria con condiciones de iluminación fijas y un buen contraste, el método puede usarse para segmentar los objetos en movimiento en una escena. Los pixeles resaltados conectados entre sí generalmente se corresponden con objetos separados y las pequeñas regiones pueden ser ignoradas. Sin embargo, cuando varios objetos en movimiento se solapan en la imagen o están demasiado próximos los unos de los otros, solamente se obtiene una región fusionada.

Esta técnica se ha usado como un primer paso en muchas aplicaciones de visión [Niy94] [Mur93]. Ambas técnicas de sustracción de imágenes son sensibles a las sombras, a los cambios en la iluminación, a las vibraciones de la cámara, al poco contraste y a las oclusiones.

Una extensión a este método es trabajar con una cámara dirijible descrita por Rowe y Blake [Row95]. La imagen de la cámara es retroproyectada en un plano de imagen de una cámara virtual que permanece fija. Se genera una imagen de fondo para la imagen de la cámara virtual efectuando un rastreo de la cámara sobre la escena. Se requiere un modelo estadístico en cada pixel para jugar con los errores en el proceso de proyección. El método es computacionalmente costoso, necesiéndose de varias horas para construir un modelo del fondo. Una vez que se ha extraído el fondo, el seguimiento del contorno puede realizarse en tiempo real.

3.1.3. Aplicación del modelo de interacción lateral a partir del conocimiento del dominio del movimiento

Se ofrece a continuación una explicación pormenorizada de qué acciones se han tomado a la hora de afrontar el conocimiento del dominio del movimiento desde la perspectiva del modelo de interacción lateral en computación acumulativa, así como las razones que han motivado dichas decisiones.

3.1.3.1. Línea general adoptada

La línea general adoptada para la aplicación del modelo de interacción lateral a la obtención de las siluetas de los objetos deformables móviles, teniendo en cuenta todo lo anteriormente expuesto en cuanto al conocimiento del dominio del movimiento, está próxima al método de detección de cambios. Es éste en efecto el método general que más se aproxima a las necesidades impuestas desde el principio, a saber, fundamentalmente:

1. una agrupación en regiones (siluetas) de una serie de píxeles resaltados (a partir del movimiento)
2. una discriminación de los distintos objetos presentes en la escena a partir de las regiones (siluetas) diferentes obtenidas

Aunque la técnica de sustracción del fondo es un método que se podía haber escogido como la línea general a seguir, no se ha valorado positivamente el hecho de tener que limitarse a la obtención de una imagen de fondo sin objetos en movimiento. Véase, a modo de explicación, la inutilidad de esta técnica en entornos reales de escenas de exteriores (paisajes) en las que el viento no deja de asombrarnos con su capacidad de mover alguna cosa no deseada (ramas de los árboles, por ejemplo). En otros entornos muy comunes, como los de la imaginería médica, tampoco es posible tomar imágenes absolutamente quietas.

Las demás aproximaciones o métodos expuestos han sido descartados por diversos motivos, aunque en todos ellos cabría hablar de falta de orientación biológica, es decir, próxima a nuestras tesis conexionistas.

Los métodos basados en la correlación de características no se han tomado en cuenta por ser excesivamente dependientes del tipo de aplicación a los que van dirigidos. El deseo de poder tocar la mayor cantidad posible de aplicaciones relacionadas con la posibilidad de obtener las

siluetas de los objetos deformables no permite acotar suficientemente las posibles características a ser correlacionadas. No se ha querido, por tanto, acotar o restringir el campo de acción del modelo de interacción lateral en computación acumulativa a su aplicación a aplicaciones bien definidas de antemano.

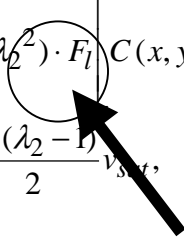
Los métodos basados en el gradiente, a su vez, sufren de excesivos problemas como para ser tenidos en cuenta en los fines perseguidos en este trabajo. El mayor inconveniente de estos métodos es que están excesivamente dirigidos a la obtención de una sola forma en la escena, y que ofrecen unos paupérrimos resultados en su uso con diversos objetos presentes. Además, el coste computacional es enorme.

Los métodos basados en regiones tampoco cumplen con los requisitos de generalidad buscados en este trabajo. Son métodos excesivamente orientados a la búsqueda en regiones muy concretas de las imágenes. Uno de los objetivos perseguidos es el de trabajar con todos los píxeles de las imágenes, sin restricción alguna en cuanto al tamaño de los objetos o su localización.

3.1.3.2. Solución al problema de la apertura

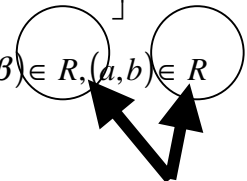
La aplicación del modelo de interacción lateral en computación acumulativa puede perfectamente afrontar el problema de la apertura efectuando un suavizado en la velocidad verdadera 2D en cada punto de la silueta a obtener de un objeto. Y ello, precisamente a través de los mecanismos de interacción lateral necesarios para el intercambio de dicha velocidad verdadera entre los puntos de imagen próximos. De ahí que no intentará la interacción lateral hallar la velocidad en cada uno de los puntos de la imagen, sino dar un valor de velocidad único a cada uno de los puntos de imagen correspondientes a un mismo objeto. Se puede pensar, en principio, en usar funciones tales como la media de la velocidad 1D de cada uno de los puntos que conforman el objeto en movimiento.

Puede verse con cierta facilidad que esto es posible hallando las funciones más adecuadas para el efecto del elemento vecino sobre el elemento o punto de imagen central (x,y) . Por lo tanto, el problema de la apertura se resuelve utilizando una función adecuada en el cálculo de la carga en la escala de tiempos local T .

$$C(x, y, T) = (1 - \lambda_2^2) \cdot F_l \left[C(x, y, T - \Delta T), \sum_{\alpha=x-1}^{x+1} \sum_{\beta=y-1}^{y+1} C(\alpha, \beta, T) \right] + \frac{\lambda_2(\lambda_2 + 1)}{2} v_{des} + \frac{\lambda_2(\lambda_2 - 1)}{2} v_{sat}, \quad \forall (\alpha, \beta) \neq (x, y), (\alpha, \beta) \in R, (a, b) \in R$$


3.1.3.3. Solución a los límites del movimiento

En el caso que nos ocupa, es éste un problema a tener muy en cuenta al intentar segmentar una imagen en todos los objetos que se mueven. Aquí también es imperativo tener en cuenta el alcance de los mecanismos de interacción lateral. Dicho de otro modo, es fundamental acotar perfectamente el campo receptivo R en la fórmula de interacción lateral en la misma escala de tiempo local T .

$$C(x, y, T) = (1 - \lambda_2^2) \cdot F_l \left[C(x, y, T - \Delta T), \sum_{\alpha=x-1}^{x+1} \sum_{\beta=y-1}^{y+1} C(\alpha, \beta, T) \right] + \frac{\lambda_2(\lambda_2 + 1)}{2} v_{des} + \frac{\lambda_2(\lambda_2 - 1)}{2} v_{sat}, \quad \forall (\alpha, \beta) \neq (x, y), (\alpha, \beta) \in R, (a, b) \in R$$


Nótese que la perfecta acotación del campo receptivo es tarea fundamental del aprendizaje en el modelo de interacción lateral en computación acumulativa. Por último debe recalcar que el campo receptivo denota perfectamente el alcance de la interacción lateral, y, por tanto, los límites del movimiento.

3.1.3.4. Solución a las oclusiones y movimiento debido al fondo

En primer lugar, cabe recalcar que el problema de las oclusiones está ampliamente superado por el modelo de interacción lateral en computación acumulativa, al ser éste un modelo orientado al pixel (punto, elemento) y no a la región (zona, objeto).

Por otra parte, para atajar el problema del movimiento debido al fondo de la imagen, se hace necesario que en el mecanismo de interacción lateral se tenga en cuenta la influencia del entorno de las zonas de detección de movimiento, de modo que el movimiento del fondo de la imagen, se diluya en el mismo. Una forma sencilla de acometer esta tarea, está en ajustar los umbrales por debajo de los cuales no se permite, bien la computación acumulativa, bien la

interacción lateral. Estamos, por lo tanto, hablando de ajustar, según convenga, el valor de disparo mínimo η , el valor de descarga mínima θ , el valor de descarga v_{des} y el valor de saturación v_{sat} .

En la escala de tiempos global t , tenemos

$$S(x, y, t) = \eta C(x, y, t)$$

$$C(x, y, t) = (1 - \lambda_1^2) \cdot F_g [C(x, y, t - \Delta T), \max(E_1(x, y, t), \dots, E_n(x, y, t))] +$$

$$+ \frac{\lambda_1(\lambda_1 + 1)}{2} v_{des} + \frac{\lambda_1(\lambda_1 - 1)}{2} v_{sat}$$

mientras que en la escala de tiempos local T , sería

$$C(x, y, T) = (1 - \lambda_2^2) \cdot F_l \left[C(x, y, T - \Delta T), \sum_{\alpha=x-1}^{x+1} \sum_{\beta=y-1}^{y+1} C(\alpha, \beta, T) \right] +$$

$$+ \frac{\lambda_2(\lambda_2 + 1)}{2} v_{des} + \frac{\lambda_2(\lambda_2 - 1)}{2} v_{sat}, \quad \forall (\alpha, \beta) \neq (x, y), (\alpha, \beta) \in R, (a, b) \in R$$

De este modo se consigue acotar perfectamente la carga válida en cada pixel que ofrezca información válida y no válida de silueta.

3.1.3.5. Solución al contexto espacio-temporal

El contexto temporal viene ya determinado en nuestro modelo de interacción lateral en computación acumulativa de un modo muy significativo por las dos temporizaciones, a saber la escala de tiempos global t y la escala de tiempos local T . En un sentido más amplio, podemos, igualmente, introducir todo el conocimiento el contexto espacio-temporal a través de la variable de estado de cada pixel de la imagen, que no es más que la carga en cada instante de tiempo, así como de las memorias FIFO, características del modelo.

$$C(x, y, t) = (1 - \lambda_1^2) \cdot F_g [C(x, y, t - \Delta T), \max(E_1(x, y, t), \dots, E_n(x, y, t))] +$$

$$+ \frac{\lambda_1(\lambda_1 + 1)}{2} v_{des} + \frac{\lambda_1(\lambda_1 - 1)}{2} v_{sat}$$

3.1.3.6. Solución a la velocidad del movimiento

Con el fin de considerar el caso de la intersección, problema expuesto con anterioridad, se ha incorporado la posibilidad de trabajar con valores intermedios de carga ($v_{des} < C(x,y,t) < v_{sat}$). Dichos valores intermedios de carga no máxima pertenecen a aquellas las partes de un objeto en movimiento que no han variado en su nivel de gris, pero que sabemos que pertenecen al objeto. La figura 3.3. muestra este hecho.

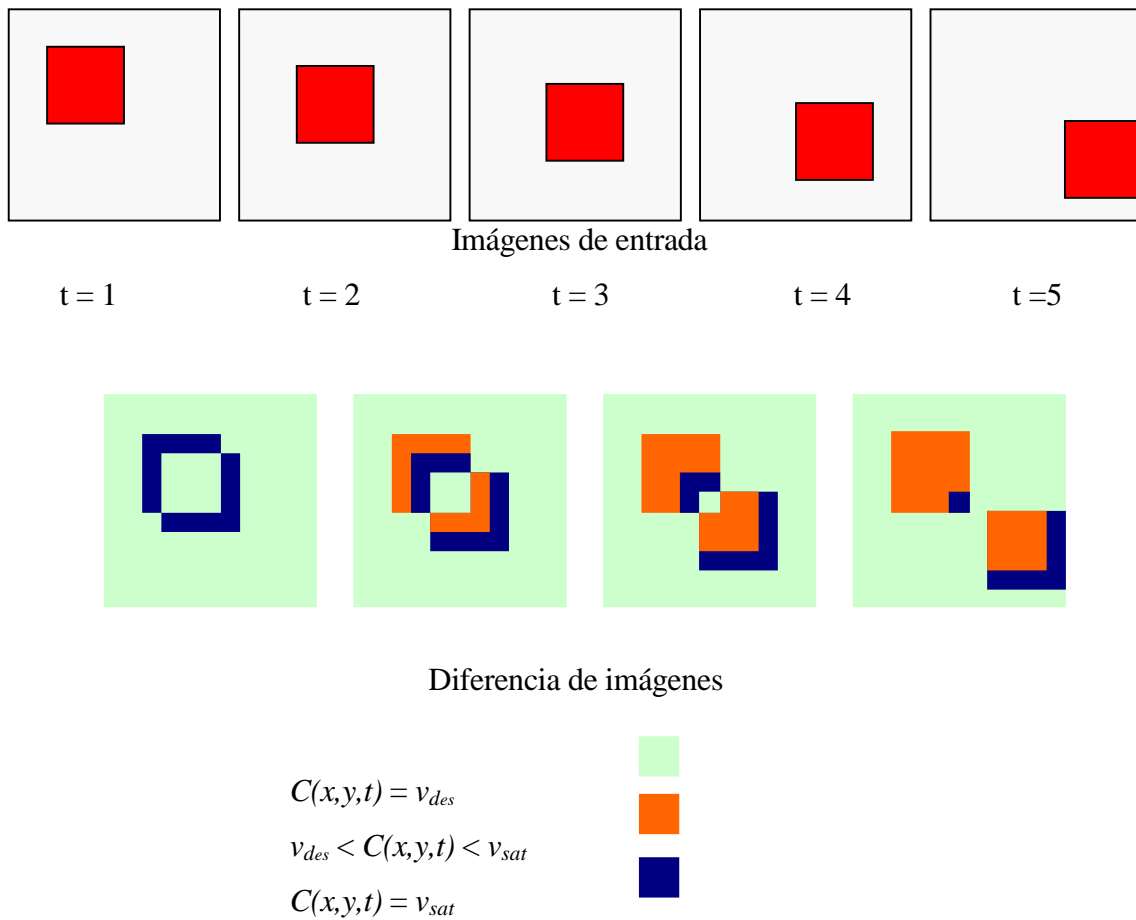


Figura 3.3. Obtención de los valores de carga

Utilizando una función de computación global F_g apropiada, podemos conseguir que la forma (o parte de la forma) vaya apareciendo mientras el elemento móvil va cambiando de posición. Por lo tanto, la solución al problema de la velocidad del movimiento pasa por incorporar un mecanismo de activación / desactivación que permita centrar la atención hacia donde el elemento se está moviendo, olvidando de algún modo de donde proviene el elemento. Y ello, mediante la búsqueda de una función de computación global F_g conveniente.

$$C(x, y, t) = (1 - \lambda_1^2) \cdot F_g [C(x, y, t - \Delta T), \max(E_1(x, y, t), \dots, E_n(x, y, t))] + \frac{\lambda_1(\lambda_1 + 1)}{2} v_{des} + \frac{\lambda_1(\lambda_1 - 1)}{2} v_{sat}$$

Tras un cierto intervalo de tiempo, podríamos encontrarnos con dos situaciones en una misma imagen, como puede apreciarse en la figura 3.4.

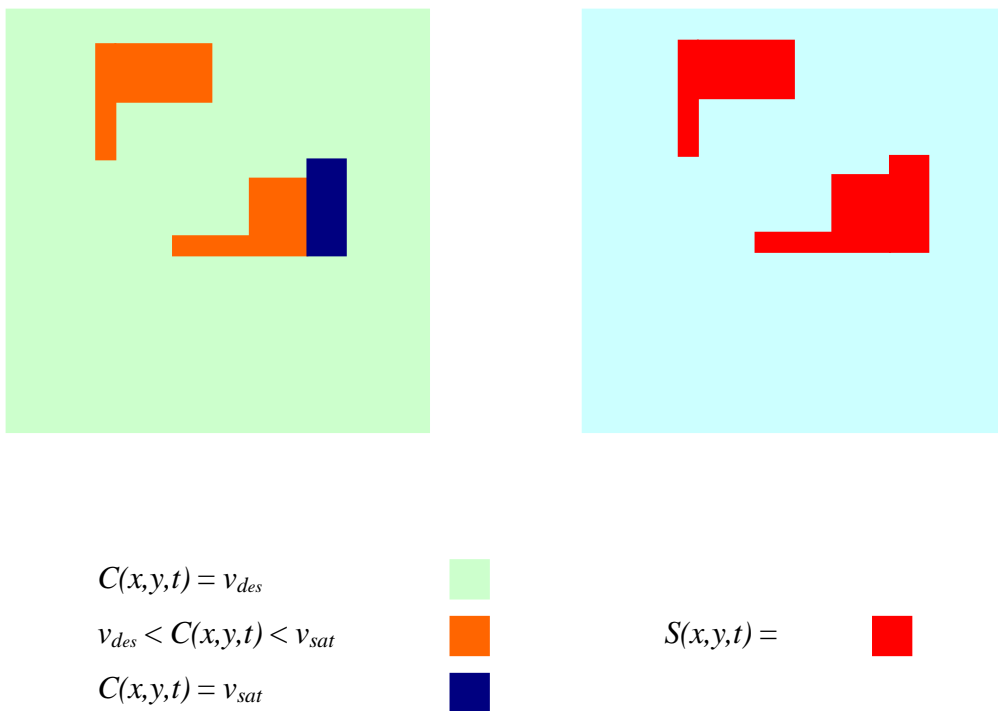


Figura 3.4. Puntos con carga no máxima aislados y unidos a puntos con carga máxima

Evidentemente, para nuestros propósitos, el grupo de pixeles con valor de carga $v_{des} < C(x, y, t) < v_{sat}$ aislados (no pegados físicamente a puntos con valor $C(x, y, t) = v_{sat}$), y, por tanto, pertenecientes al falso movimiento del fondo de la imagen, no deben captar nuestra atención. Por ello, se proponen los mecanismos de cooperación laterales necesarios para poder cumplirse las siguientes premisas:

- Los puntos con valor de carga $C(x, y, t) = v_{sat}$ ayudan a aumentar la carga de los puntos con valor $v_{des} < C(x, y, t) < v_{sat}$ directa o indirectamente conectados a ellos.
- Los puntos con valor $v_{des} < C(x, y, t) < v_{sat}$ sin conexión con puntos totalmente cargados tenderán a la descarga, y por lo tanto a la desactivación total.

Gráficamente, esto se representaría como:

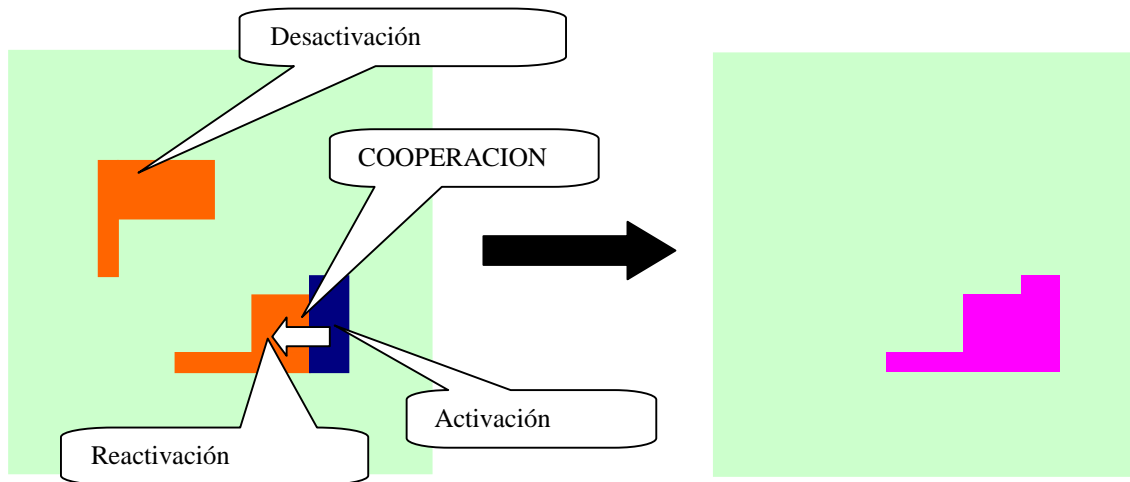


Figura 3.5. Activación, reactivación y desactivación.

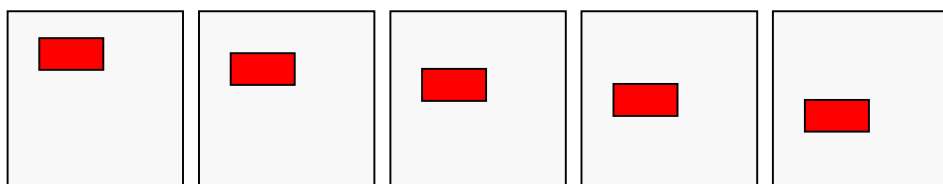
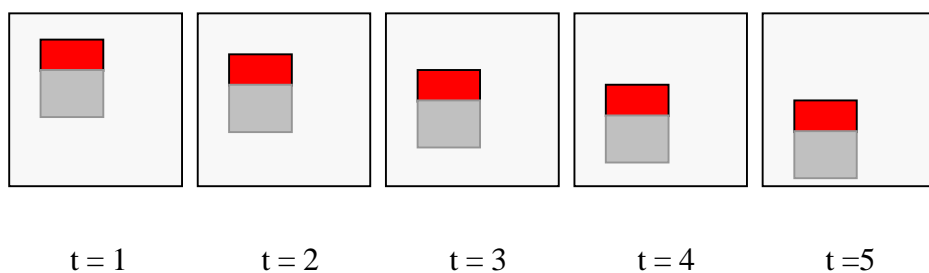
3.1.3.7. Segmentación en bandas de nivel de gris

Como sabemos que el método de detección de movimiento a partir del cambio de nivel de gris es válido cuando el objeto tiene un solo nivel de gris, se ha optado por segmentar cada imagen en un número n (típicamente 8 o 16) de niveles de gris y usar los algoritmos anteriores sobre cada una de las bandas de nivel de gris, como si tuviésemos n imágenes formadas por objetos móviles de una sola banda de nivel de gris. Nótese que utilizar todos los niveles de gris equivale a usar 256 bandas de nivel de gris, lo que no resta generalidad al argumento expuesto.

Las figuras 3.6. y 3.7. muestran la idea. En la figura 3.6. se muestra cómo una secuencia de imágenes se divide en dos secuencias de imágenes distintas (porque hay dos bandas de nivel de gris diferentes en movimiento). Se aplica el algoritmo de detección de movimiento a cada una de las secuencias por separado, obteniéndose dos mapas de movimientos distintos.

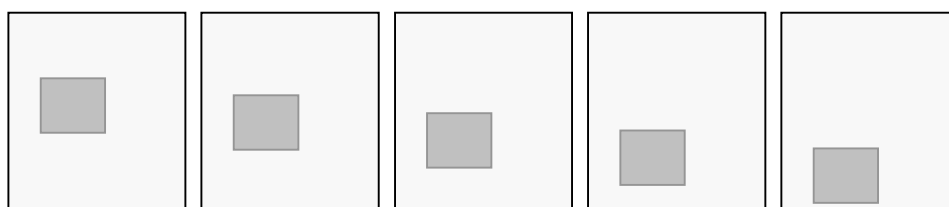
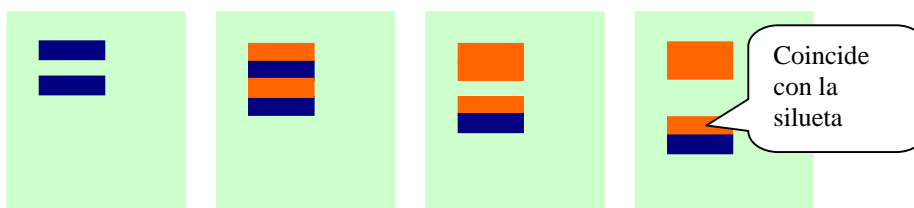
La figura 3.7., por su lado, muestra cómo se combinan los resultados de la obtención de las distintas sombras en las distintas bandas de nivel de gris para obtener la silueta completa del objeto en movimiento.

Tendremos, pues, a partir de este momento, un valor de carga por cada banda de nivel de gris por cada pixel de la imagen. Del mismo modo, se obtendrá la existencia de silueta por cada banda de nivel de gris y pixel. Al final, se unen las sombras obtenidas por cada banda de nivel de gris para obtener la silueta completa del objeto en movimiento.



Segmentación nivel de gris n

Estado de carga nivel de gris n



Segmentación nivel de gris m

Estado de carga nivel de gris m

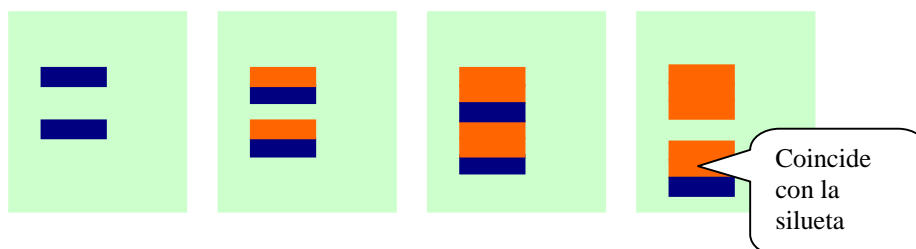


Figura 3.6. Descomposición de un elemento móvil en sus diferentes bandas de nivel de gris

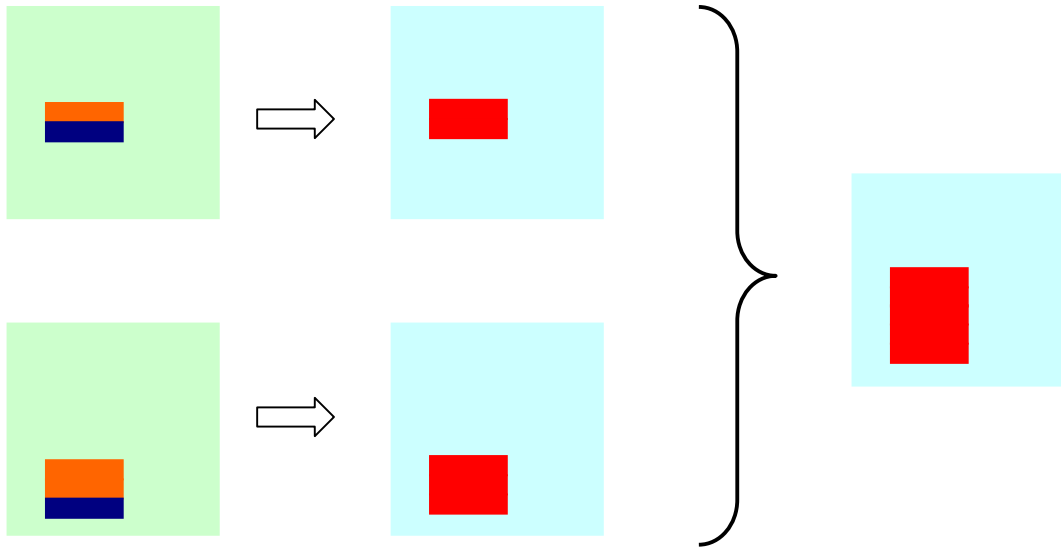


Figura 3.7. Obtención de la silueta a partir de las diferentes bandas de nivel de gris de un elemento

3.1.3.8. Clasificación de las siluetas detectadas

Se propone, igualmente, utilizar otro mecanismo de interacción lateral para diferenciar claramente las siluetas de todos los objetos que conforman el movimiento en la escena.

Podemos suponer que un objeto es un conjunto cerrado de píxeles con valor de carga superior a un mínimo exigido. Efectuando la media de la carga de todos los píxeles que conforman la silueta, se obtendrá un valor de carga en cada instante t que será la señal de identidad del objeto en movimiento. Esto es evidentemente responsabilidad de la función de acumulación espacio-temporal F_l en la escala de tiempos local T .

$$C(x, y, T) = (1 - \lambda_2) \cdot F_l \left[C(x, y, T - \Delta T), \sum_{\alpha=x-1}^{x+1} \sum_{\beta=y-1}^{y+1} C(\alpha, \beta, T) \right] + \frac{\lambda_2(\lambda_2 + 1)}{2} v_{des} + \frac{\lambda_2(\lambda_2 - 1)}{2} v_{sat}, \quad \forall (\alpha, \beta) \neq (x, y), (\alpha, \beta) \in R, (a, b) \in R$$

Así, podemos tener²:

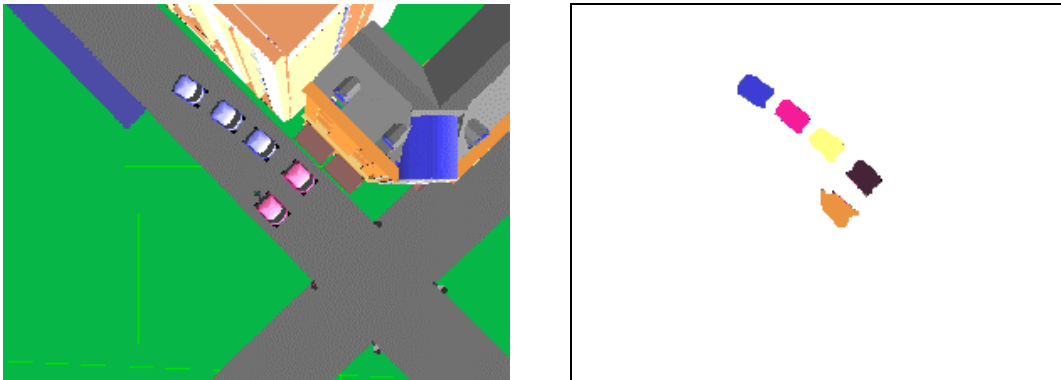


Figura 3.8. Discriminación de diferentes objetos

² Por cortesía de Institut National de Recherche en Informatique et Automatique, IRISA Project.

3.2. Arquitectura del método

3.2.1. Arquitectura multicapa del modelo de interacción lateral en computación acumulativa

Se contempla el presente apartado entroncado como apartado inspirado en la representación esquemática del proceso de visión artificial descrito por Mira y Delgado [Mir95].

En efecto, la arquitectura del método contempla básicamente la etapa de procesado de bajo nivel de un modo similar a la representación ofrecida en la figura 3.9. para las capas 0 a 3 del modelo, mientras que toma en cuenta el procesado de alto nivel en cuanto a su capa 4. Vamos a ver en este capítulo una arquitectura multicapa mínima para el modelo de interacción lateral en computación acumulativa en su aplicación a la obtención de las siluetas de los objetos deformables en una secuencia indefinida de imágenes. De la etapa de procesado de bajo nivel de Mira y Delgado, la arquitectura incluye en sus sucesivas capas una segmentación, una extracción de características y una clasificación.

El modelo de interacción lateral en sus primeras capas es independiente tanto de las restricciones propias de las escenas analizadas como del procesado de alto nivel. Podemos, y debemos, ver el modelo de interacción lateral en estas capas y aplicado al proceso de visión artificial como una pieza aislada de todo procesado inteligente.

El modelo general de interacción lateral en computación acumulativa, así como toda la arquitectura multicapa que se va a comentar en este capítulo, podrá encajarse como una pieza de puzzle en toda una serie de diferentes escenas del mundo real. No será hasta la validación del método (capítulo 5 de esta memoria) que se entoncará el modelo en un extenso conjunto de diferentes escenas. No será hasta ese momento en que se añada el procesado de alto nivel fuertemente ligado a los diferentes tipos de escenas.

La implementación del modelo, véase el capítulo 4 de esta memoria, introducirá igualmente las restricciones de las imágenes de entrada.

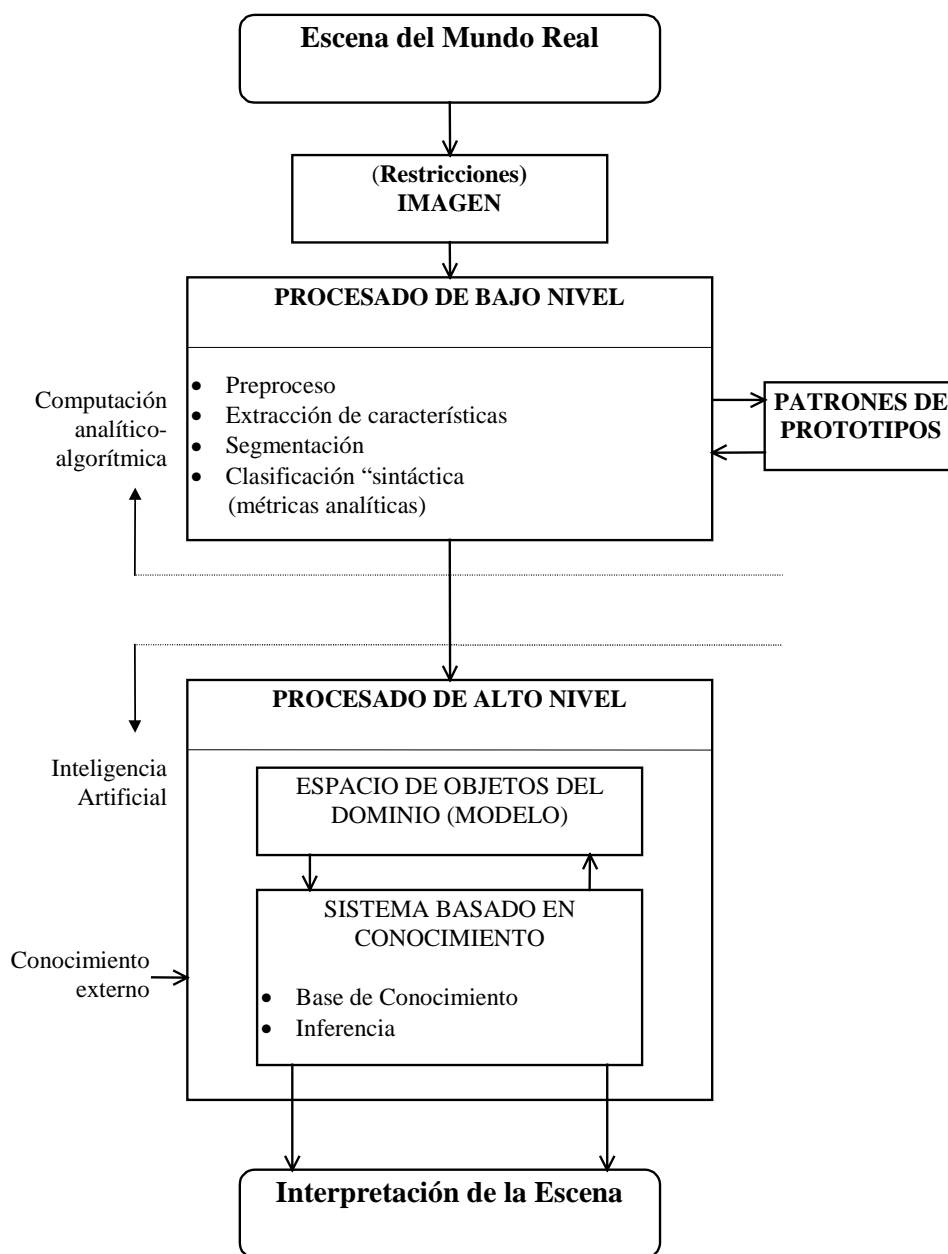


Figura 3.9. Representación esquemática del proceso de visión artificial

Se presenta, en primer lugar, y en la siguiente figura, la estructura completa elegida como solución computacional modular. En ella, se pueden apreciar cinco capas que forman la arquitectura del método de interacción lateral en computación acumulativa elegidas para abordar el problema de la obtención de las siluetas de todos los objetos deformables en movimiento en una secuencia indefinida de imágenes.

Las cinco capas son:

- a) **Capa 0: Segmentación por bandas de nivel de gris.** Esta capa cubre la necesidad anteriormente descrita de segmentar la imagen en un conjunto preestablecido de n bandas de nivel de gris. En cada uno de sus elementos, su entrada será el valor de nivel de gris del pixel correspondiente de la imagen a cada cuadro de tiempo global t . De cada uno de sus elementos salen n valores $GLL_k(x,y,t)$ hacia las n subcapas (tantas como bandas de nivel de gris se hayan establecido) de la capa 1 en la misma coordenada. Estos valores indican la pertenencia o no del pixel a cada una de las bandas de nivel de gris.
- b) **Capa 1: Interacción lateral para la computación acumulativa.** Esta capa se encarga de la obtención del valor de permanencia $PM_k(x,y,t)$ en base a una descomposición en bandas de nivel de gris. Tendremos n subcapas y cada una de ellas albergará por cada elemento el valor de la computación acumulativa presente en la escala de tiempos global t . La interacción lateral en esta capa irá orientada a reactivar la carga de permanencia de aquellos elementos con cierto valor de carga que estén directa o indirectamente conectados a elementos saturados. La carga de permanencia se ofrecerá a la siguiente capa como salida.
- c) **Capa 2: Interacción lateral para la obtención de elementos de siluetas.** La capa 2 también está formada por n subcapas, y en ellas se realiza mediante interacción lateral un reparto de la carga entre todos los vecinos con una carga mínima y conectados entre sí. Además de repartir la carga $C_k(x,y,t)$ en bandas de nivel de gris, se consigue, a este nivel, diluir la carga debida al movimiento del fondo. La nueva carga obtenida en esta capa es ofrecida como salida hacia la capa 3.
- d) **Capa 3: Interacción lateral para la fusión de objetos en movimiento.** En cada uno de los elementos de esta capa 3, tenemos una entrada desde cada uno de los elementos correspondientes de las n subcapas de la capa 2. Esta capa tiene como finalidad la unificación de las siluetas. Para ello, toma las cargas de entrada de cada una de las bandas de nivel de gris y efectúa una fusión de dichos valores, obteniendo cada una de las siluetas de la imagen origen. Su salida es el conjunto de siluetas $S(x,y,t)$.
- e) **Capa 4: Aplicación.** La capa 4 es una capa multifuncional capaz de implementar toda una serie de aplicaciones específicas de alto nivel.

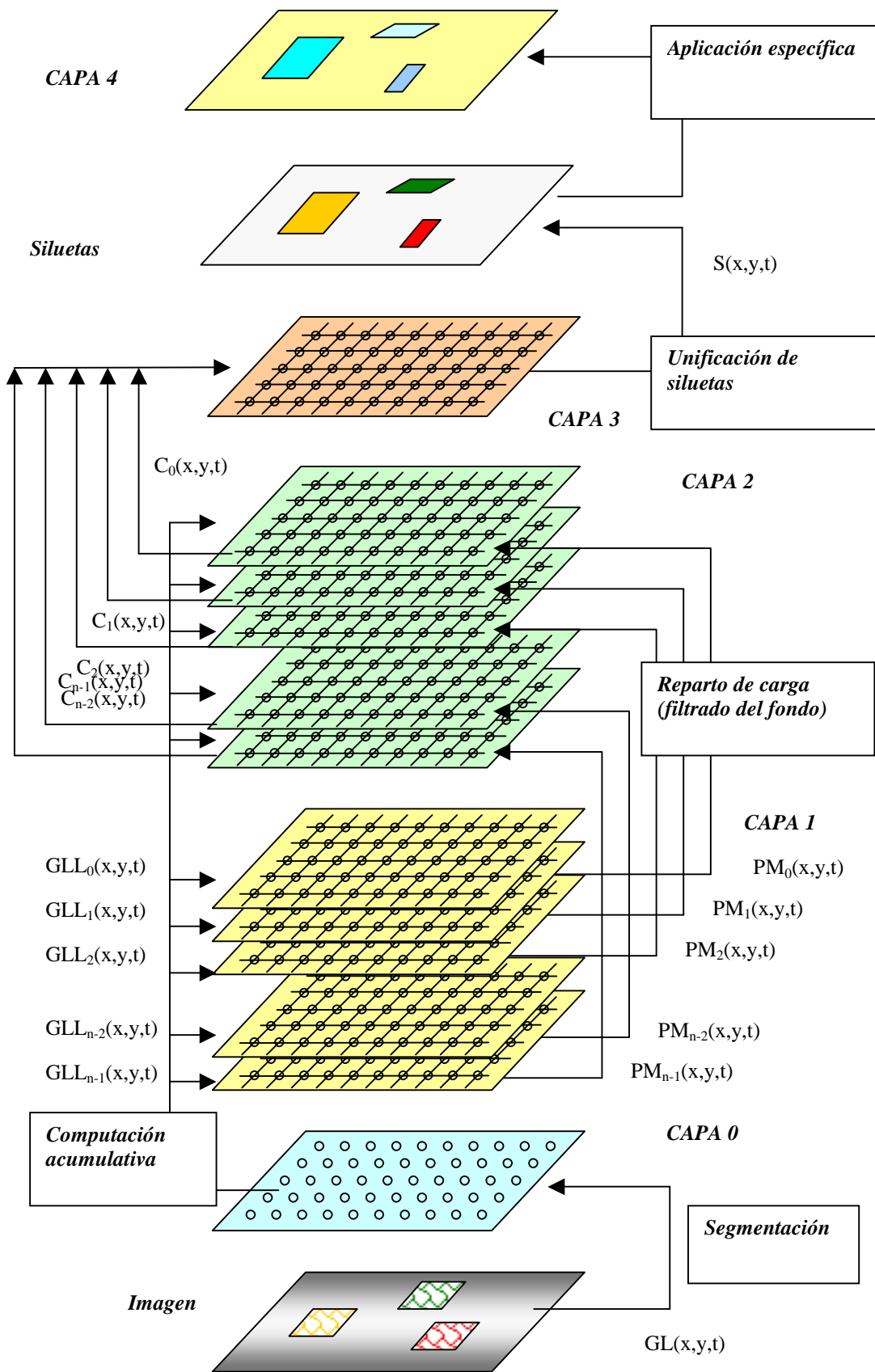


Figura 3.10. Configuración multicapa

3.2.2. Capa 0: Segmentación por bandas de nivel de gris

Una implementación en forma de computación modular de los mecanismos descritos hasta el momento nos llevan a hablar de una primera capa de hasta $i*j$ elementos (uno por cada pixel de la imagen).

Las conexiones externas de cada uno de los puntos de la imagen en esta capa serían las de la figura 3.11.

Si llamamos $GL(x,y,t)$ al nivel de gris del punto (x,y) en el instante t , por cada una de las bandas de nivel de gris k , y por cada uno de los pixeles (x,y) de la imagen, tenemos, en todo instante t , el siguiente algoritmo:

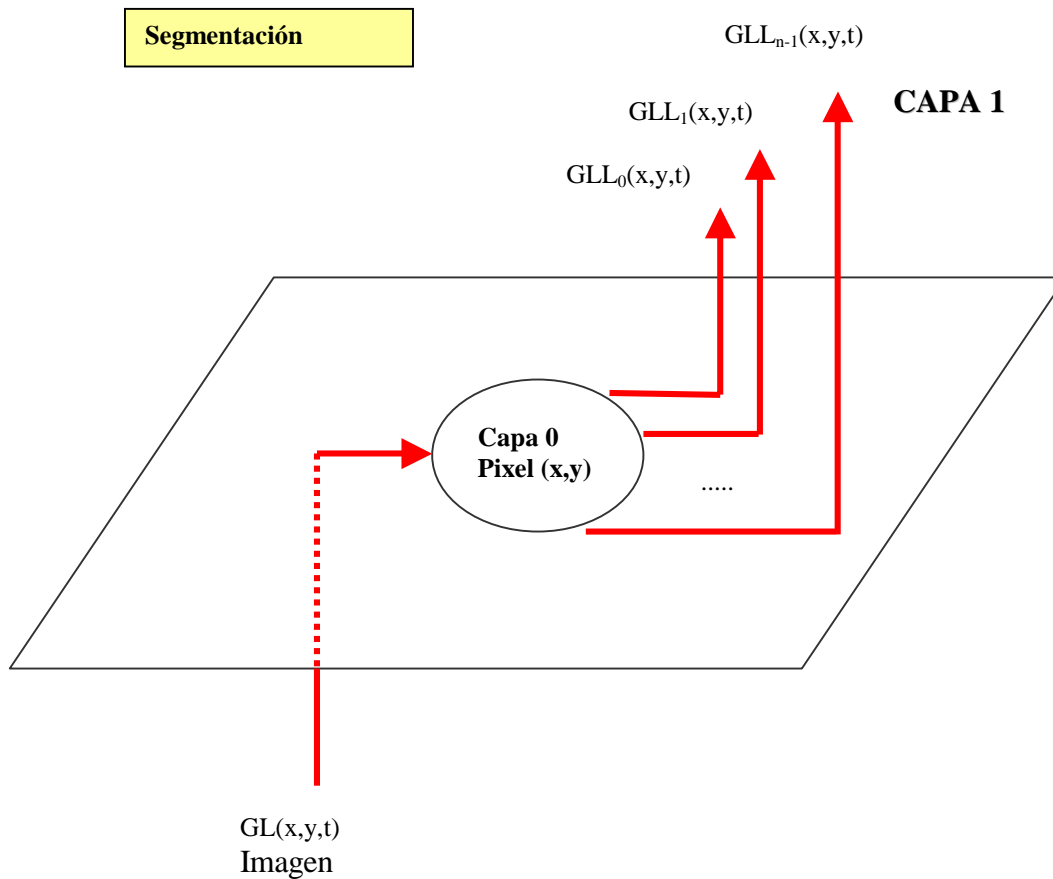


Figura 3.11. Capa 0. Conexiones externas

```
// Capa 0. Pixel (x,y).
// Segmentación por bandas de nivel de gris.
// Algoritmo.
```

$$GLL_k(x, y, t) = \begin{cases} 1, & \text{si } GL(x, y, t) \in [(256/n)k, (256/n)(k+1)[\\ 0, & \text{en caso contrario} \end{cases}$$

siendo n = número de bandas de nivel de gris
 k = banda de nivel de gris

Se trata pues de determinar en qué banda de nivel de gris se encuentra un determinado pixel. No estamos evaluando, pues, a este nivel, si existe, o no, movimiento en una banda de nivel de gris en un pixel determinado. Esta tarea se deja para el nivel siguiente.

Si suponemos n bandas de nivel de gris diferentes, tendremos la situación de la figura 3.12.

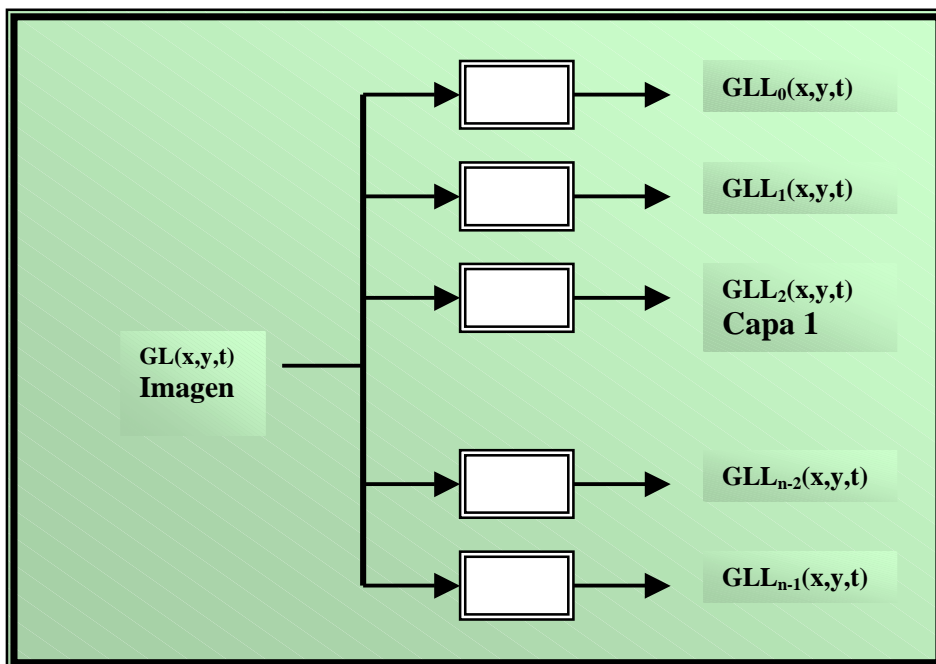


Figura 3.12. Capa 0. Determinación de la banda de nivel de gris

Puede apreciarse que siempre se activa una, y solamente una, de las salidas de todos los módulos detectores de la banda de nivel de gris. Este hecho, aunque obvio, es de gran interés en las capas superiores de la arquitectura, ya que ello evitará posibles conflictos entre los valores ofrecidos por las diferentes bandas de nivel de gris. En efecto, solamente una banda de nivel de gris tendrá valores válidos.

3.2.3. Capa 1: Interacción lateral para la computación acumulativa

Se propone usar una conexión de estructuras modulares en forma de malla. De este modo todas las filas estarán conectadas entre sí, y del mismo modo, las columnas. Habrá que tener en cuenta, además, que esta capa estará formada por n subcapas (tantas como bandas de nivel de gris se hayan elegido). Si nos centramos en un grupo concreto de estructuras de una de estas subcapas de la capa 1, tenemos la figura 3.13.

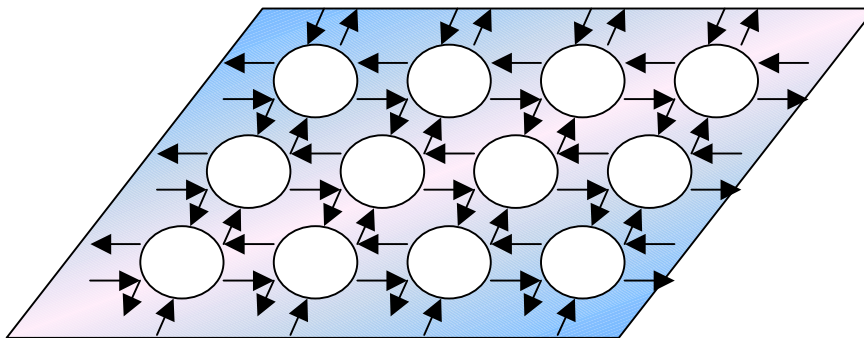


Figura 3.13. Capa 1. Subcapa k. Conexiones laterales en forma de malla

Cada uno de los nodos de la malla puede ser visto como una estructura tal como se presenta en la figura 3.14., donde las conexiones laterales se denominan $ACTI$ y tienen el siguiente significado:

- $ACTI_{in/up}$: es la conexión lateral de entrada a la estructura modular que proviene de la estructura que se encuentra una posición más arriba en la malla
- $ACTI_{in/down}$: es la conexión lateral de entrada a la estructura modular que proviene de la estructura que se encuentra una posición más abajo en la malla
- $ACTI_{in/left}$: es la conexión lateral de entrada a la estructura modular que proviene de la estructura que se encuentra una posición más a la izquierda en la malla

$ACT1_{in/right}$: es la conexión lateral de entrada a la estructura modular que proviene de la estructura que se encuentra una posición más a la derecha en la malla

$ACT1_{out/up}$: es la conexión lateral de salida de la estructura modular que va hacia la estructura que se encuentra una posición más arriba en la malla

$ACT1_{out/down}$: es la conexión lateral de salida de la estructura modular que va hacia la estructura que se encuentra una posición más abajo en la malla

$ACT1_{out/left}$: es la conexión lateral de salida de la estructura modular que va hacia la estructura que se encuentra una posición más a la izquierda en la malla

$ACT1_{out/right}$: es la conexión lateral de salida de la estructura modular que va hacia la estructura que se encuentra una posición más a la derecha en la malla

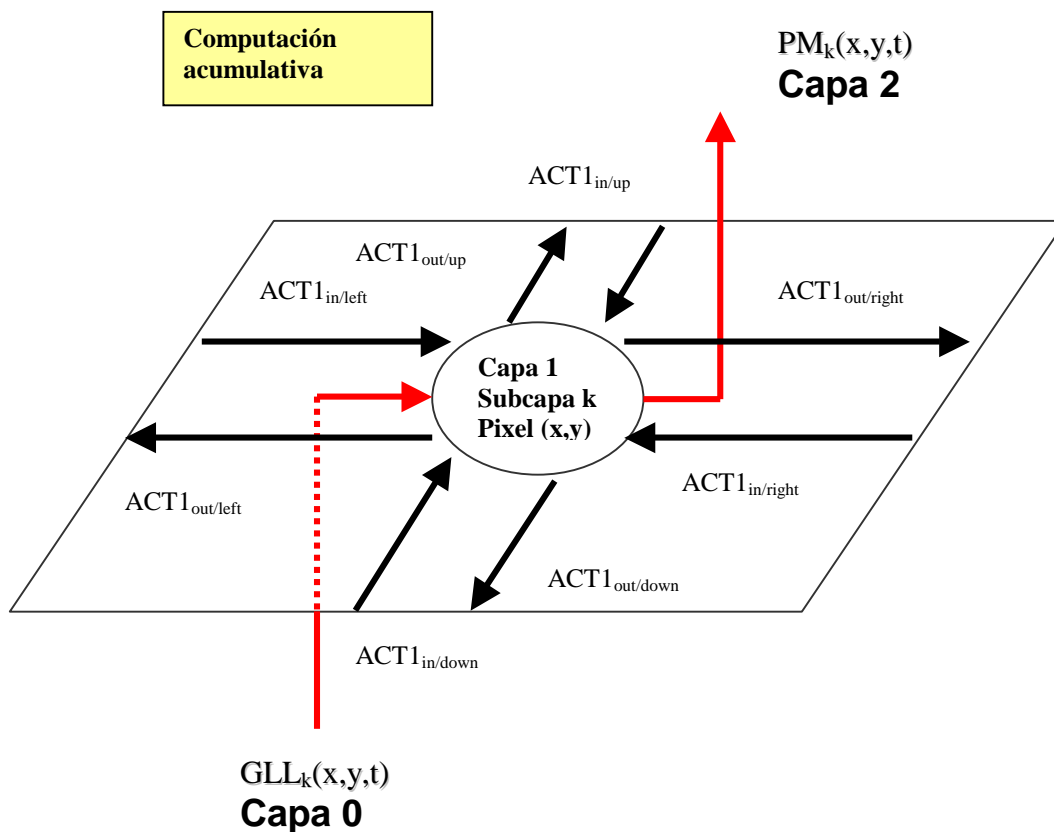


Figura 3.14. Capa 1. Subcapa k. Conexiones externas

Si nos vamos a las memorias de permanencia, podemos formular algorítmicamente el comportamiento deseado para nuestros módulos:

// Capa 1. Subcapa k . Pixel (x,y) .

// Mecanismos de interacción lateral para la computación acumulativa.

// Algoritmo.

// ACUMULACION ESPACIAL.

// Espacio de tiempos t .

// Paso 1. Carga / descarga debida a la información de movimiento.

// Apertura / cierre de canales laterales de entrada $OPEN_k$.

$$PM_k(x, y, t) = \begin{cases} v_{des}, & \text{si } GLL_k(x, y, t) \equiv 0 \\ v_{sat}, & \text{si } GLL_k(x, y, t) \equiv 1 \cap GLL_k(x, y, t - \Delta t) \equiv 0 \\ PM_k(x, y, t - \Delta t) - v_{dm}, & \text{si } GLL_k(x, y, t) \equiv 1 \cap GLL_k(x, y, t - \Delta t) \equiv 1 \end{cases}$$

siendo v_{dm} el valor de descarga por detección de movimiento.

$$PM_k(x, y, t) = \begin{cases} PM_k(x, y, t), & \text{si } PM_k(x, y, t) > v_{des} \\ v_{des}, & \text{en caso contrario} \end{cases}$$

$$OPEN_k(x, y, t) = \begin{cases} 1, & \text{si } v_{des} < PM_k(x, y, t) < v_{sat} \\ 0, & \text{en caso contrario} \end{cases}$$

// Paso 2. Envío de información de recarga.

$$ACT1_{out/up} = ACT1_{out/down} = ACT1_{out/right} = ACT1_{out/left} = \begin{cases} 1, & \text{si } PM_k(x, y, t) \equiv v_{sat} \\ 0, & \text{en caso contrario} \end{cases}$$

// ACUMULACION ESPACIO-TEMPORAL.

// Espacio de tiempos $T \ll t$.

WHILE ($T < T_{max}$) DO

{

// Paso 3. Recarga debida a la interacción lateral.

IF ($OPEN_k(x,y,T) == 1$) THEN

{

$$PM_k(x, y, T) = PM_k(x, y, T - \Delta T) + v_{ry},$$

$$OPEN_k(x, y, T) = 0,$$

$$si \ ACT1_{in/up} \equiv 1 \cup ACT1_{in/down} \equiv 1 \cup$$

$$ACT1_{in/right} \equiv 1 \cup ACT1_{in/left} \equiv 1$$

$$PM_k(x, y, T) = v_{sat}, \quad si \ PM_k(x, y, T) > v_{sat}$$

}

// Paso 4. Paso de carga a los vecinos.

$$ACT1_{out/up} = ACT1_{out/down} = ACT1_{out/right} = ACT1_{out/left} =$$

$$= \begin{cases} 1, & si \ PM_k(x, y, T) \equiv v_{sat} \cup \\ & PM_k(x, y, T) > v_{des} \cap (ACT1_{in/up} \equiv 1 \cup ACT1_{in/down} \equiv 1 \cup \\ & ACT1_{in/right}(x, y, T) \equiv 1 \cup ACT1_{in/left} \equiv 1) \\ 0, & en \ caso \ contrario \end{cases}$$

}

// DISTRIBUCION.

// Espacio de tiempos t .

// Paso 5. Carga / descarga debida a la información de movimiento.

$$PM_k(x, y, t) = \begin{cases} PM_k(x, y, t), & si \ PM_k(x, y, t) > \theta_{per} \\ \theta_{per}, & en \ caso \ contrario \end{cases}$$

A continuación se pasa a explicar, paso a paso, el algoritmo anteriormente expuesto.

Acumulación espacial.

En el paso 1, se efectúa la carga y/o descarga de la memoria de permanencia debida a la detección de movimiento. Esta información estará asociada a la subcapa k de la capa 1 (banda de nivel de gris k). Esta información nos viene dada desde la capa 0.

Si a esta subcapa k llega información de que el pixel se encuentra en la banda de nivel de gris k . En cada elemento (x,y) nos encontramos ante dos posibilidades:

- si la subcapa no se corresponde con la banda de nivel de gris de la imagen, el valor de permanencia es descargado hasta el valor v_{des}
- si la subcapa se corresponde con la banda de nivel de gris de la imagen en el instante de tiempo t , y no se correspondía con la banda de nivel de gris en el instante anterior $t-\Delta t$, el valor de permanencia es cargado al máximo de saturación v_{sat}
- si la subcapa se corresponde con la banda de nivel de gris de la imagen en el instante de tiempo t , y también se correspondía con la banda de nivel de gris en el instante anterior $t-\Delta t$, el valor de permanencia es descargado en un valor v_{dm} (valor de descarga debida al movimiento), no pudiendo bajar de un valor mínimo v_{des}

La descarga de un pixel en la cantidad v_{dm} es el modo en que se deja de prestar atención a un punto de la imagen que en el pasado había sido merecedor de nuestro interés. Como se verá más adelante, si un punto no está directa o indirectamente ligado mediante mecanismos de interacción lateral a un punto con carga máxima v_{sat} , al cabo del tiempo tiende a la descarga total.

La lógica de carga asociada a este paso es la descrita en la siguiente figura 3.16.

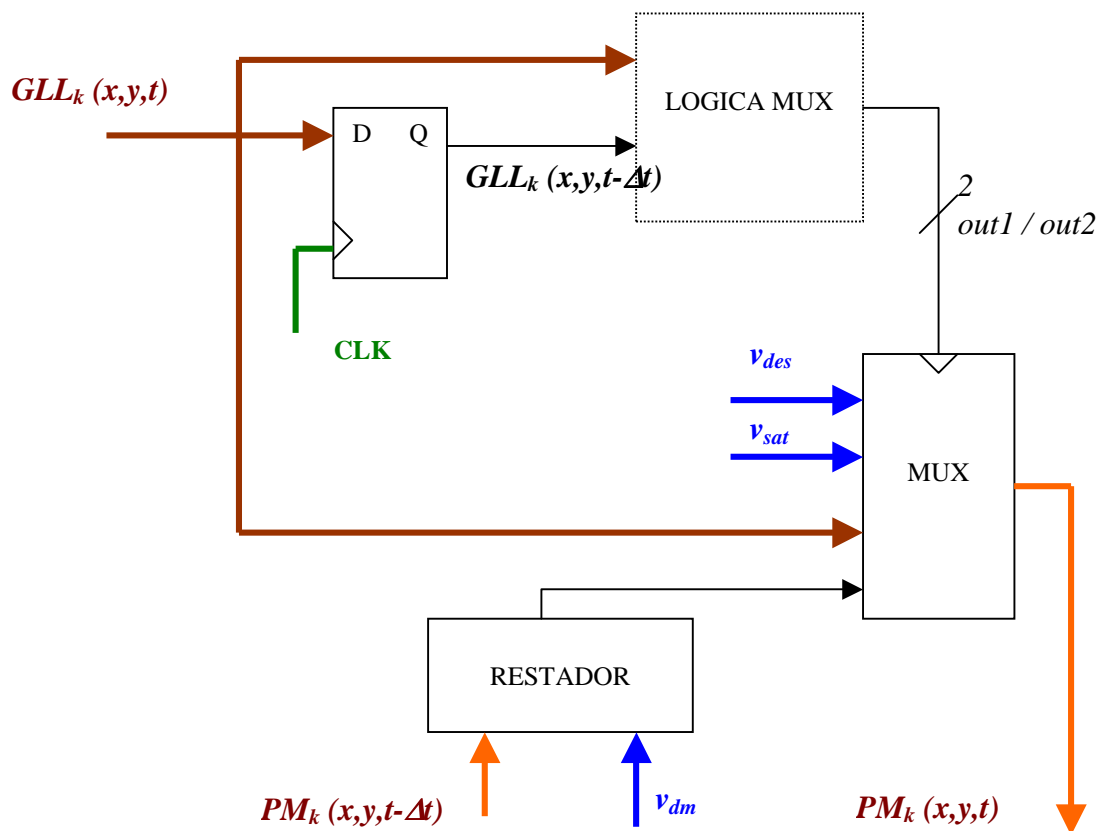
También debe apreciarse en este paso 1, la puesta a 1 / 0 de la variable $OPEN_k$, cuyo significado es el siguiente:

- la variable a 0 indica que la estructura ha cerrado sus canales de interacción lateral de entrada, y por lo tanto, no querrá recibir ningún tipo de estímulo desde los mismos; la variable toma este valor cuando el valor de la memoria de permanencia tiene un valor de carga o de descarga total

- la variable a 1 indica que la estructura ha abierto sus canales de interacción lateral de entrada para recibir cualquier tipo de estímulo desde los mismos; la variable toma este valor cuando el valor de la memoria de permanencia tiene un valor de carga no total

El paso 2 sirve para que los pixeles con valor de permanencia máxima informen de ello a sus vecinos a través de los canales habilitados a tal efecto, es decir los canales de tipo $ACTI_{out}$.

Estos dos pasos anteriores ocurren en el espacio de tiempo t normal. Los dos siguientes pasos ocurren en otro espacio de tiempo $T \ll t$.



LOGICA MUX:

$GLL_k(x,y,t)$	$GLL_k(x,y,t-\Delta t)$	out1	out2
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

Figura 3.16. Lógica de carga asociada a la capa 1 del modelo

Acumulación espacio-temporal.

El paso 4 es similar al paso 2. La diferencia estriba en que no se contemplan solamente los pixeles con carga máxima, sino también aquellos con carga intermedia, pero avisados a través de las señales laterales de entrada de que tienen que retransmitir las señales que les llegan.

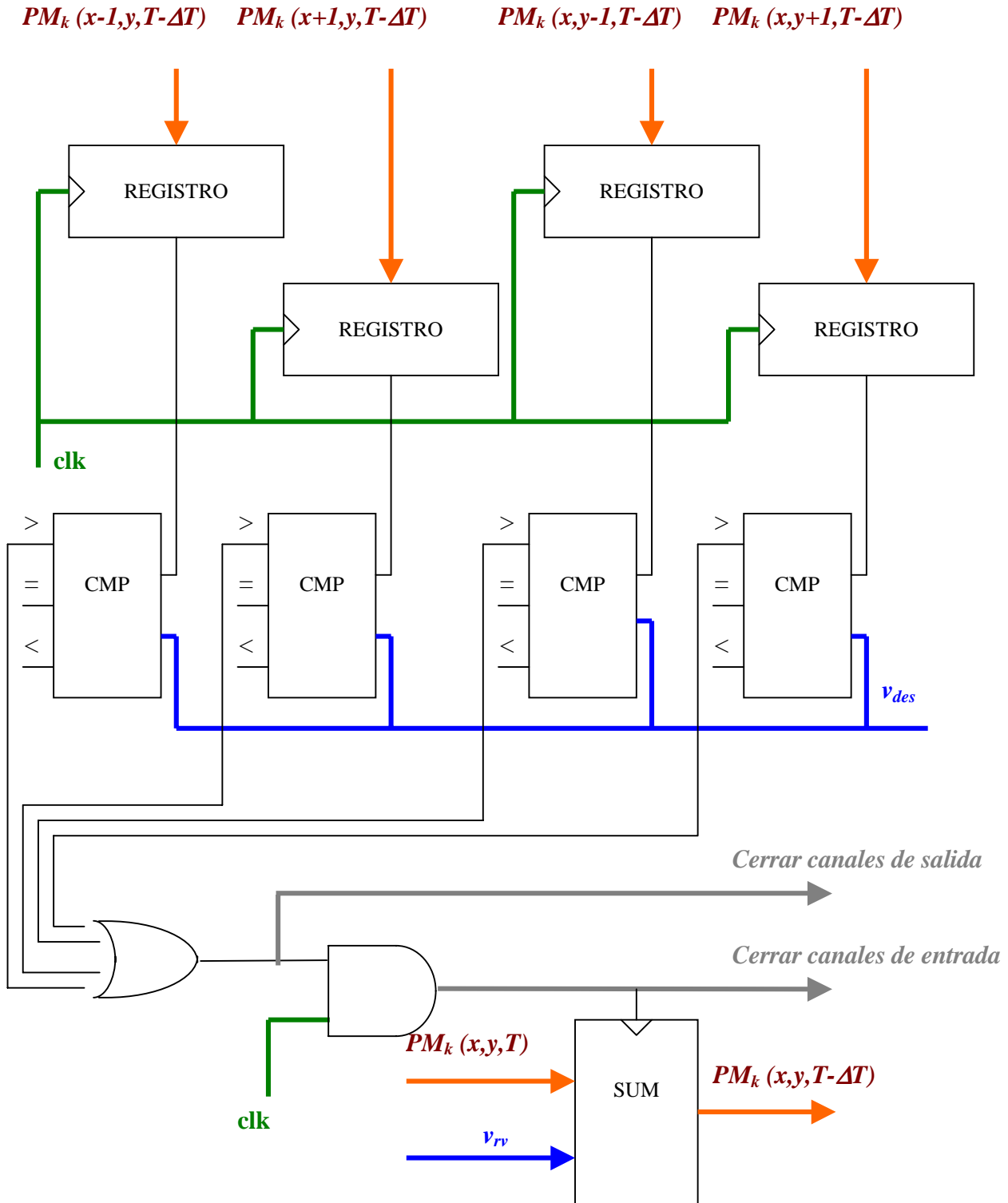


Figura 3.17. Lógica de carga lateral asociada a la capa 1 del modelo

En el paso 3 nos encontramos con una carga extra v_{rv} (carga debida a los vecinos) en la memoria de permanencia en aquellos puntos de la imagen que reciben un estímulo (de tipo $ACTI_{in}$) de cualquiera de los cuatro puntos que tienen a su alrededor. Ello, por supuesto, siempre que se cumplan una serie de requisitos. Estas condiciones son, en primer lugar, que haya llegado una activación lateral, pero, además, que la memoria de permanencia no esté descargada. Evidentemente, tampoco puede cargarse la memoria de permanencia por encima del valor máximo v_{sat} . Véase también que la recarga solamente puede ocurrir una vez. De ello se encarga el manejo de la variable $OPEN_k$. La lógica de carga lateral se muestra en la anterior figura.

Este último mecanismo de recarga es el mecanismo de interacción lateral al nivel de la capa 1 (por cada subcapa de banda de nivel de gris) que permite mantener activa nuestra atención en los puntos con cierta carga, llegando incluso, si el valor de v_{rv} es mayor que el de v_{dm} , a reforzar su valor de memoria de permanencia.

Distribución.

De nuevo en el espacio de tiempos global t , aquí en el paso 5, se efectúa una distribución hacia la siguiente capa del valor de permanencia umbralizado $PM_k(x,y,t)$.

Todo ello nos lleva a dibujar la estructura modular de la capa 1 para cada pixel de la imagen y para cada banda de nivel de gris.

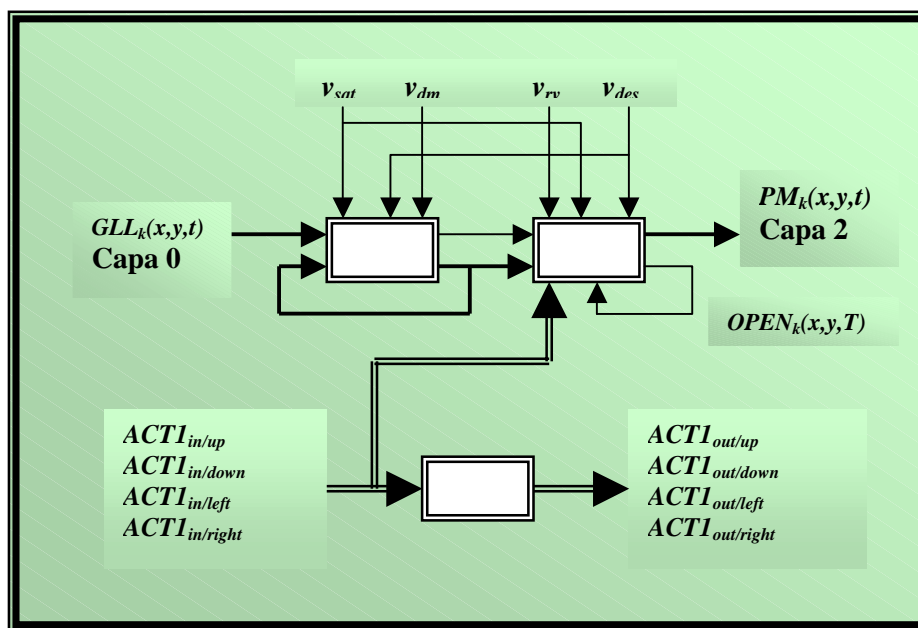


Figura 3.18. Capa 1. Subcapa k. Estructura modular para cada pixel de la imagen

Para explicar mejor la idea central de esta capa 1, diremos que la activación hacia las estructuras modulares laterales (arriba, abajo, derecha e izquierda) conlleva la siguiente idea básica:

1. todas las estructuras modulares que en esta capa tienen valor de permanencia máximo v_{sat} , informan de ello a sus vecinas (de ellas parte la carga hacia los vecinos)
2. todas las estructuras modulares que están en estado de cargado, pero no con valor máximo, y que han sido activadas desde algún vecino, dejan pasar esta información a través de ellas (son estructuras transparentes al paso de la carga)
3. las estructuras modulares con valor de permanencia igual a v_{des} (descargadas) paran el paso de la información de carga hacia los vecinos (son estructuras opacas)

Por lo tanto, se trata de una explosión de activación lateral, que parte de las estructuras con memoria de permanencia en v_{sat} , y que se propaga linealmente hacia todas las direcciones, mientras no aparezca en el camino una estructura con memoria de permanencia descargada.

La siguiente tabla nos puede mostrar cómo utilizar adecuadamente la relación entre los valores de v_{dm} y v_{rv} para la consecución de objetivos distintos.

Relación entre v_{dm} y v_{rv}	Explicación
$v_{dm} \leq v_{rv}$	Todos los píxeles con valor de permanencia comprendido entre v_{des} y v_{sat} y unidos directa o indirectamente a píxeles con valor v_{sat} , pasan a v_{sat} . El píxel forma parte de la silueta mientras se mueva algún píxel de dicha silueta.
$v_{dm} > v_{rv}$	Todos los píxeles con valor de permanencia comprendido entre v_{des} y v_{sat} , y unidos directa o indirectamente a píxeles con valor v_{sat} , se van a ir descargando lentamente. Los píxeles más alejados del “centro” del movimiento irán desapareciendo lentamente de la silueta.
$v_{dm} \gg v_{rv}$	Todos los píxeles con valor de permanencia comprendido entre v_{des} y v_{sat} , y unidos directa o indirectamente a píxeles con valor v_{sat} , se van a descargar rápidamente. La silueta estará formada por los píxeles que se han movido recientemente.

3.2.4. Capa 2: Interacción lateral para el reparto de carga por bandas de nivel de gris

A partir de los valores de la memoria de permanencia en cada pixel por banda de nivel de gris, vamos a ver cómo es posible obtener la silueta de todas las partes de un objeto en movimiento. Cuando hablamos de una parte de un objeto, nos referimos concretamente a la unión de pixeles que se encuentran juntos y en una misma banda de nivel de gris.

La discriminación de cada una de las partes que componen los objetos (que a su vez formarán el conjunto de las siluetas) se consigue igualmente por mecanismos de cooperación lateral. De nuevo vamos a conectar las estructuras modulares de esta capa en forma de malla, pero esta vez el entrelazado será doble. Nótese una vez más que tenemos tantas subcapas en esta capa 2 como bandas de nivel de gris se hayan definido.

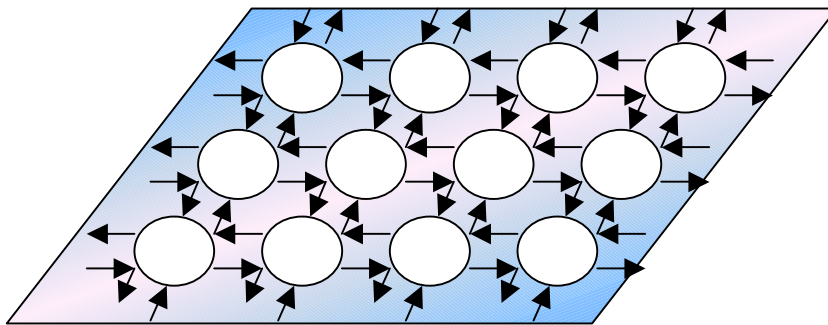


Figura 3.19. Capa 2. Subcapa k. Conexiones laterales en forma de malla

En el caso de la capa 2, se va a homogeneizar la carga entre todos los pixeles que tengan la misma banda de nivel de gris y que estén directa o indirectamente unidos entre sí.

De este modo, se conseguirá un doble objetivo:

1. El de diluir la carga debida al movimiento del fondo de la imagen entre los demás puntos del propio fondo. Así, no debería quedar constancia del movimiento propio del fondo, sino que nos quedaremos con el movimiento de los objetos de la escena.
2. El de obtener un parámetro común a todos los pixeles de la parte del objeto con misma banda de nivel de gris y que será elevado a niveles más altos (capa 3, en principio). Este parámetro nos dará una idea más o menos vaga de la velocidad del movimiento.

Las conexiones de cada una de las estructuras modulares de este nivel pueden ser vistas tal como se presentan en la figura 3.20., donde las conexiones laterales se denominan *ACT2* y tienen el siguiente significado:

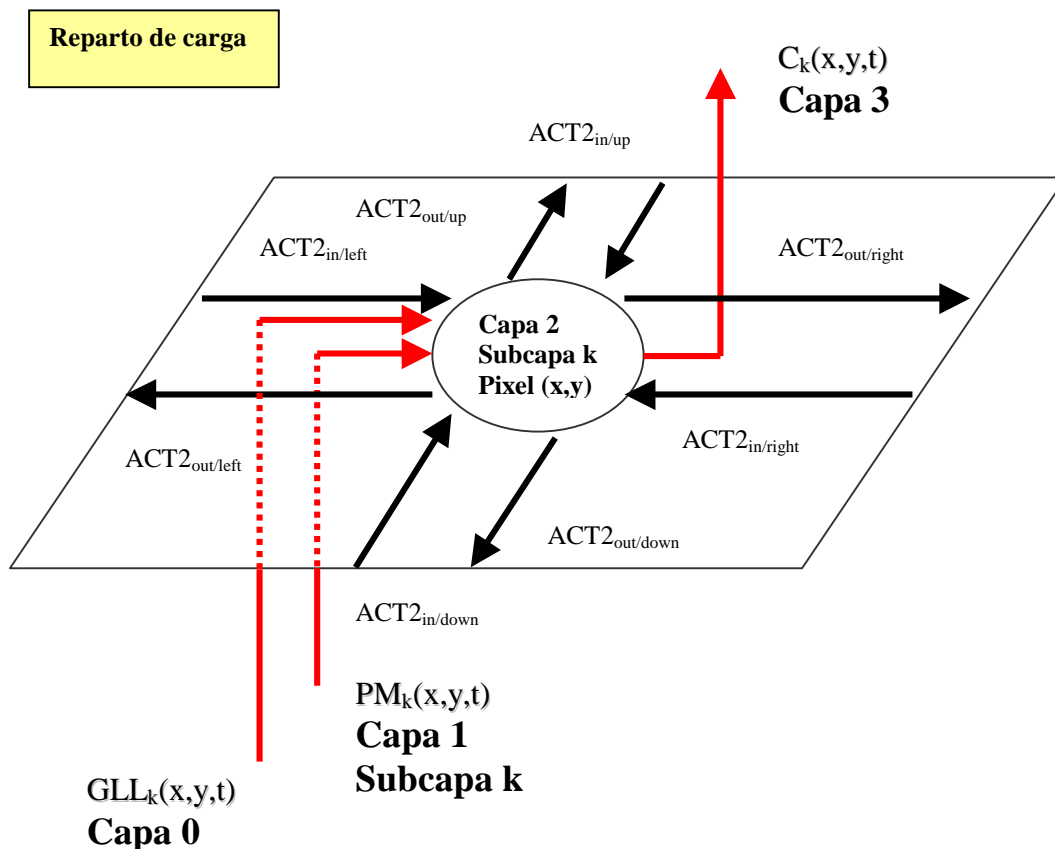


Figura 3.20. Capa 2. Subcapa k. Conexiones externas

- $ACT2_{in/up}$: es la conexión lateral de entrada a la estructura modular que proviene de la estructura que se encuentra una posición más arriba en la malla
- $ACT2_{in/down}$: es la conexión lateral de entrada a la estructura modular que proviene de la estructura que se encuentra una posición más abajo en la malla
- $ACT2_{in/left}$: es la conexión lateral de entrada a la estructura modular que proviene de la estructura que se encuentra una posición más a la izquierda en la malla
- $ACT2_{in/right}$: es la conexión lateral de entrada a la estructura modular que proviene de la estructura que se encuentra una posición más a la derecha en la malla
- $ACT2_{out/up}$: es la conexión lateral de salida de la estructura modular que va hacia la estructura que se encuentra una posición más arriba en la malla

$ACT2_{out/down}$: es la conexión lateral de salida de la estructura modular que va hacia la estructura que se encuentra una posición más abajo en la malla

$ACT2_{out/left}$: es la conexión lateral de salida de la estructura modular que va hacia la estructura que se encuentra una posición más a la izquierda en la malla

$ACT2_{out/right}$: es la conexión lateral de salida de la estructura modular que va hacia la estructura que se encuentra una posición más a la derecha en la malla

// Capa 2. Subcapa k . Pixel (i, j) .

// Mecanismos de interacción lateral para la obtención de siluetas por bandas de nivel de gris.

// Algoritmo.

// ACUMULACION ESPACIAL.

// Espacio de tiempos t .

// Paso 1. Comienzo.

$$C_k(x, y, t) = PM_k(x, y, t)$$

// ACUMULACION ESPACIO-TEMPORAL.

// Espacio de tiempos $T \ll t$.

IF $(C_k(x, y, t) > \theta_{per})$ {

 WHILE $(T < T_{max})$ DO {

 // Paso 2. Envío de información lateral.

$$ACT2_{out/up} = ACT2_{out/down} = ACT2_{out/right} = ACT2_{out/left} = C_k(x, y, T)$$

 // Paso 3. Recepción de información lateral y obtención de la nueva carga.

$$C_{up} = \begin{cases} ACT2_{in/up}, & \text{si } ACT2_{in/up} > \theta_{per} \\ 0, & \text{en caso contrario} \end{cases}$$

$$C_{down} = \begin{cases} ACT2_{in/down}, & \text{si } ACT2_{in/down} > \theta_{per} \\ 0, & \text{en caso contrario} \end{cases}$$

$$C_{right} = \begin{cases} ACT2_{in/right}, & \text{si } ACT2_{in/right} > \theta_{per} \\ 0, & \text{en caso contrario} \end{cases}$$

$$C_{left} = \begin{cases} ACT2_{in/left}, & \text{si } ACT2_{in/left} > \theta_{per} \\ 0, & \text{en caso contrario} \end{cases}$$

$$C_k(x, y, T) = \text{media} (C_k(x, y, T - \Delta T) + C_{up} + C_{down} + C_{right} + C_{left})$$

}

}

// DISTRIBUCION.

// Espacio de tiempos t.

// Paso 4. Envío del valor de carga media a la capa 3.

$$C_k(x, y, t) = \begin{cases} C_k(x, y, t), & \text{si } C_k(x, y, t) > \theta_{car} \\ \theta_{car}, & \text{en caso contrario} \end{cases}$$

Capa 2

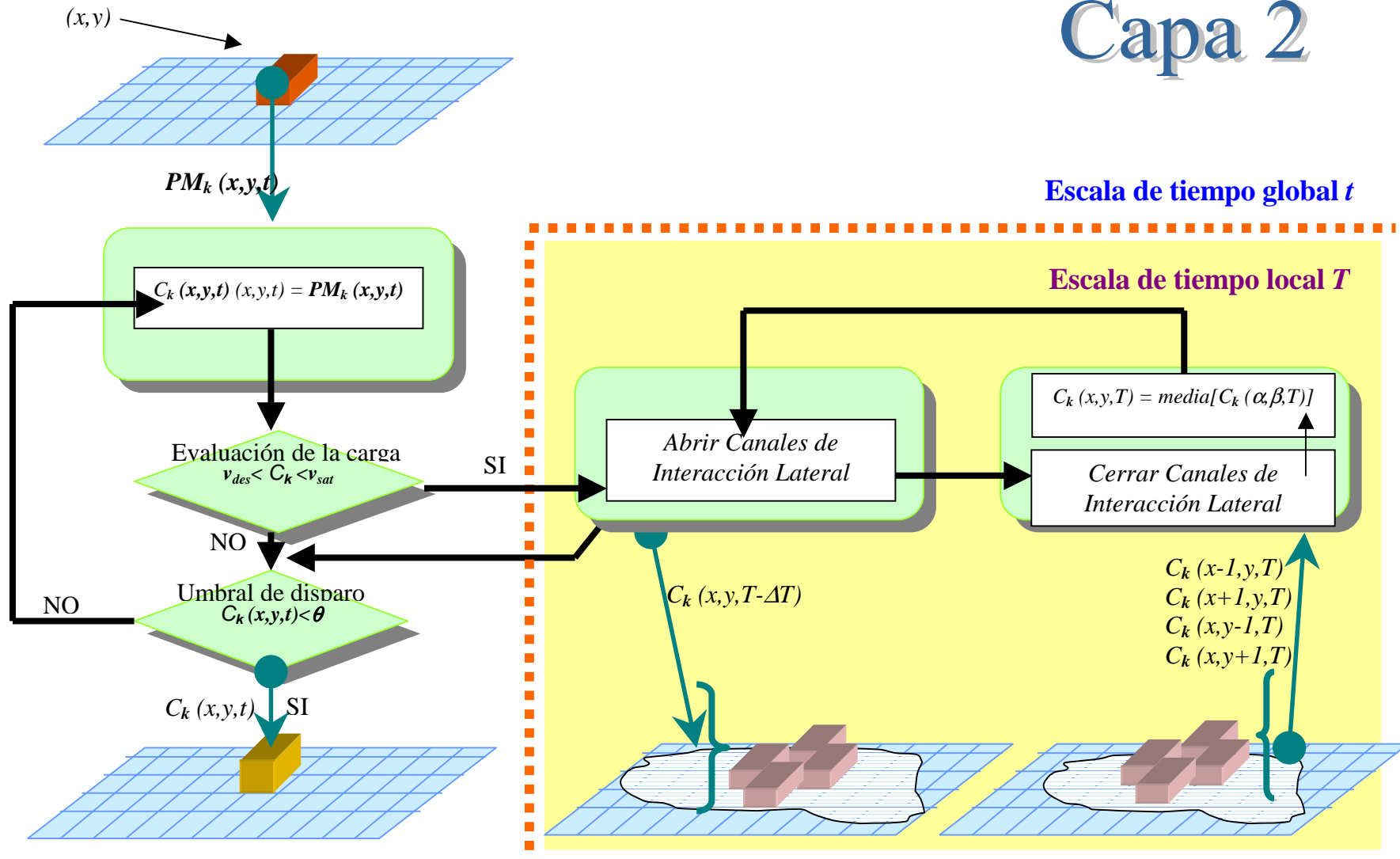
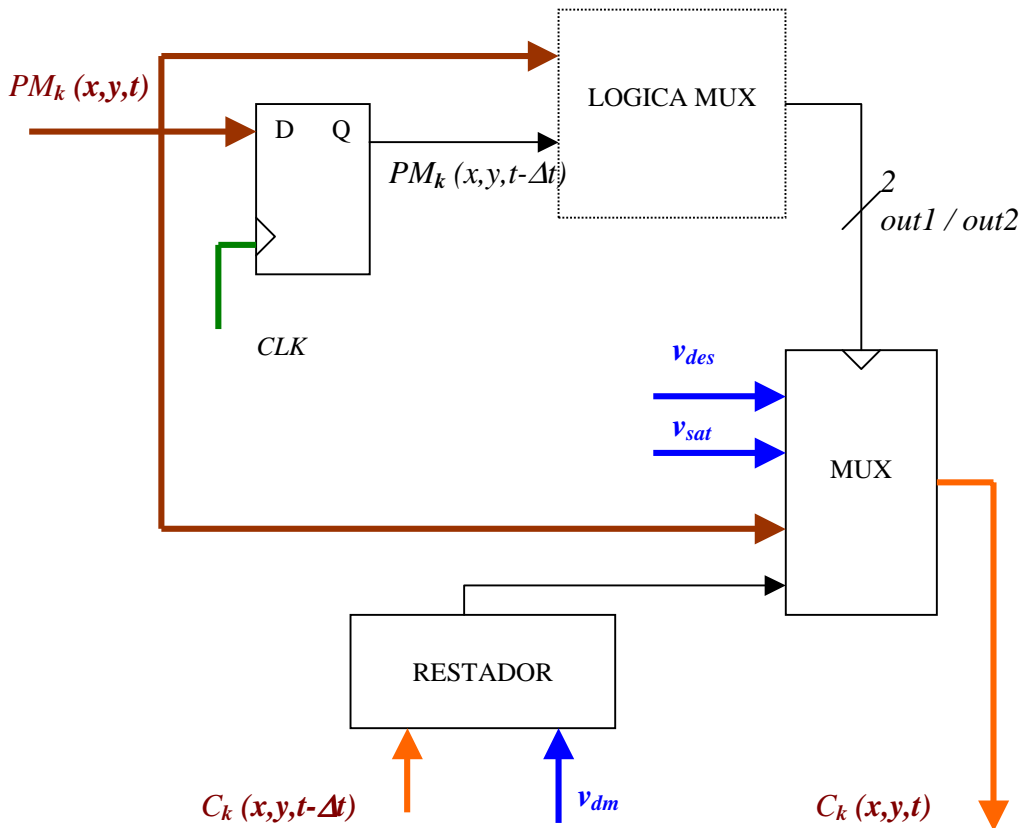


Figura 3.21. Esquema funcional del modelo en su capa 2

Se pasa a detallar cada uno de los pasos anteriores.

Acumulación espacial.

En el paso 1, la carga se iguala al valor de permanencia que proviene de la capa anterior.



LOGICA MUX:

$PM_k(x,y,t)$	$PM_k(x,y,t-\Delta t)$	out1	out2
0	0	1	0
0	1	1	0
1	0	1	0
1	1	1	0

Figura 3.22. Lógica de carga asociada a la capa 2 del modelo

Acumulación espacio-temporal.

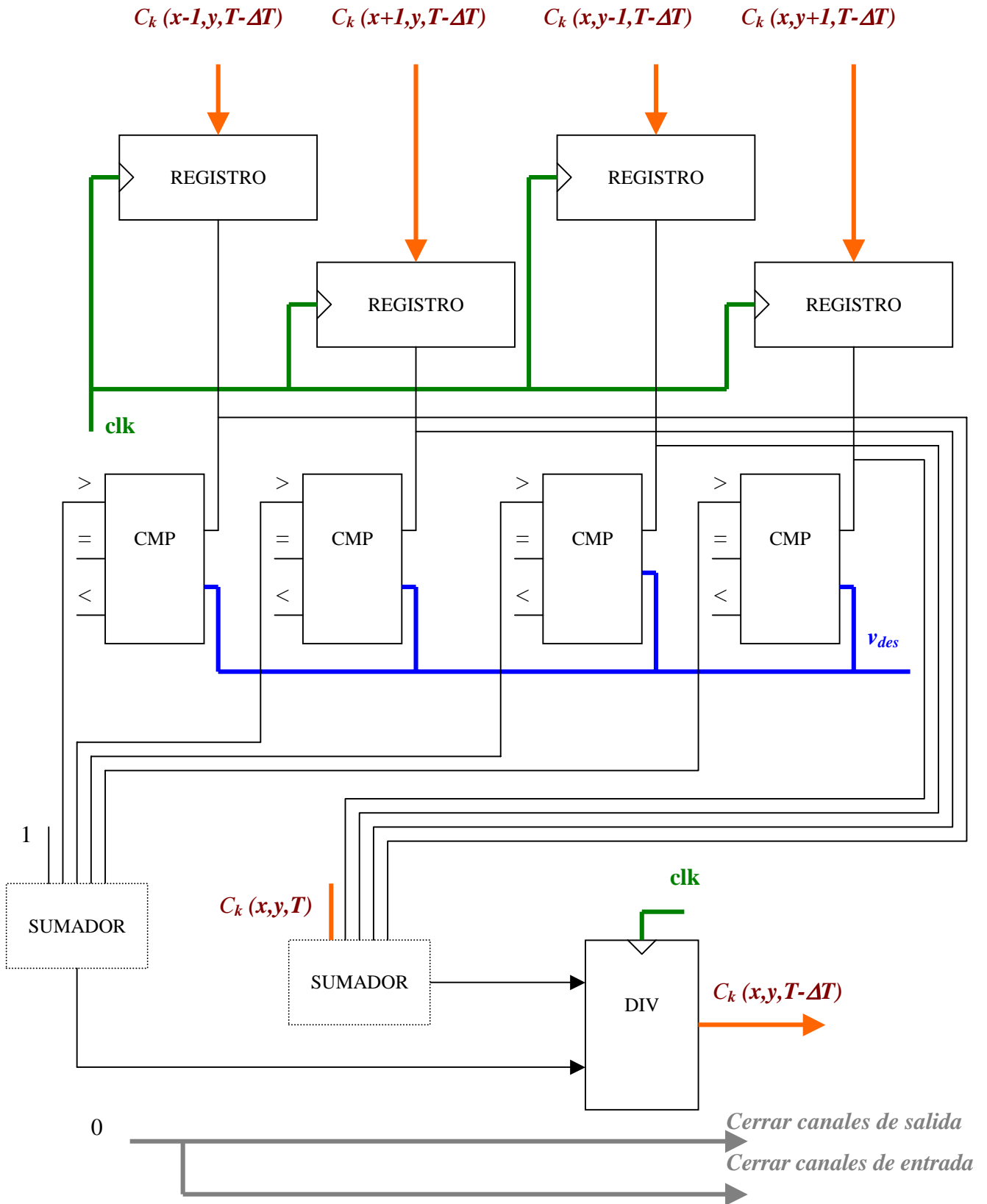


Figura 3.23. Lógica de carga asociada a la capa 2 del modelo

Los pasos 2 y 3 ocurren en la escala de tiempos $T \ll t$. En el paso 2, la información de carga es enviada a cada uno de los cuatro vecinos a través de los canales habilitados al efecto. En el paso 3, se trata de volver a calcular el nuevo valor de carga partir del valor anterior y de los valores de las cuatro vecinas de la estructura.

Distribución.

En el paso 4, una vez estabilizado el sistema, se emite un valor de carga igual a θ_{car} , si el valor de carga está por debajo del mismo, o bien el valor medio de carga obtenido por mecanismos laterales. De nuevo estamos en la escala de tiempos t .

Lo que nos lleva a la estructura general por cada pixel de la imagen y por cada banda de nivel de gris en la capa 2:

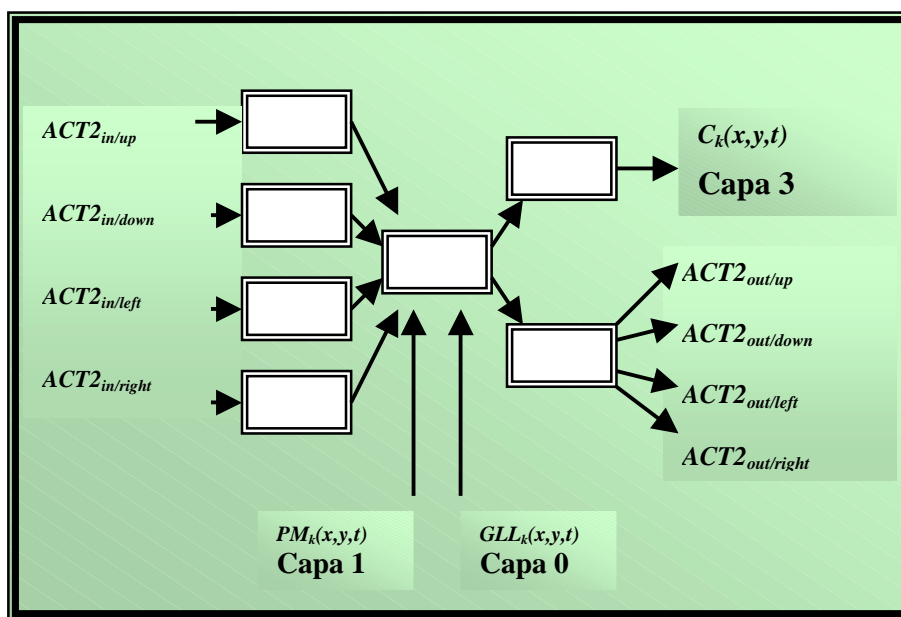


Figura 3.24. Capa 2. Subcapa k. Estructura modular para cada pixel de la imagen

3.2.5. Capa 3: Interacción lateral para la fusión de objetos en movimiento

Una vez conseguido mediante los mecanismos de computación cooperativa captar la atención sobre todo aquello que se está moviendo en alguna de las bandas de nivel de gris en la escena, y habiendo eliminado el movimiento debido al fono, cabe fijarse como nuevo objetivo el

distinguir entre los diferentes objetos que conforman las diferentes partes de las siluetas obtenidas por bandas de nivel de gris. Véase que no puede hablarse de una clasificación propiamente dicha que provenga de un aprendizaje supervisado anterior, sino que nos encontramos frente a una autoclasificación en base a las características que provienen de la capa 2. Dicho de otro modo, estamos en presencia de un aprendizaje no supervisado.

La discriminación de los objetos que forman las siluetas se consigue igualmente por mecanismos de cooperación lateral. De nuevo vamos a conectar las estructuras modulares de esta capa en forma de malla en la capa 3. Ahora ya no vamos a trabajar con subcapas, sino que toda la información de las n subcapas de la capa 2, acaban en una sola capa.

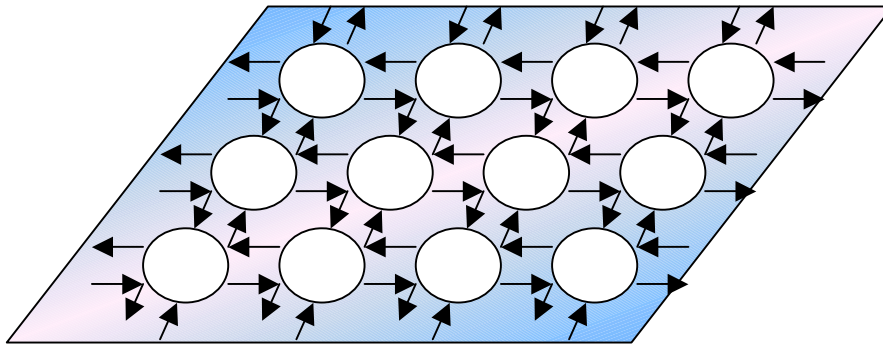


Figura 3.25. Capa 3. Conexiones laterales en forma de malla

En esta capa 3, se va a homogeneizar la carga de los valores de carga entre todos los píxeles que tengan valor de carga superior a un umbral mínimo θ_{car} y estén unidos físicamente entre sí.

Las conexiones laterales se denominan en esta capa $ACT3$ y tienen el siguiente significado:

$ACT3_{in/up}$: es la conexión lateral de entrada a la estructura modular que proviene de la estructura que se encuentra una posición más arriba en la malla

$ACT3_{in/down}$: es la conexión lateral de entrada a la estructura modular que proviene de la estructura que se encuentra una posición más abajo en la malla

$ACT3_{in/left}$: es la conexión lateral de entrada a la estructura modular que proviene de la estructura que se encuentra una posición más a la izquierda en la malla

$ACT3_{in/right}$: es la conexión lateral de entrada a la estructura modular que proviene de la estructura que se encuentra una posición más a la derecha en la malla

$ACT3_{out/up}$: es la conexión lateral de salida de la estructura modular que va hacia la estructura que se encuentra una posición más arriba en la malla

$ACT3_{out/down}$: es la conexión lateral de salida de la estructura modular que va hacia la estructura que se encuentra una posición más abajo en la malla

$ACT3_{out/left}$: es la conexión lateral de salida de la estructura modular que va hacia la estructura que se encuentra una posición más a la izquierda en la malla

$ACT3_{out/right}$: es la conexión lateral de salida de la estructura modular que va hacia la estructura que se encuentra una posición más a la derecha en la malla

Las conexiones de cada una de las estructuras modulares de este nivel pueden ser vistas tal como se presentan en la figura 3.26.

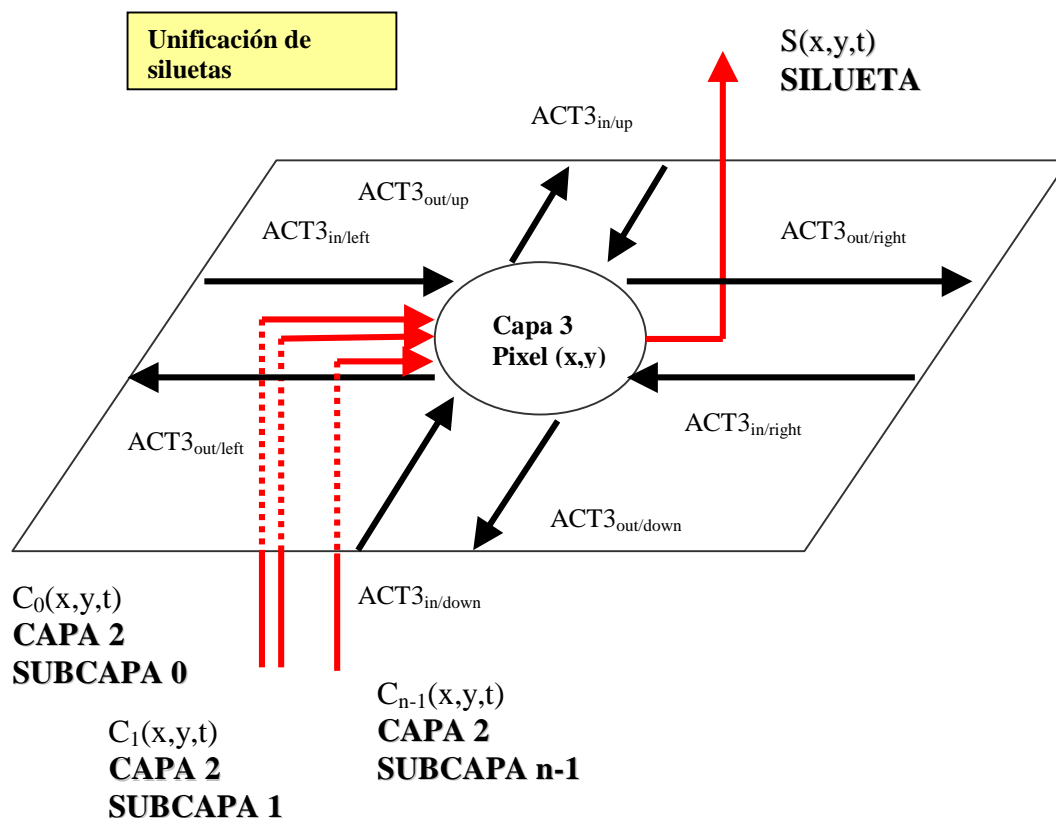


Figura 3.26. Capa 3. Conexiones externas

El comportamiento algorítmico de la estructura modular para cada uno de los píxeles de la imagen se describe igualmente a continuación.

// Capa 3. Pixel (i, j) .

// Mecanismos de interacción lateral para la obtención de objetos en movimiento.

// Algoritmo.

// ACUMULACION ESPACIAL.

// Espacio de tiempos t .

// Paso 1. Carga inicial desde la capa 2.

FOR (k =todas las bandas de nivel de gris) $S(x, y, t) = \max(C_k(x, y, t))$

// ACUMULACION ESPACIO-TEMPORAL.

// Espacio de tiempos $T \ll t$.

IF ($S(x, y, t) > \theta_{car}$)

 WHILE ($T < T_{max}$) DO

 // Paso 2. Envío de información lateral.

$$ACT3_{out/up} = ACT3_{out/down} = ACT3_{out/right} = ACT3_{out/left} = S(x, y, T)$$

 // Paso 3. Recepción de información lateral y obtención de la nueva carga.

$$C_{up} = \begin{cases} ACT3_{in/up}, & \text{si } ACT3_{in/up} > \theta_{car} \\ 0, & \text{en caso contrario} \end{cases}$$

$$C_{down} = \begin{cases} ACT3_{in/down}, & \text{si } ACT3_{in/down} > \theta_{car} \\ 0, & \text{en caso contrario} \end{cases}$$

$$C_{right} = \begin{cases} ACT3_{in/right}, & \text{si } ACT3_{in/right} > \theta_{car} \\ 0, & \text{en caso contrario} \end{cases}$$

$$C_{left} = \begin{cases} ACT3_{in/left}, & \text{si } ACT3_{in/left} > \theta_{car} \\ 0, & \text{en caso contrario} \end{cases}$$

$$S(x, y, T) = \text{media} (S(x, y, T - \Delta T) + C_{up} + C_{down} + C_{right} + C_{left})$$

// DISTRIBUCION.

// Espacio de tiempos t .

// Paso 4. Envío del valor de silueta.

$$S(x, y, t) = \begin{cases} S(x, y, t), & \text{si } S(x, y, t) > \theta_{obj} \\ \theta_{obj}, & \text{en caso contrario} \end{cases}$$

Acumulación espacial.

El módulo del paso 1 detecta, a partir la información que llega desde las n subcapas de la capa 2, el valor de carga de silueta inicial del pixel. En el paso 2, la estructura informa del valor de silueta a las cuatro vecinas.

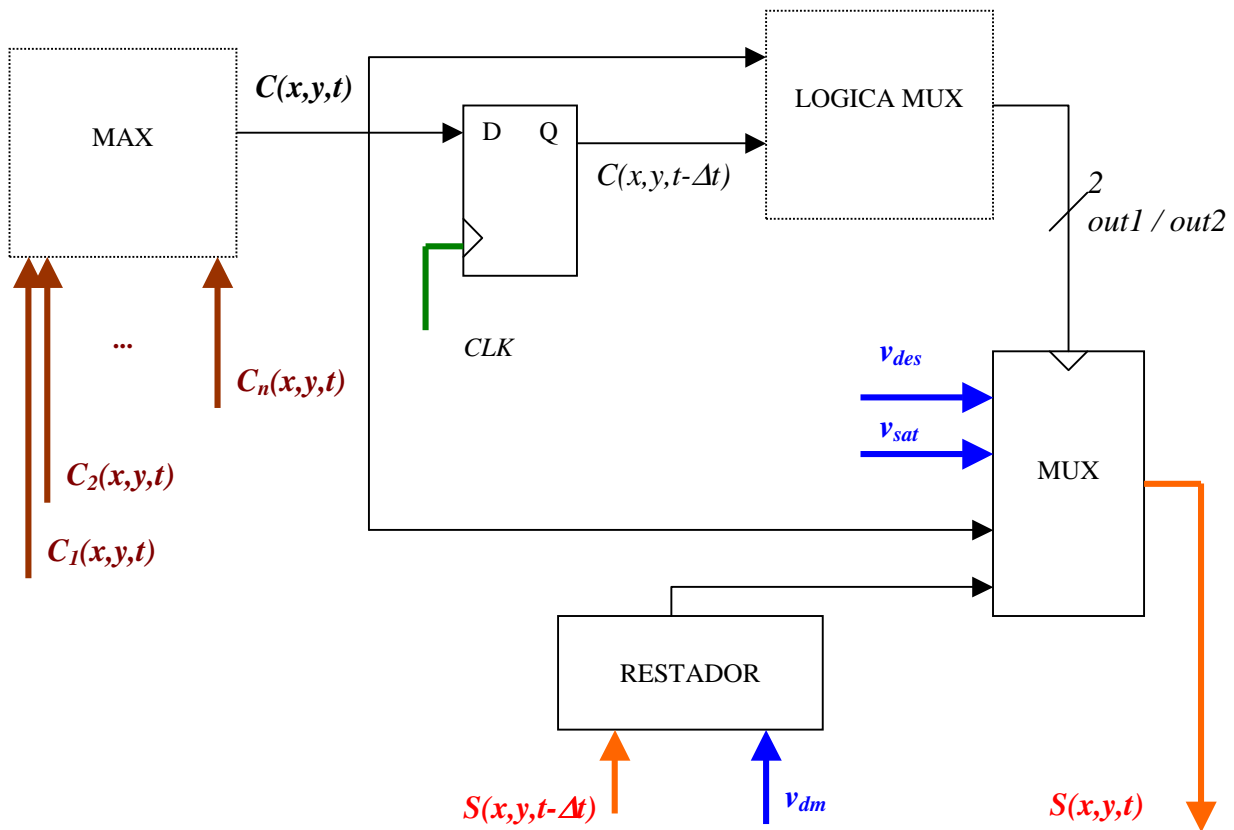


Figura 3.27. Lógica de carga asociada a la capa 3 del modelo

Capa 3

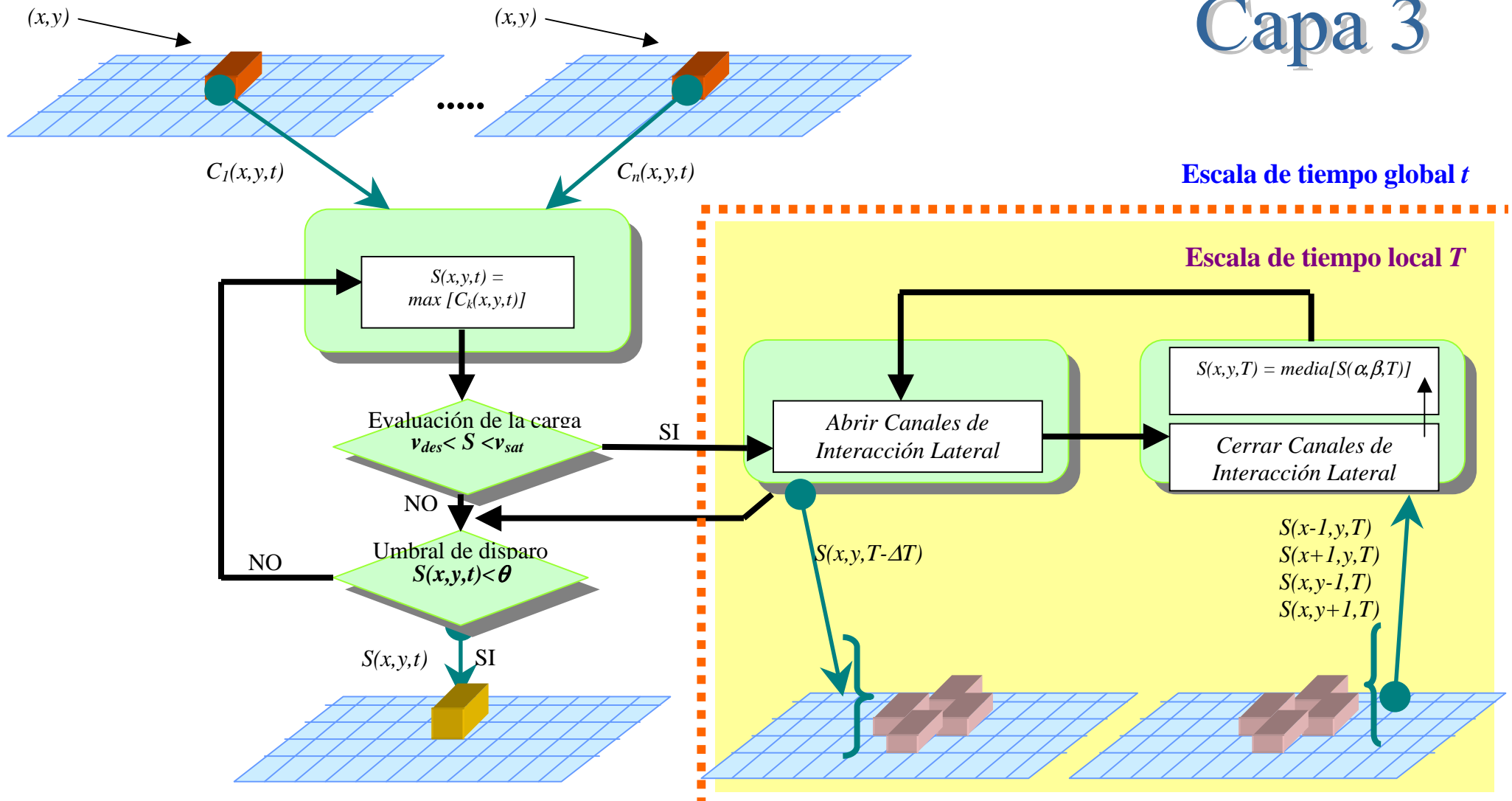


Figura 3.28. Esquema funcional del modelo en su capa 3

Acumulación espacio-temporal.

En el paso 3, se vuelve a calcular la media del valor de silueta a partir de la información propia y de las cuatro vecinas. Por lo tanto, la lógica de carga lateral es idéntica a la de la capa 2.

Distribución.

En el paso 4, se emite silueta en un punto concreto con el valor medio de la silueta para el objeto.

Todo ello nos lleva a la estructura general por cada pixel de la imagen en la capa 3:

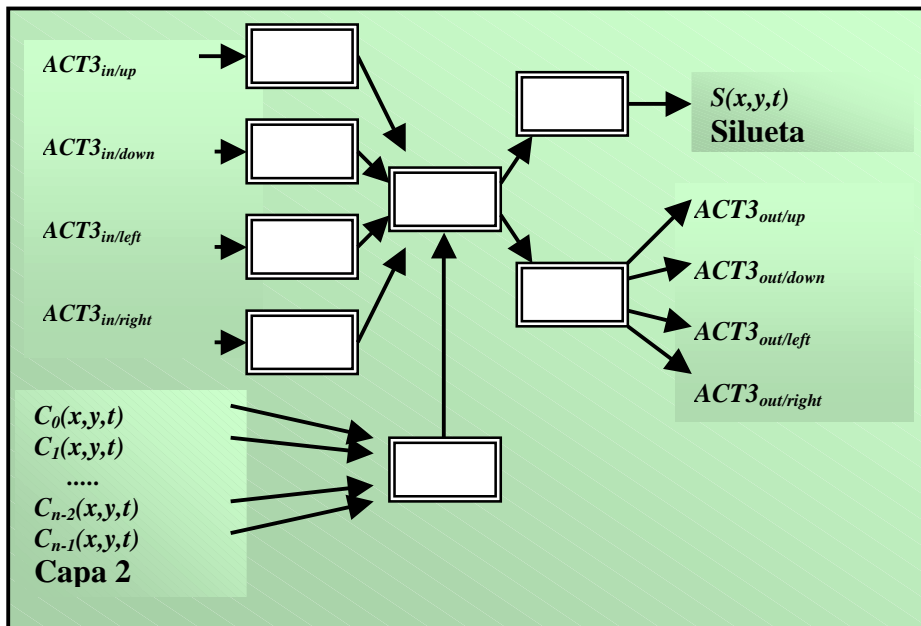


Figura 3.29. Capa 3. Estructura modular para cada pixel de la imagen

3.2.6. Capa 4: Aplicación

La capa 4 es una capa dependiente de cada una de las aplicaciones de alto nivel que se desea implementar aprovechando la generalidad del modelo de interacción lateral en computación acumulativa. De ahí que no sea hasta más adelante en la implementación cuando se hable de esta capa.

3.3. Influencia de los parámetros básicos del modelo de interacción lateral en computación acumulativa

3.3.1. Introducción

Se pasa ahora a comentar cómo debe entenderse la influencia de todos los parámetros del modelo de interacción lateral en computación acumulativa vistos en el método. Y ello se va a realizar usando los resultados de aplicar una implementación del modelo de interacción lateral a una secuencia de imágenes tomada en un laboratorio de investigación. En esta secuencia se mueven libremente tres personas.

Se va a llevar un orden claro y definido en la explicación de la influencia de los parámetros. En concreto, se hará en función de la capa en la que actúa básicamente cada uno de los parámetros.

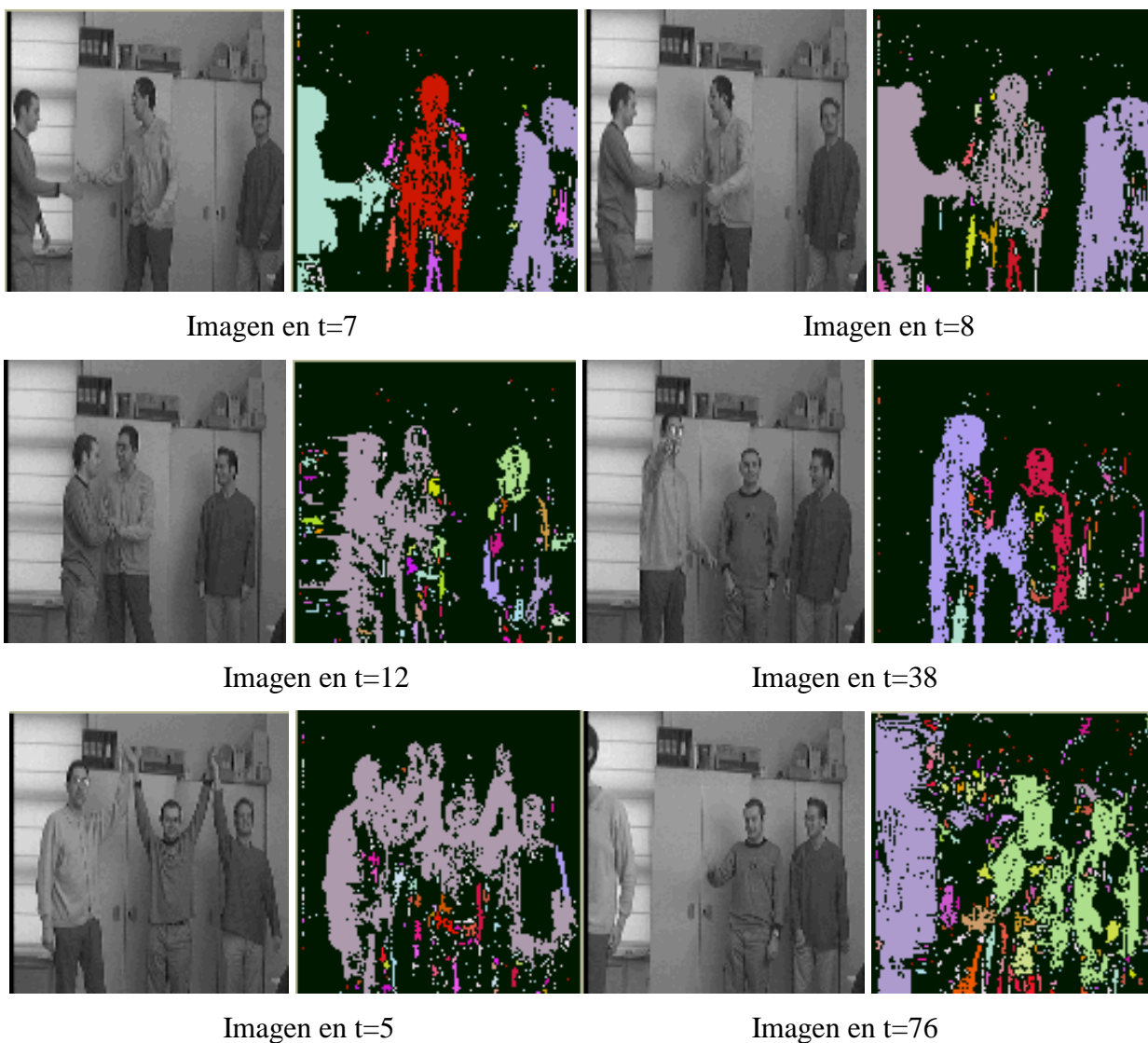
Estos parámetros, así como la capa en la que se detallan, serían:

- *Número de Bandas de Nivel de Gris (n)* - Capa 0
- *Valor de Descarga por Detección de Movimiento (v_{dm})* - Capa 1
- *Valor de Recarga por Vecindad (v_{rv})* - Capa 1
- *Número de Vecinos para la Interacción Lateral (l_{per})* - Capa 1
- *Valor Mínimo de Mancha por Banda de Nivel de Gris (θ_{per})* - Capa 1
- *Número de Vecinos para el Reparto de Carga por Bandas de Nivel de Gris (l_{car})* - Capa 2
- *Valor Mínimo de Mancha para la Fusión en Objetos (θ_{car})* - Capa 2
- *Número de Vecinos para el Reparto de Carga por Objeto (l_{obj})* - Capa 3
- *Valor Mínimo de Detección de Siluetas (θ_{obj})* - Capa 3

Se comienza, no obstante, con una disertación acerca de las oclusiones. También se incluye un apartado dedicado a la umbralización, de interés primordial para los resultados del modelo.

3.3.2. Oclusiones

El siguiente ejemplo muestra el resultado de la obtención de las siluetas de los objetos deformables móviles en una escena a partir de una implementación del modelo. Se han entresacado aquellas imágenes de interés en las que se aprecia el efecto de las oclusiones.



Ejemplo 3.1. Imágenes de ensayo para la explicación de las oclusiones

En nuestro caso, las oclusiones han de entenderse como la unión temporal de las diferentes formas en una sola silueta. Así, si echamos un vistazo a las imágenes en los tiempos $t=7$ hasta $t=12$, podemos ver cómo:

- (a) en $t=7$ las tres personas presentes en la escena aparecen como tres siluetas claramente diferenciadas; en efecto, están separadas entre sí
- (b) en $t=8$ ha ocurrido el contacto entre dos de las personas a partir de haber estrechado sus manos; como consecuencia, estas dos personas se fusionan en una sola silueta
- (c) en $t=12$ las dos personas superpuestas en la imagen siguen apareciendo como una sola silueta.

Véase, ahora, cómo unos instantes más adelante, una de las personas ha pasado a saludar a la tercera. La situación ha cambiado ahora, y son estas dos personas las que unen sus siluetas, mientras que la primera ya aparece autónomamente.

Las instantáneas entre los tiempos $t=38$ y $t=51$ también merecen nuestro interés. Pasamos aquí de una situación en la que las tres personas están separadas entre sí y aparecen como tres siluetas diferentes, y una situación en la que los tres están cogidos de las manos, fundiéndose en una sola silueta.

La imagen en $t=76$ muestra otra situación característica.

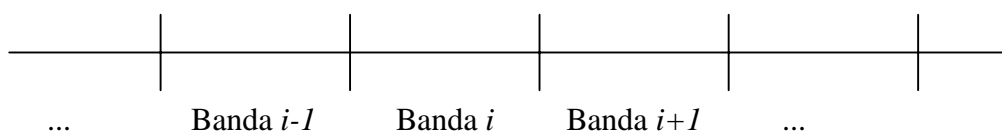
3.3.3. Umbralización Global Simple vs. Umbralización Uniforme con Solapamiento

Aplicando los algoritmos que se han contado en la Capa 0. Segmentación por bandas de nivel de gris, se ha llegado a apreciar en numerosos casos de aplicación del modelo a imágenes reales más ruido del deseado. Por ello, hablamos en este apartado de dos tipos de segmentación, la *Umbralización Global Simple* y la *Umbralización Uniforme con Solapamiento*.

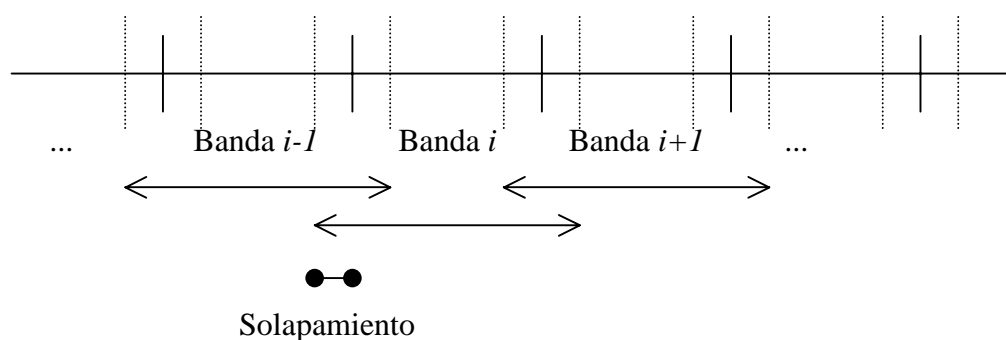
El primer tipo de segmentación, la *Umbralización Global Simple*, es el que divide los 256 posibles niveles de gris en n bandas de nivel de gris equidistantes mediante la fórmula vista hasta ahora. Es decir,

$$GLL_k(x, y, t) = \begin{cases} 1, & \text{si } GL(x, y, t) \in [(256/n)k, (256/n)(k+1)[\\ 0, & \text{en caso contrario} \end{cases}$$

De este modo, tendríamos



El segundo tipo de segmentación, la *Umbralización Uniforme con Solapamiento*, establece un margen de confianza necesario para decidir un cambio de una banda de nivel de gris a otra vecina al pasar de una imagen a la siguiente. Este margen se ofrece como un *% de Solapamiento* entre bandas. Tendríamos ahora:



Se presenta en el siguiente ejemplo el resultado final de aplicar una implementación del modelo de interacción lateral en computación acumulativa a la misma secuencia de imágenes (columna a del ejemplo).

La columna segunda del ejemplo muestra el resultado habiendo utilizado la *Umbralización Global Simple* con 16 bandas de nivel de gris. En cuanto a la columna tercera, ésta se ha obtenido como resultado de aplicar el modelo utilizando la *Umbralización Uniforme con Solapamiento*. En este caso, se ha trabajado con 16 bandas de nivel de gris, igualmente, pero con un porcentaje de solapamiento del 20%.

Puede concluirse fácilmente, observando las imágenes resultado, que la *Umbralización Uniforme con Solapamiento* elimina mucho del ruido alrededor de las siluetas que se desea obtener. Esto es algo que no ocurre al usar la *Umbralización Global Simple*.

Se observa también (ver imágenes en $t=38$ y $t=76$) que es más fácil segregar las siluetas de los diferentes objetos deformables en movimiento utilizando la *Umbralización Uniforme con Solapamiento*.

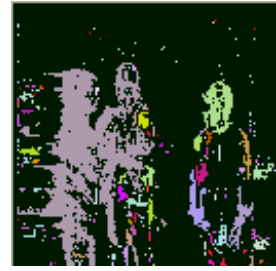


Imagen en t=12

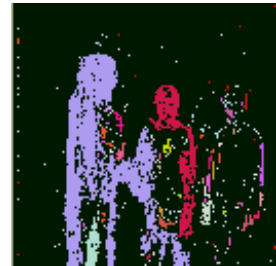


Imagen en t=38

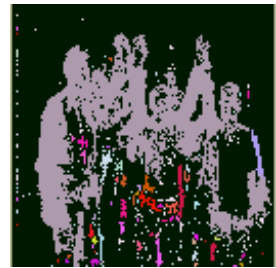


Imagen en t=51

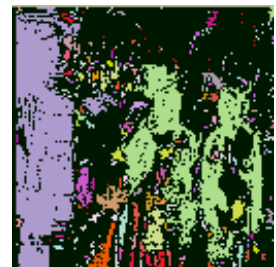


Imagen en t=76

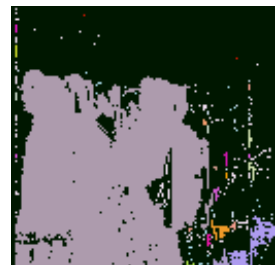
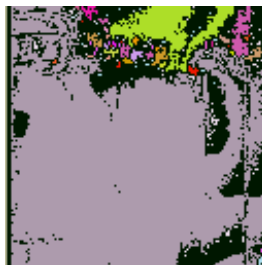


Imagen en t=88

Ejemplo 3.2. Imágenes de ensayo para la explicación de los dos tipos de segmentación

3.3.4. Influencia de los parámetros de la capa 0

3.3.4.1. Influencia del parámetro Número de Bandas de Nivel de Gris

Para explicar la influencia del parámetro *Número de Bandas de Nivel de Gris (n)* se va a trabajar con tres valores posibles de dicho parámetro y se van a ofrecer los resultados obtenidos tras finalizar la capa 3 del modelo.

Los valores con los que se ha trabajado son:

Segmentación por Bandas de Nivel de Gris (Capa 0) – Caso A	
<i>Número de Bandas de Nivel de Gris (n)</i>	4
Segmentación por Bandas de Nivel de Gris (Capa 0) – Caso B	
<i>Número de Bandas de Nivel de Gris (n)</i>	8
Segmentación por Bandas de Nivel de Gris (Capa 0) – Caso C	
<i>Número de Bandas de Nivel de Gris (n)</i>	32

Tabla 3.1. Juego de ensayo para la explicación del parámetro

Número de Bandas de Nivel de Gris

Los resultados aparecen en forma de imágenes, donde:

- en la columna primera se ofrecen las imágenes de entrada de la secuencia
- en la columna segunda se ofrecen los resultados de aplicar el caso A
- en la columna tercera se ofrecen los resultados de aplicar el caso B
- en la columna cuarta se ofrecen los resultados de aplicar el caso C

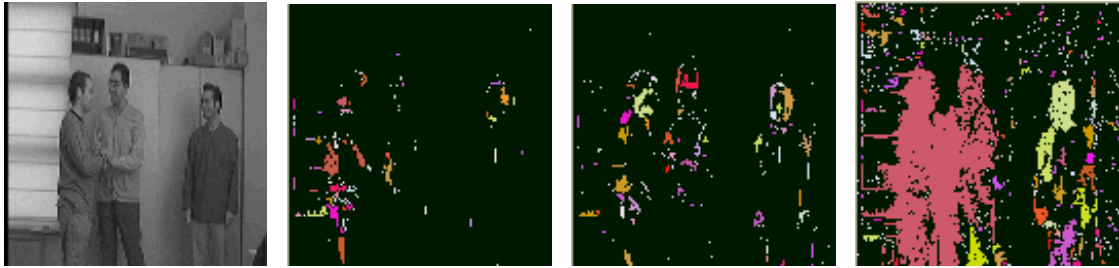


Imagen en $t=12$

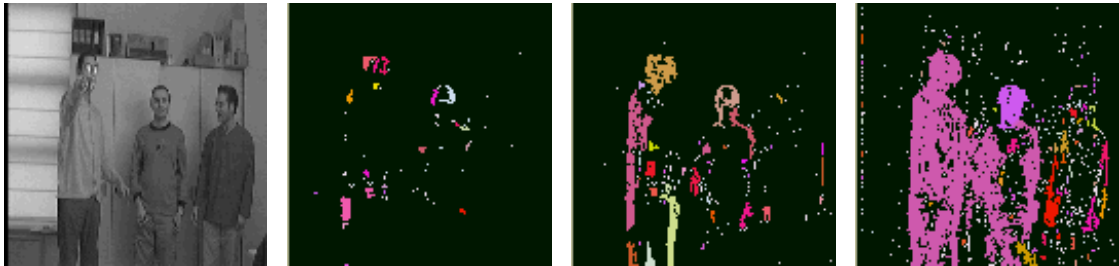


Imagen en $t=38$

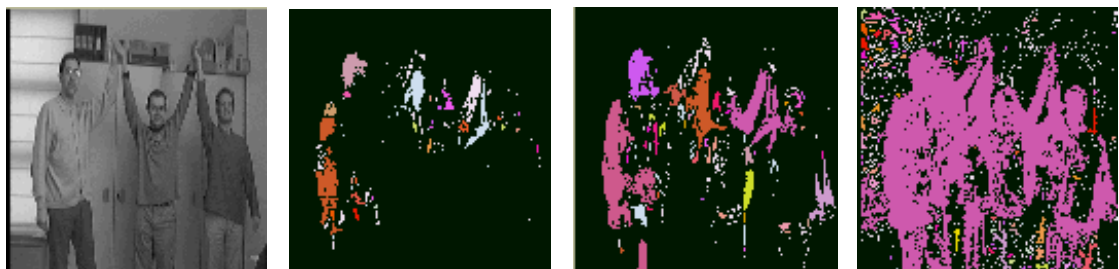


Imagen en $t=51$

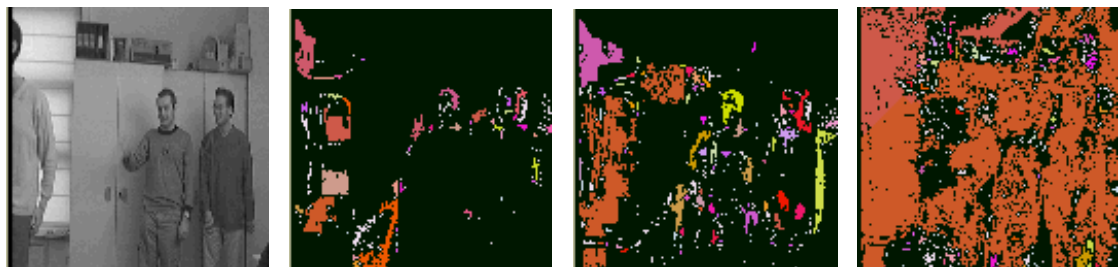


Imagen en $t=76$

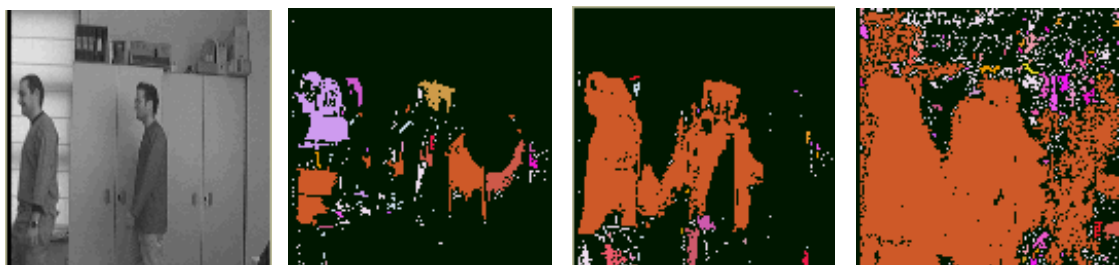


Imagen en $t=88$

Ejemplo 3.3. Imágenes de ensayo para la explicación de la influencia del parámetro *Número de Bandas de Nivel de Gris*.

3.3.5. Influencia de los parámetros de la capa 1

3.3.5.1. Influencia del parámetro Valor de Descarga por Detección de Movimiento

Para explicar la influencia del parámetro *Valor de Descarga por Detección de Movimiento* (v_{dm}) se va a trabajar con tres valores posibles de dicho parámetro y se van a ofrecer los resultados obtenidos a la salida de la capa 1 del modelo.

Segmentación por Bandas de Nivel de Gris (Capa 0) – Caso A	
<i>Número de Bandas de Nivel de Gris (n)</i>	8
Interacción Lateral para la Computación Acumulativa (Capa 1) – Caso A	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	255
<i>Valor de Recarga por Vecindad (v_{rv})</i>	0
Interacción Lateral para la Computación Acumulativa (Capa 1) – Caso B	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	127
<i>Valor de Recarga por Vecindad (v_{rv})</i>	0
Interacción Lateral para la Computación Acumulativa (Capa 1) – Caso C	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	15
<i>Valor de Recarga por Vecindad (v_{rv})</i>	0

Tabla 3.2. Juego de ensayo para la explicación del parámetro

Valor de Descarga por Detección de Movimiento

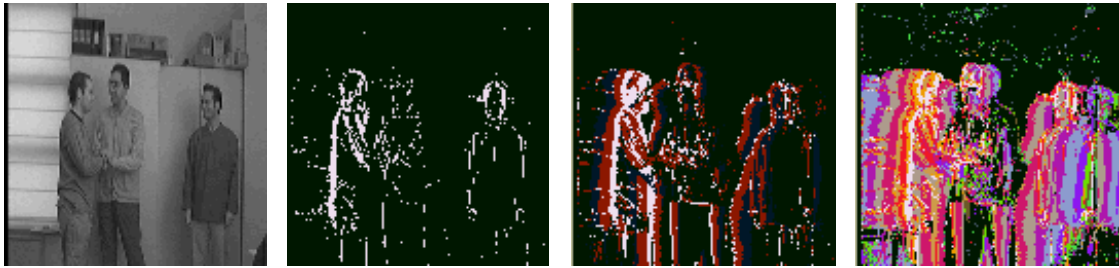


Imagen en t=12

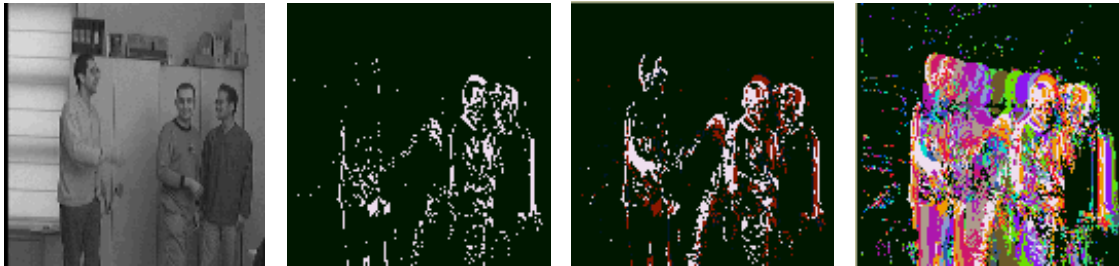


Imagen en t=30

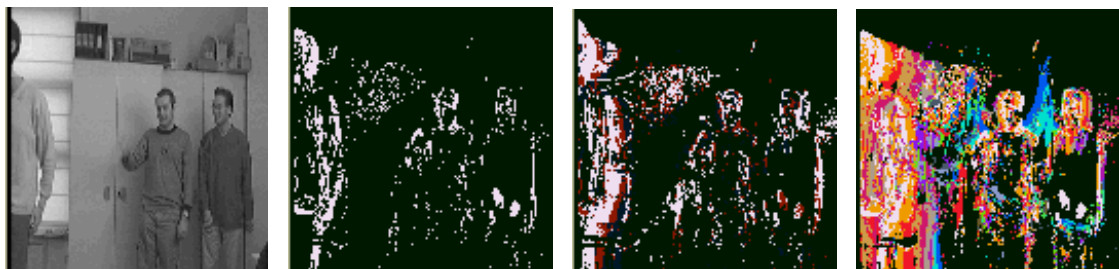


Imagen en t=76

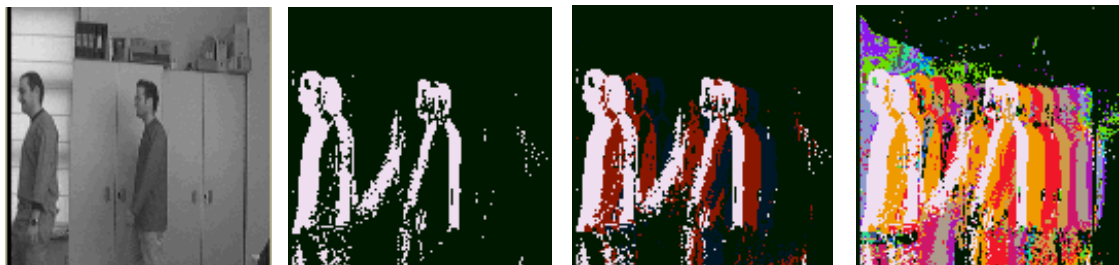


Imagen en t=88

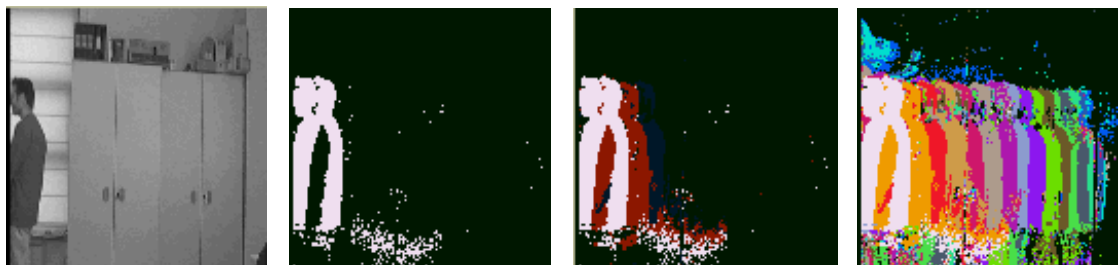


Imagen en t=97

Ejemplo 3.4. Imágenes de ensayo para la explicación de la influencia del parámetro *Valor de Descarga por Detección de Movimiento*

En la primera columna, como siempre, se ofrecen las imágenes de entrada a la implementación del modelo de interacción lateral en computación acumulativa. Las sucesivas columnas muestran el resultado de aplicar el modelo en los casos A, B y C, respectivamente.

En la columna segunda vemos únicamente, como resultado, y en color blanco los píxeles de la imagen que acaban de cambiar de una imagen a otra en su banda de nivel de nivel de gris. Bajando el *Valor de Descarga por Detección de Movimiento*, vamos obteniendo en las imágenes resultado la estela del movimiento en las últimas imágenes. Así, la columna tercera ofrece la historia de las tres últimas imágenes, a saber, en blanco la diferencia entre las imágenes en t y $t-1$, y en rojo la diferencia entre las imágenes $t-1$ y $t-2$. La columna cuarta muestra más claramente aún la historia de las últimas imágenes. En efecto, mediante un *Valor de Descarga por Detección de Movimiento* pequeño, obtenemos más información de la historia del movimiento a través de la estela ofrecida.

3.3.5.2. Influencia del parámetro Valor de Recarga por Vecindad

Para explicar la influencia del parámetro *Valor de Recarga por Vecindad* (v_r) se va a trabajar igualmente con tres valores posibles de dicho parámetro y se van a ofrecer los resultados obtenidos a la salida de la capa 1 del modelo.

En la primera columna del ejemplo ofrecido, se muestran las imágenes de entrada a la implementación del modelo de interacción lateral en computación acumulativa. Las sucesivas columnas muestran el resultado de aplicar el modelo en los casos A, B y C, respectivamente. Estos casos son los que se pueden consultar en la tabla 3.3.

Tal como se han elegido los parámetros en el caso A, y habiendo decidido no recargar las memorias, es decir, usar un *Valor de Recarga por Vecindad* igual a 0, podemos apreciar tres niveles de carga diferenciados: (a) en negro, equivalente al nivel de gris 0 o *Valor Mínimo de Permanencia* (v_{des}), vemos el fondo, donde no se ha detectado movimiento, (b) en rojo, es decir, para un nivel de gris alrededor de 128, el movimiento detectado entre los instantes $t-2$ y $t-1$, a grosso modo, y, (c) en blanco, es decir con un valor de nivel de gris de 255 o *Valor Máximo de Permanencia* (v_{sat}), el movimiento detectado entre los instantes $t-1$ y t .

Segmentación por Bandas de Nivel de Gris (Capa 0) – Caso A	
<i>Número de Bandas de Nivel de Gris (n)</i>	8
Interacción Lateral para la Computación Acumulativa (Capa 1) – Caso A	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	127
<i>Valor de Recarga por Vecindad (v_{rv})</i>	0
<i>Número de Vecinos para la Interacción Lateral (l_{per})</i>	512
Interacción Lateral para la Computación Acumulativa (Capa 1) – Caso B	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	127
<i>Valor de Recarga por Vecindad (v_{rv})</i>	63
<i>Número de Vecinos para la Interacción Lateral (l_{per})</i>	512
Interacción Lateral para la Computación Acumulativa (Capa 1) – Caso C	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	127
<i>Valor de Recarga por Vecindad (v_{rv})</i>	127
<i>Número de Vecinos para la Interacción Lateral (l_{per})</i>	512

Tabla 3.3. Juego de ensayo para la explicación del parámetro*Valor de Recarga por Vecindad*

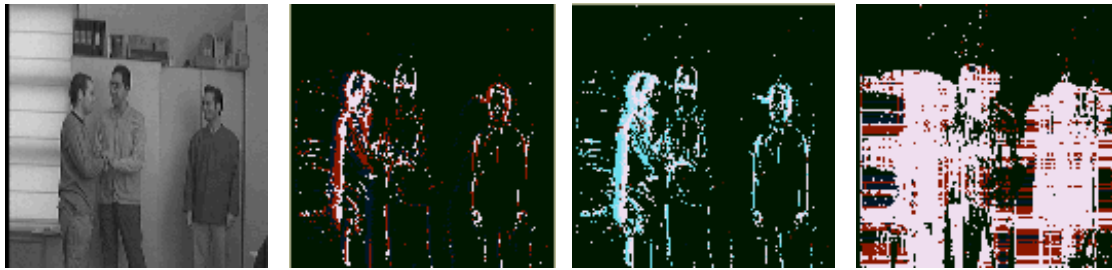


Imagen en t=12

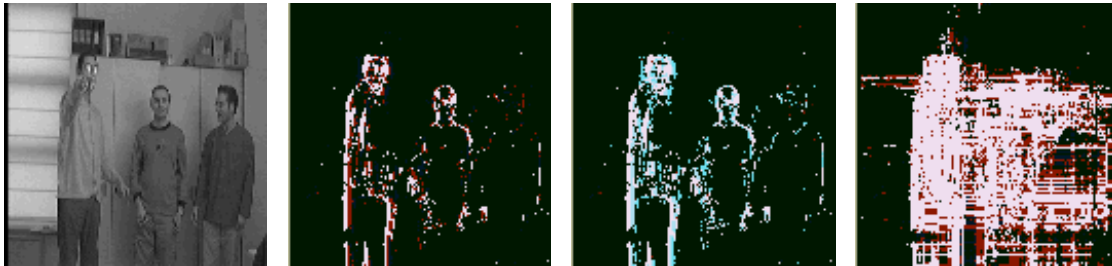


Imagen en t=38

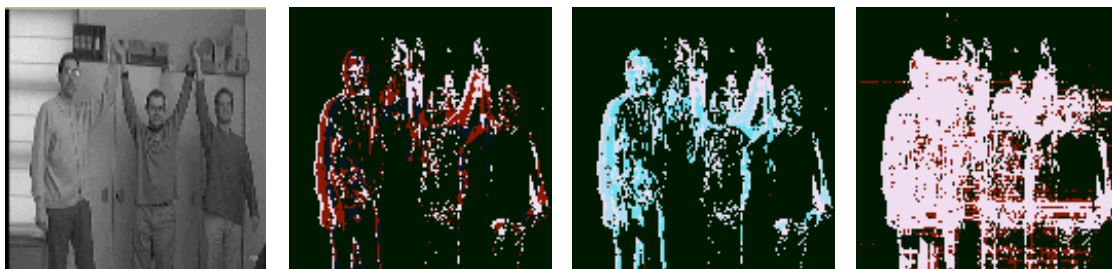


Imagen en t=51

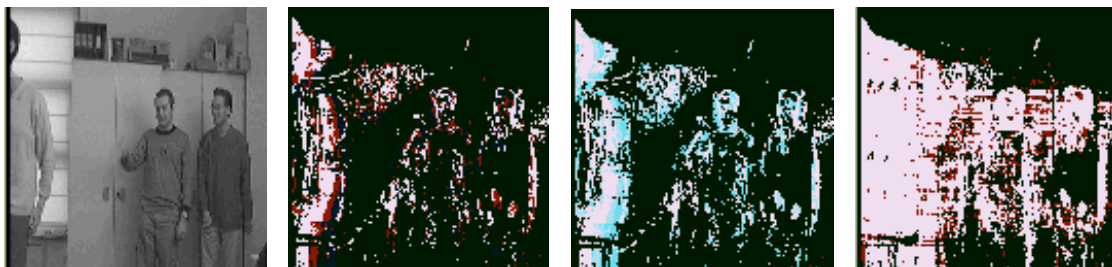


Imagen en t=76

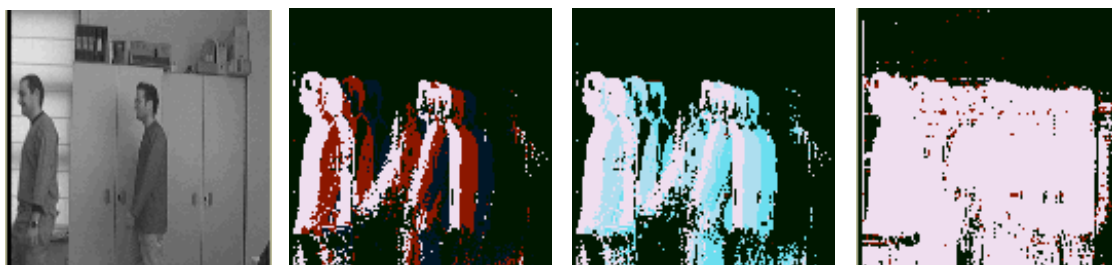


Imagen en t=88

Ejemplo 3.5. Imágenes de ensayo para la explicación de la influencia del parámetro *Valor de Recarga por Vecindad*

La columna tercera muestra los resultados de aplicar el modelo tal como se describe en el caso B, tomando un *Valor de Recarga por Vecindad* (v_{rv}) distinto de 0, pero inferior al *Valor de Descarga por Detección de Movimiento* (v_{dm}).

En este caso se aprecia un aumento de carga en aquellos pixeles que en la columna segunda estaban en rojo (valor suficiente de carga para poder ser recargado), pasando ahora en falso color a celeste. Nótese cómo la recarga tiene como efecto secundario el recuperar parte de la estela de la historia del movimiento. De ahí que aparezcan dos tonalidades de celeste debidas a dos niveles de carga diferentes.

El caso C presenta el ejemplo más extremo de recarga de todo el *Valor de Descarga por Detección de Movimiento* (v_{dm}) habido entre las parejas de imágenes. El resultado es el de grandes zonas de pixeles cargadas al máximo *Valor Máximo de Permanencia* (v_{sat}).

3.3.5.3. Influencia del parámetro Número de Vecinos para la Interacción Lateral

Para explicar la influencia del parámetro *Número de Vecinos para la Interacción Lateral* (l_{per}) se va a ofrecer un ejemplo con dos casos diferentes, denominados caso A y caso B, y reflejados en la tabla 3.4. Se ofrecen igualmente los resultados en forma de imágenes de la salida de la capa 1 del modelo.

En general, se puede afirmar que el parámetro *Número de Vecinos para la Interacción Lateral* (l_{per}) está concebido para delimitar el alcance o zona de influencia de la interacción lateral. Lógicamente, cuanto mayor sea este radio de acción, mayor va a ser la zona afectada por la interacción lateral proveniente de los pixeles con carga máxima. Por lo tanto, a mayor *Número de Vecinos para la Interacción Lateral* (l_{per}) más claramente definidas aparecerán las distintas siluetas de los objetos deformables en movimiento en la escena bajo estudio. Así, si se desea obtener siluetas claras, habrá que buscar un valor del *Número de Vecinos para la Interacción Lateral* aceptablemente alto.

Por lo contrario, si lo que se desea es influir sobre pequeñas zonas alrededor de los pixeles de máximo valor de carga, se elegirá un *Número de Vecinos para la Interacción Lateral* sensiblemente más pequeño. Este modo de operación permite captar más detalles de los objetos deformables que un valor más elevado más dirigido al bulto.

Segmentación por Bandas de Nivel de Gris (Capa 0) – Caso A	
<i>Número de Bandas de Nivel de Gris (n)</i>	8
Interacción Lateral para la Computación Acumulativa (Capa 1) – Caso A	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	63
<i>Valor de Recarga por Vecindad (v_{rv})</i>	15
<i>Número de Vecinos para la Interacción Lateral (l_{per})</i>	32
Interacción Lateral para la Computación Acumulativa (Capa 1) – Caso B	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	63
<i>Valor de Recarga por Vecindad (v_{rv})</i>	15
<i>Número de Vecinos para la Interacción Lateral (l_{per})</i>	4

Tabla 3.4. Juego de ensayo para la explicación del parámetro*Número de Vecinos para la Interacción Lateral*

Todas estas directrices se ven reflejadas en el ejemplo anexo. La columna tercera se obtiene a partir de un *Número de Vecinos para la Interacción Lateral* muy pequeño. Véase (fundamentalmente en las imágenes resultado del instante $t=88$), cómo es posible apreciar muchos más detalles en la silueta de los objetos en movimiento. Cada paso se descompone en más zonas definidas por más colores.

El fotograma en $t=38$ también merece nuestra atención. En una secuencia con poco movimiento, como es el caso, antes del instante estudiado, un *Número de Vecinos para la Interacción Lateral* muy pequeño permite acercarse más a los contornos de los objetos en movimiento que a sus siluetas.

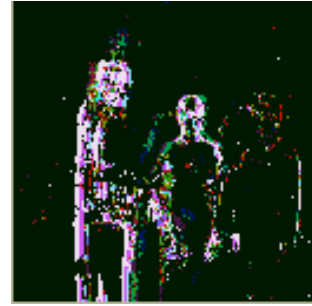
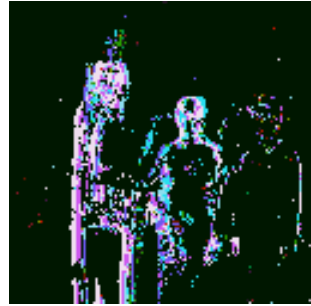


Imagen en t=38



Imagen en t=51



Imagen en t=76



Imagen en t=88

Ejemplo 3.6. Imágenes de ensayo para la explicación de la influencia del parámetro *Número de Vecinos para la Interacción Lateral*

3.3.5.4. Influencia del parámetro Valor Mínimo de Mancha por Banda de Nivel de Gris

Para explicar la influencia del parámetro *Valor Mínimo de Mancha por Banda de Nivel de Gris* (θ_{per}) se va a trabajar con dos de los valores posibles de dicho parámetro y se van a ofrecer los resultados obtenidos a la salida de la capa 2 del modelo.

Segmentación por Bandas de Nivel de Gris (Capa 0) – Caso A	
<i>Número de Bandas de Nivel de Gris (n)</i>	8
Interacción Lateral para la Computación Acumulativa (Capa 1) – Caso A	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	32
<i>Valor de Recarga por Vecindad (v_{rv})</i>	0
<i>Número de Vecinos para la Interacción Lateral (l_{per})</i>	
<i>Valor Mínimo de Mancha por Banda de Nivel de Gris (θ_{per})</i>	200
Interacción Lateral para la Computación Acumulativa (Capa 1) – Caso A	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	32
<i>Valor de Recarga por Vecindad (v_{rv})</i>	0
<i>Número de Vecinos para la Interacción Lateral (l_{per})</i>	
<i>Valor Mínimo de Mancha por Banda de Nivel de Gris (θ_{per})</i>	120

Tabla 3.5. Juego de ensayo para la explicación del parámetro *Valor Mínimo de Mancha por Banda de Nivel de Gris*

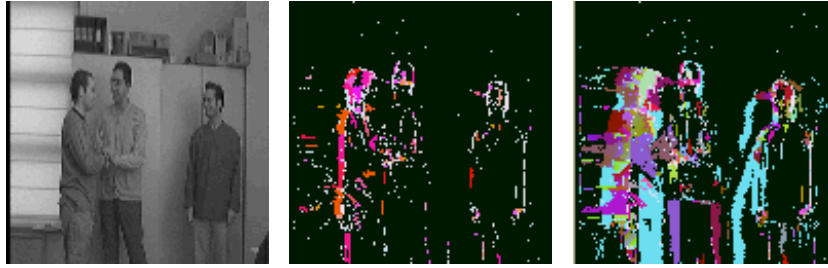


Imagen en t=12



Imagen en t=30

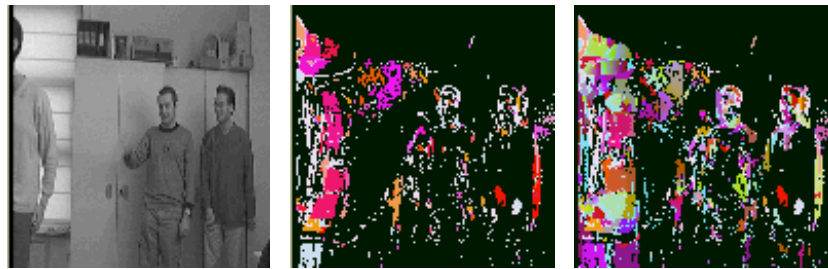


Imagen en t=76

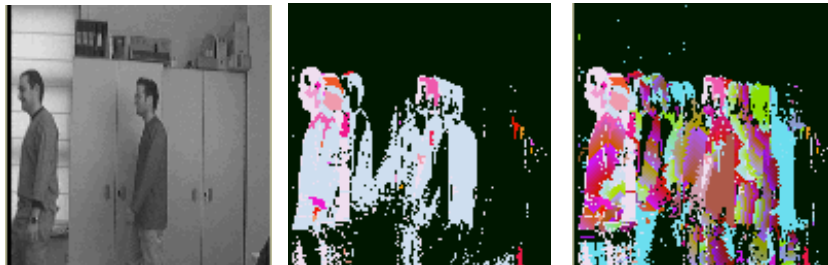


Imagen en t=88

Ejemplo 3.7. Imágenes de ensayo para la explicación de la influencia del parámetro *Valor Mínimo de Mancha por Banda de Nivel de Gris*

Como se puede apreciar fácilmente, el parámetro *Valor Mínimo de Mancha por Banda de Nivel de Gris* (θ_{per}) es un valor umbral que va a limitar el modelo de interacción lateral en computación acumulativa en su capa 1 en cuanto a la carga de aquellos pixeles de imagen con valor de carga por debajo de θ_{per} .

Un *Valor Mínimo de Mancha por Banda de Nivel de Gris* (θ_{per}) más alto discrimina más el número de píxeles a tratar a este nivel.

3.3.6. Influencia de los parámetros de la capa 2

3.3.6.1. Influencia del parámetro Número de Vecinos para el Reparto de carga por Bandas de Nivel de Gris

Para explicar la influencia del parámetro *Número de Vecinos para el Reparto de Carga por Bandas de Nivel de Gris* (l_{car}) se va a ofrecer un ejemplo con dos casos diferentes, denominados caso A y caso B, y reflejados en la tabla 3.6. Se ofrecen igualmente los resultados en forma de imágenes de la salida de la capa 2 del modelo.

En general, se puede afirmar que el parámetro *Número de Vecinos para el Reparto de Carga por Bandas de Nivel de Gris* (l_{car}) está concebido para delimitar el alcance o zona de influencia de la interacción lateral en esta capa 2 del modelo. Lógicamente, cuanto mayor sea este radio de acción, mayor va a ser la zona afectada por el reparto de la carga con los vecinos.

Si lo que se desea es fundir pequeñas zonas de píxeles entre sí (en cuanto al valor de carga), se elegirá un *Número de Vecinos para el Reparto de Carga por Bandas de Nivel de Gris* relativamente pequeño. Este modo de operación permite mantener los detalles de los objetos deformables. Al contrario, si lo que se desea es fundir mayores zonas de píxeles entre sí, igualmente en cuanto al valor de carga, se elegirá un *Número de Vecinos para el Reparto de Carga por Bandas de Nivel de Gris* mayor. Este otro modo de operación permite fundir zonas más extensas de píxeles en una sola forma.

Vemos cómo la columna tercera del ejemplo, con un valor del *Número de Vecinos para el Reparto de Carga por Bandas de Nivel de Gris* más pequeño guarda más detalles que la columna segunda.

Segmentación por Bandas de Nivel de Gris (Capa 0) – Caso A	
<i>Número de Bandas de Nivel de Gris (n)</i>	8
Interacción Lateral para la Computación Acumulativa (Capa 1)	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	63
<i>Valor de Recarga por Vecindad (v_{rv})</i>	31
<i>Número de Vecinos para la Interacción Lateral (l_{per})</i>	32
<i>Valor Mínimo de Mancha por Banda de Nivel de Gris (θ_{per})</i>	150
Interacción Lateral para el Reparto de Carga (Capa 2) – Caso A	
<i>Número de Vecinos para el Reparto de Carga por BNG (l_{car})</i>	64
<i>Valor Mínimo de Mancha para la Fusión en Objetos (θ_{car})</i>	
Interacción Lateral para el Reparto de Carga (Capa 2) – Caso B	
<i>Número de Vecinos para el Reparto de Carga por BNG (l_{car})</i>	8
<i>Valor Mínimo de Mancha para la Fusión en Objetos (θ_{car})</i>	

Tabla 3.6. Juego de ensayo para la explicación del parámetro
Número de Vecinos para el Reparto de Carga por Bandas de Nivel de Gris

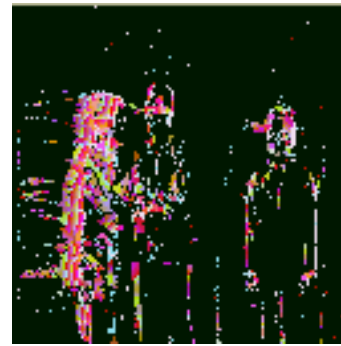
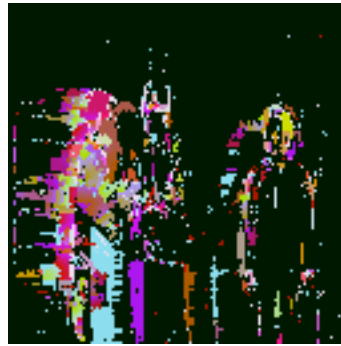


Imagen en t=12



Imagen en t=51



Imagen en t=76



Imagen en t=88

Ejemplo 3.8. Imágenes de ensayo para la explicación de la influencia del parámetro *Número de Vecinos* para el Reparto de Carga por Bandas de Nivel de Gris

3.3.6.2. Influencia del parámetro Valor Mínimo de Mancha para la Fusión en Objetos

Para explicar la influencia del parámetro *Valor Mínimo de Mancha para la Fusión de Objetos* (θ_{car}) se va a trabajar con dos de los valores posibles de dicho parámetro (ver tabla 3.7.) y se van a ofrecer los resultados obtenidos a la salida de la capa 3 del modelo (ver ejemplo 3.9).

Segmentación por Bandas de Nivel de Gris (Capa 0) – Caso A	
<i>Número de Bandas de Nivel de Gris (n)</i>	8
Interacción Lateral para la Computación Acumulativa (Capa 1)	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	63
<i>Valor de Recarga por Vecindad (v_{rv})</i>	31
<i>Número de Vecinos para la Interacción Lateral (l_{per})</i>	32
<i>Valor Mínimo de Mancha por Banda de Nivel de Gris (θ_{per})</i>	150
Interacción Lateral para el Reparto de Carga (Capa 2) – Caso A	
<i>Número de Vecinos para el Reparto de Carga por BNG (l_{car})</i>	64
<i>Valor Mínimo de Mancha para la Fusión en Objetos (θ_{car})</i>	150
Interacción Lateral para el Reparto de Carga (Capa 2) – Caso B	
<i>Número de Vecinos para el Reparto de Carga por BNG (l_{car})</i>	64
<i>Valor Mínimo de Mancha para la Fusión en Objetos (θ_{car})</i>	200

Tabla 3.7. Juego de ensayo para la explicación del parámetro

Valor Mínimo de Mancha para la Fusión en Objetos

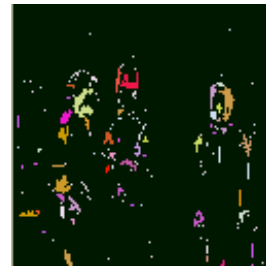
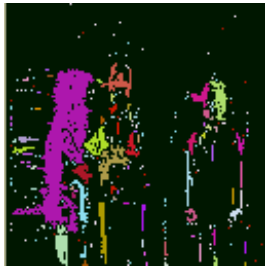


Imagen en t=12

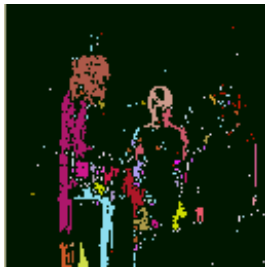


Imagen en t=38

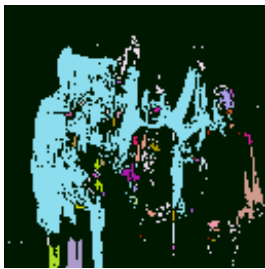


Imagen en t=51

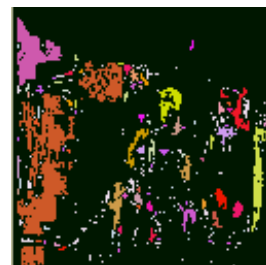
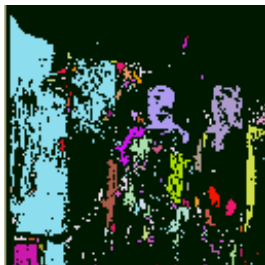


Imagen en t=76

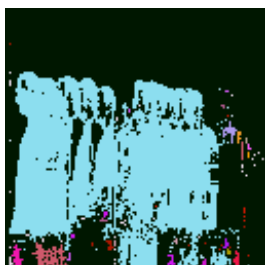


Imagen en t=88

Ejemplo 3.9. Imágenes de ensayo para la explicación de la influencia del parámetro *Valor Mínimo de Mancha para la Fusión en Objetos*

Como se puede apreciar fácilmente, el parámetro *Valor Mínimo de Mancha para la Fusión de Objetos* (θ_{car}) es un valor umbral que va a limitar el modelo de interacción lateral en computación acumulativa en su capa 2 de interacción lateral para el reparto de carga por bandas de nivel de gris a aquellos pixeles de imagen con valor de carga por debajo de θ_{car} .

Un *Valor Mínimo de Mancha para la Fusión en Objetos* (θ_{car}) más alto discrimina más el número de pixeles a tratar a este nivel.

3.3.7. Influencia de los parámetros de la capa 3

3.3.7.1. Influencia del parámetro Número de Vecinos para el Reparto de Carga por Objeto

Para explicar la influencia del parámetro *Número de Vecinos para el Reparto de Carga por Objeto* (l_{obj}) se va a ofrecer un ejemplo con dos casos diferentes, denominados caso A y caso B, y reflejados en la tabla 3.8. Se ofrecen igualmente los resultados en forma de imágenes de la salida de la capa 3 del modelo.

En general, se puede afirmar que el parámetro *Número de Vecinos para el Reparto de Carga por Objeto* (l_{obj}) está concebido para delimitar el alcance o zona de influencia de la interacción lateral en esta capa 3 del modelo. Lógicamente, cuanto mayor sea este radio de acción, mayor va a ser la zona afectada por el reparto de la carga con los vecinos.

Si lo que se desea es fundir pequeñas zonas de pixeles entre sí (en cuanto al valor de carga), se elegirá un *Número de Vecinos para el Reparto de Carga por Objeto* relativamente pequeño. Este modo de operación permite mantener los detalles de los objetos deformables. Al contrario, si lo que se desea es fundir mayores zonas de pixeles entre sí, igualmente en cuanto al valor de carga, se elegirá un *Número de Vecinos para el Reparto de Carga por Objeto* mayor. Este otro modo de operación permite fundir zonas más extensas de pixeles en una sola forma.

Vemos cómo la columna tercera del ejemplo, con un valor del *Número de Vecinos para el Reparto de Carga por Objeto* más pequeño guarda más detalles que la columna segunda.

Segmentación por Bandas de Nivel de Gris (Capa 0) – Caso A	
<i>Número de Bandas de Nivel de Gris (n)</i>	8
Interacción Lateral para la Computación Acumulativa (Capa 1)	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	63
<i>Valor de Recarga por Vecindad (v_{rv})</i>	31
<i>Número de Vecinos para la Interacción Lateral (l_{per})</i>	32
<i>Valor Mínimo de Mancha por Banda de Nivel de Gris (θ_{per})</i>	150
Interacción Lateral para el Reparto de Carga (Capa 2) – Caso A	
<i>Número de Vecinos para el Reparto de Carga por BNG (l_{car})</i>	64
<i>Valor Mínimo de Mancha para la Fusión en Objetos (θ_{car})</i>	150
Interacción Lateral para la Fusión de Objetos (Capa 3) – Caso A	
<i>Número de Vecinos para el Reparto de Carga por Objeto (l_{obj})</i>	256
<i>Valor Mínimo para la Detección de Objetos (θ_{obj})</i>	
Interacción Lateral para la Fusión de Objetos (Capa 3) – Caso B	
<i>Número de Vecinos para el Reparto de Carga por Objeto (l_{obj})</i>	16
<i>Valor Mínimo para la Detección de Objetos (θ_{obj})</i>	

Tabla 3.8. Juego de ensayo para la explicación del parámetro

Número de Vecinos para el Reparto de Carga por Objeto

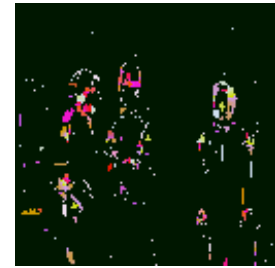


Imagen en t=12

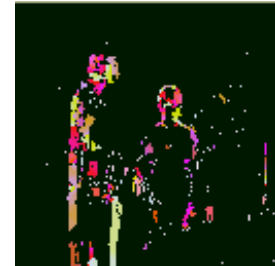
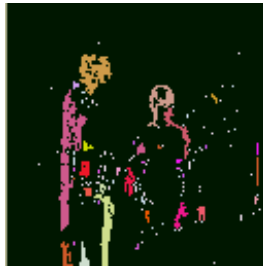


Imagen en t=38

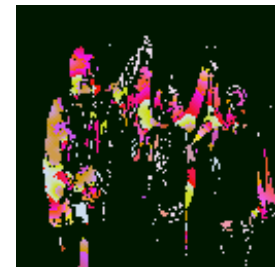


Imagen en t=51

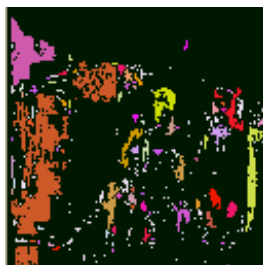


Imagen en t=76



Imagen en t=88

Ejemplo 3.10. Imágenes de ensayo para la explicación de la influencia del parámetro *Número de Vecinos para el Reparto de Carga por Objeto*

3.4. Aprendizaje en el modelo de interacción lateral en computación acumulativa

3.4.1. Introducción

El término aprendizaje [Mir95] se suele traducir por: incrementar el conocimiento adquirido, realizar tareas con mayor eficiencia o corrección, o llevar a cabo nuevas tareas. La primera de las facetas señaladas, antes o después, acaba manifestándose en alguna de las otras dos formas. La adquisición de conocimiento se entiende como una actividad de diálogo en la que un *Agente* (un sistema inteligente en interacción con el medio y en ocasiones con otros agentes) es informado de algún hecho o situación, o realiza preguntas que originan una reestructuración del conocimiento disponible.

Los tipos de aprendizaje básicos se pueden agrupar en:

1. aprendizaje de naturaleza inductiva
 - 1.1. conocimiento independiente del dominio, basado en ejemplos/observaciones
 - 1.2. conocimiento dependiente del dominio
2. aprendizaje de naturaleza deductiva, basado en la explicación
3. aprendizaje de naturaleza mixta

3.4.2. Aprendizaje en el modelo

El aprendizaje en el modelo de interacción lateral en computación acumulativa parte del conocimiento de la influencia de los parámetros básicos del modelo descritos con anterioridad. Se entiende, por tanto, por aprendizaje en el modelo de interacción lateral en computación acumulativa el ajuste de los parámetros de las diversas capas del modelo hasta ofrecer un resultado óptimo en el proceso de las secuencias de imágenes hacia la obtención de las siluetas de los elementos móviles presentes en la escena. El aprendizaje en realidad solamente afecta a la capa 0, responsable de la segmentación de las imágenes de entrada, y a las capas 1, 2 y 3, en las que cabe hablar de interacción lateral. La capa 4, fuertemente dependiente del tipo de aplicación, no requiere de aprendizaje alguno.

Durante el proceso de aprendizaje, previo al proceso normal de funcionamiento del modelo, se ofrece a la arquitectura una secuencia de imágenes de entrada, junto a los siguientes parámetros de refuerzo dependientes de la escena:

- *Número de Elementos Móviles (S_m)* a detectar en la secuencia
- *Tamaño Máximo de Silueta (S_{max})* a detectar en la secuencia
- *Tamaño Mínimo de Silueta (S_{min})* a detectar en la secuencia

No parece necesario explicar el refuerzo del parámetro *Número de Elementos Móviles (S_m)*. Los otros dos parámetros provienen del conocimiento del dominio del modelo de interacción lateral en computación acumulativa. Es imprescindible dar los parámetros de *Tamaño Máximo de Silueta (S_{max})* y de *Tamaño Mínimo de Silueta (S_{min})* para captar la atención del modelo en aquellos objetos cuya silueta esté comprendida entre estas dos magnitudes.

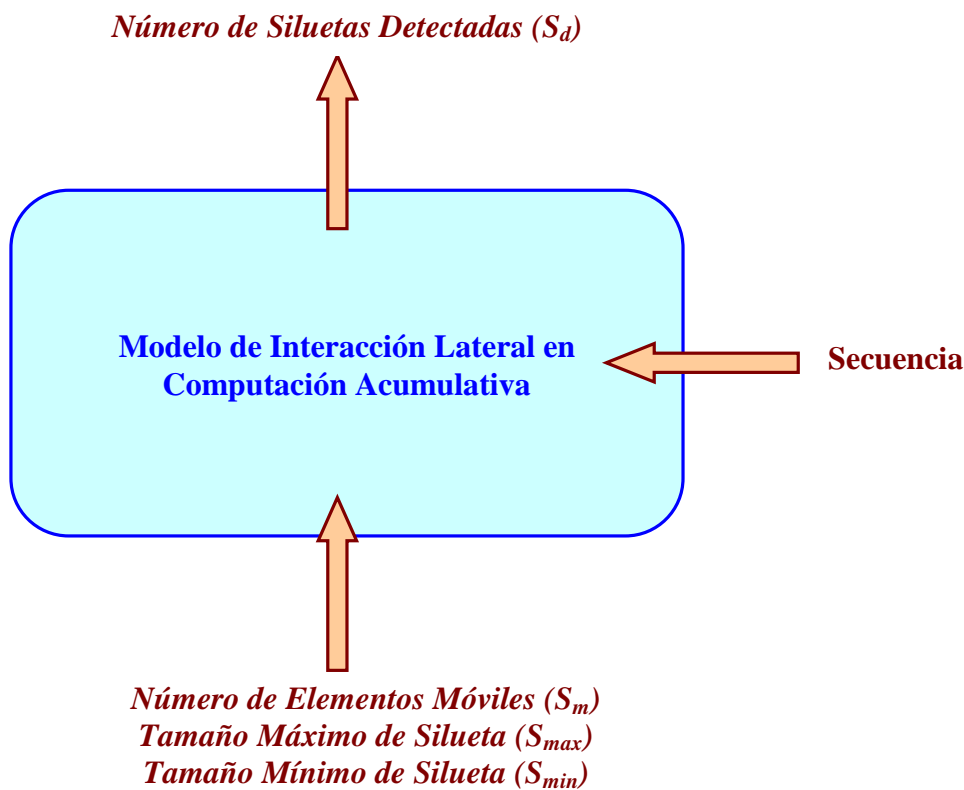


Figura 3.30. Entradas y salidas al modelo de interacción lateral durante la fase de aprendizaje

El aprendizaje se torna, en nuestro caso, en un proceso iterativo durante el cual el modelo, para un *Agente* dado, va siendo nutrido por una misma secuencia de imágenes, y va modificando los parámetros básicos del modelo hasta la obtención en la capa 3 de un número de

siluetas lo más próximo posible a *Número de Elementos Móviles* (S_m). Este nuevo parámetro de salida de la capa 3 se denomina *Número de Siluetas Detectadas* (S_d).

Se han clasificado los parámetros básicos del modelo de interacción lateral en computación acumulativa en los siguientes grupos:

a) *Parámetros con valor fijo que no evolucionan durante el aprendizaje.*

El conocimiento del dominio del modelo de interacción lateral en computación acumulativa ha mostrado que para la obtención de las siluetas de los objetos deformables en movimiento en una secuencia de imágenes algunos parámetros pueden tomar unos valores por defecto que no necesitan de ninguna evolución. Tal es el caso, por capas, de:

- Capa 1:

v_{des} *Valor Mínimo de Permanencia (descarga)*

(valor fijo, $v_{des} = 0$)

v_{sat} *Valor Máximo de Permanencia (saturación)*

(valor fijo, $v_{sat} = 255$)

b) *Parámetros con valor dependiente del Agente y que no evolucionan durante el aprendizaje.*

Igualmente, el conocimiento del dominio del modelo de interacción lateral en computación acumulativa ha mostrado que para la obtención de las siluetas de los objetos deformables en movimiento en una secuencia de imágenes otros parámetros se obtienen directamente como función de los parámetros de refuerzo propios del *Agente*.

- Capa 1:

l_{per} *Número de Vecinos para la Interacción Lateral*

(valor calculado, $l_{per} = S_{max}$)

- Capa 2:

l_{car} *Número de Vecinos para el Reparto de Carga por Bandas de Nivel de Gris*

(valor calculado, $l_{car} = S_{max}$)

- Capa 3:

l_{obj} , *Número de Vecinos para el Reparto de Carga por Objeto*

(valor calculado, $l_{obj} = S_{max}$)

- c) *Parámetros con valor inicial que evolucionan durante el aprendizaje.*

Otros parámetros básicos del modelo de interacción lateral en computación acumulativa sí que evolucionan a lo largo del aprendizaje. Por capas, tenemos:

- Capa 0:

n , *Número de Bandas de Nivel de Gris*

(rango 4, 8, 12, 16)

- Capa 1:

v_{dm} , *Valor de Descarga por Detección de Movimiento*

(rango 32, 48, 64, 80, 96, 112, 128)

v_{rv} , *Valor de Recarga por Vecindad*

(rango 32, 48, 64, 80, 96, 112)

θ_{per} , *Valor Mínimo de Mancha por Banda de Nivel de Gris (umbral)*

(rango 63, 79, 95, 111, 127, 143, 159, 175, 191)

- Capa 2:

θ_{car} , *Valor Mínimo de Mancha para la Fusión en Objetos (umbral)*

(rango 63, 79, 95, 111, 127, 143, 159, 175, 191)

- Capa 3:

θ_{obj} , *Valor Mínimo de Detección de Silueta (umbral)*

(rango 63, 79, 95, 111, 127, 143, 159, 175, 191)

3.4.3. Arquitectura del aprendizaje

La arquitectura del aprendizaje viene reflejada en la figura 3.31. Allí se ven reflejadas todas las capas que toman parte en el aprendizaje, así como los parámetros que actúan externamente sobre el modelo multicapa.

Vemos también cómo el parámetro *Número de Siluetas Detectadas* (S_d) es realimentado hacia la entrada de las capas en forma de parámetros básicos del modelo de interacción lateral en computación acumulativa modificados.

3.4.4. Algoritmo de aprendizaje

El algoritmo de aprendizaje es consecuencia directa de la arquitectura que acabamos de ver. Por tanto debe incorporar:

- a) una alimentación continua de los valores de refuerzo
- b) una alimentación continua de los parámetros con valor fijo
- c) una modificación de los parámetros que evolucionan durante el aprendizaje
- d) una realimentación de la salida de la capa 3 del modelo
- e) un control a nivel de cada una de las capas de las modificaciones operadas durante el aprendizaje

Vamos, por tanto, a utilizar para el aprendizaje en el modelo de interacción lateral en computación acumulativa una función de aprendizaje A_j de aquellas englobadas en las *funciones que minimizan el error* entre la respuesta actual de la red, $y_j(t)$, y la deseada, $d_j(t)$, para la misma entrada, $x_i(t - \Delta t)$. El problema se torna en encontrar un procedimiento de estimación del conjunto de valores que mejor ajusten la función f_j para todos los pares (x_i, y_j) , y no sólo para los que ha sido entrenada. El conjunto de entrenamiento debe ser limitado pero representativo del comportamiento de la función f_j . De lo contrario, la solución es trivial o poco útil.

Para la selección de los parámetros se usa una función de coste que minimiza directamente el error

$$e(t) = d_j(t) - y_j(t)$$

o una función del error.

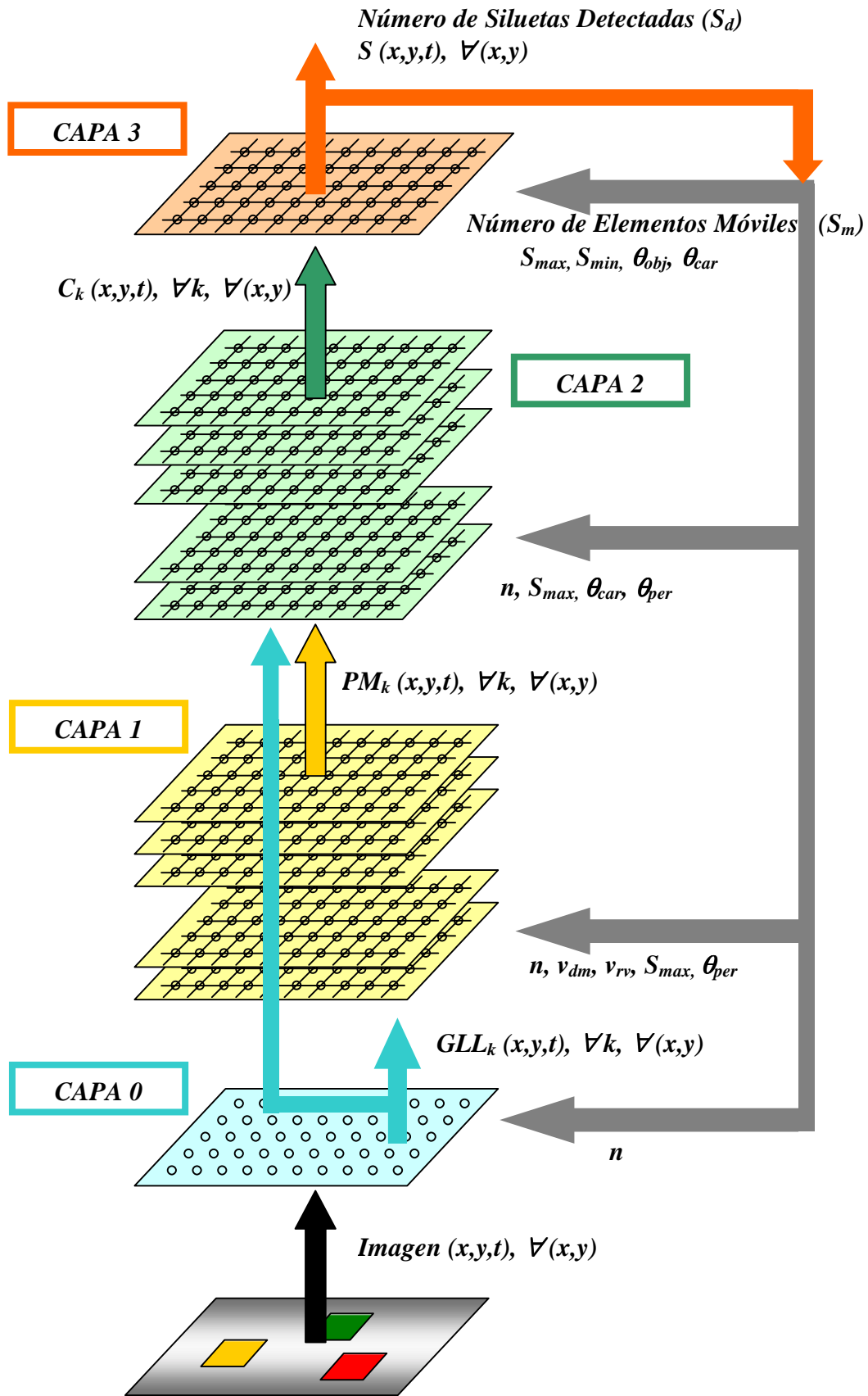


Figura 3.31. Arquitectura para el aprendizaje

La idea que subyace en cada una de las capas es la de adaptar los parámetros de la capa hasta conseguir a la salida de la capa 3 un número de siluetas óptimo, en las diferentes pasadas de una misma secuencia de imágenes de entrada.

Para ello, se ha decidido buscar el conjunto de valores óptimos

$$C^* = (n^*, v_{dm}^*, v_{rv}^*, \theta_{per}^*, \theta_{car}^*, \theta_{obj}^*)$$

que minimizan la función de error

$$E = \left| S_m - \frac{1}{k} \sum_{t=0}^k S_d(t) \right|$$

siendo k el número de imágenes de que consta la secuencia de aprendizaje,

S_m el *Número de Elementos Móviles* a detectar, fijo durante toda la secuencia de entrenamiento,

$S_d(t)$ el *Número de Siluetas Detectadas* en el instante de tiempo t .

Una primera aproximación consiste en trabajar con todas las combinaciones posibles de valores del conjunto C de parámetros del modelo. Pero ello nos lleva a tener que ofrecer $4 \times 7 \times 6 \times 9 \times 9 \times 9 = 122.472$ veces la secuencia de entrenamiento al sistema. Si suponemos una secuencia de entrada de 50 imágenes, tendríamos la friolera de 6.123.600 imágenes a mostrar al sistema. Evidentemente, este tipo de aprendizaje, aunque teóricamente correcto, es computacionalmente inviable. Eso sí, queda perfectamente garantizado que se va a hallar el mínimo global de la función de error.

Ante esta situación, una segunda aproximación es la de limitar el conjunto de valores posibles al conjunto de valores viables. Para ello, se deben tomar en cuenta las restricciones que provienen del conocimiento del dominio del propio modelo de interacción lateral en computación acumulativa. Estas son:

$$\begin{aligned} v_{dm} &\geq v_{rv} \\ \theta_{per} &\leq \theta_{car} \end{aligned}$$

Es decir, que el *Valor de Descarga debido a la Detección de Movimiento* siempre ha de ser mayor o igual al *Valor de Recarga por Vecindad*. Tampoco tiene sentido operar con un *Valor Mínimo de Descarga por Detección de Movimiento* superior a un *Valor Mínimo de Mancha por Banda de Nivel de Gris*.

De este modo se sigue garantizando la obtención del mínimo global de la función de error E . Se consigue reducir el número de casos posibles a $4 \times (1+2+3+4+5+6+6) \times (1+2+3+4+5+6+7+8+9) \times 9 = 43.740$. Ello supone una mejora en tiempo de ejecución del 64%, aunque sigue siendo excesivo. Para una secuencia de 50 imágenes, tenemos todavía 2.187.000 imágenes a procesar durante el aprendizaje.

Se propone, por ello, utilizar otra serie de restricciones que provienen igualmente del conocimiento del dominio del modelo de interacción lateral en computación acumulativa. No obstante, ya no va a quedar garantizada la consecución de un mínimo global, sino que, dependiendo de los valores iniciales dados a los parámetros del modelo, se va a llegar a un mínimo local. Estas restricciones son:

$$S_d = f \left[h_n \cdot n ; h_{dm} \cdot v_{dm} ; h_{rv} \cdot v_{rv} ; \frac{1}{h_{per}} \cdot \theta_{per} ; \frac{1}{h_{per}} \cdot \theta_{per} ; \frac{1}{h_{per}} \cdot \theta_{per} \right],$$

es decir, que el *Número de Siluetas Detectadas* es directamente proporcional al *Número de Bandas de Nivel de Gris*, al *Valor de Descarga debida al Movimiento*, al *Valor de Recarga por Vecindad*, e inversamente proporcional al *Valor de Descarga por Detección de Movimiento*, al *Valor Mínimo de Mancha por Banda de Nivel de Gris* y al *Valor Mínimo de Detección de Silueta*.

De este modo, el entrenamiento se reduce, en general, a una decena de iteraciones, con lo que se consigue una buena relación entre eficacia y tiempo de proceso.

En este último modo de aprendizaje, y tal como puede verse en la figura 3.32., tras cualquier modificación de algún parámetro en cualquiera de las capas, el algoritmo vuelve a emprenderse desde la capa 3.

Una visión general del algoritmo concreto se ofrece en los siguientes apartados de este capítulo.

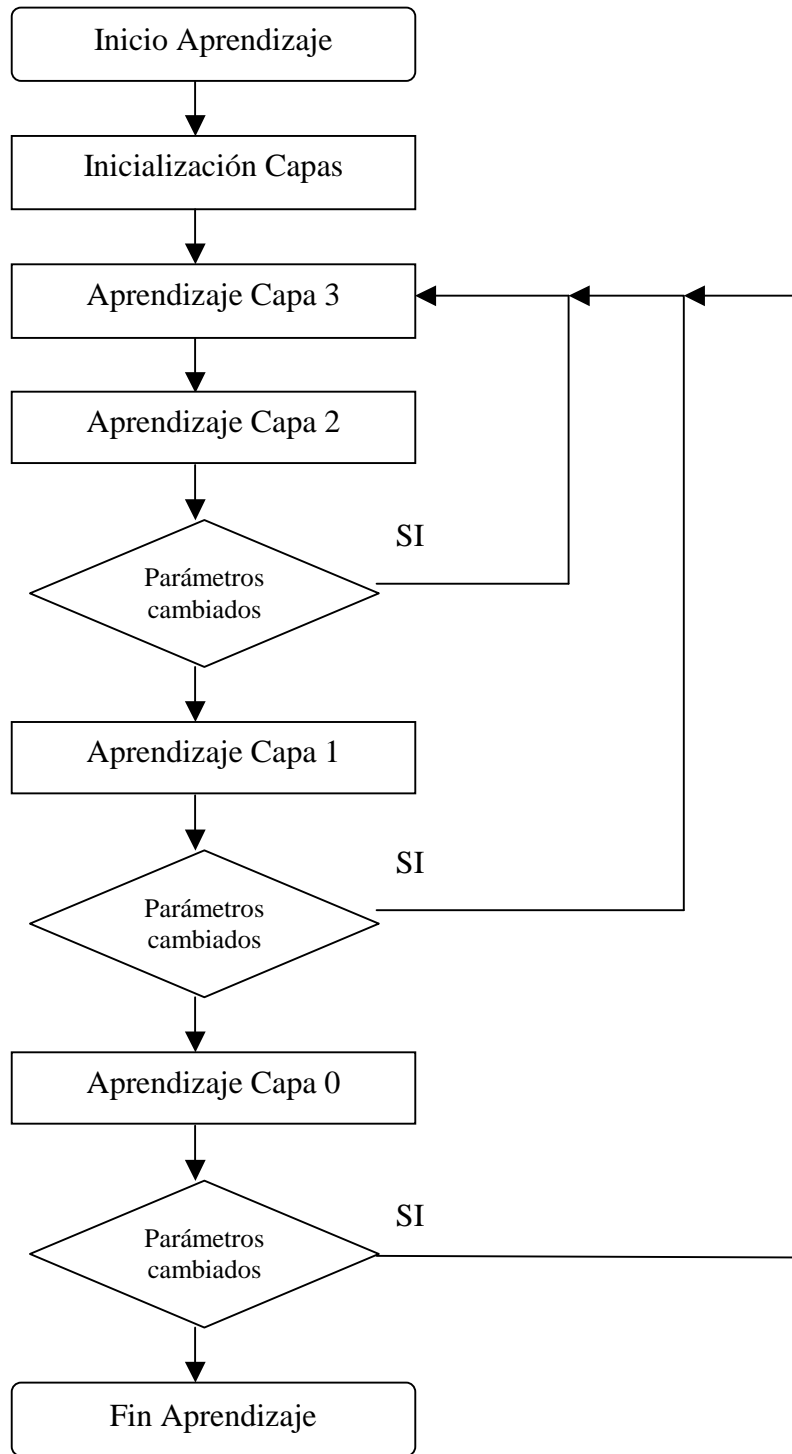


Figura 3.32. Diagrama de flujo del aprendizaje

3.4.4.1. Inicialización de las capas

// Inicialización Capas.

// Algoritmo.

// Inicialización de los parámetros de la Capa 0.

$$n = 8$$

// Inicialización de los parámetros de la Capa 1.

$$v_{des} = 0; v_{sat} = 255; v_{dm} = 64; v_{rv} = 32$$

$$l_{per} = S_{max}; \theta_{per} = 127$$

// Inicialización de los parámetros de la Capa 2.

$$l_{car} = S_{max}; \theta_{car} = 127$$

// Inicialización de los parámetros de la Capa 3.

$$l_{obj} = S_{max}; \theta_{obj} = 127$$

// Ejecución del modelo de interacción lateral en computación acumulativa

// para los parámetros iniciales.

// Obtención del *Número de Siluetas Detectadas*.

$$S_d = \sum S(x, y, t)$$

3.4.4.2. Aprendizaje en la capa 3

// Aprendizaje en la Capa 3.

// Algoritmo.

// Ajuste del parámetro de la capa 3 *Valor Mínimo de Detección de Silueta..*

DO

// Guardar el valor anterior de *Número de Siluetas Detectadas*.

$$S_{d^*} = S_d$$

// Guardar el valor anterior de *Valor Mínimo de Detección de Silueta*.

$$\theta_{obj^*} = \theta_{obj}$$

// Modificar el único parámetro de la capa dependiendo de la relación entre

// *Número de Siluetas Detectadas* y *Número de Elementos Móviles*.

IF ($S_d > S_m$) THEN θ_{obj^*} = incrementar θ_{obj}

ELSE θ_{obj^*} = decrementar θ_{obj}

// Ejecución del modelo de interacción lateral en computación acumulativa

// para los parámetros modificados.

// Obtención del nuevo *Número de Siluetas Detectadas*.

$$S_d = \sum S(x, y, t)$$

WHILE ($|S_d - S_m| \leq |S_{d^*} - S_m|$)

3.4.3.3. Aprendizaje en la capa 2

// Aprendizaje en la Capa 2.

// Algoritmo.

mejora_capa = FALSE

// Ajuste del parámetro de la capa 2 *Valor Mínimo de Mancha para la Fusión en Objetos.*

DO

// Guardar el valor anterior de *Número de Siluetas Detectadas.*

$S_{d^*} = S_d$

// Guardar el valor anterior de *Valor Mínimo de Mancha para la Fusión en Objetos.*

$\theta_{car^*} = \theta_{car}$

// Modificar el único parámetro de la capa dependiendo de la relación entre *Número de Siluetas Detectadas* y *Número de Elementos Móviles.*

IF ($S_d > S_m$) THEN $\theta_{car^*} =$ incrementar θ_{car}

ELSE $\theta_{car^*} =$ decrementar θ_{car}

// Ejecución del modelo de interacción lateral en computación acumulativa para los parámetros modificados.

// Obtención del nuevo *Número de Siluetas Detectadas.*

$S_d = \sum S(x, y, t)$

// ¿Ha habido mejora tras el ajuste?

IF ($|S_d - S_m| < |S_{d^*} - S_m|$) THEN mejora_capa = TRUE

WHILE ($|S_d - S_m| \leq |S_{d^*} - S_m|$)

3.4.3.4. Aprendizaje en la capa 1

// Aprendizaje en la Capa 1.

// Algoritmo.

// Paso 1.

mejora_capa = FALSE

// Ajuste del parámetro de la capa 1 *Valor Mínimo de Mancha por Banda de*

// *Nivel de Gris.*

DO

 // Guardar el valor anterior de *Número de Siluetas Detectadas.*

$S_{d^*} = S_d$

 // Guardar el valor anterior de *Valor Mínimo de Mancha por Banda de*

 // *Nivel de Gris.*

$\theta_{per^*} = \theta_{per}$

 // Modificar el parámetro de la capa dependiendo de la relación entre

 // *Número de Siluetas Detectadas* y *Número de Elementos Móviles.*

 IF ($S_d > S_m$) THEN θ_{per^*} = incrementar θ_{per}

 ELSE θ_{per^*} = decrementar θ_{per}

 // Ejecución del modelo de interacción lateral en computación acumulativa

 // para los parámetros modificados.

 // Obtención del nuevo *Número de Siluetas Detectadas.*

$S_d = \sum S(x, y, t)$

 // ¿Ha habido mejora tras el ajuste?

 IF ($|S_d - S_m| < |S_{d^*} - S_m|$) THEN mejora_capa = TRUE

WHILE ($|S_d - S_m| \leq |S_{d^*} - S_m|$)

// Paso 2.

Mejora = FALSE

// Ajuste del parámetro de la capa 1 *Valor de Descarga por Detección de Movimiento*

DO

// Guardar el valor anterior de *Número de Siluetas Detectadas*.

$$S_{d*} = S_d$$

// Guardar el valor anterior de *Valor de Descarga por Detección de Movimiento*

$$v_{dm*} = v_{dm}$$

// Modificar el parámetro de la capa dependiendo de la relación entre

// *Número de Siluetas Detectadas* y *Número de Elementos Móviles*.

IF ($S_d > S_m$) THEN v_{dm*} = decrementar v_{dm}

ELSE v_{dm*} = incrementar v_{dm}

// Ejecución del modelo de interacción lateral en computación acumulativa

// para los parámetros modificados.

// Obtención del nuevo *Número de Siluetas Detectadas*.

$$S_d = \sum S(x, y, t)$$

// ¿Ha habido mejora tras el ajuste?

IF ($|S_d - S_m| < |S_{d*} - S_m|$) THEN mejora = TRUE

mejora_capa = TRUE

WHILE ($|S_d - S_m| < |S_{d*} - S_m|$)

IF (mejora == TRUE) THEN Ir al paso 1.

// Paso 3.

mejora = FALSE

// Ajuste del parámetro de la capa 1 *Valor de Recarga por Vecindad*.

DO

// Guardar el valor anterior de *Número de Siluetas Detectadas*.

$$S_{d^*} = S_d$$

// Guardar el valor anterior de *Valor de Recarga por Vecindad*.

$$v_{rv^*} = v_{rv}$$

// Modificar el parámetro de la capa dependiendo de la relación entre

// *Número de Siluetas Detectadas* y *Número de Elementos Móviles*.

IF ($S_d > S_m$) THEN v_{rv^*} = decrementar v_{rv}

ELSE v_{rv^*} = incrementar v_{rv}

// Ejecución del modelo de interacción lateral en computación acumulativa

// para los parámetros modificados.

// Obtención del nuevo *Número de Siluetas Detectadas*.

$$S_d = \sum S(x, y, t)$$

// ¿Ha habido mejora tras el ajuste?

IF ($|S_d - S_m| < |S_{d^*} - S_m|$) THEN mejora = TRUE

mejora_capa = TRUE

WHILE ($|S_d - S_m| < |S_{d^*} - S_m|$)

IF (mejora == TRUE) THEN Ir al paso 1.

3.4.3.5. Aprendizaje en la capa 0

// Aprendizaje en la Capa 0.

// Algoritmo.

mejora_capa = FALSE

// Ajuste del parámetro de la capa 0 *Número de Bandas de Nivel de Gris.*

DO

 // Guardar el valor anterior de *Número de Siluetas Detectadas.*

$S_{d^*} = S_d$

 // Guardar el valor anterior de *Número de Bandas de Nivel de Gris.*

$n^* = n$

 // Modificar el único parámetro de la capa dependiendo de la relación entre

 // *Número de Siluetas Detectadas* y *Número de Elementos Móviles.*

 IF ($S_d > S_m$) THEN $n^* =$ decrementar n

 ELSE $n^* =$ incrementar n

 // Ejecución del modelo de interacción lateral en computación acumulativa

 // para los parámetros modificados.

 // Obtención del nuevo *Número de Siluetas Detectadas.*

$S_d = \sum S(x, y, t)$

 // ¿Ha habido mejora tras el ajuste?

 IF ($|S_d - S_m| < |S_{d^*} - S_m|$) THEN mejora_capa = TRUE

WHILE ($|S_d - S_m| \leq |S_{d^*} - S_m|$)

CAPITULO 4. Implementación

4.1. Esquema general del entorno

4.1.1. Antecedentes

4.1.1.1. Sistema TIRANO

El sistema **TIRANO** [Fer97] [Fer98b] es una aplicación desarrollada en el contexto de una tesis doctoral y que pretendía alcanzar el objetivo de clasificar los elementos móviles o situaciones de movimiento que aparecen en una escena y relacionarlos con los valores de refuerzo que los acompañaron durante la fase de aprendizaje.

Aunque si bien es cierto que el diseño del sistema **TIRANO** pretendía que éste trabajara en tiempo real, el simulador no tuvo esta capacidad de trabajo – fue desarrollado como aplicación MS-DOS e implementado en C estándar – aunque sí permitía la entrada de un nuevo cuadro de imagen cuando el anterior había terminado de procesarse (de aquí en adelante utilizaremos indistintamente los términos cuadro, fotograma o imagen).

El sistema **TIRANO** recibe como entrada una secuencia de imágenes blanco y negro de 512x512 puntos, que se van generando directamente sobre una tarjeta digitalizadora, y una señal de refuerzo de 3 bits. Básicamente, extrae cuadro a cuadro la característica RLV (relación longitud / velocidad) [Fer92] en cada punto de la imagen y realiza tareas de clasificación en la escena atendiendo a dicha característica.

El método de trabajo con este simulador puede dividirse en varios pasos:

1) *Diseño y montaje de las secuencias de entrenamiento*

Esta operación se realiza diseñando los fondos de escena y los elementos móviles, mediante una serie de herramientas diseñadas para trabajar con este simulador. Posteriormente, mediante rutinas software, se realiza un fichero de secuencias en

el que cada fotograma está descrito en un registro, en el que aparecen campos dedicados al nombre del fondo y a los de cada elemento de los que componen el fotograma, así como las coordenadas x e y de su posición.

2) *Diseño de los valores de refuerzo*

Los valores de refuerzo se generan al mismo tiempo que se generan los ficheros de secuencia, utilizando para ello las mismas rutinas.

3) *Preparación y definición del estado*

El simulador, por defecto, define los valores de todos los parámetros del sistema. Es posible sin embargo modificar estos valores para aplicaciones concretas.

4) *Trazado y análisis del aprendizaje y comportamiento.*

Una vez preparadas las secuencias a procesar, los valores de refuerzo y el estado inicial del sistema, puede arrancarse el proceso de simulación. Durante este proceso, el simulador permite analizar cuál va siendo el comportamiento de las variables y el estado de activación de los distintos módulos que componen el sistema.

El sistema **TIRANO** presentaba el inconveniente de poseer una única herramienta para la generación de secuencias de imágenes, la cual sólo podía generar secuencias de imágenes sintéticas, a partir de un fichero que recogía el nombre del fondo y de los elementos que se desplazaban sobre éste, teniendo que regenerarlas cada vez que quería ejecutarse la secuencia, al no almacenarlas en ningún tipo de soporte. Dado que no existía un módulo de dispensación de imágenes, este sistema se veía limitado al uso de las secuencias sintéticas arriba descritas, siendo imposible por tanto que recibiera secuencias que se estuvieran produciendo en tiempo real, y viéndose incrementado el tiempo existente entre cada procesado de imagen por el tiempo consumido en la generación de estas. Esto aumentaba aún más el tiempo de procesado total para una secuencia, que era uno de los mayores problemas que presentaba este simulador.

4.1.1.2. Sistema BRONTO

A fin de adaptar el sistema **TIRANO** a las nuevas plataformas que iban apareciendo en el mercado surgió el sistema **BRONTO** [Pon98a] [Pon98b] [Pon98c], con el que se pretendía alcanzar una serie de mejoras sobre el sistema **TIRANO**, tales como el uso de una base de datos que recogiera la información relativa al estado del sistema o la utilización de un magnetoscopio y una cámara para la recepción de secuencias de imágenes. Pero de este sistema solamente fue desarrollada la interfaz de usuario bajo el sistema operativo Windows 95.

Este sistema se apoya principalmente en una placa de computación acumulativa que realiza el procesamiento de la señal (ver figura 4.1.). Esta placa, desarrollada dentro del programa del proyecto TYC9495, se presenta como una arquitectura versátil y capaz, destinada a realizar computación acumulativa y puede ser utilizada para el proceso de extracción y selección de la característica RLV dentro del sistema **BRONTO**.

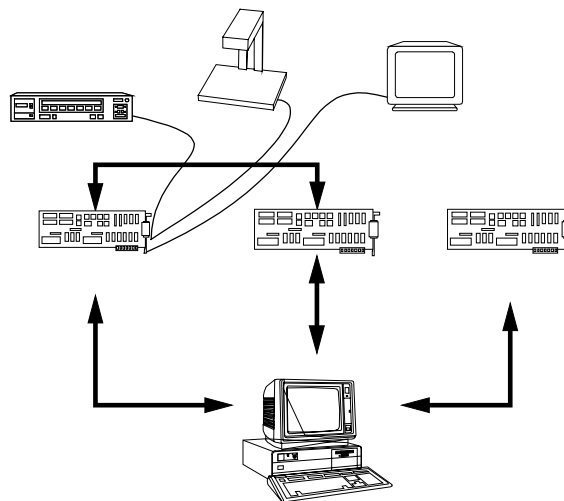


Figura 4.1. Esquema del sistema BRONTO

Las otras dos tarjetas que se muestran en el gráfico son una placa comercial de digitalización de imágenes, encargada de recibir las imágenes dispensadas por el magnetoscopio o la cámara, y una tarjeta del tipo TMB08 con 4 transputer, encargada de realizar las tareas de clasificación de las imágenes.

El sistema **BRONTO** distinguía las mismas fases de trabajo que su antecesor, el sistema **TIRANO**, y que fueron explicadas en el apartado anterior, ofreciendo además la posibilidad de recibir secuencias en tiempo real, directamente desde una cámara o desde un magnetoscopio.

Es comprensible el uso de un magnetoscopio si pensamos en la generación de secuencias sintéticas y grabadas cuadro a cuadro en la cinta magnética del magnetoscopio para ser procesadas posteriormente por el sistema en tiempo real. Un uso más elemental del magnetoscopio sería aquel que nos permitiera grabar las secuencias que, en tiempo real, se vayan recibiendo por cámara. Con esto se permite realizar distintas pruebas sobre la misma secuencia, modificando por ejemplo los parámetros del sistema, a fin de contrastar los resultados obtenidos.

Como podemos observar, independientemente del tratamiento que se vaya a aplicar sobre las imágenes, procesado software o hardware, se plantea la necesidad de acceder a distintos dispositivos con el fin de recoger las secuencias de imágenes que en ellos estén almacenadas, caso del vídeo, o que se estén recibiendo en tiempo real, como es el caso de la cámara. Este hecho vuelve a plantear la necesidad de desarrollar un módulo de dispensación de imágenes, que permita acceder a los distintos dispositivos en que las secuencias estén almacenadas, recuperar los fotogramas que constituyan la secuencia a analizar y dispensarlos al sistema de forma transparente al tipo de dispositivo de almacenamiento usado.

4.1.1.3. Sistema SkeleTton y Base de Datos SkeleTton

Teniendo en cuenta que la resolución computacional al problema de detección de elementos móviles sobre una escena – aún habiendo sido planteada desde el punto de vista del análisis de secuencias de imágenes sobre las que realizar una función de correspondencia entre los distintos elementos – puede ser abordada mediante el uso de diferentes algoritmos, se comienza a elaborar el sistema **SkeleTton** [Fer99b].

Este sistema pretende desarrollar un entorno completamente operativo, que sirva como punto de partida para la elaboración de diversas aplicaciones que permitan identificar elementos móviles en secuencias de imágenes y que se diferencien básicamente en el algoritmo usado para analizar el movimiento de los elementos sobre la escena. El diseño del sistema **SkeleTton** está enfocado por tanto, a la obtención de un conjunto de librerías que constituyan un ‘esqueleto estándar’ para el desarrollo de aplicaciones de este tipo, y cuya implementación permita la reutilización del código generado.

El sistema **SkeleTton** se encuentra estrechamente ligado a la **Base de Datos SkeleTton** [Mon99a] [Fer99a], de la que extrae la información sobre el soporte en el que se encuentran almacenadas las secuencias de imágenes que va a procesar, así como la forma y cantidad en que

los fotogramas que componen cada secuencia deben ser ofrecidos al módulo del sistema **SkeleTTon** encargado de realizar el análisis.

Durante una sesión de trabajo con la aplicación resultante, el sistema puede procesar más de una secuencia, lo que se ha dado en llamar sesión. De esta forma podemos tener sesiones que contengan las mismas secuencias de imágenes pero con características de ejecución (dispensación) distintas.

Cada una de las aplicaciones derivadas del sistema **SkeleTTon** ha seguido una especialización diferente dando lugar a un tipo de *Agente*. Este concepto hace referencia a la historia seguida por el sistema durante el procesamiento de una *Sesión* de imágenes concreta y que por tanto nos ofrece unos datos comunes, independientemente del tipo de *Agente* (esto es, del tipo de aplicación que hayamos usado) y otros específicos que recogen la evolución y los parámetros necesarios para los algoritmos utilizados en el proceso de análisis de las imágenes y sus modificaciones en función de la evolución del propio *Agente*.

Todos los datos relativos a las *Secuencias*, *Sesiones* y *Agentes*, se recogen en la **Base de Datos SkeleTTon**, que ha sido elaborada usando Microsoft Access 97.

En cuanto al desarrollo del sistema **SkeleTTon**, así como cualquiera de sus sucesores, se ha adoptado la siguiente estructura modular:

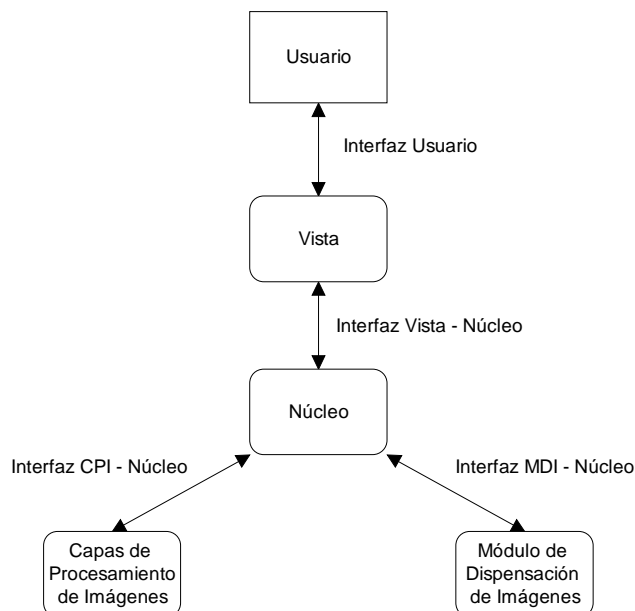


Figura 4.2. Estructura modular del sistema SkeleTTon

Tal y como vemos en el gráfico, el módulo *Vista* es el encargado de recibir y procesar las peticiones realizadas por el usuario y devolverle la información que el sistema haya generado. Esta interfaz de usuario, basada en el sistema **BRONTO**, está implementada en Visual C++ y es operativa bajo Windows 95/NT.

Las *Capas de Procesamiento de Imágenes* recogen la implementación de los modelos desarrollados, dando lugar a distintas versiones de los hijos de **SkeleTton**. Estos modelos se distribuirán en una arquitectura multicapa en la que se distinguen las siguientes fases:

- *Fase de segmentación*. Encargada de dejar la imagen recibida como entrada al sistema en n bandas de nivel de gris.
- *Fase de detección, selección y extracción de características*. En la que se extraerá un conjunto de características que se utilizará para clasificar el movimiento en la escena.
- *Fase de clasificación y asociación*. La cual clasificará los elementos detectados atendiendo al conjunto de características que se extrajeron de la fase anterior.

El *Núcleo* actúa como módulo de control entre la *Vista*, las *Capas de Procesamiento de Imágenes* y el *Módulo de Dispensación de Imágenes*, dando paso a unos u otros en función de la tarea que deba llevar a cabo el sistema.

El *Módulo de Dispensación de Imágenes (MDI)* [Mon00b] [Mon00c] [Mon00d] [Mon00e] es el encargado de ofrecer las imágenes que componen las distintas *Secuencias* de la *Sesión* cuya ejecución haya sido solicitada por el usuario del sistema. Para ello, este módulo deberá acceder a los distintos soportes en los que dichas secuencias se encuentren almacenadas e ir las dispensando al *Núcleo* atendiendo a las características de ejecución que, para la sesión a la que pertenecen, hayan sido establecidas.

4.1.1.4. Sistema Intelligent SILHFinder

Tras comprobar que el sistema **SkeleTton** alcanzaba los resultados esperados, comenzó a desarrollarse la aplicación **Intelligent SILHFinder**, que tal y como se esperaba fue creada con relativamente poco esfuerzo, gracias a la posibilidad de reutilizar el código generado y a la facilidad de mantenimiento del mismo.

Esta aplicación, a diferencia de los sistemas anteriores a **SkeleTton** (**BRONTO** y **TIRANO**) no utiliza la característica RLV como característica determinante para la detección y

selección del movimiento de elementos sobre una escena, sino que utiliza el modelo de interacción lateral para la computación acumulativa, y se aplica en un número elevado de tipos de aplicaciones.

El resto del capítulo describe el esquema general de la aplicación **Intelligent SILHFinder**, el simulador para el modelo de interacción lateral en computación acumulativa.

4.1.2. Descripción general del entorno

4.1.2.1. Descripción hardware del entorno

Al igual que los sistemas **BRONTO** y **SkeleTTon**, el sistema **Intelligent SILHFinder** usa una placa digitalizadora de imágenes para las señales de vídeo procedentes tanto del magnetoscopio como de la cámara. Incluye además un monitor de televisión que permite al usuario visualizar la evolución del procesamiento de imágenes que esté llevando a cabo el sistema.

En la siguiente relación se recogen todos aquellos componentes que constituyen el sistema **Intelligent SILHFinder** a nivel hardware, mostrándose en el gráfico inferior el modo en que se interconexionan:

- 1) Cámara JVC, modelo TK – S3000 (norma CCIR)
- 2) Magnetoscopio SONY, BETACAM-SP modelo UVW – 1700GP (Norma PAL)
- 3) Monitor de televisión SONY (norma CCIR)
- 4) Placa comercial digitalizadora Matrox PIP 1024
- 5) Ordenador de tipo PC

La comunicación entre el PC y el magnetoscopio [Mon99b] [Mon99c] [Mon99d] [Mon00a] se realiza mediante el puerto serie, atendiendo a la norma RS-232C. Del magnetoscopio usaremos los canales de salida video output 1 y video output 2 (SUPER) que serán conectados a la placa digitalizadora (canal de entrada 0) y al monitor (canal de entrada A) respectivamente. Estas señales de vídeo se diferencian en que mediante el canal SUPER podemos enviar al monitor, además de la imagen que en ese momento esté siendo reproducida información referente a los contadores de tiempo y al estado del magnetoscopio. Esta

información auxiliar aparecerá superpuesta a la imagen, facilitando al usuario el seguimiento del estado del magnetoscopio.

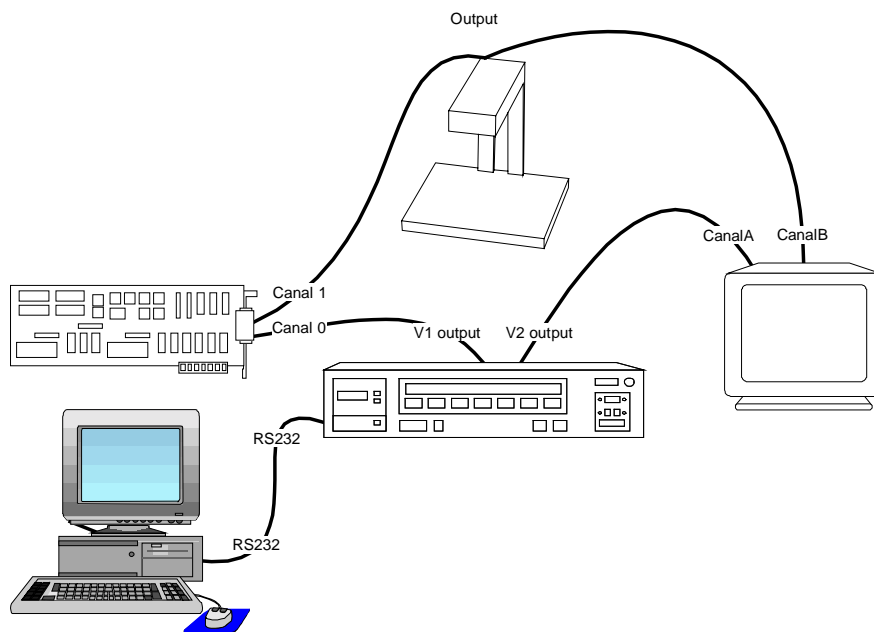


Figura 4.3. Interconexión de los elementos integrantes del sistema Intelligent SILHFinder

La cámara no posee ningún interruptor, por tanto desde el momento en que esté conectada a la red eléctrica (220V AC), estará enviando la señal de vídeo de salida a la placa digitalizadora (canal de entrada 1) y al monitor (canal de entrada B).

La tarjeta digitalizadora se muestra en el gráfico fuera de su emplazamiento natural (dentro del PC) a fin de mostrar con mayor claridad la forma en que deben conectarse los equipos. Para realizar estas conexiones se ha usado cable coaxial y conectores BNC, excepto para la conexión entre el PC y el Magnetoscopio en que se han utilizado conectores de 25 pines con su cable correspondiente. También hubo de sustituirse el conector BNC de la señal de vídeo de salida de la cámara por una T – BNC para permitir el doble direccionamiento de la señal hacia la placa y el monitor.

4.1.2.2. Entorno de desarrollo y operación

La interfaz gráfica del sistema **Intelligent SILHFinder**, módulo *Vista*, se ha desarrollado como una aplicación SDI (Single Document Interface) bajo el sistema Operativo Windows 98, habiéndose usado para su implementación Microsoft Visual C++ 6.0 .

El gestor de órdenes se encuentra situado en la clase de la vista, por lo que la comunicación mediante eventos es distribuida desde el módulo *Vista* hacia el módulo *Núcleo* que actúa como intermediario entre la vista y el resto de módulos del sistema **Intelligent SILHFinder**: el *Módulo de Dispensación de Imágenes* y las *Capas de Procesamiento de Imágenes* recibidas.

Siguiendo las directrices impuestas por el módulo *Vista*, el resto de módulos del sistema **Intelligent SILHFinder** han sido implementados también bajo Visual C++.

Se ha optado igualmente por trabajar utilizando las MFC (Microsoft Foundation Classes). La diferencia existente entre crear una aplicación Windows MFC a una que se implemente sin usar MFC es mucha. Por ejemplo, a diferencia de un programa Windows escrito en C, un programa MFC ni siquiera requiere un punto de entrada. Cabe destacar, igualmente, el soporte provisto por MFC para la conectividad a bases de datos: ODBC (Open Database Connectivity) y DAO (Data Access Objects). Estas dos variantes proveen un medio para comunicarse con un sistema DBMS usando una interfaz estándar de programación de aplicaciones y la sintaxis SQL.

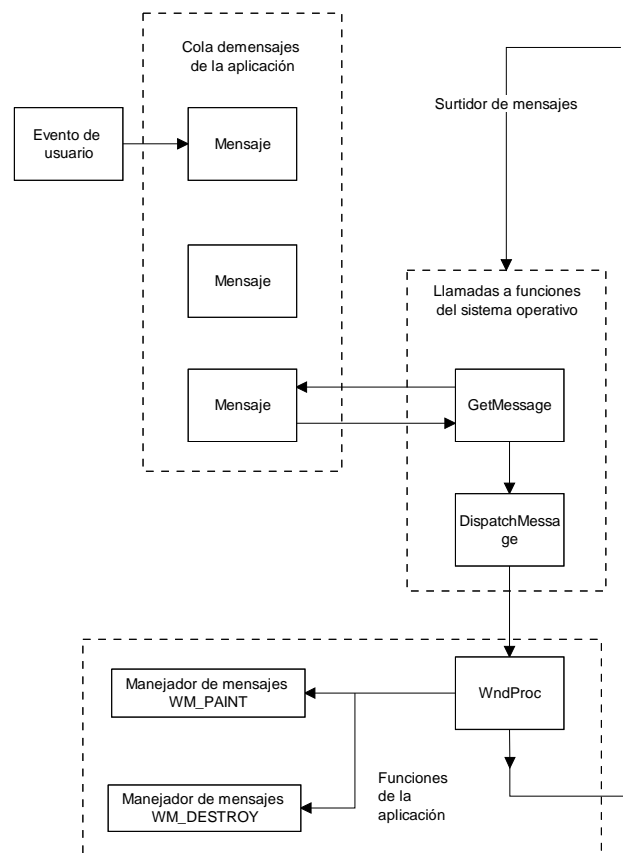


Figura 4.4. Funcionamiento del procedimiento de ventana en un programa basado en Windows

La figura 4.5. muestra el diagrama de clases básicas de toda aplicación MFC.

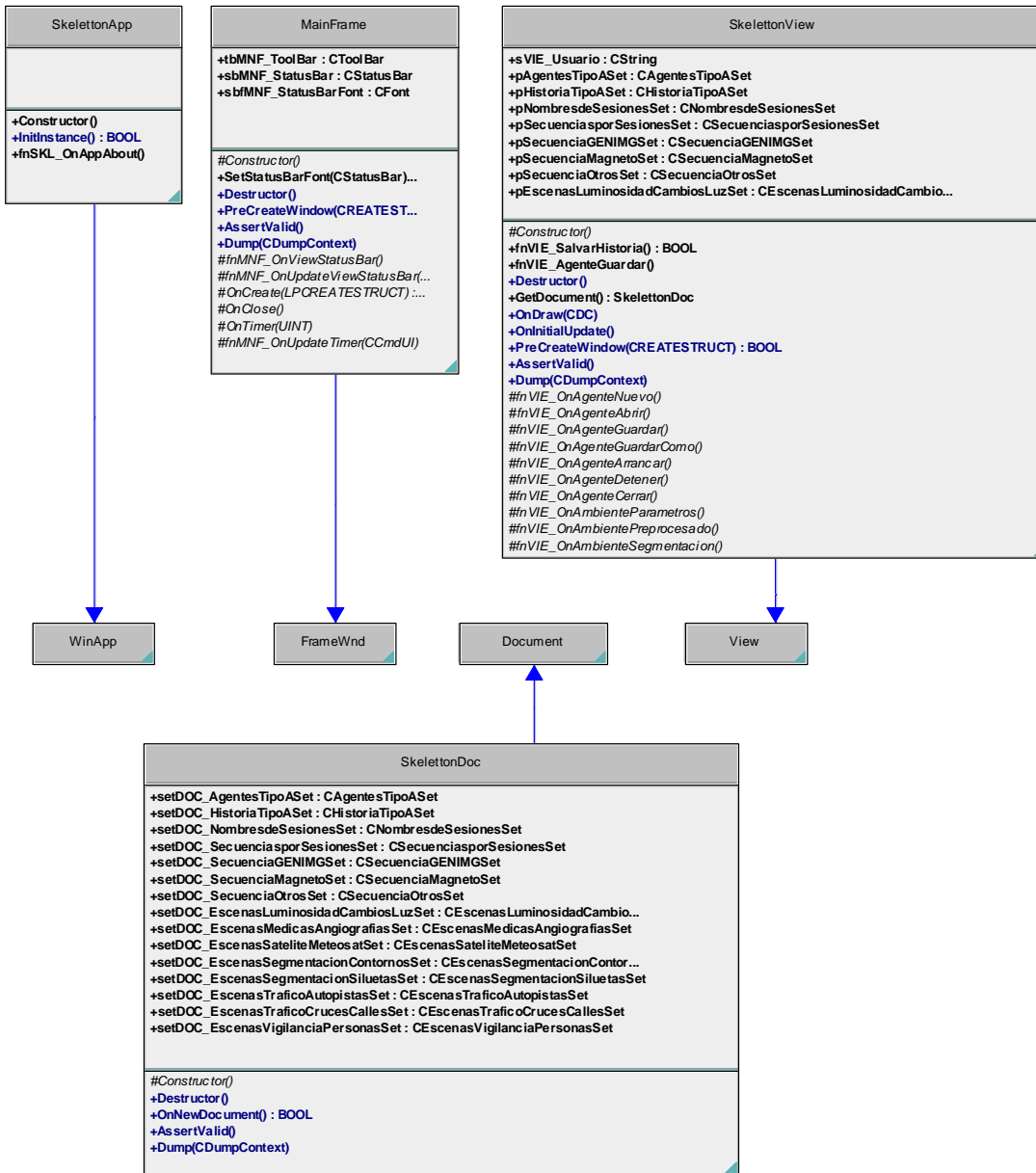


Figura 4.5. Diagrama de clases básicas de toda aplicación MFC

4.1.2.3. Conceptos básicos

4.1.2.3.1. Unidades de dispensación de imágenes

Por el nombre de *Unidades de Dispensación de Imágenes (UDIs)* haremos referencia a aquellos dispositivos internos o externos en los que están almacenadas las secuencias de

imágenes y a los que, por tanto, debe acceder el *Módulo de Dispensación de Imágenes* para obtener los cuadros que en un momento determinado deba ofrecer al módulo *Núcleo* del sistema **Intelligent SILHFinder**. Las *UDIs* disonibles en **Intelligent SILHFinder** se recogen en la siguiente relación:

- Un magnetoscopio (SONY BETACAM-SP, modelo UVW-1700GP)
- Una cámara (JVC, modelo TK-S3000)
- Unidades de disco duro y CD-ROM
- Aplicaciones de generación y manipulación de imágenes (aplicación GENIMG y aplicación CImagen)

4.1.2.3.2. Secuencias

Una *Secuencia* es un conjunto ordenado de imágenes (o fotogramas) que representan uno o varios objetos en movimiento sobre una escena y que viene determinado por un conjunto de datos que identifican la procedencia, cantidad y características de las imágenes que la constituyen. Este conjunto de datos variará en su composición en función del tipo de dispositivo (*UDI*) en que la secuencia se encuentre almacenada.

Teniendo en cuenta que las secuencias de imágenes que se pretende analizar corresponden a móviles sobre una escena, no cabe pensar que una secuencia pueda encontrarse fraccionada o almacenada en distintas *UDIs*. De este modo, en función de la *UDI* en que la *Secuencia* se encuentre almacenada distinguiremos entre los siguientes tipos de secuencias, que se detallan a continuación.

Secuencias MAGNETO

Bajo este nombre se conoce a las secuencias que se encuentran almacenadas en las cintas de vídeo usadas por un magnetoscopio; en el sistema **Intelligent SILHFinder** el magnetoscopio usado es un SONY BETACAM-SP, modelo UVW-1700GP. Este dispositivo de almacenamiento masivo permite la utilización de cintas tanto de 10 horas como de 24 horas consiguiéndose por tanto un almacenamiento máximo de 900.000 o 2.160.000 fotogramas, en función del tipo de cinta que se esté usando.

Las imágenes no son recogidas directamente desde el magnetoscopio sino que es a través de una tarjeta digitalizadora (en el sistema **Intelligent SILHFinder** la digitalizadora usada es

una Matrox PIP 1024), que recoge la señal de vídeo procedente del magnetoscopio, de donde obtenemos los fotogramas que constituyan una secuencia concreta.

Secuencias OTROS

Bajo este nombre se conoce a las secuencias compuestas por imágenes que se encuentran almacenadas en ficheros con alguno de los siguientes formatos: BMP (mapa de bits de Windows u OS/2), IMG (formato del grupo GICoNIA que recoge el fondo de las secuencias), ICN (formato del grupo GICoNIA que recoge los elementos móviles de la secuencia) y/o PGM (Portable Greymap). Considerando que los ficheros de imágenes que constituyen la secuencia pueden encontrarse en distintas unidades o directorios, la secuencia resultante es almacenada en un fichero secuencial indexado con la siguiente estructura: "Índice + Localización de la Imagen" (Fichero OTR).

Secuencias CAMARA

Este tipo de secuencias se corresponde con aquel conjunto de imágenes que son obtenidas directamente y en tiempo real de una cámara de vídeo. En el contexto actual del sistema **Intelligent SILHFinder**, la cámara de vídeo usada es una cámara JVC, modelo TK-S3000, que se conecta al PC mediante una placa comercial de digitalización de imágenes (Matrox PIP 1024).

Básicamente podemos distinguir entre dos tipos de *Secuencias CAMARA*:

- *Secuencias de Cámara FINITAS*

Son aquellas que están constituidas por un número finito de fotogramas y que está concretado de antemano antes de que comience el proceso de dispensación de imágenes.

- *Secuencias de Cámara INFINITAS*

Son aquellas que recogen los fotogramas que, en el momento en que se esté ejecutando la *Sesión*, se estén recibiendo en tiempo real y que finalizan cuando el usuario así lo indique al sistema. Debido a su especial naturaleza, dentro de una sesión no podrá encontrarse una secuencia de cámara infinita seguida de ningún otro tipo de secuencia, aunque si puede ir precedida de cualquier tipo de secuencias.

Secuencias GENIMG

Este tipo de secuencias se corresponde con aquellas constituidas por imágenes sintéticas que, mediante unas rutinas software, son creadas directamente sobre la tarjeta digitalizadora. Para este propósito, existen unos ficheros para cada secuencia (Ficheros GMG) que contienen la

localización del fondo de la secuencia (fichero IMG), y de los elementos móviles que se irán desplazando sobre este (ficheros ICN), así como las posiciones relativas de los móviles sobre el fondo.

4.1.2.3.3. Sesiones

Una *Sesión* es un conjunto ordenado de secuencias con unas características de dispensación de imágenes establecidas de forma independiente para cada una de las secuencias que componen la *Sesión*. Consideraremos por tanto que se ejecuta una *Sesión* cuando, para cada *Secuencia* que la compone y atendiendo siempre a las características de ejecución de esa *Sesión*, el **MDI** extrae las imágenes de la *UDI* en que se encuentre almacenada la secuencia y se las envía al módulo del sistema **Intelligent SILHFinder** que realizó la solicitud de ejecución (módulo *Núcleo*). La ejecución de una sesión finaliza bien cuando el **MDI** haya dispensado la última imagen de la última secuencia de la sesión o bien cuando el usuario así lo solicite.

En la siguiente relación se recogen las características de ejecución que cada secuencia puede tener asociadas cuando está asignada a una sesión:

➤ *Modo de Reproducción*

Esta característica, establecida de forma global para toda la sesión, indica la forma en que serán dispensadas las imágenes que constituyen cada secuencia de la sesión y que puede ser de dos tipos:

- Modo Directo

El **MDI**, tras recibir la solicitud de ejecución de la *Sesión* y el número de fotogramas que de esta que ya se hubieran procesado en alguna ejecución anterior, dispensará las imágenes de la *Sesión* correspondiente de forma continua. Esto es, considerando el número de fotogramas que ya se hubieran procesado, calculará cuál es el fotograma que corresponde dispensar y a qué secuencia de la sesión pertenece, comenzando en ese momento a dispensar todos los fotogramas de esa secuencia y del resto de fotogramas de las demás secuencias de la sesión que aún no se hubieran dispensado, atendiendo siempre al resto de características de ejecución que para cada secuencia se hubieran establecido en la sesión.

La ejecución de la sesión se detendrá bien cuando se haya dispensado el último

fotograma de la última secuencia de la sesión o bien cuando el usuario así lo solicite.

- Modo Diferido

Es el modo de reproducción en que el **MDI** dispensará un único fotograma tras haber calculado cuál es – atendiendo al número de fotogramas de la sesión que ya se hubieran dispensado y al resto de características de ejecución establecidas para la secuencia – y a qué secuencia de la sesión pertenece.

- *Orden*

Indica el orden con respecto al resto de secuencias que componen la sesión en que la secuencia deberá ser ejecutada. De este modo se establece un conjunto ordenado de secuencias en función del valor de esta característica (de menor a mayor) y que determina cuál será la primera y sucesivas secuencias a ejecutar.

- *Paso*

Este campo indicará el intervalo de imágenes que tomaremos a la hora de reproducir las imágenes que constituyen una *Secuencia*. Se establece de forma única para cada una de las secuencias que constituyen la *Sesión* a ejecutar.

Por ejemplo, un paso de cinco imágenes nos indicará que cada cinco imágenes de la secuencia reproduciremos una.

- *Número de Repeticiones*

Indica, para cada secuencia de una sesión, el número de veces que debe repetirse la ejecución de esta. Como mínimo toda secuencia contenida en una sesión se ejecuta al menos una vez.

4.1.2.3.4. Agentes

Desde el punto de vista del **MDI** podemos decir que un *Agente* es el conjunto de datos que, además de las variables de entorno del sistema **Intelligent SILHFinder**, recoge la evolución del sistema a medida de que se someten a análisis distintas sesiones. En cualquier momento, un *Agente* tiene asociada una única *Sesión*.

Los datos asociados a un *Agente* que ha de conocer el **MDI** para poder ejecutar la sesión que éste lleve asociada, a parte de la propia sesión, son los siguientes:

- La *Edad del Sistema*. Que como ya dijimos representa el número de fotogramas que de esa sesión ya se hubieran dispensado en anteriores ejecuciones.
- El *Modo de Reproducción*. Que establece de forma global para todas las secuencias que compongan la sesión asociada esta característica de ejecución ya comentada.

4.1.3. El Módulo de Dispensación de Imágenes

La necesidad de un módulo que acceda a los distintos dispositivos en los que las secuencias de imágenes estén almacenadas y las dispense al sistema encargado de su análisis, es una característica común para cualquier aplicación que se pretenda desarrollar, y por tanto el *Módulo de Dispensación de Imágenes (MDI)* no sólo es necesario para el sistema **Intelligent SILHFinder** sino que es una parte básica y fundamental, sin cuya existencia éste quedaría reducido a una mera interfaz de usuario.

Si bien es cierto que este módulo está englobado dentro del sistema **Intelligent SILHFinder**, como una parte fundamental para su funcionamiento, también podría ser incluido en cualquier aplicación que necesitara proveerse de secuencias de imágenes, con el único requisito de que, los datos descriptivos de estas, estuvieran recogidos en la **Base de Datos SkeleTton**.

El *Módulo de Dispensación de Imágenes* actúa básicamente como una interfaz de sesiones, ofreciendo las imágenes que el sistema **Intelligent SILHFinder** va a analizar. Para ello, el módulo *Núcleo* debe comunicar al *MDI* la *Sesión* que se desea ejecutar.

Tras recibir la solicitud de ejecución de una *Sesión* concreta, el *MDI* debe extraer la información sobre el soporte en que se encuentren las *Secuencias* que la componen – las secuencias pueden estar almacenadas o ser obtenidas a partir de distintos dispositivos: cámara, magnetoscopio, etc. – y sus características de ejecución (dispensación) se encuentran en la **Base de Datos SkeleTton**.

Una vez recabada esta información, calculada la primera imagen que se debe dispensar y a qué *Secuencia* pertenece, deberá dispensarse junto con el resto de imágenes que componen la secuencia de forma secuencial y, en caso de existir más secuencias asociadas a dicha sesión, también deberán ser dispensadas. Básicamente, la tarea que debe realizar el *MDI* es la de devolver las imágenes que constituyen cada una de las secuencias que forman la sesión a ejecutar

– independientemente del dispositivo en que se encuentren almacenadas – atendiendo a los requerimientos que haga el módulo *Núcleo*.

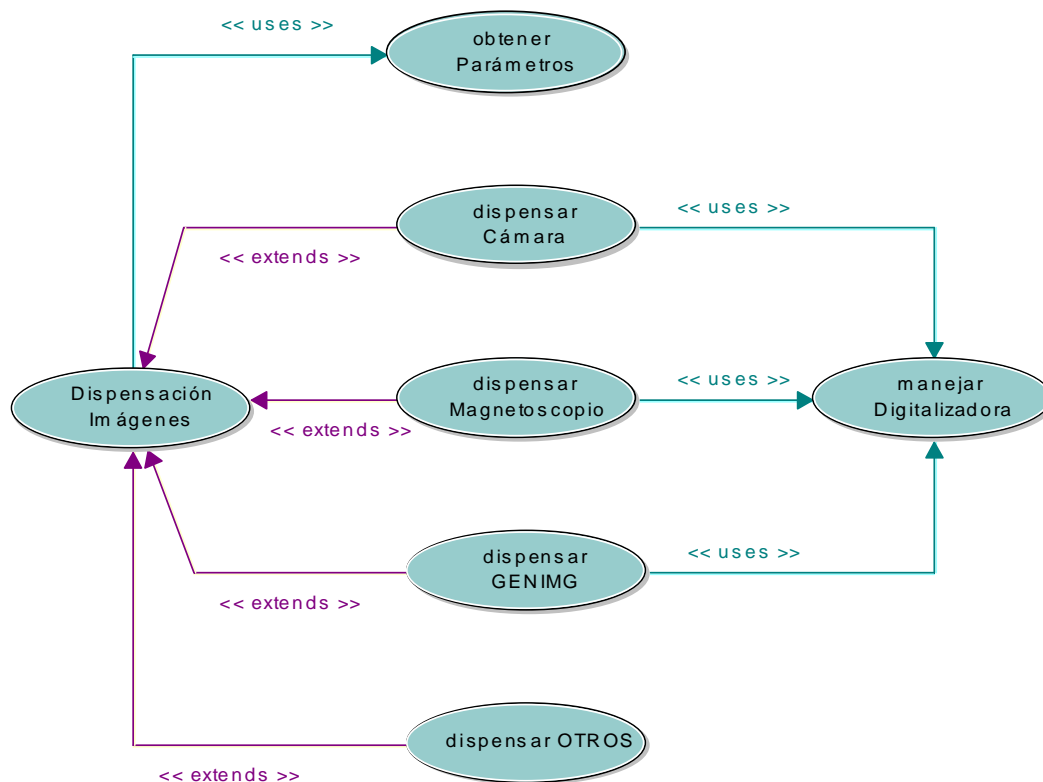


Figura 4.6. Diagrama de uso del Módulo de Dispensación de Imágenes

Cuando el usuario activa un *Agente* de la **Base de Datos SkeleTTon** a través de la interfaz de usuario *Vista*, el módulo *Núcleo*, que procesa esta solicitud de activación, accederá a la base de datos para adquirir las características asociadas al *Agente*, entre las que destacan la edad del sistema, el modo de reproducción y la sesión asociada. La edad del sistema nos indica el número de fotogramas de la sesión asociada, que ya se han procesado, puesto que el usuario puede detener la ejecución de una sesión sin necesidad de que se hayan dispensado todos los fotogramas de todas las secuencias que componen la sesión.

Llegado este momento, el usuario puede ‘arrancar’ el proceso de análisis de la *Sesión* asociada al *Agente*, momento en el cuál es activado el proceso de dispensación de imágenes. Tras recibir esta solicitud, el módulo *Núcleo*, comunicará al *MDI* cuál es la sesión que debe ejecutar (la sesión asociada al *Agente*) junto con la edad del sistema y el modo de reproducción establecidos. El *MDI* deberá extraer de la base de datos la información sobre el soporte en que

se encuentren las secuencias que la componen y sus características de ejecución. Una vez recabada esta información, calculada la primera imagen que debe procesar el sistema y a qué secuencia pertenece, irá dispensando las imágenes atendiendo al modo de reproducción establecido para la sesión y a las características de ejecución establecidas para la secuencia a la que pertenezca la imagen.

En función del tipo de dispositivo en que se encuentre almacenada la secuencia hablamos de *Secuencias MAGNETO*, *Secuencias CAMARA*, *Secuencias GENIMG*, y *Secuencias OTROS*.

En caso de que la secuencia a reproducir sea una *Secuencia OTROS*, el **MDI** deberá acceder a un fichero de tipo OTR, que contiene la información sobre la localización de las imágenes que constituyen la secuencia, y comenzar a dispensarlas en función de las características de ejecución que para esa se hubieran establecido dentro de la *Sesión*.

En el caso de que la secuencia a reproducir sea una *Secuencia MAGNETO*, el **MDI** deberá realizar, a grosso modo, las siguientes tareas: establecer la comunicación con el magnetoscopio, comprobar que la cinta insertada es la que contiene la secuencia a reproducir, localizar la secuencia y comenzar a dispensar los fotogramas que la componen en función también de las características de ejecución que en la base de datos se hubieran establecido para esta secuencia dentro de la *Sesión* que se esté ejecutando. Dado que no existe ninguna librería estándar que nos permita controlar el magnetoscopio, se ha desarrollado un conjunto de librerías que nos permita controlarlo, tanto para alcanzar los objetivos propuestos del **MDI** como para el desarrollo de futuras aplicaciones que necesiten del magnetoscopio.

Si la secuencia a reproducir es una *Secuencia GENIMG*, tras obtener las localizaciones de los ficheros que contienen los elementos móviles, el fondo y sus posiciones relativas, el **MDI** deberá ejecutar las rutinas software existentes con el fin de que se vaya generando la secuencia de imágenes sintética sobre la tarjeta digitalizadora. Tanto para el magnetoscopio y para la cámara, como para el software de generación de secuencias de imágenes (*Secuencias MAGNETO*, *CAMARA* y *GENIMG*), las imágenes son capturadas a través de la tarjeta digitalizadora, que recibe la señal de estos dispositivos. Para el acceso a la tarjeta digitalizadora que usa el sistema **Intelligent SILHFinder** (Matrox PIP 1024) se parte de una librería implementada en C estándar y distribuida por la marca comercial junto a la misma y que ha sido actualizada convenientemente para utilizar bajo Visual C++.

La ejecución de la *Sesión* finalizará bien cuando se haya dispensado la última imagen de la última *Secuencia* de la *Sesión* en curso o bien cuando el usuario así lo solicite. En caso de

producirse algún error durante el proceso de dispensación de imágenes, ficheros corruptos, acceso a los dispositivos que contienen las secuencias de imágenes,... también será dada por finalizada la ejecución de la *Sesión*.

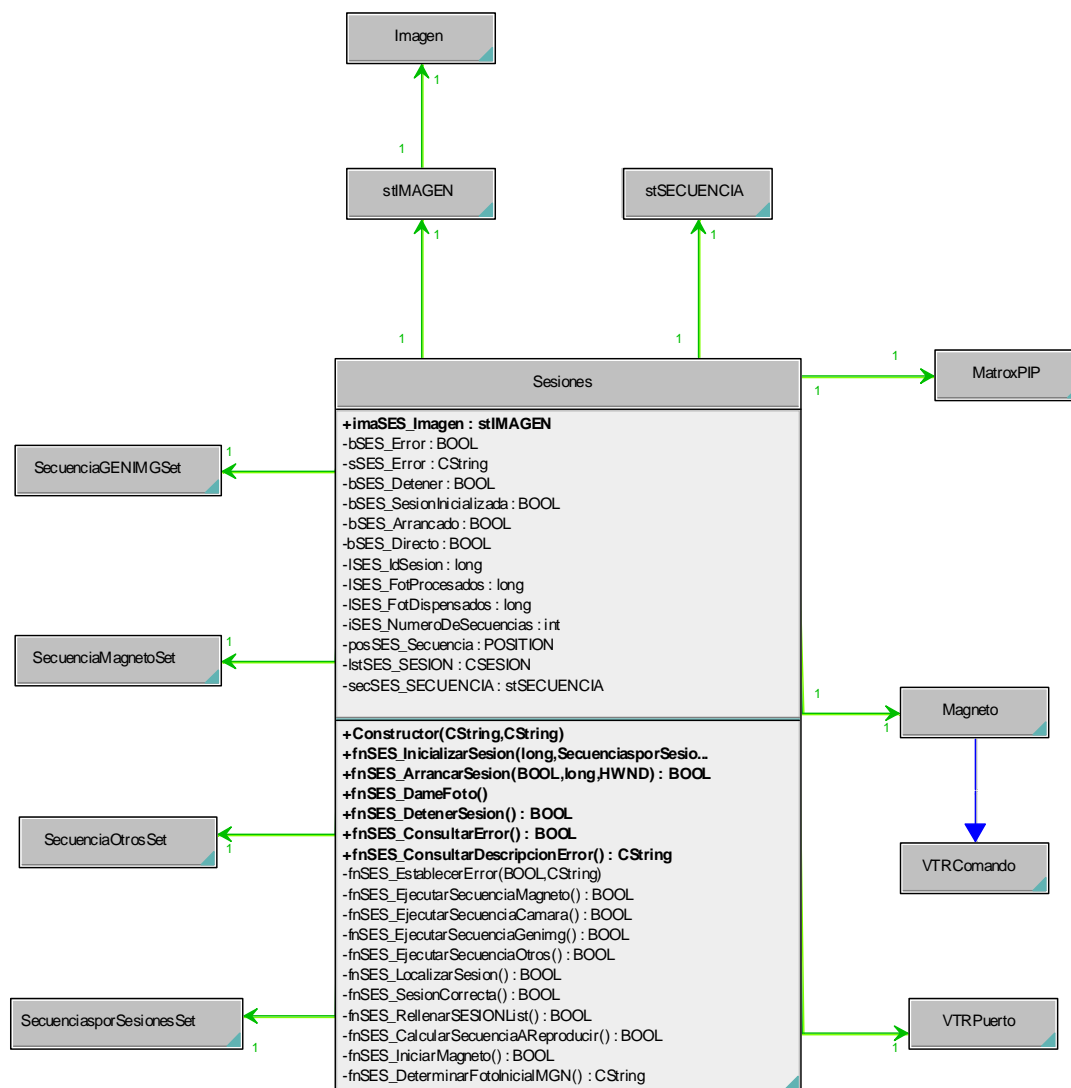


Figura 4.7. Diagrama de clases del Módulo de Dispensación de Imágenes

La figura 4.7. ofrece el diagrama de clases del *Módulo de Dispensación de Imágenes*. La clase base este modulo se denomina *Sesiones* y se relaciona básicamente con las clases de manipulación de las imágenes (*Imagen*), de la placa digitalizadora (*MatroxPIP*) y del magnetoscopio (*Magneto*, *VTRPuerto*). Para el tratamiento de la información de las secuencias que componen una sesión, la clase *Sesiones* trabaja con las clases de tipo registro de base de

datos según el tipo de secuencia ofertada (SecuenciaGENIMGSet, SecuenciaMagnetoSet, SecuenciaOtrosSet).

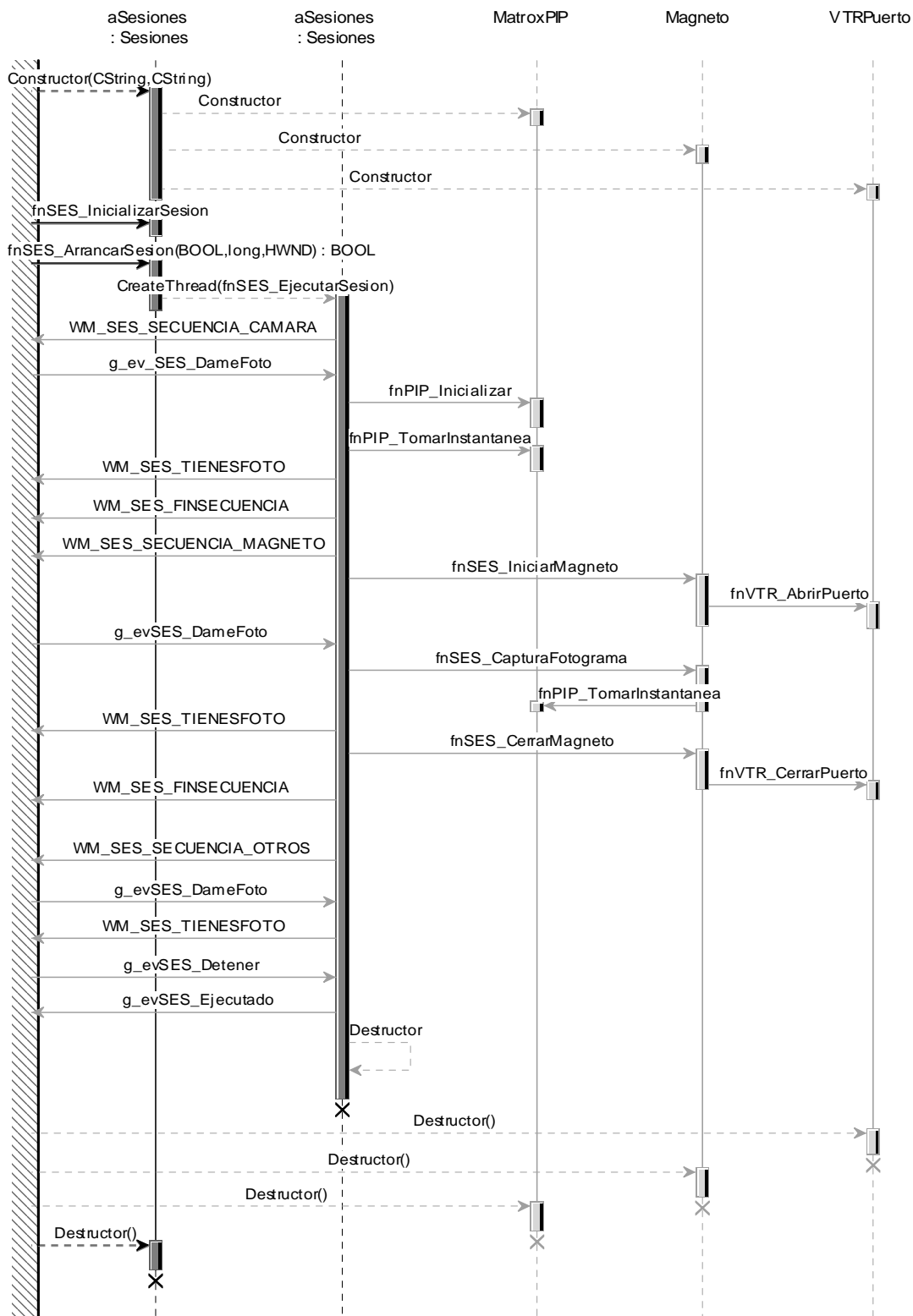


Figura 4.8. Diagrama de secuencias del Módulo de Dispensación de Imágenes

El diagrama de secuencias del *MDI* se ofrece igualmente en la figura 4.8.

4.1.4. El módulo Vista

El módulo *Vista*, como ya se ha comentado brevemente con anterioridad, ofrece el interfaz del sistema **Intelligent SILHFinder** con el usuario. Tal como se puede ver en el diagrama de uso siguiente, el actor Usuario puede realizar las siguientes acciones:

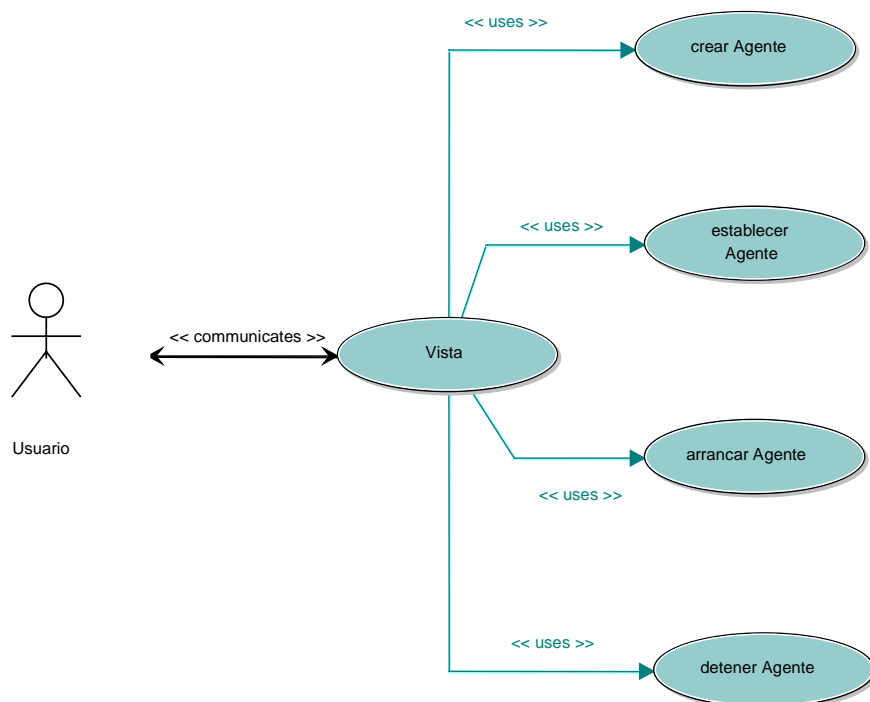


Figura 4.9. Diagrama de uso del módulo Vista

crear Agente: de este modo, el usuario tiene acceso a la **Base de Datos SkeleTton** para crear un nuevo *Agente*. Para ello, tendrá que **definir UDI**, **definir Secuencia**, **definir Sesión**, **definir Agente**. Estos son los pasos lógicos (en este mismo orden) para crear un nuevo *Agente* para el sistema **Intelligent SILHFinder**.

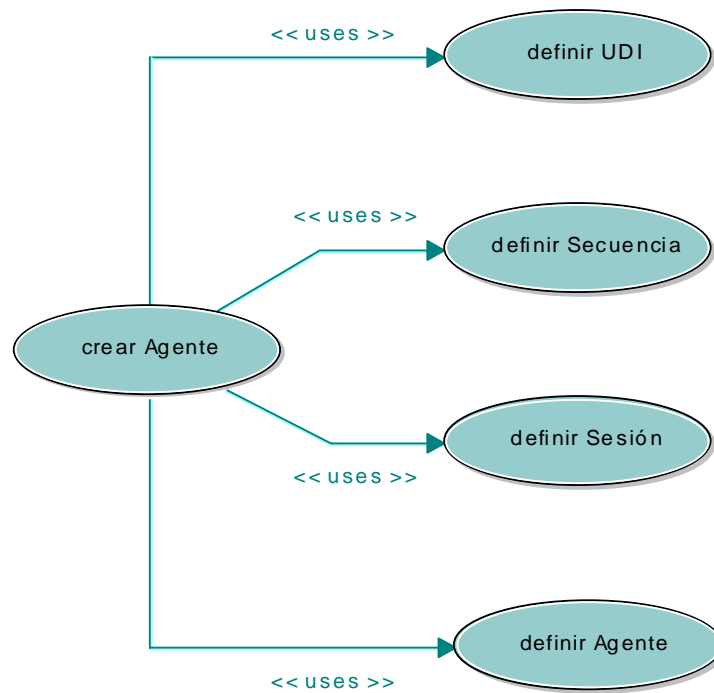


Figura 4.10. Diagrama de uso del uso crear Agente

establecer Agente: esta es la vía que ofrece el sistema para definir con qué *Agente* se desea trabajar a partir de este momento. Evidentemente, el *Agente* habrá sido creado con anterioridad en la **Base de Datos SkeleTTON**. Esta opción se describe más adelante en toda extensión.

arrancar Agente: este es el modo de poner a trabajar al *Agente* actual. Ello conlleva básicamente el arrancar los demás módulos del sistema, que hasta el momento no están realizando ninguna labor, es decir, el *Núcleo*, el *MDI* y las *Capas de Procesamiento de Imágenes*. A partir de este instante, el *Agente* empezará a sufrir cambios en sus parámetros debido al procesamiento de las imágenes de entrada.

detener Agente: se detiene automáticamente el *Agente* cuando acabe la *Sesión* asociada o a petición del actor Usuario. Esto causa que se pare el procesamiento de las imágenes, y por tanto, los módulos siguientes: el *Núcleo*, el *MDI* y las *Capas de Procesamiento de Imágenes*.

El uso **establecer Agente** es el más importante desde el punto de vista de la parametrización del *Agente*. Son varios los pasos a seguir para una correcta configuración del *Agente*. El diagrama 4.11 es la muestra de lo que se está indicando.

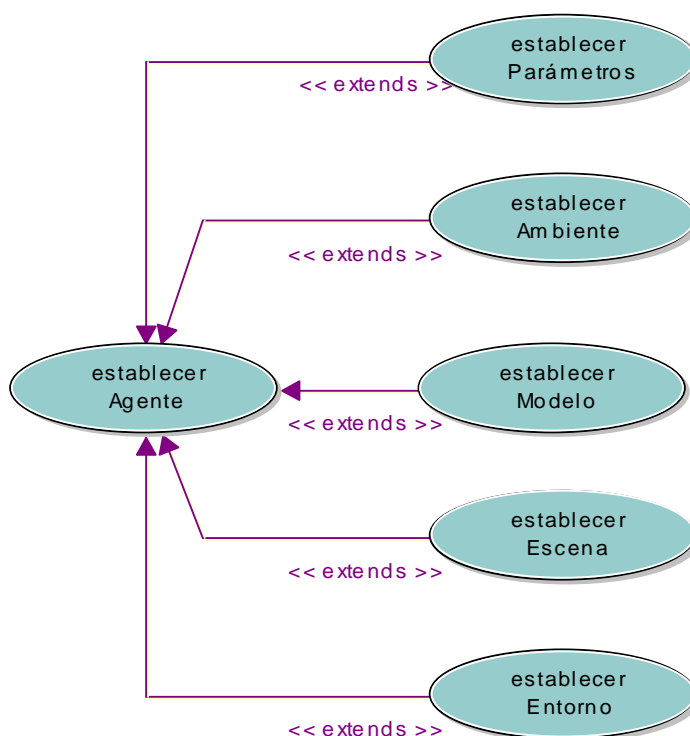


Figura 4.11. Diagrama de uso del uso establecer Agente

En efecto, vemos cómo se hace necesario **establecer Parámetros**, **establecer Ambiente**, **establecer Modelo**, **establecer Escena** y **establecer Entorno**.

El parámetro fundamental de un *Agente* es la Edad del Sistema, que va a permitir arrancarlo a partir de un fotograma determinado de las secuencias relacionadas con él.

El Ambiente de un *Agente* define todos los pasos a seguir en el procesamiento de las imágenes de las secuencias relacionadas con él. Como puede apreciarse en el diagrama de uso siguiente, pueden definirse a este nivel el preprocesamiento, la segmentación, el modelo y el postprocesamiento a realizar sobre las imágenes, así como el tipo de escena en la que se engloba el *Agente*.

El uso **establecer Modelo** está pensado para elegir el tipo de modelo que se desea aplicar a las imágenes asociadas al *Agente*. Evidentemente, el modelo que centra nuestra atención es el **Modelo de Interacción Lateral en Computación Acumulativa (ILCA)**.

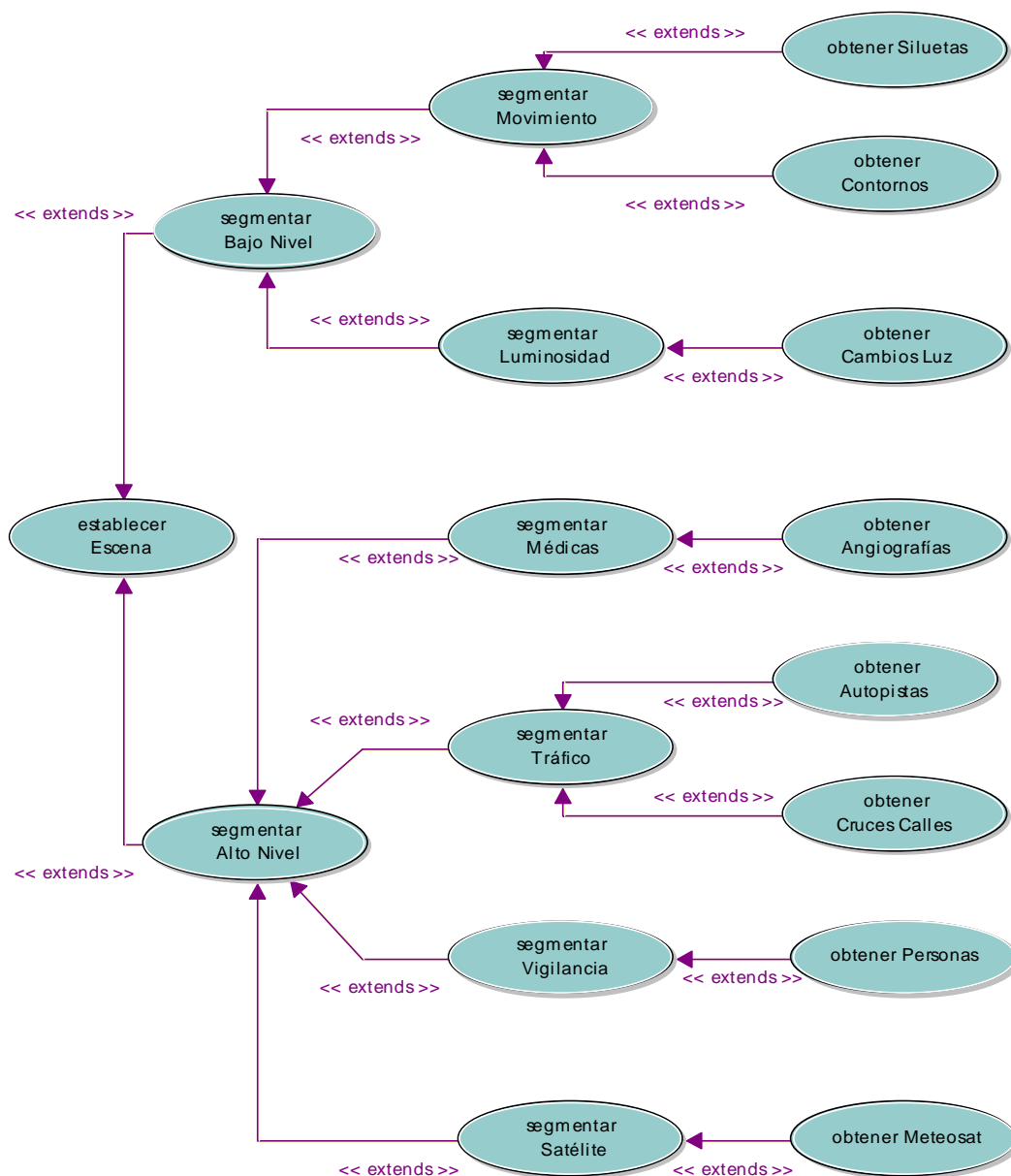


Figura 4.12. Diagrama de uso del uso establecer Escena

Esta versión del sistema **Intelligent SILHFinder** permite aplicar el modelo seleccionado a numerosas escenas. En la configuración actual es poder realizar las opciones descritas en el diagrama de uso 4.12.

Finalmente queda hablar del uso **establecer Entorno** que va a permitir:

- **establecer HW/SW:** permite elegir implementar cada una de las capas del modelo de interacción lateral en computación acumulativa mediante software o alguna placa hardware
- **establecer Directorios:** permite definir los directorios por defecto del sistema **Intelligent SILHFinder**
- **establecer Tratamiento:** permite parametrizar la interfaz de usuario frente a las imágenes de entrada al sistema
- **establecer Traza:** permite determinar las condiciones de traza del sistema para posibles depuraciones

El diagrama de clases del módulo *Vista* puede apreciarse en la figura 4.13.

Interesante es igualmente el diagrama de estados de la clase *SkeletonView*, es decir la clase principal del módulo *Vista*. Permite apreciar las transiciones de estado del sistema **Intelligent SILHFinder**.

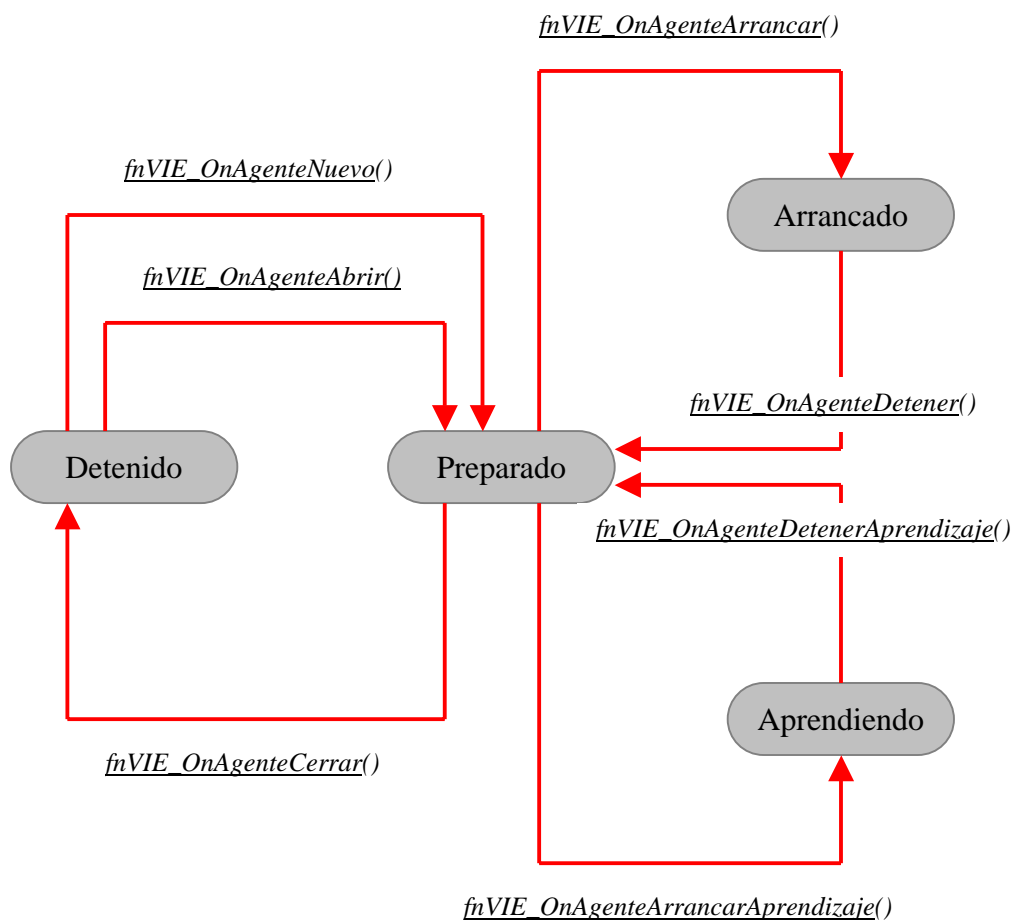


Figura 4.14. Diagrama de estados de la clase *SkeletonView*

Detenido es el estado en el cual el sistema no reconoce aún ningún *Agente*. Una vez elegido un *Agente*, se pasa:

- si estamos en fase de aprendizaje al estado *Preparado* y, finalmente, al arrancar el *Agente*, se pasa a *Aprendiendo*.
- si estamos en fase de ejecución normal al estado *Preparado* y, finalmente, al arrancar el *Agente*, se pasa a *Arrancado*.

4.1.5. El módulo Núcleo de la aplicación

El módulo *Núcleo* de la aplicación **Intelligent SILHFinder** es el nexo de unión entre los demás módulos. Para que las órdenes del usuario lleguen desde la *Vista* hasta el *Módulo de Dispensación de Imágenes* o a las *Capas de Procesamiento de Imágenes*, deben pasar por el *Núcleo*. Del mismo modo, toda la información que ha de viajar desde el *Módulo de Dispensación de Imágenes* o desde las *Capas de Procesamiento de Imágenes* para ser visualizada en pantalla a través de la *Vista* pasa por el módulo *Núcleo*.

El módulo *Núcleo*, en su vertiente de módulo de paso, es creado desde el módulo *Vista* y crea, a su vez, los módulos *MDI* y *Capas de Procesamiento de Imágenes*. La figura 4.15. presenta un diagrama de secuencia del módulo *Núcleo* de la aplicación. En este caso, como puede apreciarse se ofrece para el caso especial en que la escena elegida para el *Agente* es la de la obtención de las siluetas de los objetos deformables a partir del cambio de luminosidad.

4.1.6. El módulo Capas de Procesamiento de Imágenes

El módulo *Capas de Procesamiento de Imágenes* incluye todas aquellas tareas destinadas al tratamiento digital de las imágenes, según las especificaciones o parametrización efectuada en el módulo *Vista*. Recordemos que toda imagen de entrada al sistema **Intelligent SILHFinder** para un *Agente* dado, pasará por los siguientes pasos de proceso:

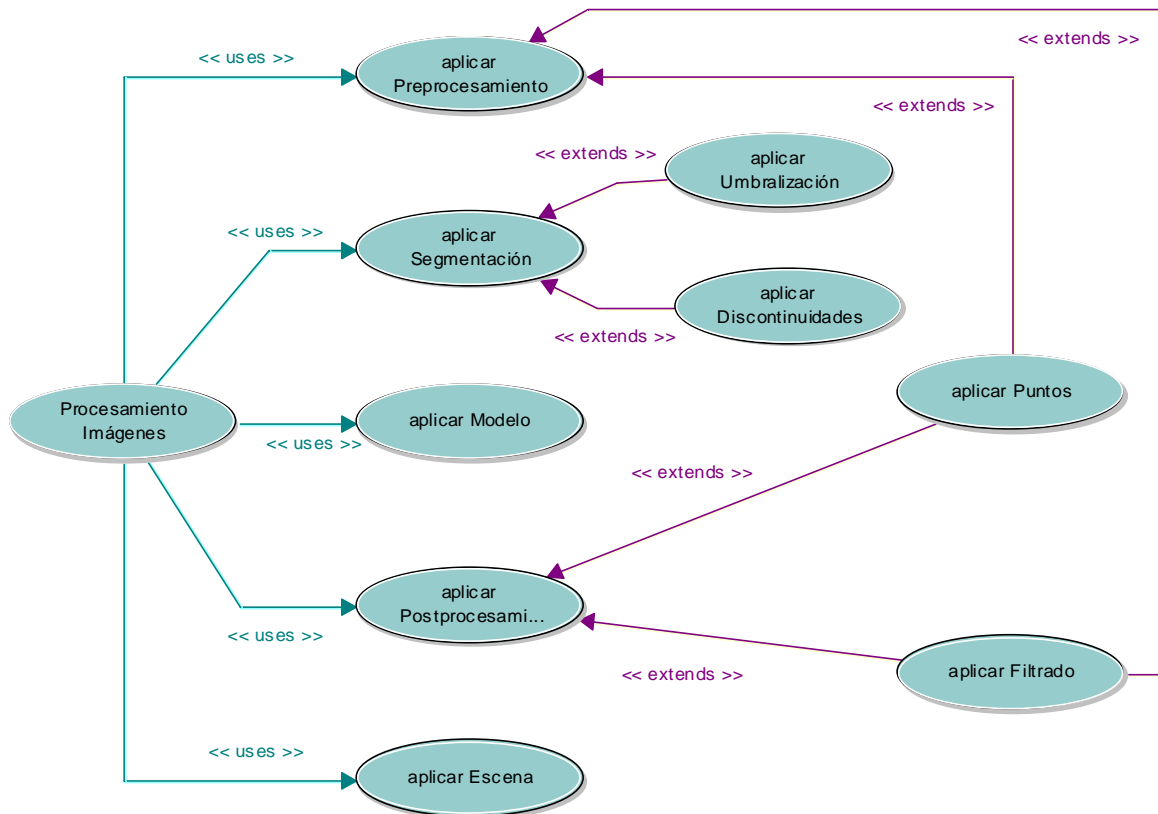


Figura 4.16. Diagrama de uso del módulo Capas de Procesamiento de Imágenes

- **Preprocesado:** permite determinar qué acción (acciones) de preprocesado se deben realizar sobre las secuencias de imágenes de entrada que nutren al *Agente*. Estas serán las primeras operaciones sobre las imágenes de entrada (tras las pertinentes validaciones de formatos, tamaños, ...). El preprocesado tiene una amplia finalidad en este contexto, destinado fundamentalmente a eliminar ruidos en las imágenes. El preprocesado se ha dividido en dos grandes bloques, Procesamiento de Puntos y Filtrado Espacial.
- **Segmentación:** permite determinar qué acción (acciones) de segmentación se deben realizar sobre las secuencias de imágenes de entrada previamente preprocesadas y que nutren al *Agente*. La segmentación subdivide una imagen en sus partes constituyentes u objetos. La segmentación se ha dividido en el sistema **Intelligent SILHFinder** dos grandes bloques, a saber la Umbralización y la Detección de Discontinuidades.
- **Modelo** (ver apartado 4.1.6.1)

- **Postprocesado:** permite determinar qué acción (acciones) de postprocesado se deben realizar sobre las secuencias de imágenes que ya han pasado por el modelo. El postprocesado también se ha dividido en dos grandes bloques, Procesamiento de Puntos y Filtrado Espacial.
- **Escena** (ver apartado 4.1.6.2)

4.1.6.1. El nivel de Interacción Lateral en Computación Acumulativa

En el diagrama de uso anterior no se ha entrado en detalle en el uso **aplicar Modelo**, sino que se dedica este apartado al mismo.

En primer lugar, cabe señalar que el sistema **Intelligent SILHFinder** está concebido para incorporar cualquier tipo de modelo dedicado a la obtención de siluetas. Pero a la fecha, solamente se ha incorporado el **Modelo de Interacción Lateral en Computación Acumulativa (ILCA)**. También es necesario tener en cuenta que la **Capa 0. Segmentación por Bandas de Nivel de Gris** ha pasado en el sistema **Intelligent SILHFinder** al uso aplicar Segmentación, con lo que aquí nos resta hablar únicamente de las capas 1 a 3 del modelo. Por ello, tendremos:

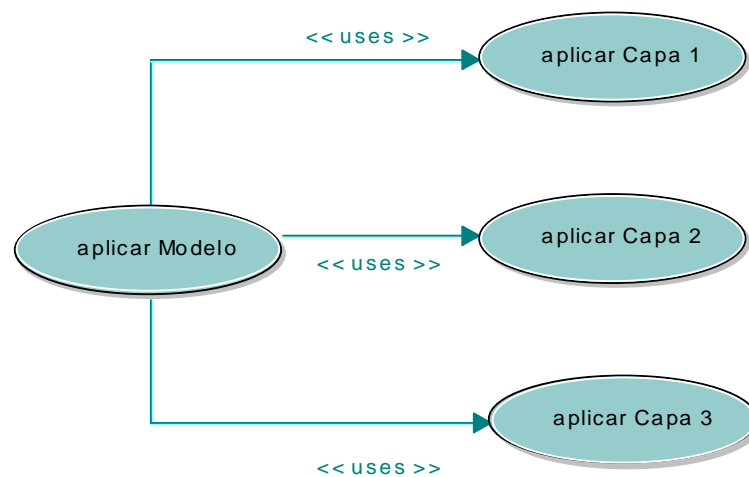


Figura 4.17. Diagrama de uso del uso aplicar Modelo

donde **aplicar Capa 1** es el uso para **Capa 1. Interacción Lateral para la Computación Acumulativa**, **aplicar Capa 2** es el uso para **Capa 2. Interacción Lateral para el Reparto de**

Carga por Bandas de Nivel de Gris y **aplicar Capa 3** es el uso para **Capa 3. Interacción Lateral para la Fusión en Objetos.**

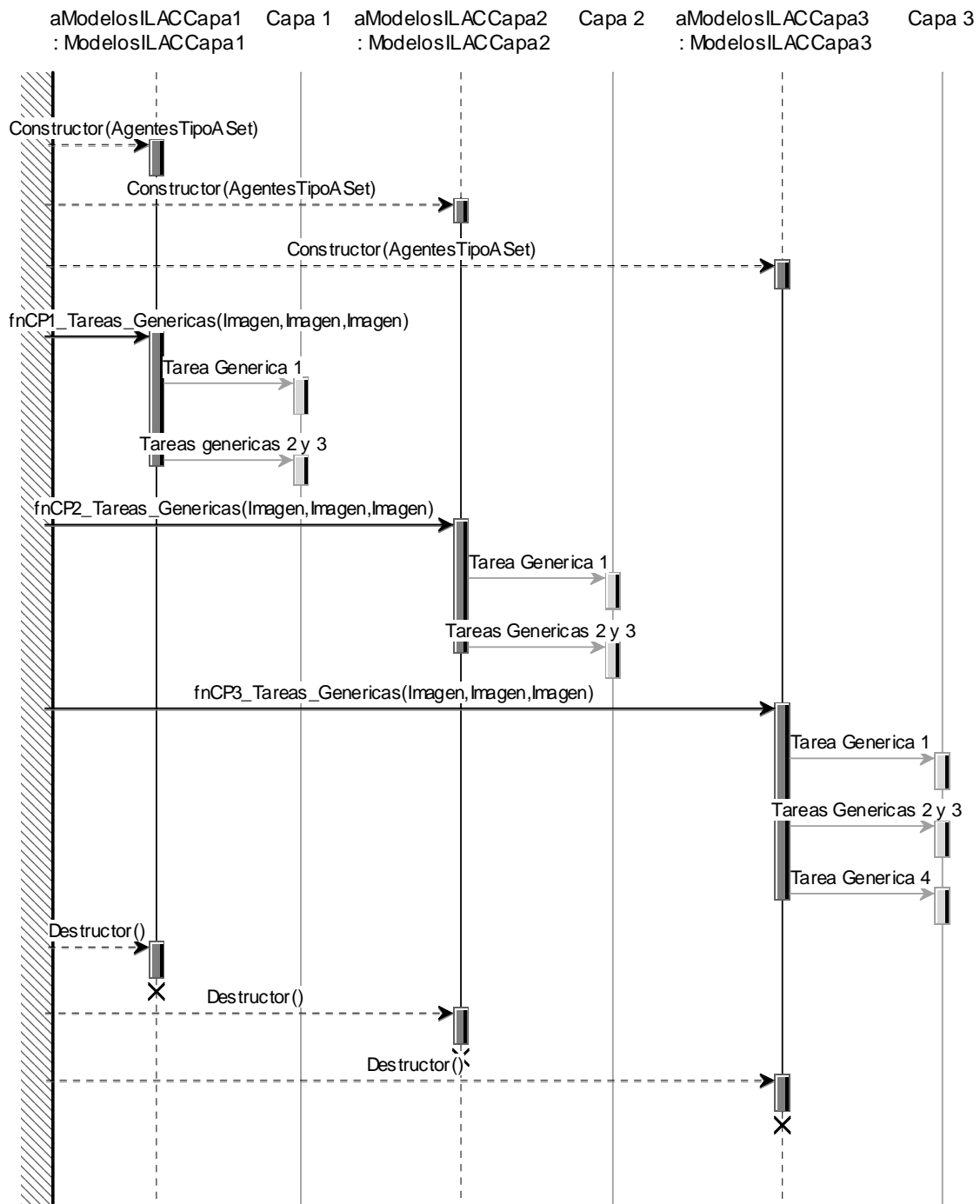


Figura 4.18. Diagrama de secuencias del Modelo de Interacción Lateral en Computación Acumulativa (ILCA)

Las tres clases del **Modelo de Interacción Lateral en Computación Acumulativa (ILCA)** son ModelosILCACapa1, ModelosILCACapa2 y ModelosILCACapa3, respectivamente.

También es de interés el diagrama de secuencias del **Modelo de Interacción Lateral en Computación Acumulativa (ILCA)**, ya que en él se ofrece el modo de llamar a las diferentes tareas genéricas en cada una de las capas del modelo.

4.1.6.2. El nivel de Escena

Dada la gran variedad de posibles escenas a las que se enfrenta el sistema **Intelligent SILHFinder**, se ofrece una explicación de las mismas en este apartado.

Las Escenas actualmente en vigencia son:

- **Aplicaciones Generales:** Aquí irán aquellos *Agentes* que no están asociados a un tipo de aplicación claro y específico. Por ello, se ha optado por realizar la siguiente diferenciación:
 - **Segmentación:** La tarea que se acomete a este nivel es la obtención (sin mayores especificaciones de alto nivel) de: (a) **Siluetas:** Este tipo de Escena obtiene las siluetas de todos los objetos móviles presentes en una secuencia indefinida de imágenes. (b) **Contornos:** Este tipo de Escena aporta un paso más respecto de la obtención de las siluetas, a saber, la obtención de los contornos de todas las siluetas obtenidas.
 - **Luminosidad:** Se ha llegado a la conclusión que ciertos modelos, entre los que se encuentra el **Modelo de Interacción Lateral en Computación Acumulativa**, permiten obtener las siluetas de los objetos presentes en una sola imagen a partir del cambio forzado de la iluminación de la imagen. Ello se engloba en la opción: **Cambios de Luz.**
- **Aplicaciones Específicas:** En este apartado se incluyen algunas aplicaciones de alto nivel, en las que nos hemos adentrado en el dominio de conocimiento, y hemos inyectado este conocimiento al *Agente*. Se ofrecen en esta versión de **Intelligent SILHFinder** algunas aplicaciones de interés, como pueden ser el manejo de:

- **Imágenes Médicas. Angiografías:** Esta es una aproximación volumétrica a las imágenes vasculares, que permite ver las estructuras vasculares, buscando posibles lesiones.
- **Imágenes Satélite. Meteosat:** La aplicación elegida en este caso es el seguimiento de la nubosidad a partir de imágenes tomadas por el satélite Meteosat.
- **Escenas de Tráfico. (a) Autopistas:** Aplicación de clasificación y conteo de vehículos en una autopista. **(b) Cruces de Calles:** La misma aplicación en cruces de calles, pero más enfocada a conocer la dirección que toman los vehículos.
- **Escenas de Vigilancia. Personas:** Esta aplicación clasifica las personas en base a su velocidad de paso.

4.2. Vista del usuario: Aplicación Base de Datos SkeleTTon

4.2.1. Menús de la aplicación

La pantalla inicial que se muestra al lanzar la aplicación realiza las tareas propias de un diálogo de acceso, ofreciendo al usuario la posibilidad de continuar la ejecución de la aplicación (botón *Menú de la aplicación*) o bien de abandonar la misma (botón *Salir de la aplicación*).

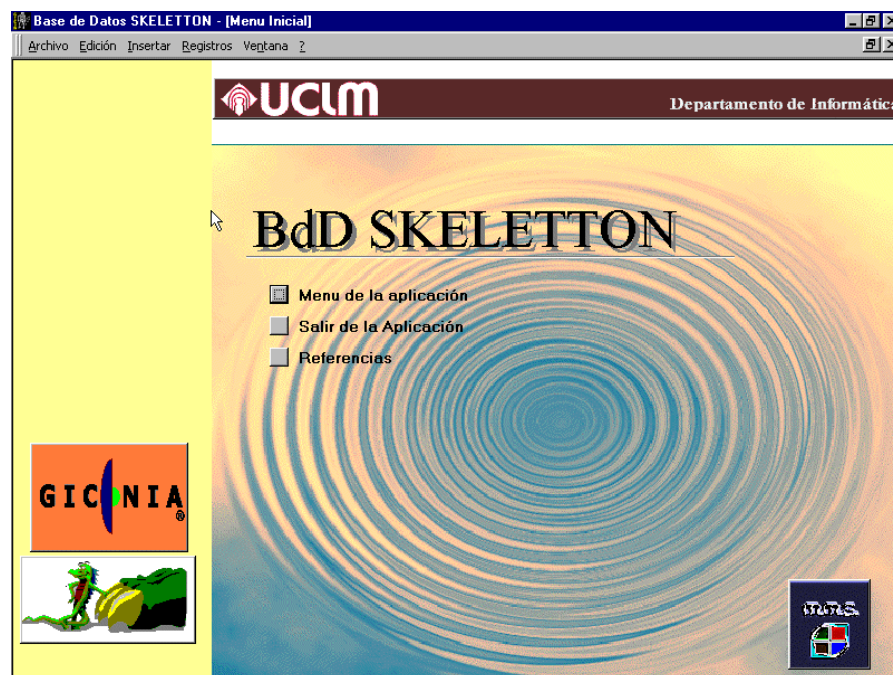


Figura 4.19. Menú inicial

Si selecciona el botón *Menú de la aplicación*, accederá a un formulario similar al que se muestra en la figura 4.20. que ofrece la posibilidad de acceder al sistema de menús con los que podrá acceder los distintos formularios que permiten al usuario la introducción, visualización y modificación de los datos así como la impresión o visualización de informes.

Como podemos observar, los cuatro primeros botones nos permiten acceder a los datos contenidos en la base de datos a distintos niveles. Podemos considerar que los datos contenidos en la base de datos se irán almacenando atendiendo a una estructura jerárquica en la que los elementos de menor rango serían las *Secuencias* de imágenes y los de mayor rango serían los *Proyectos* y los *Agentes*, siendo las *Sesiones* un conjunto ordenado de *Secuencias*.

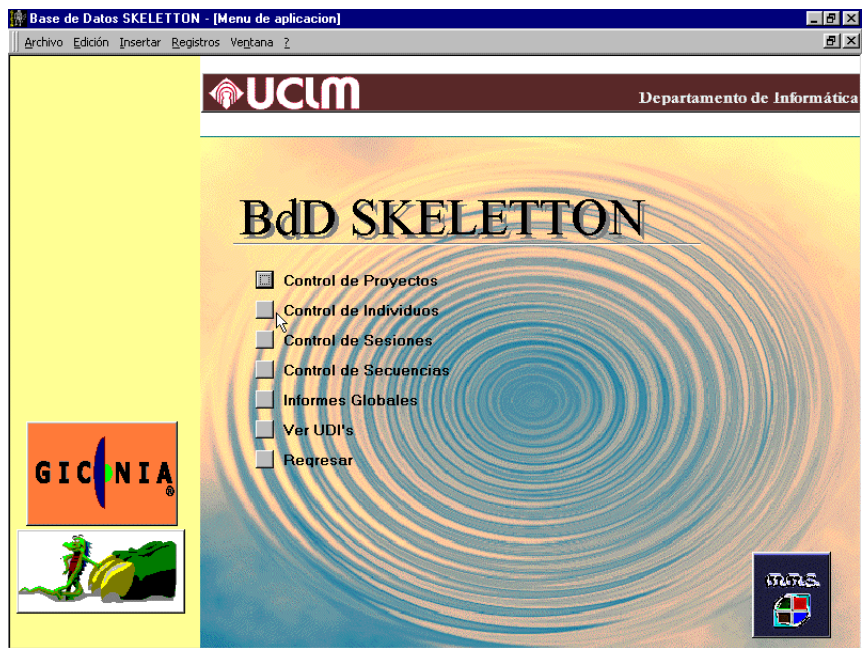


Figura 4.20. Menú de la aplicación

Las *Secuencias* de imágenes, como ya sabemos, podrán estar almacenadas en diversos dispositivos (Unidades de dispensación de imágenes (UDI's)) cuya relación podemos ver pulsando el botón *Ver UDI's*.

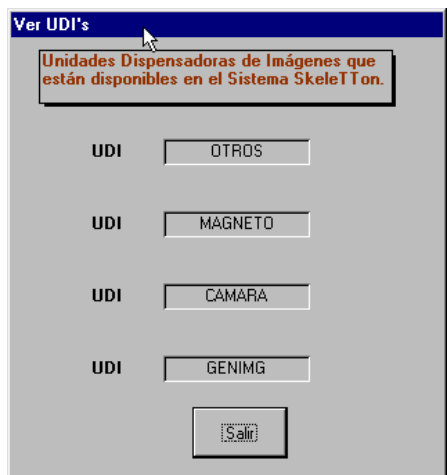


Figura 4.21. Unidades de dispensación de imágenes

- OTROS hace referencia a aquellas secuencias de imágenes que se encuentran almacenadas en ficheros del tipo OTR.

- MAGNETO identifica aquellas secuencias de imágenes que se encuentran almacenadas en cintas de vídeo (magnetoscopio SONY BETACAM-SP, modelo UVW-1700GP).
- CAMARA hace referencia a aquellas secuencias que se reciben en tiempo real desde una cámara digital.
- La UDI GENIMG identifica aquellas secuencias que se generan en tiempo de ejecución a partir de un software de generación de imágenes sintéticas.

4.2.2. Menú Control de secuencias

Pulsando sobre el botón *Control de Secuencias* del *Menú de la aplicación* accederá al formulario que se muestra en la siguiente figura:

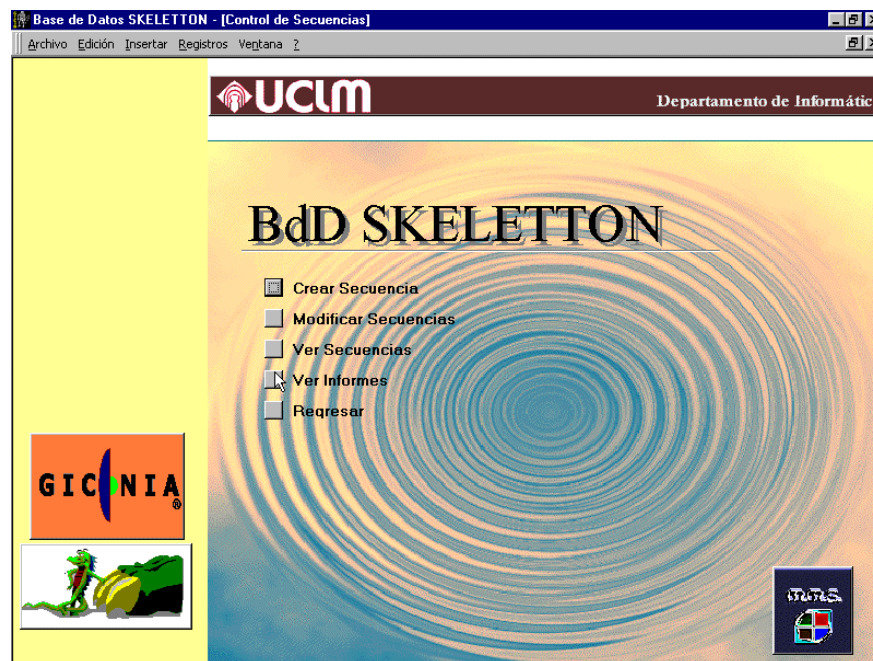


Figura 4.22. Control de secuencias

Como podemos observar, desde este formulario podremos realizar las tareas básicas de control sobre los datos relativos a las secuencias, como son crear nuevas secuencias, modificar secuencias existentes o simplemente visualizar los datos relativos a las secuencias que hayan sido creadas previamente.

4.2.2.1. Opción Crear secuencias

Para crear una nueva secuencia, esto es, darla de alta en la base de datos, debemos pulsar el botón *Crear Secuencia* del menú *Control de Secuencias* (ver figura 4.23.). Desde el formulario que se muestre en pantalla podremos definir el “Nombre” que queremos asignar a la secuencia y la UDI en la que se encuentra almacenada, además de seleccionar el autor de nuestra secuencia (ver figura 4.24.). En caso de que el autor no exista, bastará con pulsar la tecla INTRO, momento en que se nos mostrará el siguiente cuadro para dar de alta al nuevo autor:

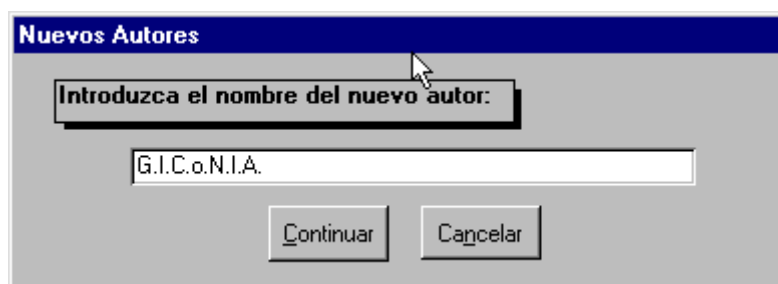


Figura 4.23. Nuevos autores

Una vez hayamos introducido todos los datos podremos salvar la *Secuencia* pulsando sobre el botón *Guardar*, tras lo cual la aplicación solicitará que procedamos a detallar la secuencia. En función del tipo de UDI que hayamos seleccionado, se nos mostrará el correspondiente formulario de detalle para la secuencia que estamos creando. Estos formularios de detalle son los encargados de recoger aquellas características con las que el sistema Intelligent SILHFinder podrá acceder a la UDI en que la secuencia se encuentre almacenada y localizarla.

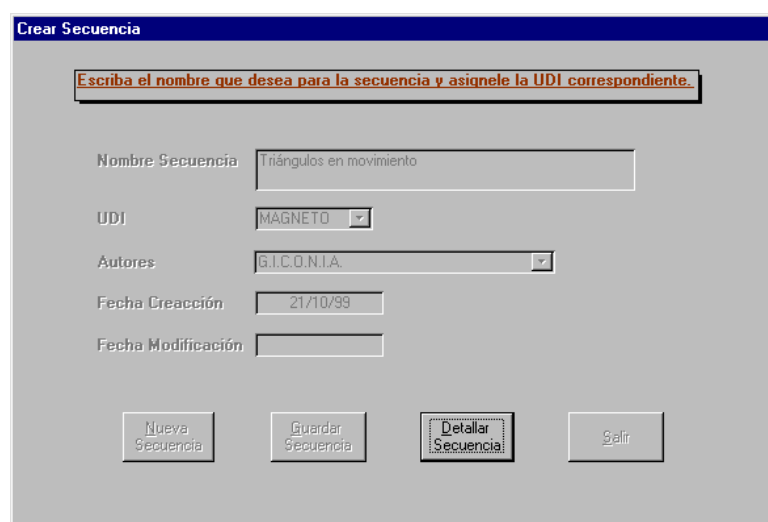


Figura 4.24. Formulario de creación de secuencias

4.2.2.1.1 Detalle de secuencias MAGNETO

Si la secuencia que hemos creado es del tipo *Magneto* (UDI = MAGNETO), al pulsar el botón *Detallar Secuencia*, el formulario de detalle solicitará que introduzcamos los siguientes parámetros (ver figura 4.25.):

Figura 4.25. Detallar secuencias Magneto

- **Nombre de cinta / secuencia** : Se corresponde con los cuatro bytes de mayor peso del Dato UBIT almacenado en la cinta de vídeo para la secuencia a dar de alta.
- **Posición Inicial**: Indica la posición LTC en la que comienza la secuencia. Por consenso todas las secuencias Magneto deben comenzar en la posición 00:00:00:00. El sistema Intelligent SILHFinder considerará que las imágenes correspondientes a la secuencia comienzan a partir de la posición 00:00:07:00, destinándose desde la posición 00:00:00:00 hasta la posición 00:00:07:00 para el almacenamiento de diversos datos de control.
- **Posición Final**: Indica la posición LTC en la que finaliza la secuencia Magneto.

4.2.2.1.2. Detalle de secuencias OTROS

En caso de que la secuencia que hayamos creado sea del tipo OTROS (UDI = OTROS) los datos solicitados serán los siguientes (ver figura 4.26.):

- **Path**: Indica la ruta completa en que se encuentra el fichero OTR que contiene la localización de las distintas imágenes que componen la Secuencia (por ejemplo, C:\Secuencias)
- **Nombre del Fichero**: Indica el nombre completo (nombre + extensión) del fichero OTR que contiene la localización de las distintas imágenes que componen la Secuencia (por ejemplo, Triangu1.OTR)

- **Número de Imágenes:** Representa el número de localizaciones de imágenes que contiene el fichero OTR.

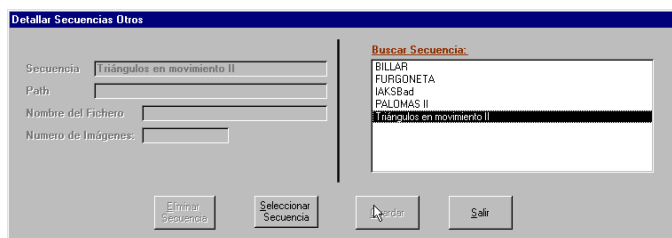


Figura 4.26. Detallar secuencias Otros

4.2.2.1.3. Detalle de secuencias GENIMG

En caso de que la secuencia que hayamos creado sea del tipo GENIMG (UDI = GENIMG) los datos solicitados serán los siguientes (ver figura 4.27.):

- **Path:** Indica la ruta completa en que se encuentra el fichero GMG que contiene la localización de los distintos elementos que serán usados por el software de generación de imágenes a fin de generar la secuencia (fondo y elementos móviles).
- **Nombre del fichero:** Determina el nombre completo del fichero GMG (por ejemplo, Triangu1.GMG)
- **Fotograma Inicial & Fotograma Final:** Indican el número del fotograma en que comienza/finaliza la secuencia Genimg.



Figura 4.27. Detallar secuencias Genimg

4.2.2.1.4. Detalle de secuencias CAMARA

En caso de que la secuencia que hayamos creado sea del tipo CAMARA (UDI = CAMARA) no se solicitará ningún dato auxiliar, ya que las secuencias *Cámara* son recibidas en

tiempo real y por tanto el número de fotogramas a procesar puede variar en función de la *Sesión* a la que pertenezca la *Secuencia*.

4.2.2.2. Opción Modificar secuencias

Desde el menú de control de Sesiones, pulsando sobre el botón *Modificar Secuencias* accederá al formulario que se muestra en la siguiente figura, desde el cuál podrá seleccionar la secuencia que desea modificar, bien pulsando sobre el botón *Modificar Secuencia* una vez la haya seleccionado o bien haciendo doble click sobre la misma. Con esto conseguirá que se despliegue el formulario de detalle correspondiente para la el tipo de secuencia elegida.

Si tras elegir una secuencia pulsa sobre el botón *Imprimir Secuencia* obtendrá un listado en que se le describan las características que en ese momento tiene asociada dicha *secuencia*.

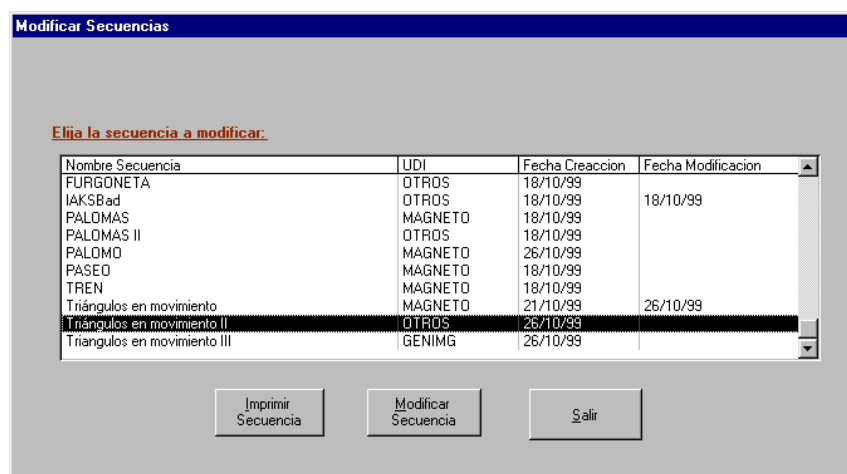


Figura 4.28. Modificar secuencias Genimg

Es fundamental no modificar ninguna secuencia que esté siendo usada por alguna sesión, ya que en caso contrario, las características asociadas a esa secuencia dentro de la sesión correspondiente pueden inducir a un mal funcionamiento de los algoritmos de dispensación de imágenes que posteriormente vayan a ser aplicados sobre dichas sesiones. Considérese, por ejemplo, el caso más elemental en que se elimine una secuencia que ha sido asignada previamente a una sesión.

4.2.2.3. Opción Ver secuencias

Si lo único que desea es ver las secuencias que hay generadas para un tipo concreto de UDI, puede pulsar sobre el botón *Ver Secuencias* del menú de control de secuencias. En este formulario, puede seleccionar en la caja de diálogo de la esquina superior derecha el tipo de UDI asociado a las secuencias que desea visualizar, además de modificar las características asociadas a dichas secuencias a partir de sus formularios de detalle.

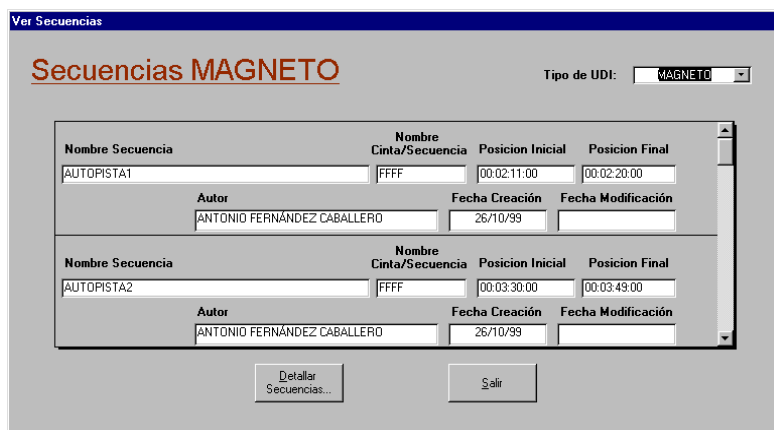


Figura 4.29. Ver secuencias

4.2.2.4. Opción Ver informes de secuencias

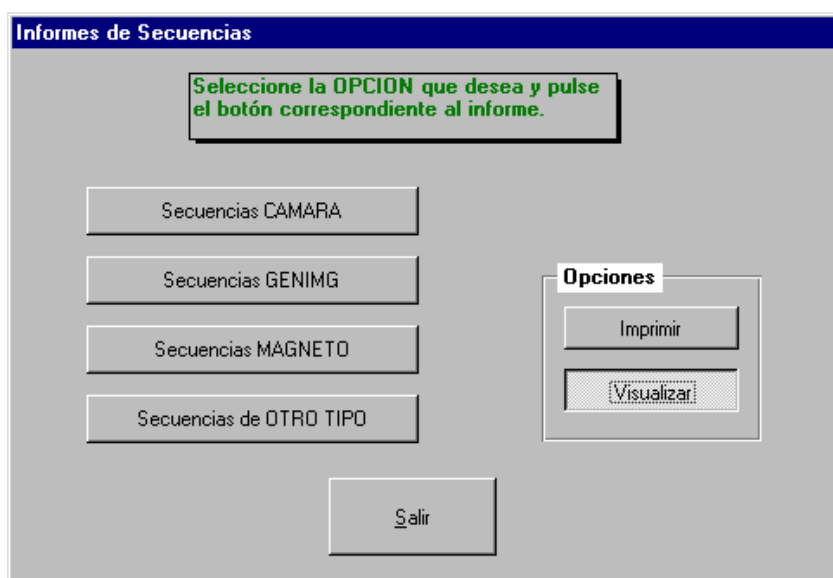


Figura 4.30. Informes de secuencias

Desde el formulario de *Informes de Secuencias* se puede visualizar o imprimir una relación completa de todas las secuencias generadas atendiendo al tipo de UDI que utilizan. Para ello bastará con seleccionar la acción que se desea realizar - Visualizar o Imprimir - y pulsar sobre el tipo de secuencia que deseamos activar.

4.2.3. Menú Control de sesiones

Una vez que se ha creado alguna *Secuencia* podemos proceder a crear *Sesiones*. Las *sesiones*, definidas en el apartado 4.2.1 como un conjunto ordenado de *sesiones*, poseen unas características peculiares debido a su especial naturaleza. Teniendo en cuenta la definición dada, las sesiones deben atenerse a las siguientes restricciones:

- Deben poseer como mínimo una secuencia asociada.
- En caso de que exista alguna secuencia de cámara asociada a dicha sesión deberá ser *finita*, esto es, que posea un número concreto de fotogramas a procesar, ya que, en caso contrario, sólo podrá existir una secuencia de cámara infinita que irá situada como cierre de la sesión y tras la cuál no podrá insertarse ninguna otra secuencia.

Desde el menú de Control de Sesiones que podemos ver en el siguiente gráfico tenemos acceso a los distintos formularios con los que podremos realizar las tareas de creación, modificación, visualización e impresión de sesiones.

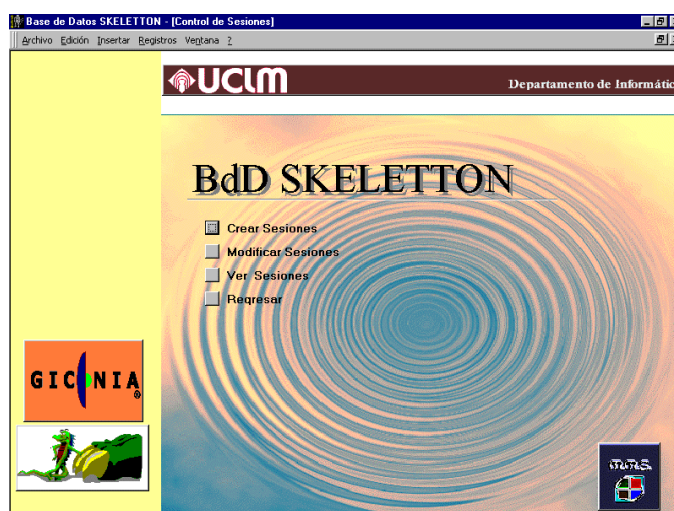


Figura 4.31. Control de sesiones

4.2.3.1. Opción Crear sesiones

Para crear una nueva sesión, pulse en el botón *Crear Sesiones* desde el menú de *Control de Sesiones*. Una vez en el formulario de creación, deberá pulsar el botón *Nueva Sesión* con lo que se activarán los distintos campos que deberá rellenar:

- **Autor:** Seleccione de la lista desplegable el nombre del autor de la sesión. En caso de que el autor no exista pulse el botón *Nuevo Autor*, tras esto se le mostrará el diálogo de "Nuevos Autores", cuyo funcionamiento fue visto en el apartado 4.2.2.1.
- **Nombre Sesión:** Aquí debe introducir un nombre descriptivo para la sesión que va a generar, de forma que posteriormente pueda reconocerlo fácilmente.
- **Fecha Creación:** Automáticamente la aplicación tomará la fecha de su sistema.

Orden	Nombre Secuencia	UDI	Nº Fotogramas
-------	------------------	-----	---------------

Figura 4.32. Crear sesión

Una vez se haya situado el cursor sobre este campo podrá observar que el formulario adopta la forma que se mostraba en la figura anterior. Dejándole únicamente la posibilidad de guardar los datos que ha introducido. Tras guardarlos deberá pulsar sobre el botón *Insertar Secuencia*; desde el formulario que se le mostrará podrá asociar, a la sesión, la secuencia que desee (véase el apartado *Edición de Sesiones*).

4.2.3.2. Opción Modificar sesiones

Si pulsa sobre el botón *Modificar Sesiones* desde el menú de *control de sesiones* se le mostrará un formulario como el siguiente, desde el cuál podrá buscar la sesión a editar de entre el conjunto de sesiones existente atendiendo a diversos criterios.

Autores	Nombre Sesion	Fecha Creacion	Fecha Modificacion
ANTONIO FERNÁNDEZ CABA	AUTOPISTA1	26/10/99	
ANTONIO FERNÁNDEZ CABA	AUTOPISTA2	26/10/99	
ANTONIO FERNÁNDEZ CABA	CIRCUNVALACION1	26/10/99	
ANTONIO FERNÁNDEZ CABA	CIRCUNVALACION3	26/10/99	
G.I.C.O.N.I.A.	Sesión de Triángulos	27/10/99	28/10/99

Buscar:
 Todas
 Autor
 Nombre Sesión

Parámetro de la búsqueda:

Figura 4.33. Modificar sesiones

- **Búsqueda Visual** .- Para ello no tendrá más que recorrer la lista de sesiones que se le muestra, seleccionar la sesión a editar y pulsar sobre el botón *Editar Sesión*.
- **Búsqueda por Autor**.- Para lo cuál no tiene más que seleccionar la casilla *Autor* en la parte inferior del formulario. Tras esto, escriba el nombre completo del autor a buscar y pulse sobre el botón *Buscar*. Esto actuará a modo de criba sobre la lista de sesiones mostrándole únicamente aquellas en que el autor coincida con el que usted haya introducido, facilitándole así la posterior realización de una búsqueda visual.
- **Búsqueda por Nombre Sesión**: Su funcionamiento es similar al de las búsquedas por autor con la salvedad de introducir el nombre de la sesión que se pretenda editar en lugar del autor de la misma.

Una vez haya localizado la sesión y pulsado sobre el botón *Editar sesión* se le mostrará un formulario idéntico al de creación de sesiones, con el que podrá realizar las tareas básicas referentes a las secuencias de dicha sesión, tales como la inserción y eliminación de secuencias.

4.2.3.3. Opción Edición de sesiones

La pantalla de edición de sesiones permite realizar y/o modificar las asociaciones establecidas entre una *sesión* y las *secuencias* que la constituyen. Como ya sabemos, cada una de las secuencias que compone la *sesión* está definida por un conjunto de parámetros que permiten determinar el modo en que esta será dispensada.

A la hora de eliminar una secuencia de la sesión, bastará con seleccionar dicha secuencia y pulsar sobre el botón eliminar. En este momento la aplicación nos pedirá que confirmemos la eliminación antes de proceder a su ejecución.

Del mismo modo que ocurría con las secuencias, es fundamental que tengamos la absoluta certeza, en caso de que vayamos a eliminar una sesión, de que ésta no está siendo utilizada por ningún Agente. En caso contrario, si eliminamos una sesión e intentamos ejecutar el Agente que la contiene, estaremos forzando a la aplicación que use la BDD SkeleTTON a que cometa un error.

Si por el contrario, lo que se desea es asociar una nueva secuencia a la sesión existente, bastará con pulsar sobre el botón *Insertar Secuencia*. Cuando inserta una secuencia debe tener en cuenta que el orden de inserción se corresponderá con el orden en que dichas secuencias serán dispensadas.

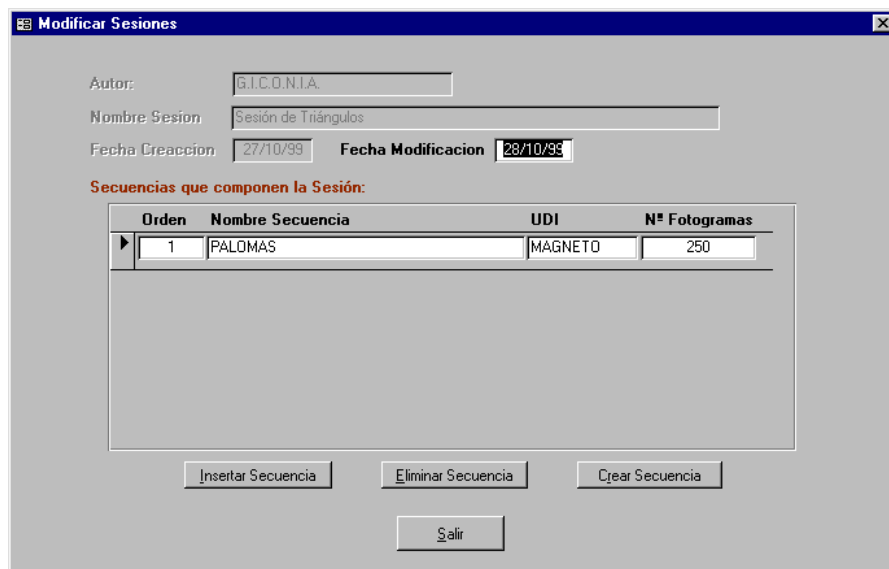


Figura 4.34. Editar sesiones

Una vez que se le haya desplegado el formulario de *Inserción de Secuencias* (figura 4.35.) deberá escoger la secuencia que desea insertar de la lista de secuencias que se le muestra en la parte superior del formulario y pulsar sobre el botón *Seleccionar Secuencia* o bien hacer doble click sobre ésta.

Nombre Secuencia	UDI	Creación	Modificación
AUTOPISTA1	MAGNETO	26/10/99	
AUTOPISTA2	MAGNETO	26/10/99	
BILLAR	OTROS	18/10/99	
CIRCUNVALACION1	MAGNETO	26/10/99	
FURGONETA	OTROS	18/10/99	
IAKSBad	OTROS	18/10/99	18/10/99
PALOMAS	MAGNETO	18/10/99	
PALOMAS II	OTROS	18/10/99	
PALOMO	MAGNETO	26/10/99	

Refuerzo

No
 Teclado
 Fichero
 Audio
 Cinta Video

Numero Fotogramas:
 Posicion Comienzo:
 Numero Repeticiones:
 Paso:
 Path:
 Nombre del Fichero:

Figura 4.35. Insertar secuencia

Si no está seguro de cuál es la secuencia que deseaba insertar puede visualizar las características asociadas a dicha secuencia previamente a su selección. Para ello no tiene más que escoger la secuencia que desea visualizar y pulsar sobre el botón *Ver Detalles Secuencia*, con lo que conseguirá que se le muestre el correspondiente formulario de detalle.

Una vez que haya seleccionado la secuencia se activará el resto de controles del formulario a fin de que pueda insertar las características de ejecución que desea asociar a dicha secuencia. Estas características no se asocian de forma permanente a la secuencia sino que solamente la afectan para esa instancia concreta de la secuencia dentro de la sesión. En caso de que deseara volver a utilizar la misma secuencia dentro de la misma sesión deberá volver a establecer las características de ejecución deseadas para cada nueva instancia.

Las características de ejecución que se pueden asociar a una secuencia están en función del tipo de secuencia (MAGNETO, OTROS, GENIMG o CAMARA) insertada, esto es:

- ◆ **Numero Fotogramas:** Excepto para las secuencias CAMARA, este valor es calculado de forma automática por la aplicación teniendo en cuenta las características: *Posición Comienzo, Número Repeticiones y Paso*.

En las secuencias CAMARA, este dato es el usado para determinar el número de fotogramas que se desea analizar. En caso de que se indique el valor 0 (cero) se considerará que se trata de una secuencia de cámara *infinita* y que por tanto no pueden asociarse más secuencias a la *sesión*.

- ◆ **Posición Comienzo:** Este dato indica a partir de qué fotograma de la secuencia debe empezar a procesarse las imágenes de la misma. En el caso de secuencias MAGNETO, este dato se corresponde con el dato LTC (este dato debe estar comprendido entre la posición inicial y la posición final de la secuencia). Para las secuencias OTROS y GENIMG la máscara es obviada y se considera el dato como un número entero y para las secuencias CAMARA este dato carece de significado.

- ◆ **Número de Repeticiones:** Valor numérico que indica el número de veces que será repetida la ejecución de la secuencia. El valor 0 indica por tanto que la secuencia será ejecutada una sola vez, el valor 1, que será ejecutada dos veces y así sucesivamente.

- ◆ **Paso:** Valor numérico que establece el intervalo conque serán tomadas las imágenes, esto es, un intervalo 5 indica que, de las imágenes que componen la secuencia, solamente será tomada una de cada cinco.

- ◆ **Refuerzo:** En caso de que las imágenes que componen la secuencia deban ser dispensadas junto con su valor de refuerzo asociado, deberá escogerse de entre las casillas de verificación aquella que identifica el soporte en que éste se encuentra almacenado:

- Teclado.- Si el refuerzo será introducido de manualmente por el usuario durante el proceso de análisis de las imágenes.
- Cinta de Vídeo.- Si se trata de una secuencia MAGNETO y el refuerzo asociado debe ser tomado a partir de los 4 bytes de menor peso del dato UBIT asociado a cada fotograma.
- Audio.- Si el refuerzo asociado a las imágenes consiste en una señal de audio.
- Fichero.- Si el refuerzo asociado a las imágenes se encuentra almacenado en un fichero de tipo REF. Cuando se selecciona esta posibilidad se activan

automáticamente los campos *Path* y *Nombre Fichero* de forma que se pueda introducir la localización exacta de dicho fichero.

Si, una vez comprobados que los datos establecidos se corresponden con los que realmente deseaba establecer, se da cuenta de que ha cometido algún error puede reiniciar toda la operación pulsando sobre el botón *Reiniciar Selección*. O, en caso de que los datos establecidos sean correctos, puede proceder a guardar los datos establecidos pulsando sobre el botón *Guardar Datos*.

4.2.3.4. Opción Ver sesiones

Desde el menú de *Control de Sesiones*, pulsando sobre el botón *Ver Sesiones*, podrá visualizar las secuencias que haya asociado previamente a cada una de las sesiones así como las características asociadas a estas. Para ello no tiene más que seleccionar de la lista de la esquina superior derecha la sesión que desea ver, tras lo cuál se mostrarán en el cuadro inferior la relación de secuencias que dicha sesión tiene asociadas.

Desde este formulario puede obtener un informe para cada sesión que le muestre una relación detallada de las secuencias que la componen. Usando los botones de la parte inferior izquierda podrá visualizar, guardar, imprimir o enviar por correo electrónico el informe deseado.

Ver Sesiones

Nombre Sesión:

Autor:

F. Creación / Modificación:

Buscar Sesión:

- AUTOPISTA1
- AUTOPISTA2
- BILLAR
- CIRCUNVALACION1
- CIRCUNVALACION3
- FERRONETA

Secuencias que la componen:

Nombre Secuencia	LIDI	Nº Fotogramas	Nº Repeticiones Paso
1 PASEO	MAGNETO	35	0

Pos. Comienzo: Refuerzo: Path: Nombre de Fichero: Autor:

Ver Informe Sesión Enviar Informe Sesión Salir

Imprimir Informe Sesión Salvar Informe Sesión

Figura 4.36. Ver sesiones

4.2.4. Menú Control de agentes

Una vez que ya hemos creado una sesión, el siguiente paso a realizar es generar un *Agente* al que asociar dicha sesión, de forma que, a través de dicho agente, podamos procesar la imágenes contenidas en las distintas secuencias que conforman la sesión asociada.

Figura 4.37. Crear agente

A la hora de generar un nuevo *Agente* podemos basarnos en las características asociadas a un *agente* anterior, para ello, no tenemos nada más que pulsar sobre el botón *Nuevo Agente*, la aplicación nos preguntará si deseamos basarnos en un agente anterior. Si aceptamos, tras elegir el agente en el que deseamos basarnos, tendremos que todos los campos del formulario habrán tomado los mismos valores que el agente *padre*. Esto indica que el agente padre a pasado al historial y que a partir de ahora, el nombre del agente identificará al nuevo agente que estemos creando.

Una vez que el formulario está preparado para la inserción de nuevos datos, deberemos rellenarlos de la siguiente forma:

- **Nombre Agente:** Es el nombre descriptivo que usaremos para identificar el *agente* que estemos creando en este preciso instante.
- **Tipo de Agente:** Seleccionaremos "Tipo A" si el agente que estamos creando va a ser procesado con el software *Intelligent SILHFinder*.

- **Procede de:** Este campo será rellenado automáticamente por la aplicación si le indicamos, tal y como hemos explicado anteriormente, que deseamos basar el nuevo agente en uno existente.
- **Sesión Asociada:** Indica la sesión que deseamos asociar al agente y que contendrá las secuencias que deseamos procesar.
- **Digitalizadora Presente:** Indica la presencia de un hardware específico encargado de realizar el proceso de digitalización de imágenes.
- **Extractor Presente:** Indica la presencia de un hardware específico encargado de realizar el proceso de extracción de características.
- **Clasificadora Presente:** Indica la presencia de un hardware específico encargado de las tareas de clasificación.
- **Ejecutar Directo:** Indica si el modo de ejecución va a ser Directo o Diferido. En el modo de ejecución directo, las imágenes serán dispensadas de forma secuencial y sin interrupciones, mientras que en el modo diferido, al software encargado de analizar las secuencias de imágenes será el encargado de solicitar cada imagen una vez haya terminado su proceso de análisis.
- **Edad Sistema:** Indica el número de imágenes que ya han sido procesadas de la sesión asociada al agente. Este valor puede ser reseteado para, de esta forma, poder procesar de nuevo la sesión.
- **Detener ejecución cada:** Indica el intervalo de fotogramas tras el cuál se detendrá el proceso de análisis.
- **Salvar estado cada:** Establece la frecuencia (en fotogramas) con que será salvado el estado del agente durante su ejecución.
- **Número Fotogramas:** Este valor indica el número de fotogramas que un agente a procesado a lo largo de su vida, independientemente de que haya variado la sesión asociada.
- **Tiempo ejecución en Background:** Tiempo que queda libre de CPU entre cada procesamiento de imagen.

Una vez establecidos estos datos y salvado el agente, la aplicación nos pedirá que insertemos las características asociadas al individuo, para lo cuál nos mostrará un formulario como el siguiente:

Detallar Agente TipoA		
IdIndividuo	<input type="text" value="10"/>	
Número de bandas de nivel de gris	<input type="text" value="3"/>	
Valor máximo de permanencia	<input type="text" value="255"/>	
Valor mínimo de permanencia	<input type="text" value="0"/>	
Número de vecinos para la interacción lateral	<input type="text" value="1"/>	
Valor de descarga por detección de movimiento	<input type="text" value="0"/>	
Valor de recarga por vecindad	<input type="text" value="0"/>	
	Número de vecinos para el reparto de carga por bandas de nivel de gris	<input type="text" value="1"/>
	Valor mínimo de mancha por banda de nivel de gris	<input type="text" value="0"/>
	Número de vecinos para el reparto de carga por objeto	<input type="text" value="1"/>
	Valor mínimo de mancha para la fusión en objetos	<input type="text" value="0"/>
	Valor umbral mínimo para la detección de objetos	<input type="text" value="0"/>
<input type="button" value="Continuar"/> <input type="button" value="Cancelar"/>		

Figura 4.38. Detallar agente

Desde este formulario podremos establecer aquellas características específicas para el software Intelligent SILHFinder usado para el procesamiento de las secuencias de imágenes contenidas en la sesión asociada al Agente que acabamos de crear.

4.2.4.1. Opción Eliminación de agentes

Desde el menú de *Control de Agentes* se nos da la opción de eliminar un agente. Esta opción no es tan crítica como la de eliminación de Secuencias o de Sesiones, al ser este el último eslabón en la cadena. De esta forma, el hecho de eliminar un *agente* no implica la eliminación de la sesión asociada ni de las secuencias contenidas en esta. ***Solamente se eliminará el agente en sí y los datos asociados a la ejecución del mismo.***

Por tanto, bastará con seleccionar el agente que deseemos y proceder a su eliminación.

4.2.5. Menú Informes globales

Desde esta opción del menú de la aplicación (ver figura 4.39.), tendremos acceso a los diversos informes que nos serán de gran utilidad a nivel documental de la base de datos.

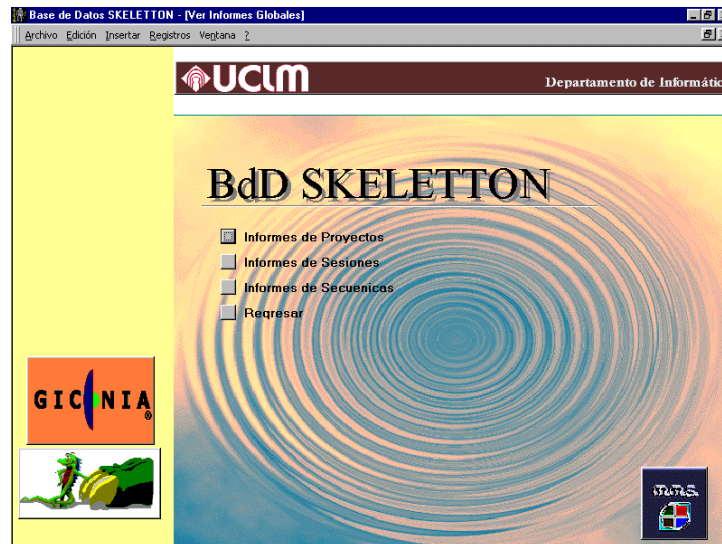


Figura 4.39. Menú de informes globales

Destaquemos del menú que se presenta en la anterior imagen los que hacen referencia a las Sesiones y a las Secuencias, ya que desde ellos será a los que accederemos para obtener relaciones de las distintas secuencias que tenemos asociadas a cada una de las sesiones que hayamos generado así como de las características asociadas a cada una de las secuencias existentes, que tal y como vemos en la figura 4.40. pueden ser obtenidos tanto de forma global o en función del tipo de UDI que tienen asociada.

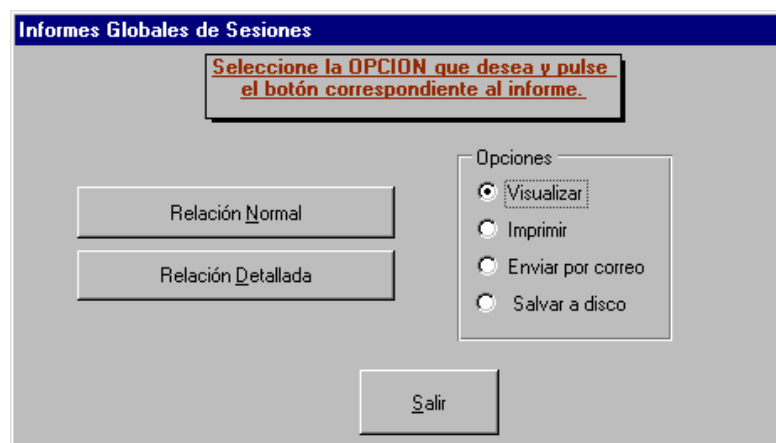


Figura 4.40. Informes globales por sesiones

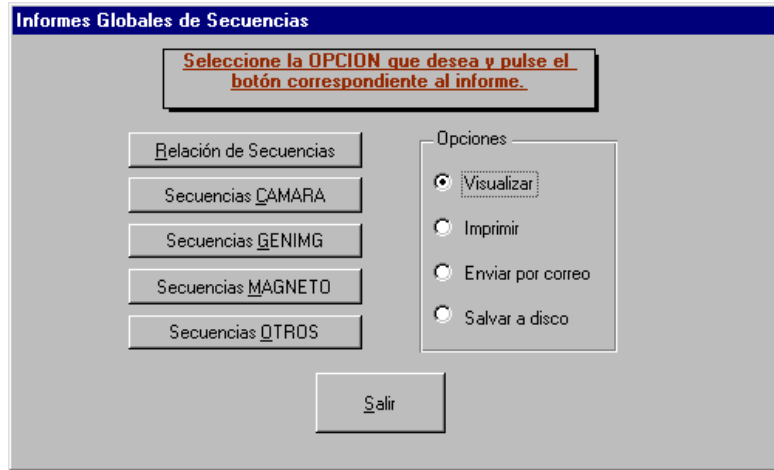


Figura 4.41. Informes globales por secuencia

4.3. Vista del usuario: Aplicación Intelligent SILHFinder

4.3.1. Control de acceso

El control de acceso a la aplicación Intelligent SILHFinder - Procesamiento Inteligente de Secuencias de Imágenes, se realiza mediante una ventana en la que se pide el nombre de un usuario.

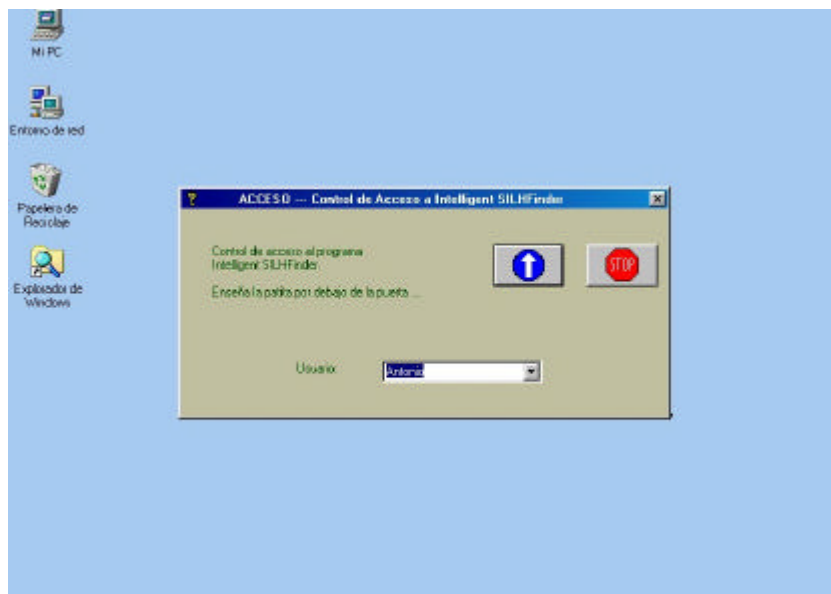



Figura 4.42. ACCESO --- Control de acceso a Intelligent SILHFinder

El nombre de usuario introducido en este control de acceso facilitará al sistema una serie de directorios de trabajo por defecto para cada uno de los usuarios preestablecidos. Se elige uno de los nombres de usuario autorizados para entrar en la aplicación de entre una lista desplegable de usuarios autorizados.



Figura 4.43. ACCESO --- Control de acceso a Intelligent SILHFinder. Selección de un usuario

Tenemos dos opciones en el control de acceso a la aplicación Intelligent SILHFinder:

??  **Aceptar:** pulsando este botón se toma el nombre de usuario y entramos al marco principal de la aplicación con este nombre de usuario


??  **Cancelar:** pulsando este botón no llegamos a entrar en la aplicación. En este caso aparece el siguiente mensaje:



Figura 4.44. ACCESO --- Control de acceso a Intelligent SILHFinder. Despedida

4.3.2. Entorno del sistema

El entorno general del sistema Intelligent SILHFinder se muestra en el marco principal de la aplicación Intelligent SILHFinder escrita en lenguaje MS Visual C++ versión 6.0 bajo entorno Windows 98.

Puede apreciarse en este marco: a) la barra de menús, b) la barra de herramientas compuesta por un conjunto de botones, c) la ventana marco de la aplicación y d) la barra de estado.

La ventana marco de la aplicación consta de un mapa de bits que contiene información sobre el nombre del software y el grupo de desarrollo de la aplicación.

La barra de estado tiene como función principal el mostrar una ayuda sensible a la posición del ratón sobre las diferentes áreas del marco principal del sistema. Las siguientes dos áreas centrales de la barra de estado muestran cómo trabaja el sistema y el estado de la sesión, respectivamente. Por fin, se muestra en todo momento la hora del sistema a la derecha de la barra de estado.

En cuanto a la información mostrada en la barra de estado sobre cómo trabaja el sistema, tendremos los siguientes casos posibles:

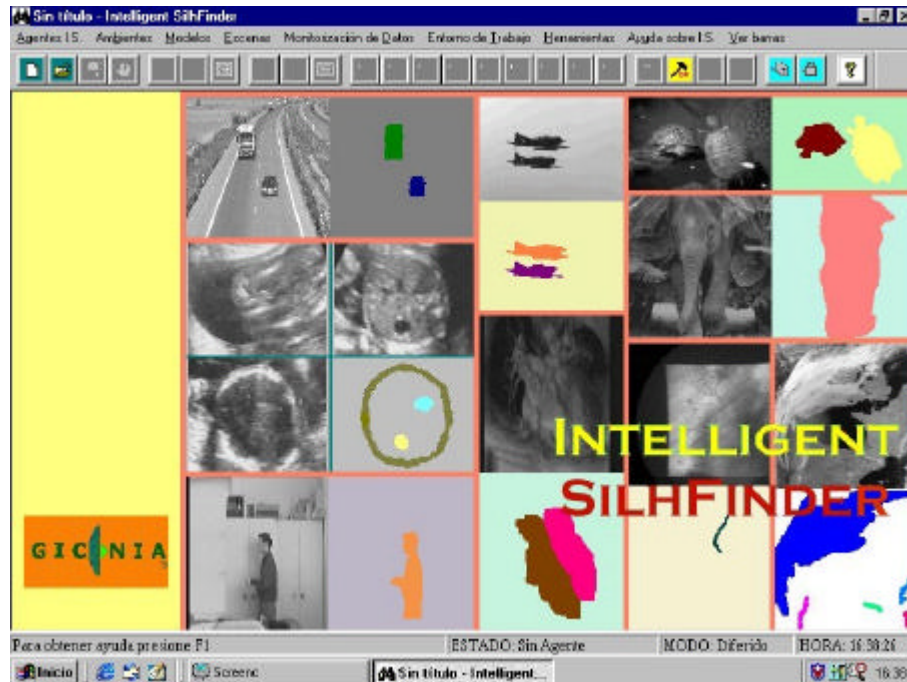


Figura 4.45. Marco principal de la aplicación Intelligent SILHFinder

- **DIRECTO:** el sistema trabaja procesando en tiempo real la señal de vídeo (este estado aparece en la barra de estado escrito como DIFERIDO)
- **DIFERIDO:** después de la captura de cada fotograma, hay una espera hasta que el proceso de la imagen se ha completado (este estado aparece en blanco en la barra de estado)

En cuanto al estado de la sesión, tenemos:





- **SIN AGENTE:** no se ha abierto ningún agente
- **DETENIDO:** la sesión está detenida y el control está en manos del usuario (aparece en blanco en la barra de estado)
- **ARRANCADO:** la sesión está arrancada; se está ejecutando el proceso
- **APRENDIENDO:** la sesión está arrancada, pero en fase de aprendizaje; se está buscando el juego de parámetros óptimo para el tipo de aplicación elegida

En el menú de la aplicación podemos apreciar las siguientes opciones generales que se irán comentando más adelante:










- **Agente Intelligent SILHFinder:** acceso a las ventanas de manipulación de Agentes Intelligent SILHFinder

- **Ambientes:** permite definir los parámetros del ambiente en que se va a mover un agente Intelligent SILHFinder
- **Modelos:** permite definir los parámetros del modelo a aplicar al agente Intelligent SILHFinder
- **Escenas:** permite definir la escena (o tipo de aplicación) relacionada con el agente Intelligent SILHFinder
- **Monitorización de Datos:** acceso a las ventanas de definición de parámetros y condiciones monitorizados durante la ejecución del sistema Intelligent SILHFinder
- **Entorno de Trabajo:** acceso a las ventanas de definición de parámetros y condiciones generales de trabajo del sistema Intelligent SILHFinder para un Agente determinado
- **Herramientas:** opciones que permiten lanzar aplicaciones de utilidad directamente relacionadas con Intelligent SILHFinder.
- **Ayuda sobre Intelligent SILHFinder:** acceso a las ayudas sensibles al contexto y a la obtención de información adicional acerca del producto
- **Ver barras:** permite mostrar/ocultar las barras de herramientas y de estado

Los botones de la barra de herramientas están ligados a algunas opciones de los menús vistos anteriormente. En concreto tenemos (de izquierda a derecha):

-  **Nuevo** (opción **Nuevo Agente**, opción perteneciente al menú **Agente Intelligent SILHFinder** que permite crear un nuevo Agente de tipo Intelligent SILHFinder)
-  **Abrir** (opción **Abrir Agente**, opción perteneciente al **menú Agente Intelligent SILHFinder** que permite acceder a la información de un Agente existente en la base de datos de Agentes de tipo Intelligent SILHFinder)
-  **Guardar** (opción **Guardar Agente**, opción perteneciente al menú **Agente Intelligent SILHFinder** que permite salvar el estado del Agente activo)
-  **Cerrar** (opción **Cerrar Agente**, opción perteneciente al **menú Agente Intelligent SILHFinder** que permite cerrar el Agente activo)

-  **Arrancar** (opción **Arrancar Agente**, opción perteneciente al menú **Agente Intelligent SILHFinder**, pensada para lanzar el proceso de la aplicación)
-  **Detener** (opción **Detener Agente**, opción perteneciente al **menú Agente Intelligent SILHFinder**, pensada para parar el proceso de la aplicación)
- **Imágenes** (opción **Imágenes**, opción del menú **Monitorización de datos**, permite ver la salida en forma de imágenes de las capas 1, 2 y 3 de proceso de la aplicación Intelligent SILHFinder para el Agente activo)
-  **Arrancar Aprendizaje** (opción **Arrancar Aprendizaje Agente**, opción perteneciente al menú **Agente Intelligent SILHFinder**, pensada para lanzar la fase de aprendizaje de la aplicación)
-  **Detener Aprendizaje** (opción **Detener Aprendizaje Agente**, opción perteneciente al **menú Agente Intelligent SILHFinder**, pensada para parar la fase de aprendizaje de la aplicación)
-  **Imágenes Aprendizaje** (opción **Imágenes Aprendizaje**, opción del menú **Monitorización de datos**, permite ver la salida en forma de imágenes de las capas 1, 2 y 3 durante el proceso de aprendizaje de la aplicación Intelligent SILHFinder para el Agente activo)
-  **Parámetros Globales** (opción **Parámetros Globales**, opción del menú **Monitorización de datos**, muestra la información de tipo genérico de la aplicación para el Agente activo)
-  **Restricciones de Imágenes** (opción **Parámetros Globales**, opción del menú **Monitorización de datos**, muestra la información de tipo genérico de la aplicación para el Agente activo)
-  **Preprocesado** (opción **Parámetros Globales**, opción del menú **Monitorización de datos**, muestra la información de tipo genérico de la aplicación para el Agente activo)

-  **Segmentación** (opción **Parámetros Globales**, opción del menú **Monitorización de datos**, muestra la información de tipo genérico de la aplicación para el Agente activo)
-  **Modelo Interacción Lateral para la Computación Acumulativa** (opción **Parámetros Globales**, opción del menú **Monitorización de datos**, muestra la información de tipo genérico de la aplicación para el Agente activo)
-  **Modelo Interacción Lateral para el Reparto de Carga** (opción **Parámetros Globales**, opción del menú **Monitorización de datos**, muestra la información de tipo genérico de la aplicación para el Agente activo)
-  **Modelo Interacción Lateral para la Fusión en Objetos** (opción **Parámetros Globales**, opción del menú **Monitorización de datos**, muestra la información de tipo genérico de la aplicación para el Agente activo)
-  **Postprocesado** (opción **Parámetros Globales**, opción del menú **Monitorización de datos**, muestra la información de tipo genérico de la aplicación para el Agente activo)
-  **Escena** (opción **Parámetros Globales**, opción del menú **Monitorización de datos**, muestra la información de tipo genérico de la aplicación para el Agente activo)
-  **Base de Datos SkeleTTon** (opción **Acceso a la Base de Datos**, que contiene la información de estado actual e histórica de los Agentes, opción del menú **Herramientas**)
-  **Interfaz de Vídeo** (opción **Acceso a la Interfaz de Vídeo**, opción del menú **Herramientas**, que permite manejar fuera de la aplicación Intelligent SILHFinder el magnetoscopio dispensador de imágenes de vídeo)
-  **Acerca de** (opción **Acerca de Intelligent SILHFinder**, opción del menú **Ayuda sobre Intelligent SILHFinder**, muestra una información extensiva acerca del producto, su versión, los grupos de desarrollo, ...)

4.3.3. Menús de la aplicación

Para comenzar a trabajar con el sistema es necesario definir un nombre de Agente donde recoger el estado de la aplicación. Esto se puede hacer desplegando la opción del menú **Agente Intelligent SILHFinder** como se muestra a continuación en la figura 4.46.



Figura 4.46. Menú Agente Intelligent SILHFinder

4.3.3.1. Menú Agente Intelligent SILHFinder

Observamos dos opciones para definir el nombre de un Agente:

- **Nuevo Agente**, opción que permite crear un nuevo Agente de tipo Intelligent SILHFinder
- **Abrir Agente**, opción que permite acceder a la información de un Agente existente en la base de datos de Agentes de tipo Intelligent SILHFinder

En este menú pueden realizarse también las siguientes funciones:

- **Guardar Agente**: salva el estado del Agente en la base de datos SkeleTTon, machacando la información anterior; una copia del estado actual permanece en una tabla histórica de Agentes
- **Guardar Agente como**: guarda el estado del Agente con un nombre distinto al actual; una copia del estado actual permanece en una tabla histórica de Agentes
- **Cerrar Agente**: cierra el Agente, y por tanto, para poder seguir trabajando con Intelligent SILHFinder será necesario definir otro Agente
- **Arrancar Agente**: arranca la sesión asociada al Agente, es decir, se lanza el proceso Intelligent SILHFinder para el Agente activo

- **Detener Agente:** detiene la sesión asociada al Agente; para la ejecución de Intelligent SILHFinder en cuanto a proceso de imágenes
- **Arrancar Aprendizaje Agente:** arranca la sesión asociada al aprendizaje del Agente, es decir, se lanza el proceso de aprendizaje Intelligent SILHFinder para el Agente activo
- **Detener Aprendizaje Agente:** detiene la sesión asociada al aprendizaje del Agente; para la ejecución de Intelligent SILHFinder en cuanto a proceso de aprendizaje de imágenes
- **Salir de Intelligent SILHFinder:** permite salir de la aplicación Intelligent SILHFinder

Es interesante hacer notar que algunas opciones del menú no están accesibles al entrar en la aplicación. Véase que solamente se puede elegir entre **Nuevo Agente**, **Abrir Agente** y **Salir de Intelligent SILHFinder**. Las demás opciones **Guardar Agente**, **Guardar Agente como**, **Cerrar Agente**, **Arrancar Agente**, **Detener Agente**, **Arrancar Aprendizaje Agente** y **Detener Aprendizaje Agente**, en efecto, no tienen sentido si aún no ha sido definido un Agente de trabajo.

Una vez hayamos definido un Agente dispondremos de las opciones ahora no autorizadas. Entonces, el marco de la aplicación Intelligent SILHFinder, sin desplegar ningún menú, tendrá el aspecto de la figura 4.47.

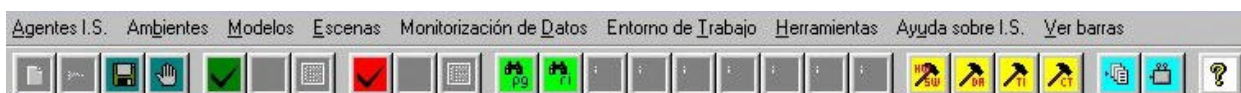




Figura 4.47. Marco de Intelligent SILHFinder. Agente abierto

Para salir de cualquier caja de diálogo de este menú, tenemos dos opciones:

- ??  **Aceptar:** pulsando este botón se actualizan las modificaciones introducidas y volvemos al marco principal de la aplicación
- ??  **Cancelar:** pulsando este botón se deshacen las modificaciones introducidas y volvemos al marco principal de la aplicación

Opción Nuevo Agente

La opción **Nuevo Agente** nos lleva a otra aplicación, a saber a la Base de Datos SkeleTTon, de la que se ofrece el marco principal en la figura 4.48. De este modo se ha evitado reescribir un código redundante y se ha posibilitado el usar toda la potencia de la base de datos de Agentes.

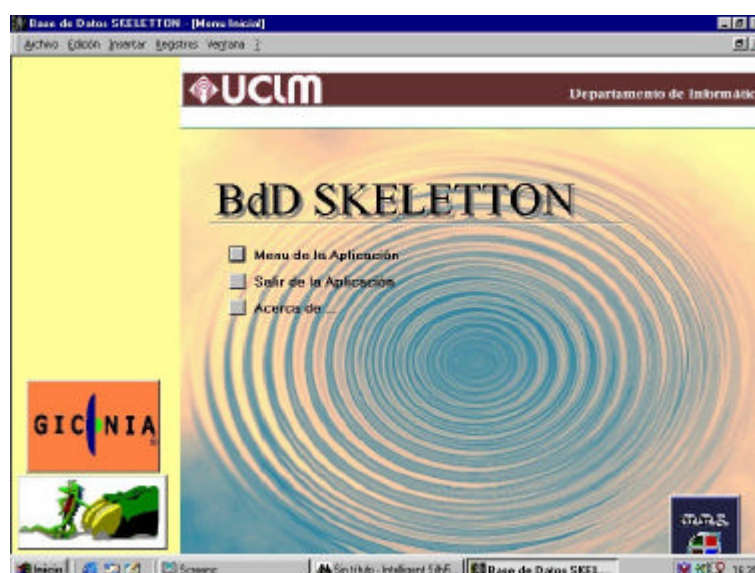


Figura 4.48. Base de Datos SkeleTTon

Si desea seguir los pasos para crear un nuevo Agente, vea el capítulo Vista del usuario: Base de Datos SkeleTTon.

Opción Abrir Agente

La opción **Abrir Agente**, lleva a una caja de diálogo en la que podemos seleccionar un nombre de Agente de entre todos aquellos dados de alta en la Base de Datos SkeleTTon.

Si se necesita más información antes de seleccionar un Agente determinado, es posible pasar desde esta caja de diálogo directamente a la aplicación Base de Datos SkeleTTon, pulsando

el botón  BDD SkeleTTon.

La selección del Agente se realiza pulsando dos veces (doble click) sobre uno de los nombres de Agente disponibles en la caja de diálogo. Entonces aparecerá el nombre de Agente seleccionado.



Figura 4.49. AGENTES I.S. --- Abrir Agente



Figura 4.50. AGENTES I.S. --- Abrir Agente. Resultado de la selección

Una vez definido un nombre de Agente, las opciones disponibles desde el menú **Agente Intelligent SILHFinder** son los que se muestran en la figura 4.51.



Figura 4.51. Menú Agente Intelligent SILHFinder. Agente definido

Las opciones **Nuevo Agente**, **Abrir Agente** y **Salir de Intelligent SILHFinder** están ahora, lógicamente, deshabilitadas. Ahora sí que tiene sentido **Guardar Agente**, **Guardar Agente como** y **Cerrar Agente**. Nótese que de entre las dos posibilidades mutuamente excluyentes de **Arrancar Agente** y **Detener Agente**, solamente está activa la opción **Arrancar Agente**. Lo mismo ocurre con las opciones **Arrancar Aprendizaje Agente** y **Detener Aprendizaje Agente**.

Opción Guardar Agente

Se abre una caja de diálogo donde aparece el nombre del Agente actual de modo informativo. No se permite cambiar aquí el nombre del Agente, ya que para ello está la opción alternativa **Guardar Agente como**.



Figura 4.52. AGENTES I.S. --- Guardar Agente

Opción Guardar Agente como

La opción Guardar como, permite cambiar el nombre del Agente. Por lo tanto, se abre una caja de diálogo que permite introducir un nuevo nombre de Agente.

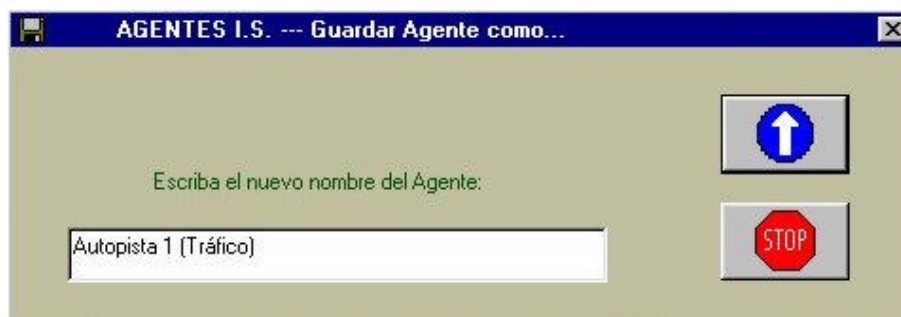


Figura 4.53. AGENTES I.S. --- Guardar Agente como

Opción Arrancar Aprendizaje Agente

Esta opción arranca la sesión de trabajo asociada al aprendizaje del Agente activo. El sistema pasa a estado APRENDIENDO. En este momento comienza a funcionar el sistema tratando las imágenes de entrada. Quedan invalidadas todas las opciones de menú salvo **Detener Aprendizaje Agente** del menú **Agente Intelligent SILHFinder** y la opción **Imágenes Aprendizaje** del menú **Monitorización de datos**.



Figura 4.54. Marco de Intelligent SILHFinder. Aprendizaje del Agente arrancado

La sesión de aprendizaje del Agente puede detenerse por dos motivos diferentes según puede apreciarse en las figuras 4.55. y 4.56., respectivamente:

1. porque el usuario detiene la sesión manualmente mediante la opción **Detener Aprendizaje Agente** del menú **Agente Intelligent SILHFinder**
2. porque se ha procesado la totalidad de las imágenes que componen la sesión

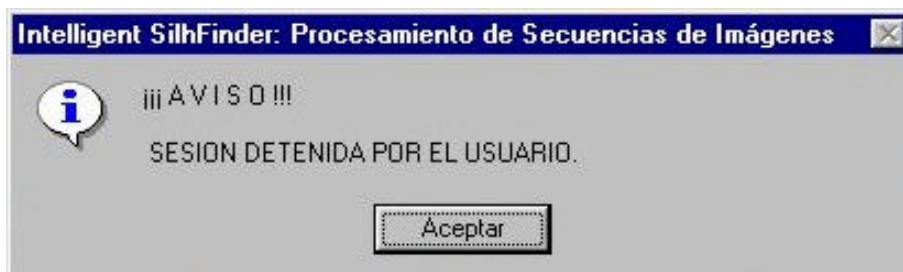



Figura 4.55. Mensaje de sesión detenida por el usuario durante el Aprendizaje del Agente



Figura 4.56. Mensaje de sesión detenida por Fin de Aprendizaje del Agente

Para proseguir con cualquier nueva opción de Intelligent SILHFinder será necesario pulsar el botón  **Aceptar**, tanto para el mensaje informativo SESION DETENIDA POR EL USUARIO (figura 4.55.) como para el mensaje igualmente de carácter informativo SESION DETENIDA POR FIN DE APRENDIZAJE (figura 4.56.).

Opción Detener Aprendizaje Agente

Esta opción detiene la sesión de trabajo asociada al aprendizaje del Agente. El sistema pasa de este modo a OFF-LINE. En este momento se detiene el proceso y pasa el control al usuario. Recordemos que el mensaje que aparece es el de la figura 4.55. vista con anterioridad.

Opción Arrancar Agente

Esta opción arranca la sesión de trabajo asociada al Agente activo. El sistema pasa a estado ON-LINE. En este momento comienza a funcionar el sistema tratando las imágenes de entrada. Quedan invalidadas todas las opciones de menú salvo **Detener Agente** del menú **Agente** en todos los casos y todas las opciones del menú **Monitorización de datos**, siempre que se esté trabajando en modo DIFERIDO. Recordemos que:

- ?? Si trabajamos en DIRECTO, el sistema procesa la señal de vídeo en tiempo real. Por lo tanto, no se monitoriza ninguna información para dejar todo el procesador al análisis de las imágenes.
- ?? Si trabajamos en DIFERIDO, entre dos fotogramas consecutivos el usuario recibe información acerca de lo que está ocurriendo con el estado del Agente.

El marco de la aplicación es ahora el siguiente, en el que vemos las opciones disponibles (botones de la barra de herramientas):



Figura 4.57. Marco de Intelligent SILHFinder. Agente arrancado

En el caso de que estuviésemos trabajando en real con imágenes tomadas desde la cámara, aparecería igualmente una caja de diálogo como la que se ofrece en la figura 4.58.

La sesión de Agente puede detenerse por dos motivos diferentes según puede apreciarse en las figuras 4.59. y 4.60., respectivamente:

3. porque el usuario detiene la sesión manualmente mediante la opción **Detener Agente** del menú Agente Intelligent SILHFinder
4. porque se ha procesado la totalidad de las imágenes que componen la sesión

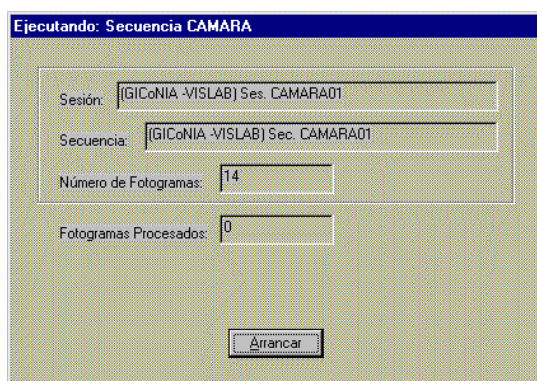


Figura 4.58. Ejecutando Secuencia Cámara

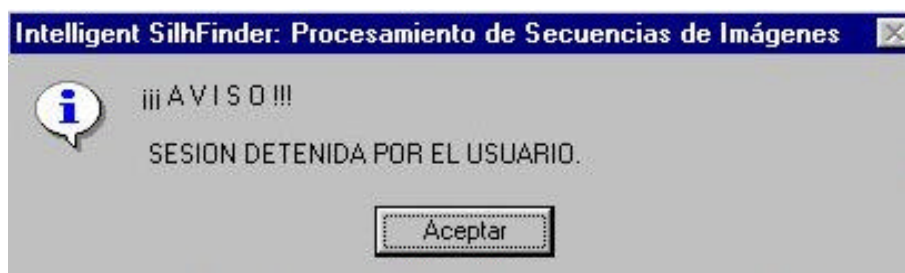


Figura 4.59. Mensaje de sesión detenida por el usuario

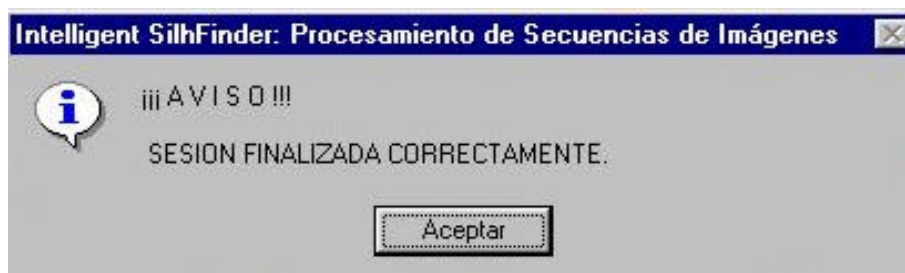



Figura 4.60. Mensaje de sesión finalizada correctamente

Para proseguir con cualquier nueva opción de Intelligent SILHFinder será necesario pulsar el botón  **Aceptar**, tanto para el mensaje informativo SESION DETENIDA POR EL USUARIO (figura 4.59.) como para el mensaje igualmente de carácter informativo SESION FINALIZADA CORRECTAMENTE (figura 4.60.).

Opción Detener Agente

Esta opción detiene la sesión de trabajo asociada al Agente. El sistema pasa de este modo a OFF-LINE. En este momento se detiene el proceso y pasa el control al usuario. Recordemos que el mensaje que aparece es el de la figura 4.59. vista con anterioridad.

Opción Cerrar Agente

Esta opción cierra el Agente activo, por lo tanto, no se puede lanzar el proceso de tratamiento de imágenes de Intelligent SILHFinder.

Opción Salir de Intelligent SILHFinder

Mediante esta opción, se abandona la aplicación Intelligent SILHFinder. Antes de salir, aparece un mensaje que pregunta si el Agente está cerrado o no.

Si se contesta con la opción del botón **No**, no salimos de la aplicación Intelligent SILHFinder. Con **Si** conseguimos salir de Intelligent SILHFinder.

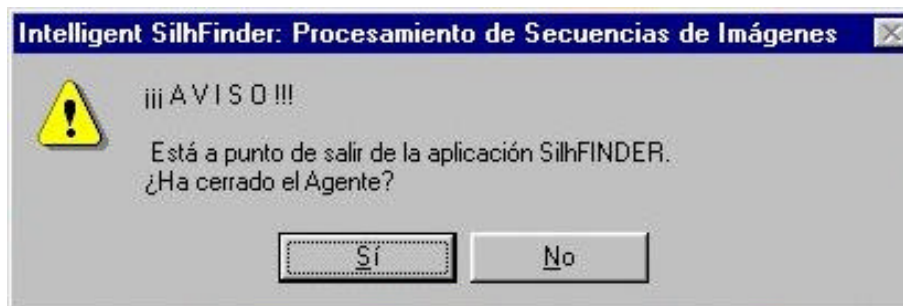


Figura 4.61. Mensaje de confirmación de Salida de Intelligent SILHFinder

4.3.3.2. Menú Ambientes

El menú **Ambientes** permite determinar el ambiente en el que el Agente se va a mover. Al elegir el menú **Ambientes** se nos abre la posibilidad de ir a alguna de las siguientes opciones:

- ?? **Parámetros Globales:** permite actualizar los parámetros globales de funcionamiento del Agente
- ?? **Preprocesado:** permite especificar qué tipo de preprocesado se desea aplicar a las imágenes de entrada al Agente

- ?? **Segmentación:** permite elegir qué tipo de segmentación se desea realizar sobre las imágenes de entrada preprocesadas
- ?? **Modelo:** esta opción tiene como finalidad elegir de entre todos los modelos de procesamiento disponibles, aquél que se desea aplicar al Agente
- ?? **Postprocesado:** permite elegir qué tipo de postprocesado aplicar a las imágenes tras aplicarles el modelo elegido en la opción anterior
- ?? **Escena:** está destinado a elegir uno de los tipos predefinidos de escenas para el Agente
- ?? **Parámetros del Aprendizaje:** permite actualizar los parámetros propios de la fase de aprendizaje del Agente

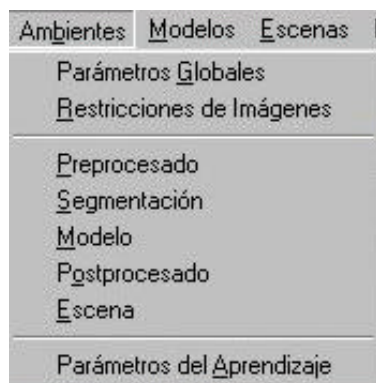




Figura 4.62. Menú Ambientes

Para salir de cualquier caja de diálogo de este menú, tenemos dos opciones:

- ??  **Aceptar:** pulsando este botón se actualizan las modificaciones introducidas y volvemos al marco principal de la aplicación
- ??  **Cancelar:** pulsando este botón se deshacen las modificaciones introducidas y volvemos al marco principal de la aplicación

Opción Parámetros Globales

Esta caja de diálogo tiene dos zonas claramente diferenciadas. Por una parte podemos observar (figura 4.63.) un cuadro que contiene la información básica del Ambiente del Agente. Este cuadro ayuda a englobar, de un solo vistazo, el Agente en su Ambiente completo. La información incluye:

- ?? **Preprocesado:** presencia o no de preprocesado a nivel de **Procesamiento de Puntos** y/o de **Filtrado Espacial**
- ?? **Segmentación:** presencia o no de segmentación a nivel de **Umbralización** y/o **Detección de Discontinuidades**
- ?? **Modelo:** tipo de modelo elegido para el Agente
- ?? **Postprocesado:** presencia o no de postprocesado a nivel de **Procesamiento de Puntos** y/o de **Filtrado Espacial**
- ?? **Escena:** tipo de Escena donde se encuadra el Agente

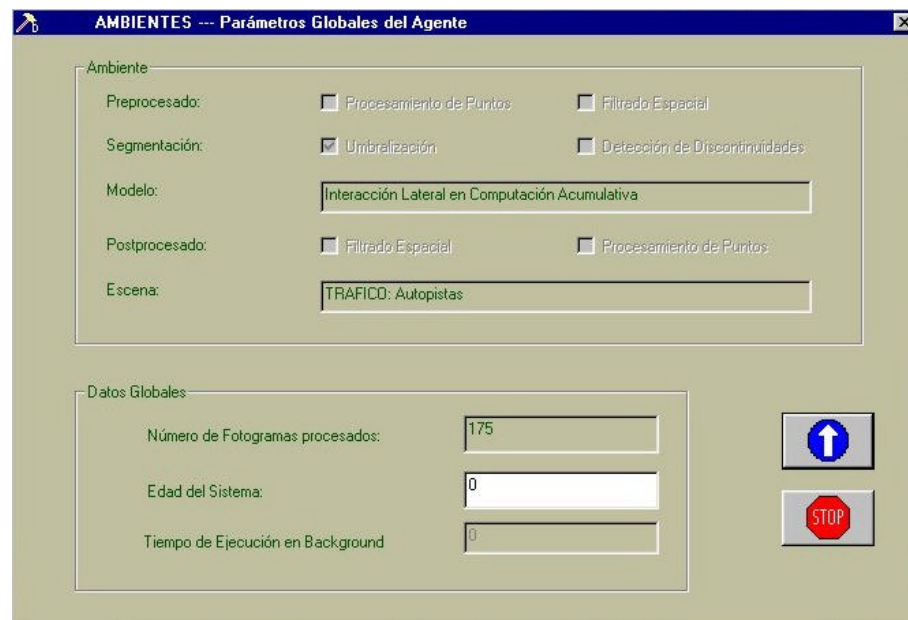


Figura 4.63. AMBIENTES --- Parámetros Globales del Agente

En esta caja de diálogo se muestran igualmente, en otro cuadro separado, los parámetros globales del sistema:

- ?? **Número de fotogramas procesados:** este campo nos muestra el número real de fotogramas que ha procesado el sistema. Se incrementa en 1 a cada fotograma procesado y no puede modificarse. Se trata, por tanto, de un campo de consulta.
- ?? **Edad del sistema:** marca la edad del funcionamiento del sistema. A cada cuadro procesado se incrementa en 1. Inicialmente la edad del sistema es 0. Puede ser modificada por el usuario para retroceder o avanzar en la vida del Agente.

?? **Tiempo de ejecución en background:** indica el tiempo que le queda libre al procesador para trabajar entre dos interrupciones de cuadro. Se trata igualmente de un campo de consulta, y no puede modificarse manualmente.

Opción Restricciones de Imágenes

Esta caja de diálogo tiene dos zonas claramente diferenciadas. Por una parte podemos observar (figura 4.64.) un cuadro que contiene la información básica de las Imágenes de Entrada. Este cuadro ayuda a englobar, de un solo vistazo, la información más relevante de las imágenes de entrada al sistema. La información incluye:

- ?? **Procedencia:** MAGNETO, CAMARA u OTROS
- ?? **Profundidad de Color:** número de bits / pixel
- ?? **Niveles de Gris:** número máximo de niveles de gris
- ?? **Tamaño:** tamaño total de cada imagen de entrada
- ?? **Número Columnas:** número de columnas de cada imagen
- ?? **Número Filas:** número de filas de cada imagen



Figura 4.64. AMBIENTES --- Restricciones de las Imágenes

El segundo cuadro permite elegir la **Resolución deseada de Proceso**. Evidentemente, la resolución deberá menor o igual al valor mínimo de **Número Columnas** y **Número Filas**. Una resolución menor permite obtener una velocidad de proceso mayor en detrimento de una calidad menor en el algoritmo.

Opción Preprocesado

La opción Preprocesado permite determinar qué acción (acciones) de preprocesado se deben realizar sobre las secuencias de imágenes de entrada que nutren al Agente. Estas serán las primeras operaciones sobre las imágenes de entrada (tras las pertinentes validaciones de formatos, tamaños, ...). El preprocesado tiene una amplia finalidad en este contexto, destinado fundamentalmente a eliminar ruidos en las imágenes.

Puede apreciarse en la figura 4.65. que el preprocesado se divide en dos grandes bloques, **Procesamiento de Puntos** y **Filtrado Espacial**.

Bajo **Procesamiento de Puntos** se engloban aquellas técnicas de mejora de la imagen basadas únicamente en la intensidad de los píxeles individuales. Estos procesos punto a punto se encuentran entre las técnicas más simples de la mejora de la imagen.

A la fecha se han incorporado las siguientes opciones de preprocesamiento por procesamiento de puntos:

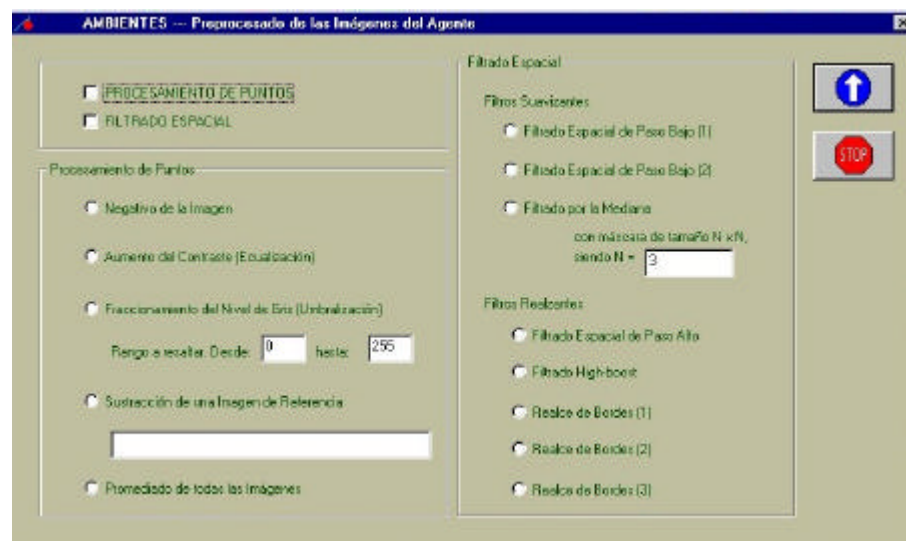


Figura 4.65. AMBIENTES --- Preprocesado de las Imágenes del Agente

?? **Negativo de la imagen:** Los negativos de las imágenes digitalizadas son útiles en numerosas aplicaciones, como la representación de imágenes médicas y en la obtención de fotografías de una pantalla con película monocroma con la idea de emplear los negativos resultantes como diapositivas. La idea es invertir el orden de blanco a negro, de forma que la intensidad de la imagen de salida disminuya conforme la intensidad de la imagen de entrada aumente.

- ?? **Aumento del contraste:** Las imágenes con poco contraste pueden ser debidas a diversas causas, entre las que destacamos una iluminación deficiente. La idea subyacente en las técnicas de aumento del contraste consiste en incrementar el rango dinámico de los niveles de gris de la imagen que se está procesando. En esta opción es imprescindible rellenar el valor del **% Contraste**.
- ?? **Fraccionamiento del Nivel de Gris:** A menudo se desea destacar un rango específico del nivel de gris de una imagen. Entre estas aplicaciones se encuentran la mejora de rangos como las masas de aguas en las imágenes de satélite o la mejora de defectos en las imágenes de rayos X. Tenemos que rellenar el rango a resaltar, es decir los valores de **Rango a resaltar. Desde y Hasta**.
- ?? **Sustracción de imágenes con imagen de referencia:** La sustracción de imágenes tiene numerosas e importantes aplicaciones en la mejora de las imágenes. Una aplicación clásica se tiene en un área de imágenes médicas conocida como la radiografía en modo máscara. Debemos incluir la localización y el nombre de la **Imagen de referencia**.
- ?? **Promediado de la imagen:** La imagen de salida está formada por la media de todas las imágenes de entrada que han llegado al Agente hasta el momento. Esta técnica se usa básicamente para eliminar el ruido.

De todas las opciones de Procesamiento de Puntos, podemos elegir a lo sumo una.

El empleo de máscaras espaciales para el procesamiento de las imágenes se denomina **Filtrado Espacial** y las propias máscaras se denominan filtros espaciales. Tenemos para el preprocesamiento por filtrado espacial:

?? **Filtros Suavizantes:**

Los filtros suavizantes se emplean para que la imagen aparezca algo borrosa y también para reducir el ruido. Es útil que la imagen aparezca algo borrosa en algunas etapas de preprocesado, como la eliminación de los pequeños detalles de una imagen antes de la extracción de un objeto, y el relleno de pequeños espacios entre líneas o curvas. Los filtros suavizantes actualmente presentes en Intelligent SILHFinder son: **Filtrado Espacial de Paso Bajo (1), Filtrado Espacial de Paso Bajo (2), Filtrado por la Mediana**.

?? **Filtros Realzantes:**

El objetivo principal del realce es el de destacar los detalles finos de una imagen o intensificar detalles que han sido difuminados, bien sea por error o bien por efecto natural del método de adquisición de la imagen. Intelligent SILHFinder ofrece los siguientes filtros realzantes: **Filtrado Espacial de Paso Alto**, **Filtrado High-boost**, **Realce de Bordes (1)**, **Realce de Bordes (2)**, **Realce de Bordes (3)**.

De todas las opciones de Filtrado Espacial, podemos elegir a lo sumo una.

El hecho de haber elegido alguna de las opciones de Procesamiento de Puntos o Filtrado Espacial se ve reflejado en el cuadro que contiene las opciones **PROCESAMIENTO DE PUNTOS** y **FILTRADO ESPACIAL**. Para eliminar toda opción de Procesamiento de Puntos o Filtrado Espacial, se ha de eliminar la muela de **PROCESAMIENTO DE PUNTOS** y **FILTRADO ESPACIAL**, respectivamente.

Opción Segmentación

La opción Segmentación permite determinar qué acción (acciones) de segmentación se deben realizar sobre las secuencias de imágenes de entrada previamente preprocesadas y que nutren al Agente. La segmentación subdivide una imagen en sus partes constituyentes u objetos.

Puede apreciarse en la figura 4.66. que la segmentación se ha dividido en dos grandes bloques: **Umbralización** y **Detección de Discontinuidades**.

La **Umbralización** es uno de los métodos más importantes de la segmentación de imágenes. Intelligent SILHFinder ofrece las siguientes técnicas de umbralización:

?? **Umbralización Global Simple:** Una imagen es segmentada en franjas equidistantes de bandas de nivel de gris. Un nivel de gris cae siempre en una única banda que lo incluye. Por ello, se requiere rellenar el campo **Número de Bandas de Nivel de Gris**.

?? **Umbralización Uniforme con Solapamiento:** En este caso, la idea es semejante a la anterior. La diferencia está en que para que un pixel cambie de banda de nivel de gris desde una imagen anterior hasta la actual, debe existir un cambio suficiente en el nivel de gris detectado. Por ello, se incorpora, además del campo **Número de Bandas de Nivel de Gris**, el parámetro **% Solapamiento**.

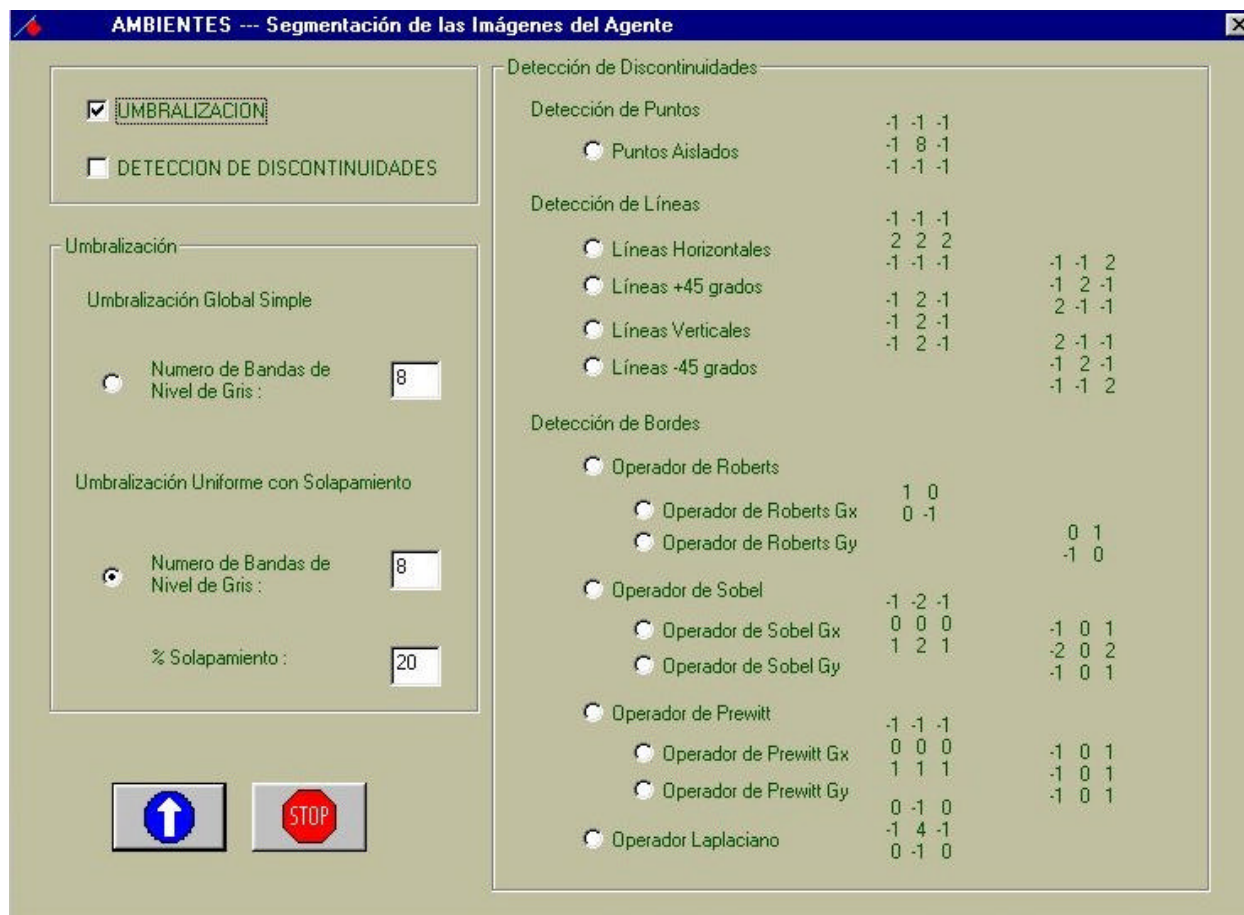


Figura 4.66. AMBIENTES --- Segmentación de las Imágenes del Agente

No puede elegirse más de una opción de las dos disponibles.

Ahora, bajo **Detección de Discontinuidades** se engloban varias técnicas para la detección de los tres tipos básicos de discontinuidades de una imagen digital: puntos, líneas y bordes. La forma más común de ver las discontinuidades es pasar una máscara (al igual que se hacía en el Filtrado Espacial). Intelligent SILHFinder se ha centrado en la detección de las siguientes discontinuidades: a) **Detección de Puntos (Puntos Aislados)**, b) **Detección de Líneas (Líneas Horizontales, Líneas orientadas a + 45°, Líneas Verticales, Líneas orientadas a - 45°)**, c) **Detección de Bordes (Operador de Roberts, Operador de Roberts Gx, Operador de Roberts Gy, Operador de Sobel, Operador de Sobel Gx, Operador de Sobel Gy, Operador de Prewitt, Operador de Prewitt Gx, Operador de Prewitt Gy, Operador Laplaciano)**.

Se permite elegir únicamente hasta una de las opciones anteriores de detección de discontinuidades.

El hecho de haber elegido alguna de las opciones de Umbralización o Detección de Discontinuidades se ve reflejado en el cuadro que contiene las opciones **UMBRALIZACION** y **DETECCION DE DISCONTINUIDADES**. Para eliminar toda opción de Umbralización o Detección de Discontinuidades, se ha de eliminar la muela de **UMBRALIZACION** y **DETECCION DE DISCONTINUIDADES**, respectivamente.

Opción Modelo

La opción Modelo lleva a una pantalla (ver figura 4.67.) que permite determinar el modelo a aplicar para la obtención de las siluetas. Actualmente, solamente existe la posibilidad de elegir el **Modelo de Interacción Lateral en Computación Acumulativa (ILCA)**.

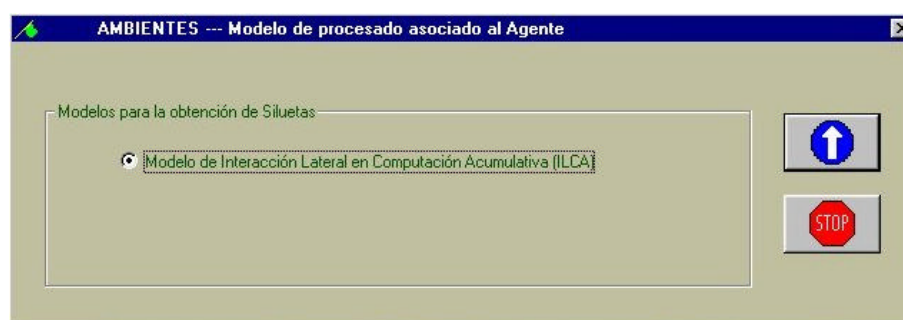


Figura 4.67. AMBIENTES --- Modelo de procesado asociado al Agente

Opción Postprocesado

La opción Postprocesado permite determinar qué acción (acciones) de postprocesado se deben realizar sobre las secuencias de imágenes que ya han pasado por el modelo relacionado con el Agente.

Puede apreciarse en la figura 4.68. que el postprocesado se divide en dos grandes bloques: **Filtrado Espacial** y **Procesamiento de Puntos**.

El empleo de máscaras espaciales para el procesamiento de las imágenes se denomina **Filtrado Espacial** y las propias máscaras se denominan filtros espaciales. Tenemos para el preprocesamiento por filtrado espacial: a) **Filtros Suavizantes (Filtrado Espacial de Paso Bajo (1), Filtrado Espacial de Paso Bajo (2), Filtrado por la Mediana)**, b) **Filtros Realzantes (Filtrado Espacial de Paso Alto, Filtrado High-boost, Realce de Bordes (1), Realce de Bordes (2), Realce de Bordes (3))**.

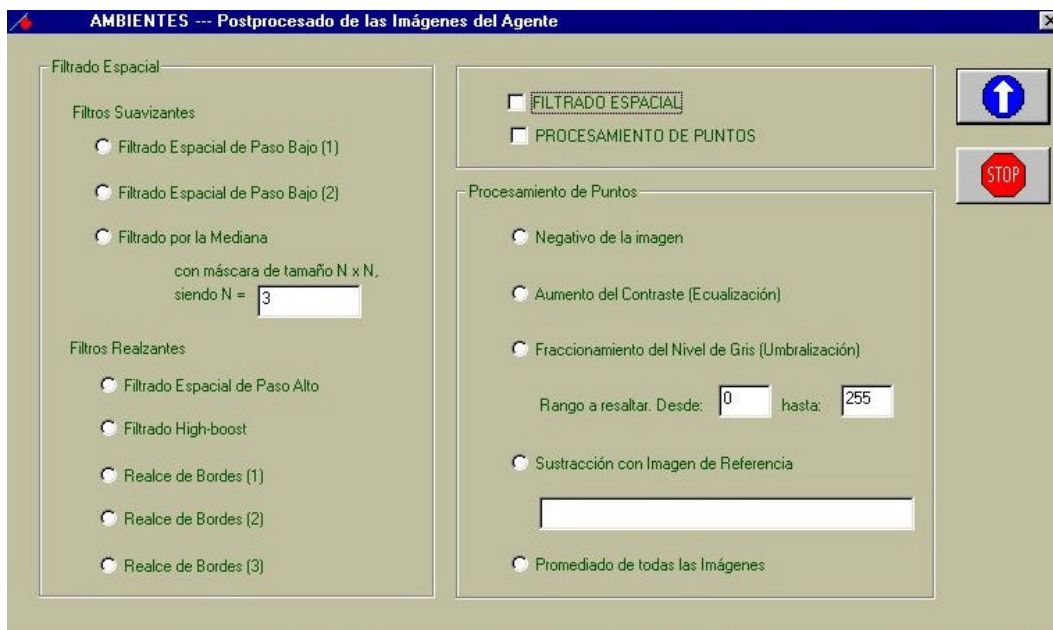


Figura 4.68. AMBIENTES --- Postprocesado de las Imágenes del Agente

De todas las opciones de Filtrado Espacial, podemos elegir a lo sumo una.

Bajo **Procesamiento de Puntos** se engloban aquellas técnicas de mejora de la imagen basadas únicamente en la intensidad de los píxeles individuales. Estos procesos punto a punto se encuentran entre las técnicas más simples de la mejora de la imagen.

A la fecha se han incorporado las siguientes opciones de postprocesamiento por procesamiento de puntos:

?? **Negativo de la imagen**

?? **Aumento del contraste:** En esta opción es imprescindible rellenar el valor del % **Contraste**.

?? **Fraccionamiento del Nivel de Gris:** Tenemos que rellenar el rango a resaltar, es decir los valores de **Rango a resaltar. Desde y Hasta**.

?? **Sustracción de imágenes con imagen de referencia:** Debemos incluir la localización y el nombre de la **Imagen de referencia**.

?? **Promediado de la imagen**

De todas las opciones de Procesamiento de Puntos, podemos elegir a lo sumo una.

El hecho de haber elegido alguna de las opciones de Procesamiento de Puntos o Filtrado Espacial se ve reflejado en el cuadro que contiene las opciones **PROCESAMIENTO DE PUNTOS** y **FILTRADO ESPACIAL**. Para eliminar toda opción de Procesamiento de Puntos o Filtrado Espacial, se ha de eliminar la muela de **PROCESAMIENTO DE PUNTOS** y **FILTRADO ESPACIAL**, respectivamente.

Opción Escena

Esta opción nos lleva a un caja de diálogo donde podemos elegir una de las posibles escenas dadas de alta en Intelligent SILHFinder. Esta Escena (íntimamente relacionada con el tipo de aplicación, o capa 4 del modelo) será parte integrante del Agente.

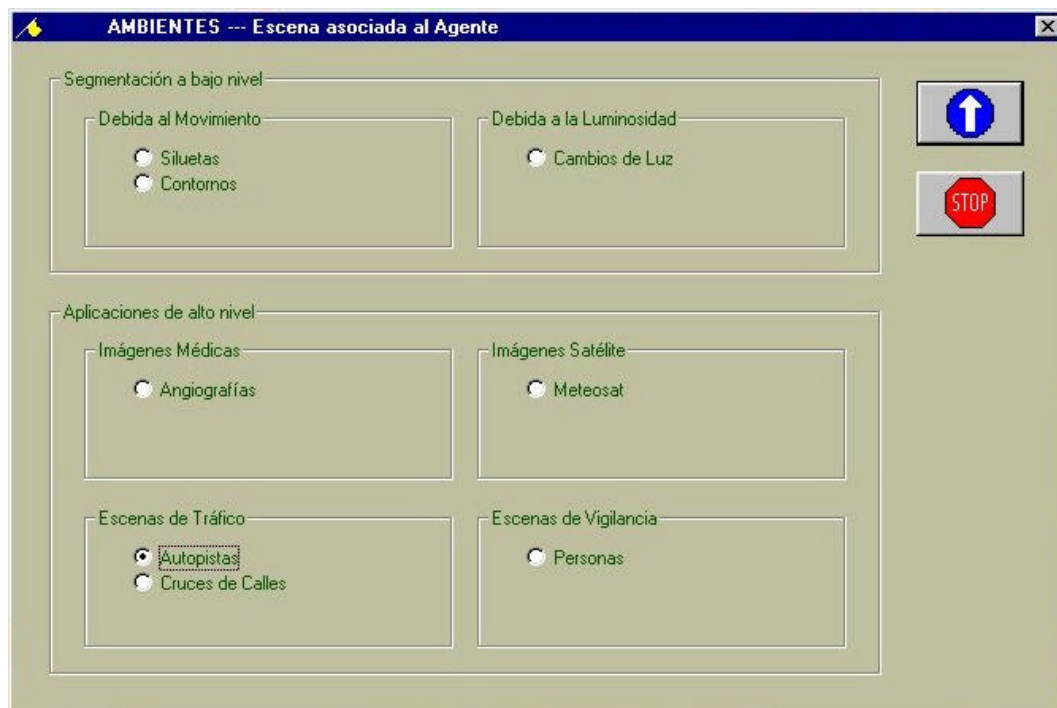


Figura 4.69. AMBIENTES --- Escena asociada al Agente

Las Escenas actualmente en vigencia son:

?? **Aplicaciones Generales:** Aquí irán aquellos Agentes que no están asociados a un tipo de aplicación claro y específico. Por ello, se ha optado por realizar la siguiente diferenciación:

?? **Segmentación:** La tarea que se acomete a este nivel es la obtención (sin mayores especificaciones de alto nivel) de:

?? **Siluetas:** Este tipo de Escena obtiene las siluetas de todos los objetos móviles presentes en una secuencia indefinida de imágenes.

?? **Contornos:** Este tipo de Escena aporta un paso más respecto de la obtención de las siluetas, a saber, la obtención de los contornos de todas las siluetas obtenidas.

?? **Luminosidad:** Se ha llegado a la conclusión que ciertos modelos, entre los que se encuentra el Modelo de Interacción Lateral en Computación Acumulativa, permiten obtener las siluetas de los objetos presentes en una sola imagen a partir del cambio forzado de la iluminación de la imagen. Ello se engloba en la opción:

?? **Cambios de Luz**

?? **Aplicaciones Específicas:** En este apartado se incluyen algunas aplicaciones de alto nivel, en las que nos hemos adentrado en el dominio de conocimiento, y hemos inyectado este conocimiento al Agente. Se ofrecen en esta versión de Intelligent SILHFinder algunas aplicaciones de interés, como pueden ser el manejo de:

?? **Imágenes Médicas**

?? **Angiografías:** Esta es una aproximación volumétrica a las imágenes vasculares, que permite ver las estructuras vasculares, buscando posibles lesiones.

?? **Imágenes Satélite**

?? **Meteosat:** La aplicación elegida en este caso es el seguimiento de la nubosidad a partir de imágenes tomadas por el satélite Meteosat.

?? **Escenas de Tráfico**

?? **Autopistas:** Aplicación de clasificación y conteo de vehículos en una autopista.

?? **Cruces de Calles:** La misma aplicación en cruces de calles, pero más enfocada a conocer la dirección que toman los vehículos.

?? **Escenas de Vigilancia**

?? **Personas:** Esta aplicación clasifica las personas en base a su velocidad de paso.

Evidentemente, solamente puede seleccionarse una Escena para el Ambiente en el que incluir el Agente.

Opción Parámetros de Aprendizaje del Agente

Esta caja de diálogo tiene dos zonas claramente diferenciadas. Por una parte podemos observar (figura 4.70.) un cuadro que contiene el tipo de aprendizaje a elegir para el Agente. Los posibles tipos de aprendizaje son:

- ?? **Mínimo Global sin restricciones del conocimiento:** explota toda la combinatoria posible de los parámetros fundamentales del Agente; por tanto, requiere de un tiempo muy elevado de cómputo (típicamente, más de 15 días)
- ?? **Mínimo Global con restricciones del conocimiento:** restringe la combinatoria anterior a los casos plausibles según el conocimiento del dominio; baja sensiblemente el tiempo del aprendizaje, pero aún así, es elevado (típicamente, más de un día)
- ?? **Mínimo Local con restricciones del conocimiento:** aplicando restricciones debidas al conocimiento del dominio se llega a un mínimo local que no garantiza a solución óptima; el tiempo de aprendizaje se reduce drásticamente (unos pocos minutos)

En esta caja de diálogo se muestran igualmente, en otro cuadro separado, los siguientes parámetros globales del sistema:

- ?? **Número de fotogramas procesados:** este campo nos muestra el número real de fotogramas que ha procesado el sistema. Se incrementa en 1 a cada fotograma procesado y no puede modificarse. Se trata, por tanto, de un campo de consulta.
- ?? **Edad del sistema:** marca la edad del funcionamiento del sistema. A cada cuadro procesado se incrementa en 1. Inicialmente la edad del sistema es 0. Puede ser modificada por el usuario para retroceder o avanzar en la vida del Agente.
- ?? **Tiempo de ejecución en background:** indica el tiempo que le queda libre al procesador para trabajar entre dos interrupciones de cuadro. Se trata igualmente de un campo de consulta, y no puede modificarse manualmente.



Figura 4.70. AMBIENTES --- Parámetros de Aprendizaje del Agente

4.3.3.3. Menú Modelos

A través del menú Modelos accedemos a todos los modelos dados de alta en la aplicación Intelligent SILHFinder, es decir a todas las variables o parámetros específicos de cada una de las capas que componen los modelos. Hasta la fecha solamente se ha incluido el **Modelo de Interacción Lateral en Computación Acumulativa (ILCA)**, habiéndose efectuado una descomposición del modelo en tres capas (véase la figura 4.71.):

?? **Interacción Lateral para la Computación Acumulativa**

?? **Interacción Lateral para el Reparto de Carga**

?? **Interacción Lateral para la Fusión de Objetos**

El acceso a cada una de estas tres opciones nos va a permitir cambiar los parámetros propios de estas tres capas del modelo.

No obstante, como se podrá apreciar en las pantallas asociadas a las tres capas del modelo, se dará la siguiente advertencia:

Los parámetros del modelo de interacción lateral en computación acumulativa se establecen inteligentemente durante la fase de aprendizaje.



Un cambio manual en alguno de los parámetros puede distorsionar gravemente el resultado obtenido.

Se ruega, por tanto, hacer caso de la advertencia. Así, se cambiarán los parámetros en casos excepcionales y bajo la responsabilidad del usuario.



Figura 4.71. Menú Modelos, Submenú Interacción Lateral en Computación Acumulativa

Para salir de cualquier caja de diálogo de este menú, tenemos dos opciones:

- ??  **Aceptar:** pulsando este botón se actualizan las modificaciones introducidas y volvemos al marco principal de la aplicación
- ??  **Cancelar:** pulsando este botón se deshacen las modificaciones introducidas y volvemos al marco principal de la aplicación

Opción Interacción Lateral para la Computación Acumulativa

La opción **Interacción Lateral para la Computación Acumulativa** del menú **Modelos** permite establecer aquellos parámetros requeridos por el hardware y/o software de la capa 1 del modelo de interacción en computación acumulativa de procesamiento de imágenes de la aplicación Intelligent SILHFinder. Esta capa 1 se encarga de la interacción lateral para la computación acumulativa por bandas de nivel de gris, de ahí que los parámetros propios de esta capa sean los siguientes:

- ?? **Número de Bandas de Nivel de Gris:** el número de bandas de gris en las que vienen segmentadas las imágenes (parámetro compartido con la fase de segmentación (umbralización) y las capas 2 y 3; típicamente toma el valor 8)
- ?? **Valor Mínimo de Permanencia:** el valor mínimo que puede tomar el valor de la memoria de permanencia, o sea, el valor de descarga (típicamente 0)
- ?? **Valor Máximo de Permanencia:** el valor máximo que puede tomar el valor de la memoria de permanencia, o sea, el valor de saturación (típicamente 255)
- ?? **Valor de Descarga por Detección de Movimiento:** el valor de descarga decrementado a la memoria de permanencia debido a la detección de movimiento
- ?? **Valor de Recarga por Vecindad:** el valor aplicado como suma debido a la interacción lateral en forma de recarga por vecindad
- ?? **Número de Vecinos para la Interacción Lateral:** el número de vecinos lineales a los que se puede llevar la interacción lateral de recarga
- ?? **Valor Mínimo de Mancha por Banda de Nivel de Gris:** el valor mínimo de la memoria de permanencia que permite efectuar un reparto de carga con los vecinos de la misma banda de nivel de gris

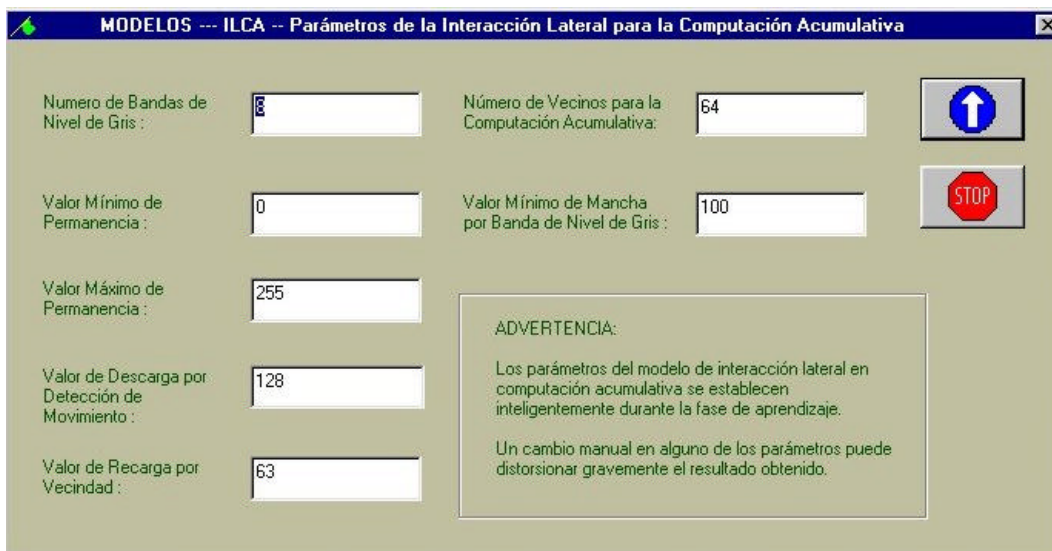


Figura 4.72. MODELOS --- ILCA -- Parámetros de la Interacción Lateral para la Computación Acumulativa

Opción Interacción Lateral para el Reparto de Carga

La opción **Interacción Lateral para el Reparto de Carga** del menú **Modelos** permite establecer aquellos parámetros requeridos por el hardware y/o software de la capa 2 del modelo de interacción lateral en computación acumulativa de procesamiento de imágenes de la aplicación Intelligent SILHFinder. Esta capa 2 se encarga de la interacción lateral para la obtención de los elementos de las siluetas por bandas de nivel de gris, de ahí que los parámetros propios de esta capa sean los siguientes:

- ?? **Número de Bandas de Nivel de Gris:** el número de bandas de gris en las que vienen segmentadas las imágenes (parámetro compartido con la fase de segmentación (umbralización) y las capas 1 y 3; típicamente toma el valor 8)
- ?? **Número de Vecinos para el Reparto de Carga por Bandas de Nivel de Gris:** el número de vecinos lineales a los que se puede llevar la interacción lateral de reparto de carga en la misma banda de nivel de gris
- ?? **Valor Mínimo de Mancha para la Fusión en Objetos:** el valor mínimo de la memoria de permanencia que permite efectuar un reparto de carga con los vecinos independientemente de su banda de nivel de gris (parámetro compartido con la capa 3)

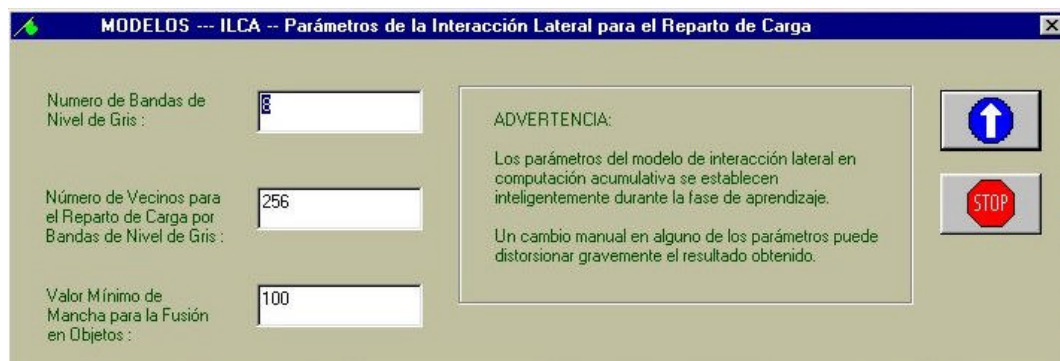


Figura 4.73. MODELOS --- ILCA -- Parámetros de la Interacción Lateral para el Reparto de Carga

Opción Interacción Lateral para la Fusión de Objetos

La opción **Interacción Lateral para la Fusión de Objetos** del menú **Modelos** permite establecer aquellos parámetros requeridos por el hardware y/o software de la capa 3 del modelo de interacción lateral en computación acumulativa de procesamiento de imágenes de la aplicación Intelligent SILHFinder. Esta capa 3 se encarga de la interacción lateral para la fusión de siluetas, de ahí que los parámetros propios de esta capa sean los siguientes:

- ?? **Número de Bandas de Nivel de Gris:** el número de bandas de gris en las que vienen segmentadas las imágenes (parámetro compartido con la fase de segmentación (umbralización) y las capas 1 y 2; típicamente toma el valor 8)
- ?? **Número de Vecinos para el Reparto de Carga por Objeto:** el número de vecinos lineales a los que se puede llevar la interacción lateral de reparto de carga independientemente de su banda de nivel de gris
- ?? **Valor Umbral Mínimo para la Detección de Objetos:** el valor mínimo de la memoria de permanencia que permite indicar la existencia de un objeto hacia el exterior de la aplicación Intelligent SILHFinder

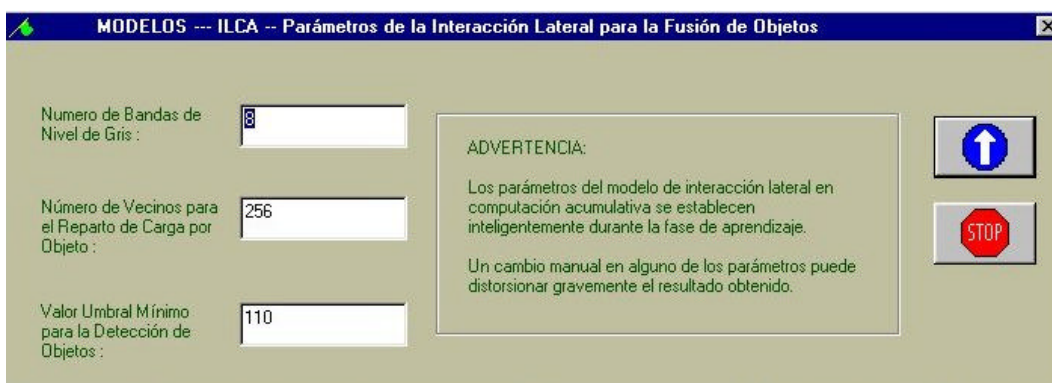


Figura 4.74. MODELOS --- ILCA -- Parámetros de la Interacción Lateral para la Fusión de Objetos

4.3.3.4. Menú Escenas

El menú Escenas nos lleva a definir los parámetros para el tipo de aplicación o escena relacionada con el Agente actualmente activo. La escena ya se habrá definido anteriormente en la opción Escena del menú Ambientes.

Los parámetros de Escenas vienen clasificados en dos grandes grupos, como puede apreciarse en la figura 4.75.: **Segmentación a bajo nivel** y **Aplicaciones de alto nivel**.

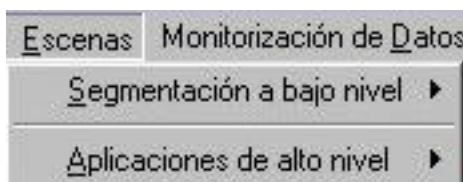




Figura 4.75. Menú Escenas

Para salir de cualquier caja de diálogo de este menú, tenemos dos opciones:

- ??  **Aceptar:** pulsando este botón se actualizan las modificaciones introducidas y volvemos al marco principal de la aplicación
- ??  **Cancelar:** pulsando este botón se deshacen las modificaciones introducidas y volvemos al marco principal de la aplicación

Submenú Segmentación a bajo nivel

Aquí irán los parámetros de aquellos Agentes que no están asociados a un tipo de aplicación claro y específico. Por ello, se ha optado por realizar la siguiente diferenciación:

- ?? **Segmentación:** La tarea que se acomete a este nivel es la obtención (sin mayores especificaciones de alto nivel) de:
 - ?? **Siluetas:** Este tipo de Escena obtiene las siluetas de todos los objetos móviles presentes en una secuencia indefinida de imágenes.
 - ?? **Contornos:** Este tipo de Escena aporta un paso más respecto de la obtención de las siluetas, a saber, la obtención de los contornos de todas las siluetas obtenidas.
- ?? **Luminosidad:** Se ha llegado a la conclusión que ciertos modelos, entre los que se encuentra el Modelo de Interacción Lateral en Computación Acumulativa, permiten obtener las siluetas de los objetos presentes en una sola imagen a partir del cambio forzado de la iluminación de la imagen. Ello se engloba en la opción:
 - ?? **Cambios de Luz**



Figura 4.76. Menú Escenas, Submenú Segmentación a bajo nivel

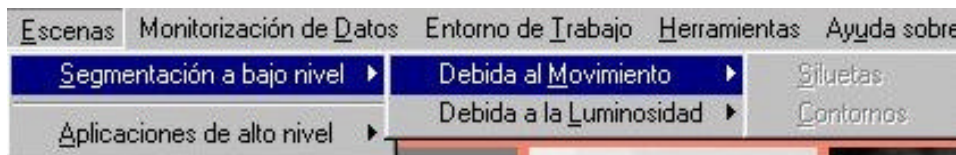


Figura 4.77. Menú Escenas, Submenú Segmentación a bajo nivel – Debida al Movimiento

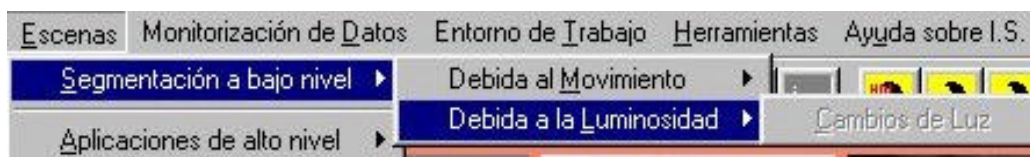


Figura 4.78. Menú Escenas, Submenú Segmentación a bajo nivel – Debida a la Luminosidad

Opción Siluetas

La opción Siluetas lleva a una caja de diálogo donde parametrizar el modo de obtener las siluetas. En estos momentos, esta opción no requiere de parámetros, por lo que el diálogo está vacío.



Figura 4.79. ESCENAS --- Segmentación a bajo nivel – Debida al Movimiento - Siluetas

Opción Contornos

La opción Contornos lleva a una caja de diálogo donde parametrizar el modo de obtener los contornos. En estos momentos, esta opción no requiere de parámetros, por lo que el diálogo está vacío.

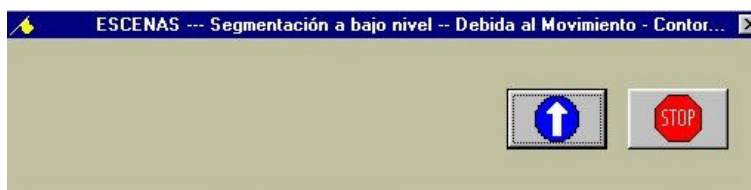


Figura 4.80. ESCENAS --- Segmentación a bajo nivel – Debida al Movimiento - Contornos

Opción Cambios de Luz

La opción Cambios de Luz admite un parámetro, que es el **Número de Bandas para el Cambio de Luminosidad**. Este parámetro va a determinar cuantas imágenes con distinta luminosidad se van a generar para la obtención de las siluetas a partir del cambio debido a la luminosidad.

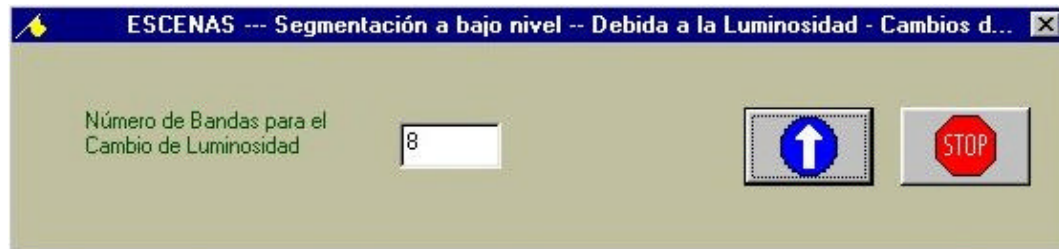


Figura 4.81. ESCENAS --- Segmentación a bajo nivel – Debida a la Luminosidad – Cambios de Luz

Submenú Aplicaciones de alto nivel

En este apartado se incluyen algunas aplicaciones de alto nivel, en las que nos hemos adentrado en el dominio de conocimiento, y hemos inyectado este conocimiento al Agente. Se ofrecen en esta versión de Intelligent SILHFinder algunas aplicaciones de interés, como pueden ser el manejo de:



Figura 4.82. Menú Escenas, Submenú Aplicaciones de alto nivel

?? Imágenes Médicas

?? **Angiografías:** Esta es una aproximación volumétrica a las imágenes vasculares, que permite ver las estructuras vasculares, buscando posibles lesiones.

?? Imágenes Satélite

?? **Meteosat**: La aplicación elegida en este caso es el seguimiento de la nubosidad a partir de imágenes tomadas por el satélite Meteosat.

?? Escenas de Tráfico

?? **Autopistas**: Aplicación de clasificación y conteo de vehículos en una autopista.

?? **Cruces de Calles**: La misma aplicación en cruces de calles, pero más enfocada a conocer la dirección que toman los vehículos.

?? Escenas de Vigilancia

?? **Personas**: Esta aplicación clasifica las personas en base a su velocidad de paso.



Figura 4.83. Menú Escenas, Submenú Aplicaciones de alto nivel – Imágenes Médicas



Figura 4.84. Menú Escenas, Submenú Aplicaciones de alto nivel – Escenas de Tráfico



Figura 4.85. Menú Escenas, Submenú Aplicaciones de alto nivel – Escenas de Vigilancia



Figura 4.86. Menú Escenas, Submenú Aplicaciones de alto nivel – Imágenes Satélite

Opción Angiografías

La opción Angiografías lleva a una caja de diálogo donde parametrizar el modo de obtener los resultados de la aplicación.

El parámetro actualmente en vigor es el del **Número de Imágenes que se desea visualizar**. Este parámetro indica cuantas imágenes queremos ver simultáneamente durante la monitorización de este tipo de Escena.



Figura 4.87. ESCENAS --- Aplicaciones de alto nivel -- Imágenes Médicas - Angiografías

Opción Autopistas

La pantalla asociada a esta opción se ofrece en la figura 4.88.

La opción Autopistas incorpora en el diálogo correspondiente a la entrada de los parámetros de la aplicación relacionada los siguientes campos de interés:

- ?? **Aplicación:** Permite elegir (activando o desactivando la opción correspondiente) el (los) tipo(s) de aplicación que se desea lanzar en el contexto de la vigilancia en autopistas. Las posibilidades ofrecidas son:
 - ?? **Clasificación de Vehículos:** Aplicación de clasificación de los vehículos en función de su tamaño (ver características de los vehículos).
 - ?? **Conteo de Vehículos:** Aplicación de conteo de los vehículos en función de su tamaño.

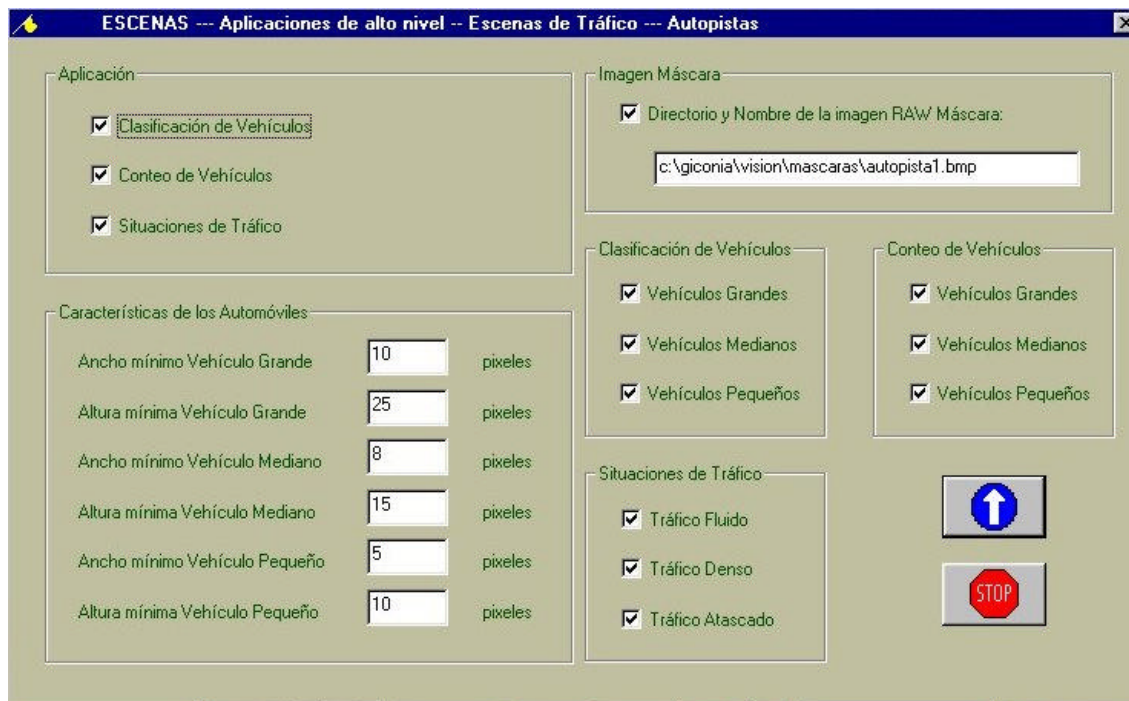


Figura 4.88. ESCENAS --- Aplicaciones de alto nivel – Escenas de Tráfico - Autopistas

?? **Situaciones de Tráfico:** Aplicación que ofrece la situación del tráfico en un momento dado, pudiéndose hablar de tráfico fluido, tráfico denso y tráfico atascado.

?? **Características de los Vehículos**

?? **Ancho mínimo Vehículo Grande:** Este parámetro da (en pixeles) el ancho mínimo para poder clasificar un vehículo como vehículo grande (véase camión, trailer, autocar, ...).

?? **Altura mínima Vehículo Grande:** Este parámetro da (en pixeles) la altura mínima para poder clasificar un vehículo como vehículo grande (véase camión, trailer, autocar, ...).

?? **Ancho mínimo Vehículo Mediano:** Este parámetro da (en pixeles) el ancho mínimo para poder clasificar un vehículo como vehículo mediano (véase coche, furgoneta, ...).

?? **Altura mínima Vehículo Mediano:** Este parámetro da (en pixeles) la altura mínima para poder clasificar un vehículo como vehículo mediano (véase coche, furgoneta, ...).

- ?? **Ancho mínimo Vehículo Pequeño:** Este parámetro da (en píxeles) el ancho mínimo para poder clasificar un vehículo como vehículo pequeño (véase moto, ...).
- ?? **Altura mínima Vehículo Pequeño:** Este parámetro da (en píxeles) la altura mínima para poder clasificar un vehículo como vehículo pequeño (véase moto, ...).
- ?? **Imagen Máscara:** Si solamente queremos considerar una zona de la imagen para su estudio, aplicamos una máscara en forma de imagen con formato RAW. Tendremos que rellenar el:
- ?? **Directorio y Nombre de la Imagen RAW Máscara**
- ?? **Clasificación de Vehículos:** Si hemos elegido la aplicación de clasificación de vehículos, podemos optar por clasificar: **Vehículos Grandes, Vehículos Medianos, Vehículos Pequeños.**
- ?? **Conteo de Vehículos:** Si hemos elegido la aplicación de conteo de vehículos, podemos optar por contar: **Vehículos Grandes, Vehículos Medianos, Vehículos Pequeños.**
- ?? **Situaciones de Tráfico:** Si hemos elegido la aplicación de obtención de las situaciones de tráfico, podemos optar por centrarnos en los casos de: **Tráfico Fluido, Tráfico Denso, Tráfico Atascado.**

Opción Cruces de Calles

La opción Cruces de Calles incorpora en el diálogo correspondiente a la entrada de los parámetros de la aplicación relacionada los siguientes campos de interés:

- ?? **Aplicación:** Permite elegir (activando o desactivando la opción correspondiente) el (los) tipo(s) de aplicación que se desea lanzar en el contexto de la vigilancia en cruces de calles. Las posibilidades ofrecidas son:
- ?? **Conteo de Vehículos:** Aplicación de conteo de los vehículos.

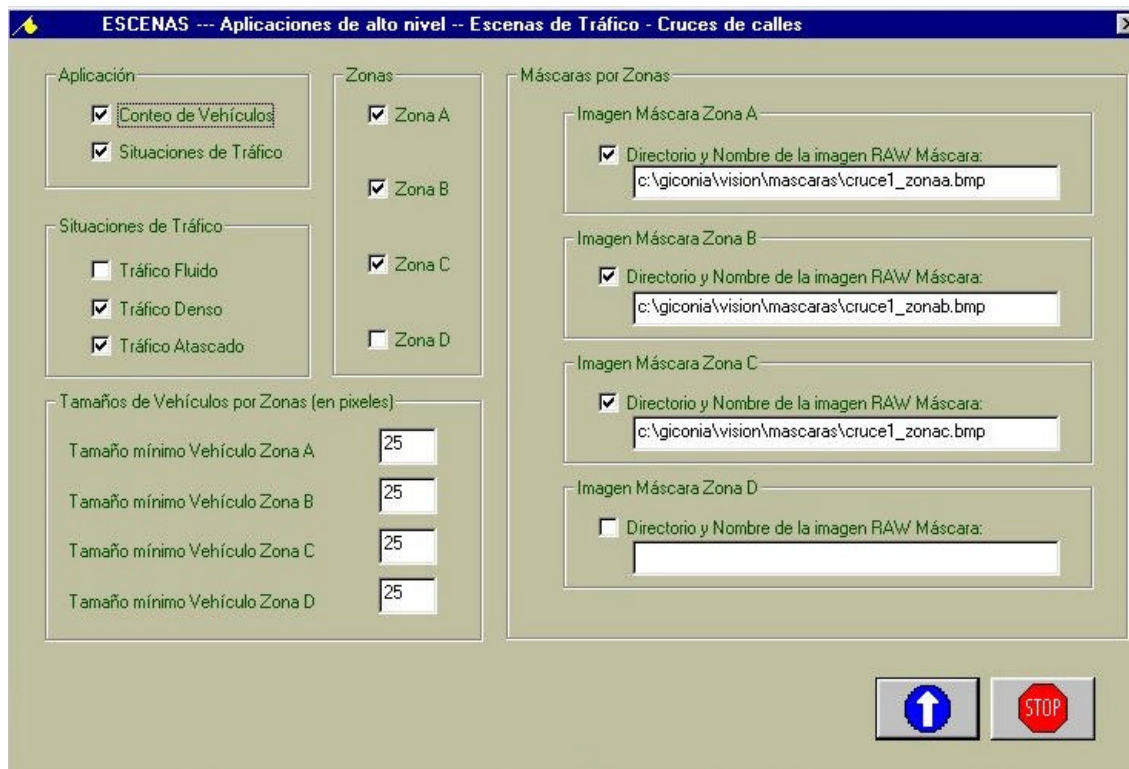


Figura 4.89. ESCENAS --- Aplicaciones de alto nivel – Escenas de Tráfico - Cruces de Calles

?? **Situaciones de Tráfico:** Aplicación que ofrece la situación del tráfico en un momento dado, pudiéndose hablar de tráfico fluido, tráfico denso y tráfico atascado.

?? **Situaciones de Tráfico:** Si hemos elegido la aplicación de obtención de las situaciones de tráfico, podemos optar por centrarnos en los casos de: **Tráfico Fluído, Tráfico Denso, Tráfico Atascado.**

?? **Zonas:** Esta aplicación permite centrar nuestra atención en hasta 4 zonas distintas, de modo que pueda controlarse perfectamente el flujo del tráfico de unas zonas a otras. Estas zonas se denominan: **Zona A, Zona B, Zona C, Zona D.**

?? **Tamaños de Vehículos por Zonas**

?? **Tamaño mínimo Vehículo Zona A:** Este parámetro da (en pixeles) el tamaño mínimo para poder determinar que un objeto es un vehículo en la Zona A.

?? **Tamaño mínimo Vehículo Zona B:** Este parámetro da (en pixeles) el tamaño mínimo para poder determinar que un objeto es un vehículo en la Zona B.

- ?? **Tamaño mínimo Vehículo Zona C:** Este parámetro da (en pixeles) el tamaño mínimo para poder determinar que un objeto es un vehículo en la Zona C.
- ?? **Tamaño mínimo Vehículo Zona D:** Este parámetro da (en pixeles) el tamaño mínimo para poder determinar que un objeto es un vehículo en la Zona D.
- ?? **Máscaras por Zonas:** Las zonas de interés en la imagen se obtienen a partir de unas máscara en forma de imágenes con formato RAW. Por cada una de las zonas (hasta 4) tendremos:
 - ?? **Imagen Máscara Zona A (Directorio y Nombre de la Imagen RAW Máscara)**
 - ?? **Imagen Máscara Zona B (Directorio y Nombre de la Imagen RAW Máscara)**
 - ?? **Imagen Máscara Zona C (Directorio y Nombre de la Imagen RAW Máscara)**
 - ?? **Imagen Máscara Zona D (Directorio y Nombre de la Imagen RAW Máscara)**

Opción Personas

La opción Personas relacionada con la aplicación de vigilancia de personas incorpora en el diálogo correspondiente a la entrada de los parámetros de la aplicación relacionada los siguientes campos de interés:

- ?? **Situaciones de Riesgo:** Podemos optar por centrarnos en los casos de: **Velocidad de paso Baja, Velocidad de paso Media, Velocidad de paso Alta.**

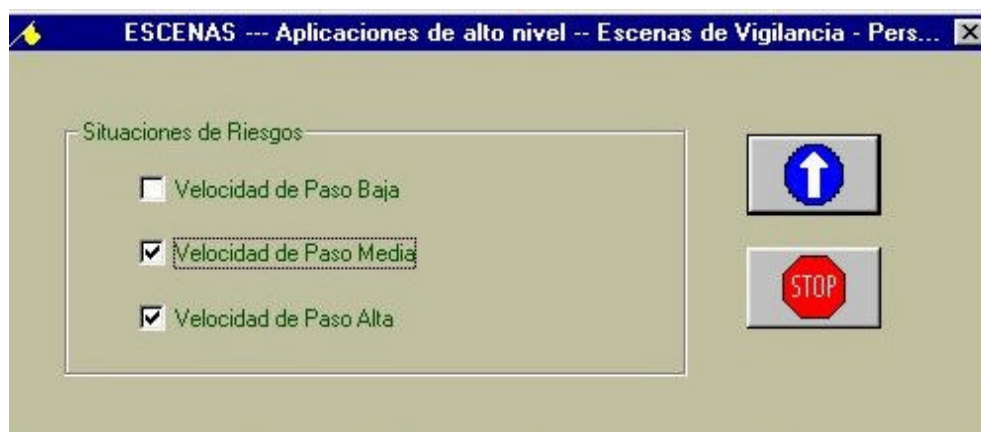


Figura 4.90. ESCENAS --- Aplicaciones de alto nivel – Escenas de Vigilancia - Personas

Opción Meteosat

La opción Meteosat incorpora en el diálogo correspondiente a la entrada de los parámetros de la aplicación relacionada los siguientes campos de interés:



Figura 4.91. ESCENAS --- Aplicaciones de alto nivel – Imágenes Satélite - Meteosat

?? **Aplicación:** Permite elegir (activando o desactivando la opción correspondiente) el (los) tipo(s) de aplicación que se desea lanzar en el contexto del tratamiento de imágenes obtenidas por el satélite Meteosat y destinadas a la obtención de información meteorológica. Las posibilidades ofrecidas son:

?? **Conteo de Nubes:** Aplicación de conteo de las nubes en función de su tamaño.

?? **Situaciones de Nubosidad:** Aplicación que ofrece la situación de la nubosidad en un momento dado, pudiéndose hablar de despejado y cubierto.

?? **Características de las Nubes**

?? **Nubes Grandes:** Este parámetro da (en píxeles) el tamaño mínimo para poder clasificar una nube como grande.

?? **Nubes Medianas:** Este parámetro da (en píxeles) el tamaño mínimo para poder clasificar una nube como mediana.

?? **Nubes Pequeñas:** Este parámetro da (en píxeles) el tamaño mínimo para poder clasificar una nube como pequeña.

?? **Conteo de Nubes:** Si hemos elegido la aplicación de conteo de nubes, podemos optar por contar: **Nubes Grandes, Nubes Medianas, Nubes Pequeñas.**


?? **Situaciones de Nubosidad:** Si hemos elegido la aplicación de obtención de las situaciones de nubosidad, podemos optar por centrarnos en los casos de: **Despejado y Cubierto.**

?? **Imagen Máscara:** Si solamente queremos considerar una zona de la imagen para su estudio, aplicamos una máscara en forma de imagen con formato RAW. Tendremos que rellenar el:

?? **Directorio y Nombre de la Imagen RAW Máscara**

4.3.3.5. Menú Monitorización de datos

Este menú está destinado a la monitorización de los parámetros del sistema Intelligent SILHFinder, fundamentalmente mientras éste está arrancado en modo DIFERIDO. Gracias a las opciones de este menú podemos seguir todas las incidencias del programa en proceso.

Nótese que todos los cuadros de diálogo de este menú son no modales, es decir, que pueden mostrarse con cualquier otro cuadro de diálogo no modal. Para salir de cualquiera de estas cajas de diálogo púlsese el botón  **Cerrar.**

Monitorización de datos durante el Aprendizaje del Agente

Durante la fase de aprendizaje, solamente se tiene acceso a las dos opciones:

?? **Parámetros Globales**

?? **Imágenes Aprendizaje**

La primera opción se verá más adelante.



Figura 4.92. Menú Monitorización de datos durante el Aprendizaje del Agente

Opción Imágenes del Aprendizaje

En cuanto la segunda, como puede apreciarse en la figura 4.93., se puede ver, por un lado, en forma de imágenes:

?? **La entrada**

?? **La salida de la Segmentación (capa 0 del modelo)**

?? **La salida de la capa 1 del modelo**, es decir de la Interacción Lateral para la Computación Acumulativa

?? **La salida de la capa 2 del modelo**, es decir de la Interacción Lateral para el Reparto de Carga

?? **La salida de la capa 3 del modelo**, es decir de la Interacción Lateral para la Fusión en Objetos

Tenemos también los tres parámetros fundamentales de entrada al aprendizaje para una correcta evolución del Agente hacia los parámetros óptimos: **Tamaño Máximo de Silueta, Tamaño Mínimo de Silueta, Número de Elementos Móviles a Detectar.**

Nos encontramos también con los parámetros que evolucionan a lo largo del aprendizaje. En todo momento, la pantalla nos ofrece los valores actuales y los óptimos encontrados hasta el momento actual. Estos parámetros son: **Número de Bandas de Nivel de Gris, Valor de Descarga por Detección de Movimiento, Valor de Recarga por Vecindad, Valor Umbral Mínimo de Mancha por Banda de Nivel de Gris, Valor Umbral Mínimo para la Fusión en Objetos, Valor Umbral Mínimo para la Detección de Objetos.**

La pantalla ofrece también el importante parámetro resultado del aprendizaje, denominado **Numero de Siluetas Detectadas**.

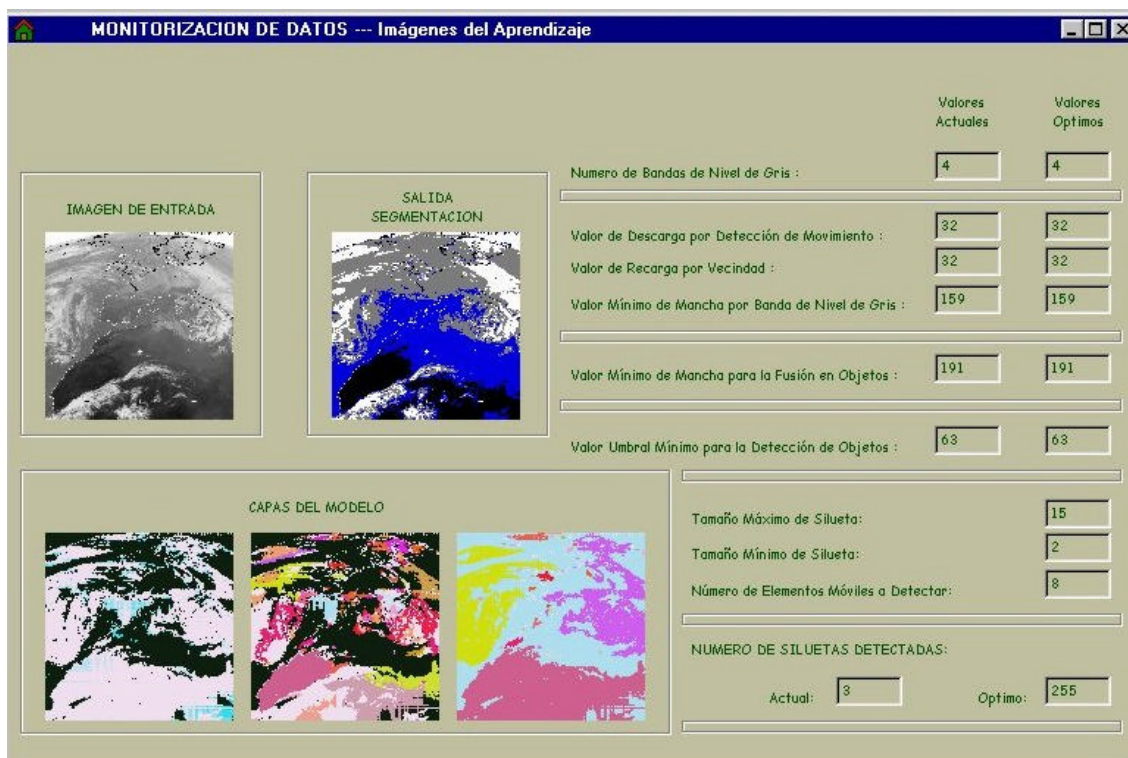


Figura 4.93. MONITORIZACION DE DATOS --- Imágenes del Aprendizaje

Monitorización de datos

Las opciones son siete como puede verse en la figura 4.94.: **Parámetros Globales, Preprocesado, Segmentación, Modelos, Postprocesado, Escena e Imágenes.**

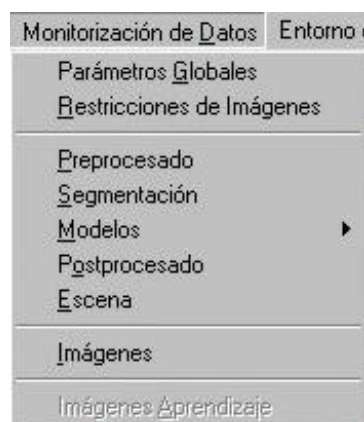


Figura 4.94. Menú Monitorización de datos

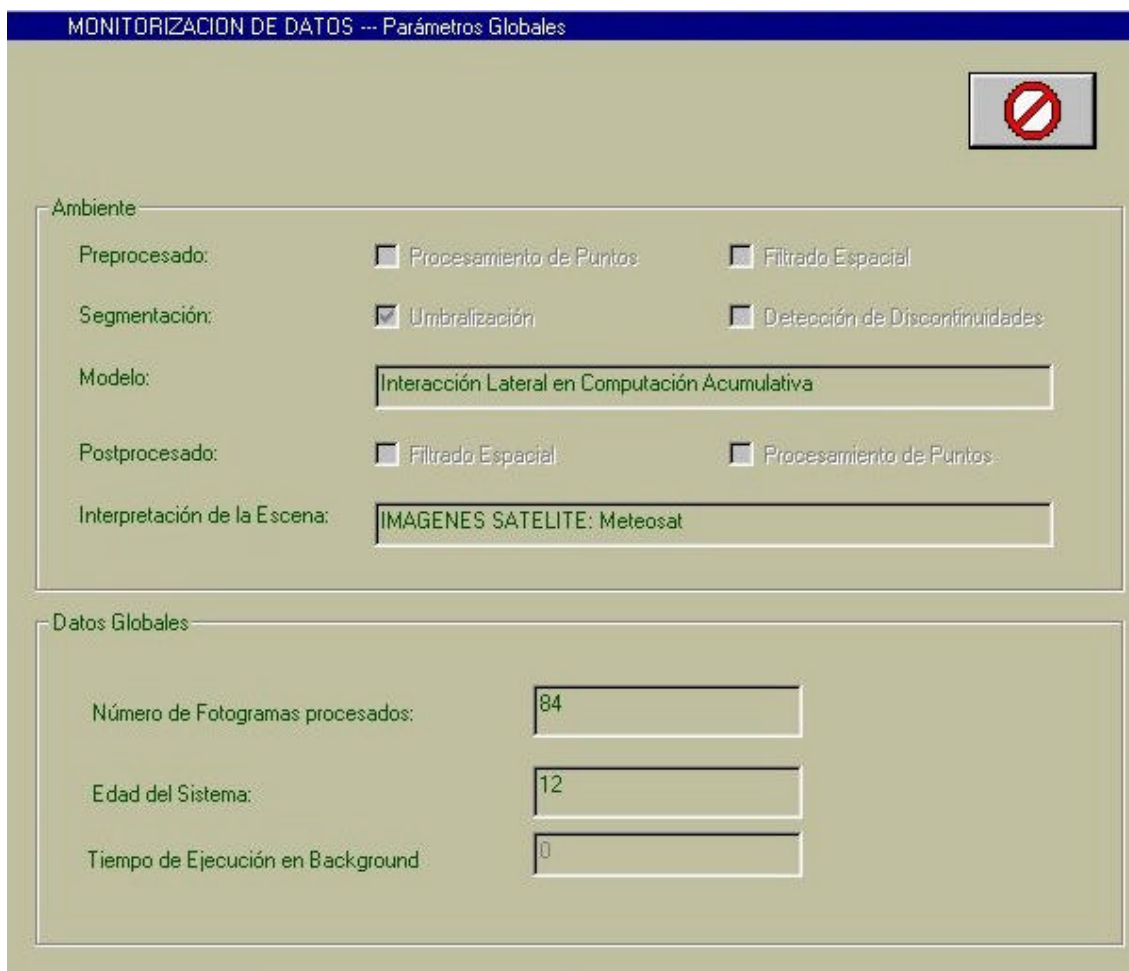
Opción Parámetros Globales

Esta caja de diálogo tiene dos zonas claramente diferenciadas. Por una parte podemos observar (figura 4.95.) un cuadro que contiene la información básica del Ambiente del Agente. Este cuadro ayuda a englobar, de un solo vistazo, el Agente en su Ambiente completo. La información incluye:

- ?? **Preprocesado:** presencia o no de preprocesado a nivel de **Procesamiento de Puntos** y/o de **Filtrado Espacial**
- ?? **Segmentación:** presencia o no de segmentación a nivel de **Umbralización** y/o **Detección de Discontinuidades**
- ?? **Modelo:** tipo de modelo elegido para el Agente
- ?? **Postprocesado:** presencia o no de postprocesado a nivel de **Procesamiento de Puntos** y/o de **Filtrado Espacial**
- ?? **Interpretación de la Escena:** tipo de escena donde se encuadra el Agente

En esta caja de diálogo se muestran igualmente, en otro cuadro separado, los parámetros globales del sistema:

- ?? **Número de fotogramas procesados:** este campo nos muestra el número real de fotogramas que ha procesado el sistema. Se incrementa en 1 a cada fotograma procesado.
- ?? **Edad del sistema:** marca la edad del funcionamiento del sistema. A cada cuadro procesado se incrementa en 1. Inicialmente la edad del sistema es 0.
- ?? **Tiempo de ejecución en background:** indica el tiempo que le queda libre al procesador para trabajar entre dos interrupciones de cuadro.



MONITORIZACION DE DATOS --- Parámetros Globales

Ambiente

Preprocesado: Procesamiento de Puntos Filtrado Espacial

Segmentación: Umbralización Detección de Discontinuidades

Modelo:

Postprocesado: Filtrado Espacial Procesamiento de Puntos

Interpretación de la Escena:

Datos Globales

Número de Fotogramas procesados:

Edad del Sistema:

Tiempo de Ejecución en Background:

Figura 4.95. MONITORIZACION DE DATOS --- Parámetros Globales

Opción Restricciones de Imágenes

Esta caja de diálogo tiene dos zonas claramente diferenciadas. Por una parte podemos observar (figura 4.96.) un cuadro que contiene la información básica de las Imágenes de Entrada. Este cuadro ayuda a englobar, de un solo vistazo, la información más relevante de las imágenes de entrada al sistema. La información incluye:

- ?? **Procedencia:** MAGNETO, CAMARA u OTROS
- ?? **Profundidad de Color:** número de bits / pixel
- ?? **Niveles de Gris:** número máximo de niveles de gris
- ?? **Tamaño:** tamaño total de cada imagen de entrada
- ?? **Número Columnas:** número de columnas de cada imagen

?? **Número Filas:** número de filas de cada imagen

Información de las Imágenes de Entrada	
Procedencia	OTROS
Profundidad de Color (nº bits / pixel)	8
Niveles de Gris	256
Tamaño (bytes)	262144
Número Columnas	512
Número Filas	512

Resolución deseada de Proceso:	
Anchura / Altura Máxima:	
<input type="radio"/>	512 pixeles
<input type="radio"/>	256 pixeles
<input checked="" type="radio"/>	128 pixeles
<input type="radio"/>	64 pixeles

Datos Globales	
Número de Fotogramas procesados:	100
Edad del Sistema:	28
Tiempo de Ejecución en Background	0

Figura 4.96. MONITORIZACION DE DATOS --- Restricciones de las Imágenes

El segundo cuadro permite visualizar la **Resolución deseada de Proceso**.

Opción Preprocesado

La opción Preprocesado permite ver qué acción (acciones) de preprocesado se van a realizar sobre las secuencias de imágenes de entrada que nutren al Agente. Estas serán las primeras operaciones sobre las imágenes de entrada (tras las pertinentes validaciones de formatos, tamaños, ...). El preprocesado tiene una amplia finalidad en este contexto, destinado fundamentalmente a eliminar ruidos en las imágenes.

Puede apreciarse en la figura 4.97. que el preprocesado se divide en dos grandes bloques: **Procesamiento de Puntos y Filtrado Espacial**.

Bajo **Procesamiento de Puntos** se engloban aquellas técnicas de mejora de la imagen basadas únicamente en la intensidad de los píxeles individuales. Estos procesos punto a punto se encuentran entre las técnicas más simples de la mejora de la imagen.

A la fecha se han incorporado las siguientes opciones de preprocesamiento por procesamiento de puntos:

- ?? **Negativo de la imagen:** Los negativos de las imágenes digitalizadas son útiles en numerosas aplicaciones, como la representación de imágenes médicas y en la obtención de fotografías de una pantalla con película monocroma con la idea de emplear los negativos resultantes como diapositivas. La idea es invertir el orden de blanco a negro, de forma que la intensidad de la imagen de salida disminuya conforme la intensidad de la imagen de entrada aumente.
- ?? **Aumento del contraste:** Las imágenes con poco contraste pueden ser debidas a diversas causas, entre las que destacamos una iluminación deficiente. La idea subyacente en las técnicas de aumento del contraste consiste en incrementar el rango dinámico de los niveles de gris de la imagen que se está procesando. En esta opción es imprescindible rellenar el valor del **% Contraste**.
- ?? **Fraccionamiento del Nivel de Gris:** A menudo se desea destacar un rango específico del nivel de gris de una imagen. Entre estas aplicaciones se encuentran la mejora de rangos como las masas de aguas en las imágenes de satélite o la mejora de defectos en las imágenes de rayos X. Tenemos que rellenar el rango a resaltar, es decir los valores de **Rango a resaltar. Desde y Hasta**.
- ?? **Sustracción de imágenes con imagen de referencia:** La sustracción de imágenes tiene numerosas e importantes aplicaciones en la mejora de las imágenes. Una aplicación clásica se tiene en un área de imágenes médicas conocida como la radiografía en modo máscara. Debemos incluir la localización y el nombre de la **Imagen de referencia**.
- ?? **Promediado de la imagen:** La imagen de salida está formada por la media de todas las imágenes de entrada que han llegado al Agente hasta el momento. Esta técnica se usa básicamente para eliminar el ruido.

De todas las opciones de Procesamiento de Puntos, se habrá elegido a lo sumo una.

El empleo de máscaras espaciales para el procesamiento de las imágenes se denomina **Filtrado Espacial** y las propias máscaras se denominan filtros espaciales. Tenemos para el preprocesamiento por filtrado espacial:

?? **Filtros Suavizantes:**

Los filtros suavizantes se emplean para que la imagen aparezca algo borrosa y también para reducir el ruido. Es útil que la imagen aparezca algo borrosa en algunas etapas de preprocesado, como la eliminación de los pequeños detalles de una imagen antes de la extracción de un objeto, y el relleno de pequeños espacios entre líneas o curvas. Los filtros suavizantes actualmente presentes en Intelligent SILHFinder son: **Filtrado Espacial de Paso Bajo (1), Filtrado Espacial de Paso Bajo (2) y Filtrado por la Mediana.**

?? **Filtros Realzantes:**

El objetivo principal del realce es el de destacar los detalles finos de una imagen o intensificar detalles que han sido difuminados, bien sea por error o bien por efecto natural del método de adquisición de la imagen. Intelligent SILHFinder ofrece los siguientes filtros realzantes: **Filtrado Espacial de Paso Alto, Filtrado High-boost, Realce de Bordes (1), Realce de Bordes (2) y Realce de Bordes (3).**

De todas las opciones de Filtrado Espacial, se habrá elegido a lo sumo una.

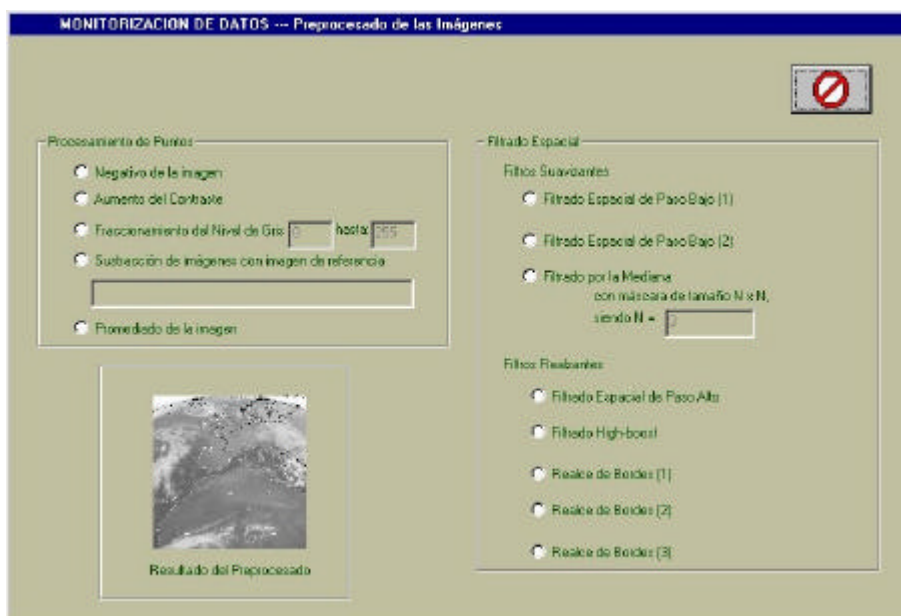


Figura 4.97. MONITORIZACION DE DATOS --- Preprocesado de las Imágenes

Opción Segmentación

La opción Segmentación permite ver qué acción (acciones) de segmentación se ha decidido realizar sobre las secuencias de imágenes de entrada previamente preprocesadas y que nutren al Agente. La segmentación subdivide una imagen en sus partes constituyentes u objetos.

Puede apreciarse en la figura 4.98. que la segmentación se ha dividido en dos grandes bloques: **Umbralización** y **Detección de Discontinuidades**.

La **Umbralización** es uno de los métodos más importantes de la segmentación de imágenes. Intelligent SILHFinder ofrece las siguientes técnicas de umbralización:

- ?? **Umbralización Global Simple**: Una imagen es segmentada en franjas equidistantes de bandas de nivel de gris. Un nivel de gris cae siempre en una única banda que lo incluye. Por ello, se requiere rellenar el campo **Número de Bandas de Nivel de Gris**.
- ?? **Umbralización Uniforme con Solapamiento**: En este caso, la idea es semejante a la anterior. La diferencia está en que para que un pixel cambie de banda de nivel de gris desde una imagen anterior hasta la actual, debe existir un cambio suficiente en el nivel de gris detectado. Por ello, se incorpora, además del campo **Número de Bandas de Nivel de Gris**, el parámetro **% Solapamiento**.

No se habrá elegirse más de una opción de las dos disponibles.

Ahora, bajo **Detección de Discontinuidades** se engloban varias técnicas para la detección de los tres tipos básicos de discontinuidades de una imagen digital: puntos, líneas y bordes. La forma más común de ver las discontinuidades es pasar una máscara (al igual que se hacía en el Filtrado Espacial). Intelligent SILHFinder se ha centrado en la detección de las siguientes discontinuidades: **Detección de Puntos (Puntos Aislados)**, **Detección de Líneas (Líneas Horizontales, Líneas orientadas a + 45°, Líneas Verticales, Líneas orientadas a - 45°)**, **Detección de Bordes (Operador de Roberts, Operador de Roberts Gx, Operador de Roberts Gy, Operador de Sobel, Operador de Sobel Gx, Operador de Sobel Gy, Operador de Prewitt, Operador de Prewitt Gx, Operador de Prewitt Gy y Operador Laplaciano)**.

Se habrá permitido elegir únicamente hasta una de las opciones anteriores de detección de discontinuidades.

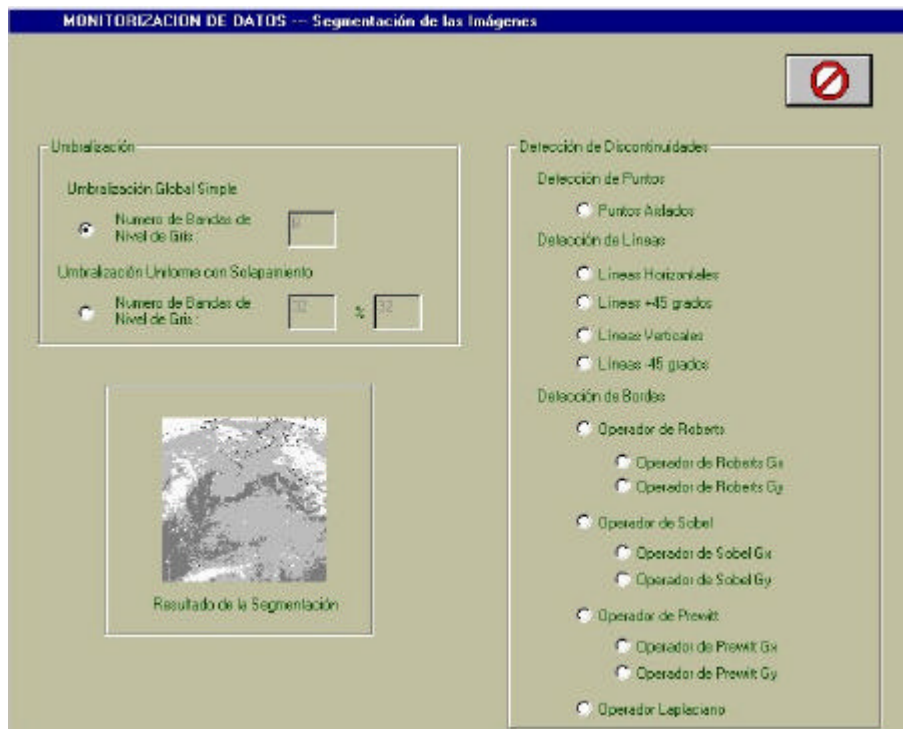


Figura 4.98. MONITORIZACION DE DATOS --- Segmentación de las Imágenes

Submenú Modelo

Gracias a este submenú podemos acceder a los parámetros de las diferentes capas de los modelos dados de alta en Intelligent SILHFinder. Como ya se ha comentado con anterioridad, solamente se dispone del modelo de Interacción Lateral en Computación Acumulativa, a la fecha de hoy.

Por ello, solamente se puede acceder a las opciones: **Interacción Lateral para la Computación Acumulativa**, **Interacción Lateral para el Reparto de Carga** e **Interacción Lateral para la Fusión de Objetos**.

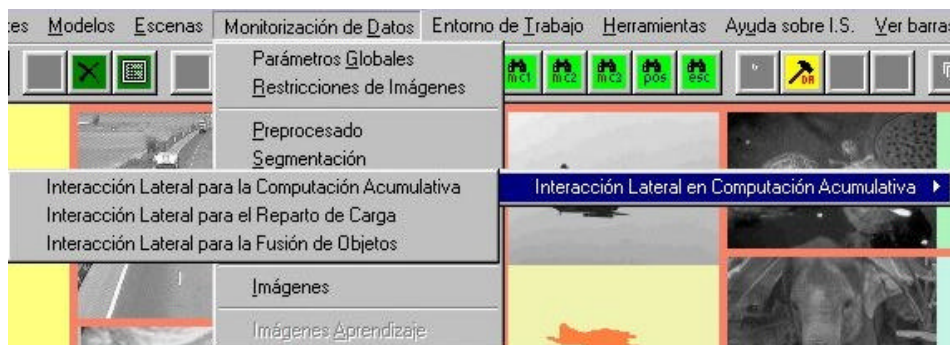


Figura 4.99. Menú Monitorización de datos, Submenú Modelo – Interacción Lateral en Computación Acumulativa

Opción Interacción Lateral para la Computación Acumulativa

La opción **Interacción Lateral para la Computación Acumulativa** del menú **Monitorización de datos** permite monitorizar aquellos parámetros requeridos por el hardware y/o software de la capa 1 del modelo de Interacción Lateral en Computación Acumulativa de procesamiento de imágenes de la aplicación Intelligent SILHFinder. Esta capa 1 se encarga de la interacción lateral para la computación acumulativa por bandas de nivel de gris, de ahí que los parámetros propios de esta capa sean los siguientes:

Figura 4.100. MONITORIZACION DE DATOS --- MODELOS – ILCA - Interacción Lateral para la Computación Acumulativa

- ?? **Número de Bandas de Nivel de Gris:** el número de bandas de gris en las que vienen segmentadas las imágenes (parámetro compartido con las capas 2 y 3; típicamente tomará el valor 8)
- ?? **Valor Mínimo de Permanencia:** el valor mínimo que puede tomar el valor de la memoria de permanencia, o sea, el valor de descarga (típicamente 0)
- ?? **Valor Máximo de Permanencia:** el valor máximo que puede tomar el valor de la memoria de permanencia, o sea, el valor de saturación (típicamente 255)
- ?? **Valor de Descarga por Detección de Movimiento:** el valor de descarga decrementado a la memoria de permanencia debido a la detección de movimiento

- ?? **Valor de Recarga por Vecindad:** el valor aplicado como suma debido a la interacción lateral en forma de recarga por vecindad
- ?? **Número de Vecinos para la Interacción Lateral:** el número de vecinos lineales a los que se puede llevar la interacción lateral de recarga
- ?? **Valor Mínimo de Mancha por Banda de Nivel de Gris:** el valor mínimo de la memoria de permanencia que permite efectuar un reparto de carga con los vecinos de la misma banda de nivel de gris

Opción Interacción Lateral para el Reparto de Carga

La opción **Interacción Lateral para el Reparto de Carga** del menú **Monitorización de datos** permite monitorizar aquellos parámetros requeridos por el hardware y/o software de la capa 2 de procesamiento de imágenes de la aplicación Intelligent SILHFinder. Esta capa 2 se encarga de la interacción lateral para la obtención de los elementos de siluetas por bandas de nivel de gris, de ahí que los parámetros propios de esta capa sean los siguientes:

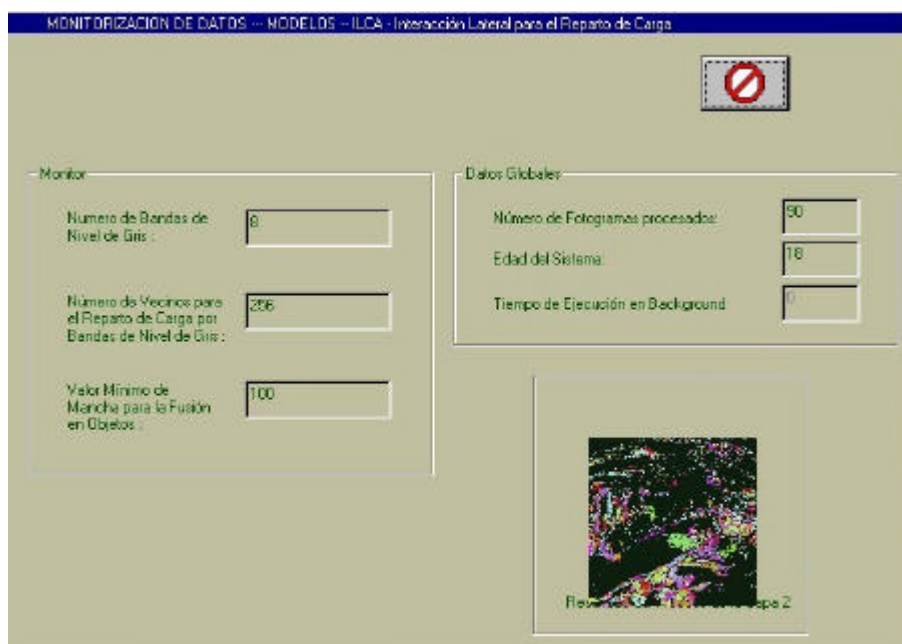


Figura 4.101. MONITORIZACION DE DATOS --- MODELOS – ILCA - Interacción Lateral para el Reparto de Carga

- ?? **Número de Bandas de Nivel de Gris:** el número de bandas de gris en las que vienen segmentadas las imágenes (parámetro compartido con las capas 1 y 3; típicamente toma el valor 8)
- ?? **Número de Vecinos para el Reparto de Carga por Bandas de Nivel de Gris:** el número de vecinos lineales a los que se puede llevar la interacción lateral de reparto de carga en la misma banda de nivel de gris
- ?? **Valor Mínimo de Mancha para la Fusión en Objetos:** el valor mínimo de la memoria de permanencia que permite efectuar un reparto de carga con los vecinos independientemente de su banda de nivel de gris (parámetro compartido con la capa 3)

Opción Interacción Lateral para la Fusión de Objetos

La opción **Interacción Lateral para la Fusión de Objetos** del menú **Monitorización de datos** permite monitorizar aquellos parámetros requeridos por el hardware y/o software de la capa 3 de procesamiento de imágenes de la aplicación Intelligent SILHFinder. Esta capa 3 se encarga de la interacción lateral para la fusión de siluetas, de ahí que los parámetros propios de esta capa sean los siguientes:



Figura 4.102. MONITORIZACION DE DATOS --- MODELOS – ILCA - Interacción Lateral para la Fusión de Objetos

- ?? **Número de Bandas de Nivel de Gris:** el número de bandas de gris en las que vienen segmentadas las imágenes (parámetro compartido con las capas 1 y 2; típicamente toma el valor 8)
- ?? **Número de Vecinos para el Reparto de Carga por Objeto:** el número de vecinos lineales a los que se puede llevar la interacción lateral de reparto de carga independientemente de su banda de nivel de gris
- ?? **Valor Umbral Mínimo para la Detección de Objetos:** el valor mínimo de la memoria de permanencia que permite indicar la existencia de un objeto hacia el exterior de la aplicación Intelligent SILHFinder

Opción Postprocesado

La opción Postprocesado permite ver qué acción (acciones) de postprocesado se ha decidido realizar sobre las secuencias de imágenes que ya han pasado por el modelo relacionado con el Agente. Puede apreciarse en la figura 4.103. que el postprocesado se divide en dos grandes bloques: **Filtrado Espacial** y **Procesamiento de Puntos**.

El empleo de máscaras espaciales para el procesamiento de las imágenes se denomina **Filtrado Espacial** y las propias máscaras se denominan filtros espaciales. Tenemos para el preprocesamiento por filtrado espacial: **Filtros Suavizantes (Filtrado Espacial de Paso Bajo (1), Filtrado Espacial de Paso Bajo (2), Filtrado por la Mediana)**, **Filtros Realzantes (Filtrado Espacial de Paso Alto, Filtrado High-boost, Realce de Bordes (1), Realce de Bordes (2) y Realce de Bordes (3))**.

De todas las opciones de Filtrado Espacial, se habrá elegido a lo sumo una.

Bajo **Procesamiento de Puntos** se engloban aquellas técnicas de mejora de la imagen basadas únicamente en la intensidad de los píxeles individuales. Estos procesos punto a punto se encuentran entre las técnicas más simples de la mejora de la imagen.

A la fecha se han incorporado las siguientes opciones de postprocesamiento por procesamiento de puntos:

- ?? **Negativo de la imagen**
- ?? **Aumento del contraste:** En esta opción es imprescindible rellenar el valor del % **Contraste**.

?? **Fraccionamiento del Nivel de Gris:** Tenemos que rellenar el rango a resaltar, es decir los valores de **Rango a resaltar. Desde y Hasta.**

?? **Sustracción de imágenes con imagen de referencia:** Debemos incluir la localización y el nombre de la **Imagen de referencia.**

?? **Promediado de la imagen**

De todas las opciones de Procesamiento de Puntos, se habrá elegido a lo sumo una.

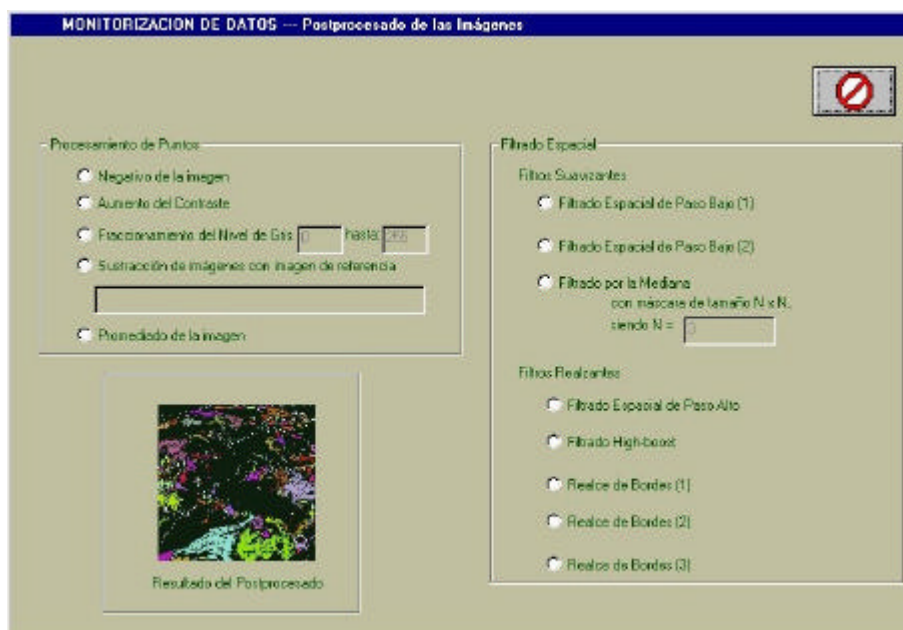


Figura 4.103. MONITORIZACION DE DATOS - Postprocesado de las Imágenes

Submenú Escena

El submenú Escena permite acceder a la monitorización de la última capa de procesamiento de las imágenes en función del tipo de aplicación elegida para el Agente.

Así, tendremos dos grandes grupos de Escenas a las que acceder:

1. Segmentación a bajo nivel, donde se podrá elegir entre:
 - a) Debida al movimiento. (a1) Siluetas. (a2) Contornos.
 - b) Debida a la luminosidad. (b1) Cambios de luz
2. Aplicaciones de alto nivel, entre las que figuran:
 - a) Imágenes médicas. (a1) Angiografías.

- b) Escenas de tráfico. (b1) Autopistas. (b2) Cruces de calles.
- c) Escenas de vigilancia. (c1) Personas.
- d) Imágenes satélite. (d1) Meteosat.

Opción Siluetas

Este diálogo muestra la salida de la opción Siluetas, escena de obtención de las siluetas de todos los objetos móviles en la escena, en el contexto de la segmentación a bajo nivel.

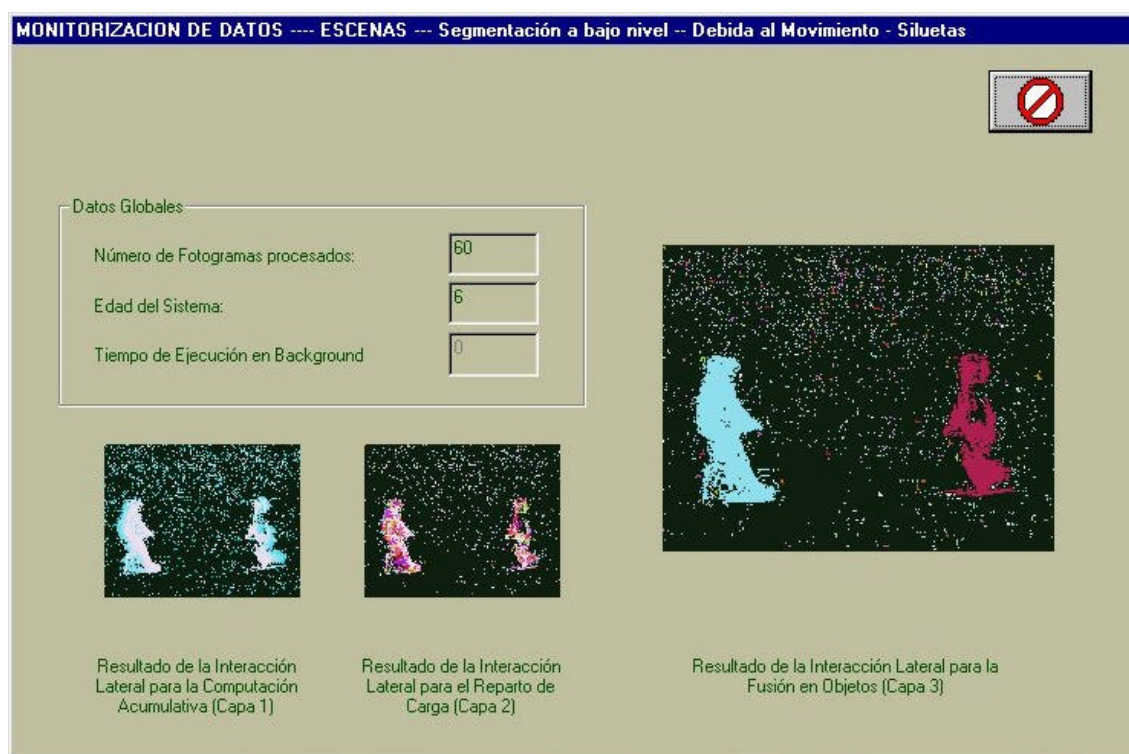


Figura 4.104. MONITORIZACION DE DATOS ---- ESCENAS --- Segmentación a bajo nivel – Debida al Movimiento - Siluetas

Opción Contornos

Este diálogo muestra la salida de la opción Contornos, escena de obtención de los contornos de las siluetas de todos los objetos móviles en la escena, en el contexto de la segmentación a bajo nivel.

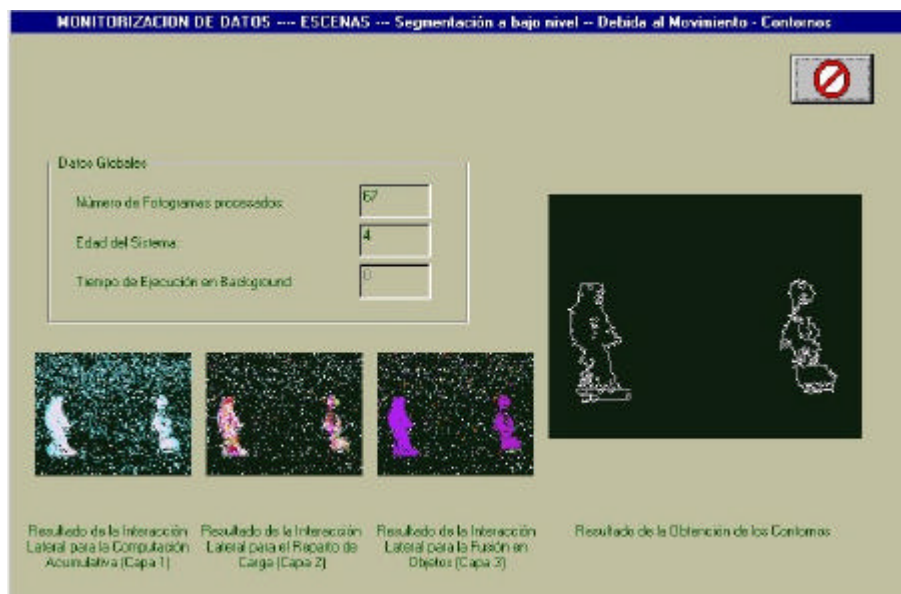


Figura 4.105. MONITORIZACION DE DATOS ---- ESCENAS --- Segmentación a bajo nivel – Debida al Movimiento - Contornos

Opción Cambios de Luz

Este diálogo muestra la salida de la opción Cambios de Luz, escena de obtención de las siluetas de todos los objetos móviles en la escena a partir del cambio en la luminosidad, en el contexto de la segmentación a bajo nivel.



Figura 4.106. MONITORIZACION DE DATOS ---- ESCENAS --- Segmentación a bajo nivel – Debida a la Luminosidad – Cambios de Luz

Opción Angiografías

Este diálogo muestra la salida de la opción Angiografías, escena de obtención del resultado de aplicar el modelo a la técnica de angiografía, en el contexto de las Aplicaciones de alto nivel, Imágenes Médicas.



Figura 4.107. MONITORIZACION DE DATOS ---- ESCENAS --- Aplicaciones de alto nivel – Imágenes Médicas - Angiografías

Opción Autopistas

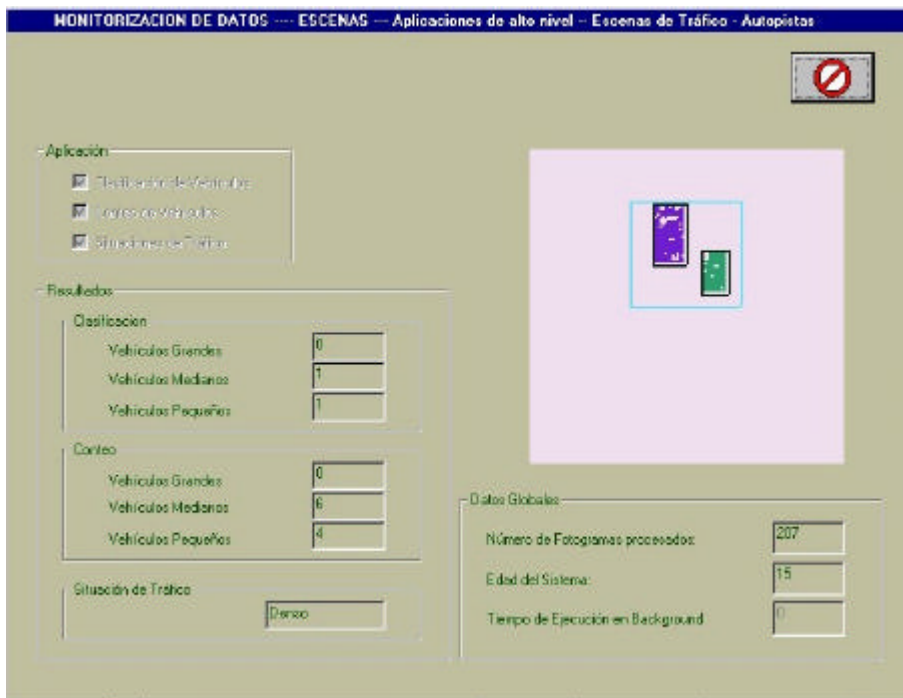


Figura 4.108. MONITORIZACION DE DATOS ---- ESCENAS --- Aplicaciones de alto nivel – Escenas de Tráfico - Autopistas

Este diálogo muestra la salida de la opción Autopistas, escena de obtención de los vehículos presentes en la escena de tráfico en autopistas, en el contexto de las Aplicaciones de alto nivel, Escenas de Tráfico.

Opción Cruces de Calles

Este diálogo muestra la salida de la opción Cruces de Calles, escena de obtención de los vehículos presentes en la escena de tráfico en cruces de calles, en el contexto de las Aplicaciones de alto nivel, Escenas de Tráfico.

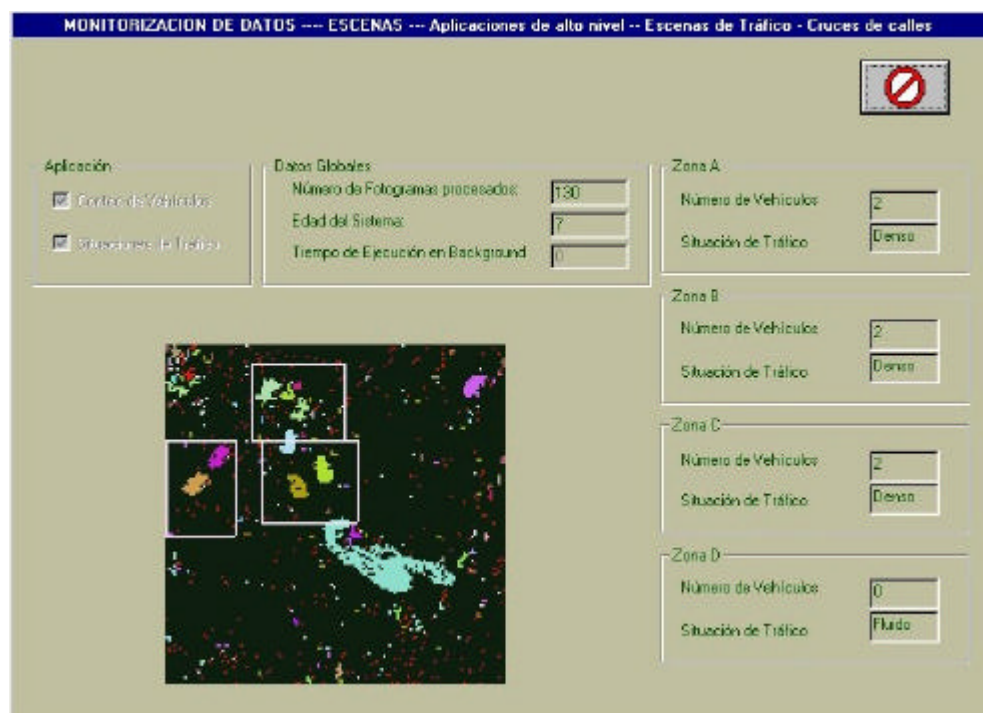


Figura 4.109. MONITORIZACION DE DATOS ---- ESCENAS --- Aplicaciones de alto nivel – Escenas de Tráfico – Cruces de Calles

Opción Personas

Este diálogo muestra la salida de la opción Personas, escena de obtención de las personas presentes en una escena de vigilancia, en el contexto de las Aplicaciones de alto nivel, Escenas de Vigilancia.



Figura 4.110. MONITORIZACION DE DATOS ---- ESCENAS --- Aplicaciones de alto nivel – Escenas de Vigilancia - Personas

Opción Meteosat

Este diálogo muestra la salida de la opción Meteosat, escena de obtención de las nubes presentes en la secuencia de imágenes, en el contexto de las Aplicaciones de alto nivel, Imágenes Satélite.

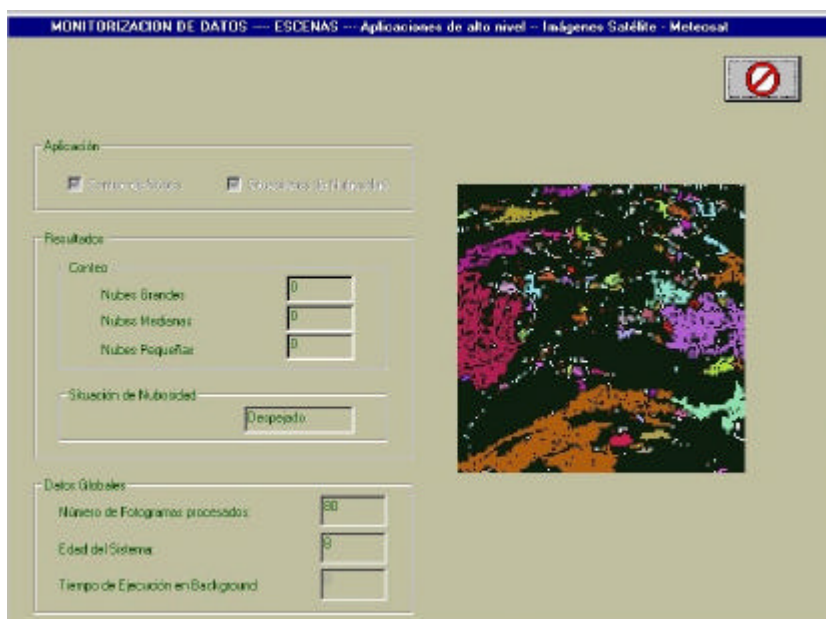


Figura 4.111. MONITORIZACION DE DATOS ---- ESCENAS --- Aplicaciones de alto nivel – Imágenes Satélite - Meteosat

Opción Imágenes

Esta opción permite monitorizar en un solo diálogo aquellos tipos de imágenes que se eligieron en la caja de diálogo de la opción **Tratamiento de Imágenes** del menú **Entorno de trabajo** en su parámetro **Visualizar Imágenes en Pantalla**.



Figura 4.112. MONITORIZACION DE DATOS --- Imágenes

En la figura 4.112. vemos el máximo de imágenes monitorizables en esta opción, a saber:

1. Si elegimos Imágenes de Entrada:
 - 1.1. La imagen actual.
 - 1.2. La imagen anterior.
2. Si elegimos Imágenes de Salida de Preprocesado, la imagen de salida del proceso de preprocesado.
3. Si elegimos Imágenes de Salida de Segmentación, la imagen de salida del proceso de segmentación.

4. Si elegimos Imágenes de Modelo, las imágenes de salida de las capas 1, 2 y 3, pero coloreadas para remarcar más los valores de carga.
5. Si elegimos Imágenes de Salida del Postprocesado, la imagen de salida del proceso de postprocesado, pero coloreada para remarcar más los valores de carga.
6. Si elegimos Imágenes de Escena, la imagen de salida del proceso de la Escena:

Todas estas imágenes de salida pueden monitorizarse una a una en las opciones anteriormente mencionadas de este mismo menú de **Monitorización de datos**.

4.3.3.6. Menú Entorno de trabajo

Al entrar en el menú **Entorno de trabajo** vamos a tener a nuestro alcance cuatro opciones diferentes de trabajo. Este menú está destinado a la configuración y control del entorno de trabajo para un Agente determinado. Por ello, cuando no hay ningún Agente seleccionado aún, solamente una de las opciones está disponible. En efecto, esta opción de **Directorios de Aplicaciones** es independiente del Agente. En este caso concreto, tenemos la visión del menú desplegable según la figura 4.113.

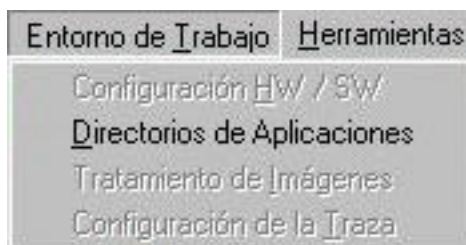


Figura 4.113. Menú Entorno de trabajo. Ningún Agente seleccionado

Las cuatro opciones disponibles con un Agente abierto son:

- ?? **Configuración HW / SW**
- ?? **Directorios de Aplicaciones**
- ?? **Tratamiento de Imágenes**
- ?? **Configuración de la Traza**

como se puede ver en la figura 4.114.

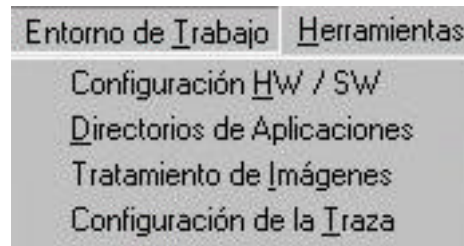





Figura 4.114. Menú Entorno de trabajo

Para salir de cualquier caja de diálogo de este menú, tenemos dos opciones:

- ??  **Aceptar:** pulsando este botón se actualizan las modificaciones introducidas y volvemos al marco principal de la aplicación
- ??  **Cancelar:** pulsando este botón se deshacen las modificaciones introducidas y volvemos al marco principal de la aplicación

Opción Configuración HW / SW

En esta caja de diálogo se establece la configuración del hardware y/o del software disponibles y se permite realizar un  **TEST** para verificar las condiciones establecidas. También podemos comprobar si la interrupción de cuadro ha sido reconocida, es decir, si funciona correctamente la señal de vídeo.

Intelligent SILHFinder está preparado para admitir tres placas hardware:

- ?? 1 placa digitalizadora
- ?? 1 placa extractora de características
- ?? 1 placa clasificadora

Caso de no estar presente alguna de ellas, el proceso específico que realizaría la placa pasa a realizarse bajo software.

En la parte de la izquierda de la caja de diálogo, puede apreciarse la información de la presencia o no de cada una de las placas hardware que el sistema admite según lo que consta en la Base de Datos SkeleTton para el Agente actual.



Figura 4.115. ENTORNO DE TRABAJO --- Configuración HW / SW

Si se desea cambiar el modo de trabajo (hardware o software), se debe seleccionar para cada una de las placas si está instalada o no. No obstante, si se desea detectar las placas realmente, se debe pulsar el botón de **TEST**, obteniéndose una información fidedigna al respecto.

Opción Directorios de Aplicaciones

Esta opción permite modificar una serie de directorios por defecto utilizados por Intelligent SILHFinder. En estos directorios se supone que están localizados una serie de ficheros. Si alguna instalación específica requiere un cambio de directorio, se puede cambiar mediante esta pantalla. Estos directorios se refieren a:

- ?? **Interfaz de Vídeo:** localización de la aplicación Interfaz de Vídeo llamada desde el menú **Herramientas**
- ?? **Configuración de Vídeo:** localización del fichero que contiene la información de la parametrización del magnetoscopio conectado a la aplicación Intelligent SILHFinder
- ?? **Aplicación de Configuración de Vídeo:** localización de la aplicación de parametrización del magnetoscopio conectado a la aplicación Intelligent SILHFinder

?? **Base de Datos SkeleTTon:** localización de la Base de Datos SkeleTTon, llamada desde el menú **Herramientas** para trabajar en OFF-LINE, y desde múltiples puntos de la propia aplicación Intelligent SILHFinder

NOTA: Es fundamental comprobar esta localización antes de abrir un Agente, ya que Intelligent SILHFinder accede a esta base de datos.

?? **Aplicación MS Access:** localización de la aplicación Microsoft Access; imprescindible si se desea acceder a la Base de Datos SkeleTTon, ya que está escrita bajo Access

?? **Parámetros de la Aplicación MS Access:** parámetro de llamada a MS Access para acceder a la Base de Datos SkeleTTon

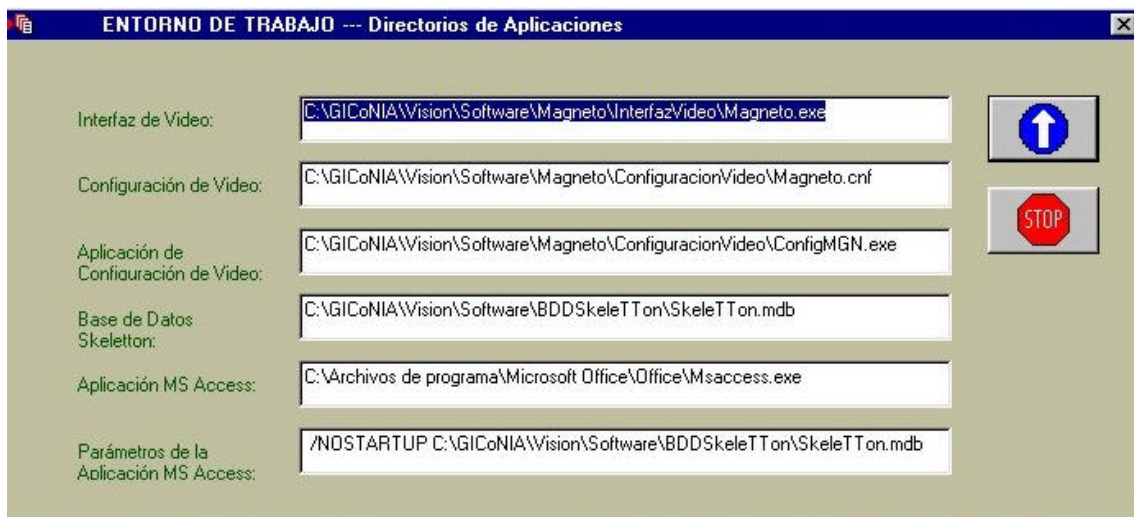


Figura 4.116. ENTORNO DE TRABAJO --- Directorios de Aplicaciones

Opción Tratamiento de Imágenes

En esta caja de diálogo se permite:

- a) controlar el almacenamiento de las imágenes tratadas
- b) controlar la visualización de las imágenes tratadas en pantalla

Todo ello, como puede verse en la figura 4.117., diferenciando entre imágenes de entrada e imágenes de salida de cualquiera de las capas de proceso.

La información de interés, en el caso del control de almacenamiento de las imágenes, es:

?? **Guardar Imágenes en Directorio:** directorio donde se van a salvar las imágenes deseadas; el directorio por defecto está relacionado con el nombre de usuario seleccionado en el control de acceso, comenzando siempre con C:\GICoNIA\Vision\Salidas\

?? **Guardar Imágenes como Grupo de Ficheros:** prefijo del nombre asociado a todas las imágenes salvadas en la sesión de trabajo; por defecto toma el nombre del Agente.

?? **Guardar Imágenes De:** permite seleccionar qué tipo de imágenes se salvan; las opciones son Imágenes de Entrada, Imágenes de Salida de Preprocesado, Imágenes de Salida de Segmentación, Imágenes de Salida de Modelo, Imágenes de Salida de Postprocesado e Imágenes de Salida de Escena.

A la derecha de la selección vemos con qué nombre se van a grabar las imágenes, donde ??? debe entenderse como número de fotograma en una serie de fotogramas.

?? **Guardar Imágenes Cada número de Fotogramas:** permite indicar la frecuencia de grabación de las imágenes seleccionadas; el número 0 indica que no se desea salvar ninguna de las imágenes.

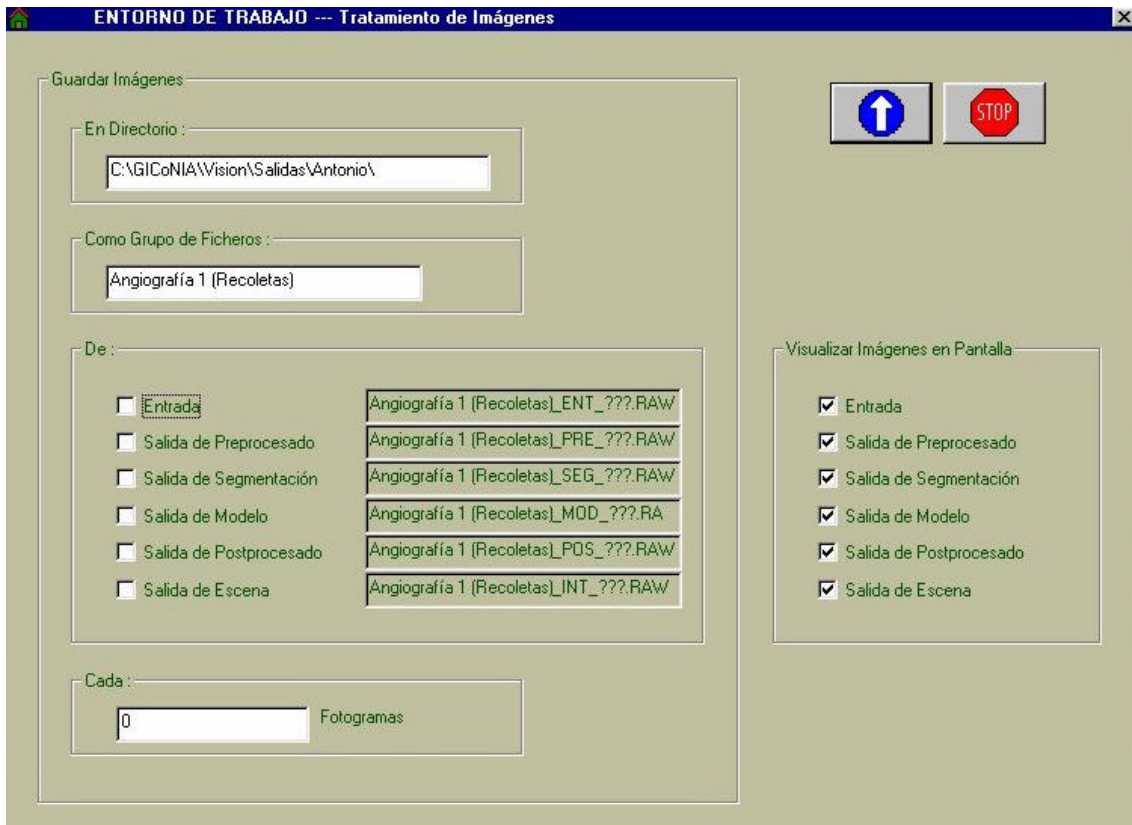


Figura 4.117. ENTORNO DE TRABAJO --- Tratamiento de Imágenes

Para el caso del control de visualización de las imágenes en pantalla, solamente se puede elegir qué tipo de imágenes se mostrarán en la opción **Monitorización de Imágenes** del menú **Monitorización de datos**. Las opciones son Imágenes de Entrada, Imágenes de Salida de Preprocesado, Imágenes de Salida de Segmentación, Imágenes de Salida de Modelo, Imágenes de Salida de Postprocesado e Imágenes de Salida de Escena.

A resaltar de modo muy especial que estas configuraciones no están asociadas a los Agentes en la Base de Datos SkeleTTon, sino que toman los valores por defecto que se contemplan en la figura 4.117. cada vez que se entra en la aplicación Intelligent SILHFinder. De ahí que sea fundamental el entrar en esta caja de diálogo antes de **Arrancar Agente** si estos parámetros no son los deseados. Véase pues que los valores por defecto son:

?? **Guardar Imágenes en Directorio**

C:\GICoNIA\Vision\Salidas\'nombre_de_usuario\'

?? **Guardar Imágenes como Grupo de Ficheros:**

“nombre_de_Agente”

?? **Guardar Imágenes De:**

Entrada	no
Salida de Preprocesado	no
Salida de Segmentación	no
Salida de Modelo	no
Salida de Postprocesado	no
Salida de Escena	no

?? **Guardar Imágenes Cada número de Fotogramas:**

cada 0 fotogramas (no salvar)

?? **Visualizar Imágenes en Pantalla De:**

Entrada	si
Salida de Preprocesado	si
Salida de Segmentación	si
Salida de Modelo	si

Salida de Postprocesado	si
Salida de Escena	si

Opción Configuración de la Traza

Esta caja de diálogo está destinada al control de la traza que realiza el sistema y se muestran los siguientes datos:

?? **Detener ejecución:** introducimos un valor que indica que la ejecución debe detenerse cuando el número de fotogramas procesados sea múltiplo de este dato. El valor 0 indica que no se ha de detener la ejecución.

?? **Salvar estado:** introducimos un valor que indica que el estado del Agente debe salvarse automáticamente en la Base de Datos SkeleTTon cuando el número de fotogramas procesados sea múltiplo de este dato. El valor 0 indica que no se salve el estado de forma automática.

?? **Ejecutar:** tenemos dos opciones:

?? **Paso a paso:** la ejecución se realiza en modo diferido, es decir, entre el proceso de dos fotogramas consecutivos, se permite la monitorización de los datos solicitados.

?? **En continuo:** se trabaja en tiempo real. No se permite monitorizar entre fotogramas.

?? **Monitorizar estado:** al monitorizar el estado debemos tener en cuenta si la ejecución se realiza:

?? **Paso a paso:** todos los parámetros del sistema son accesibles.

?? **En continuo:** no se pueden monitorizar los parámetros del sistema en tiempo real.

Tenemos dos opciones:

?? **Lo definido.**

?? **Nada:** aún pudiendo monitorizar, no lo deseamos en estos momentos.

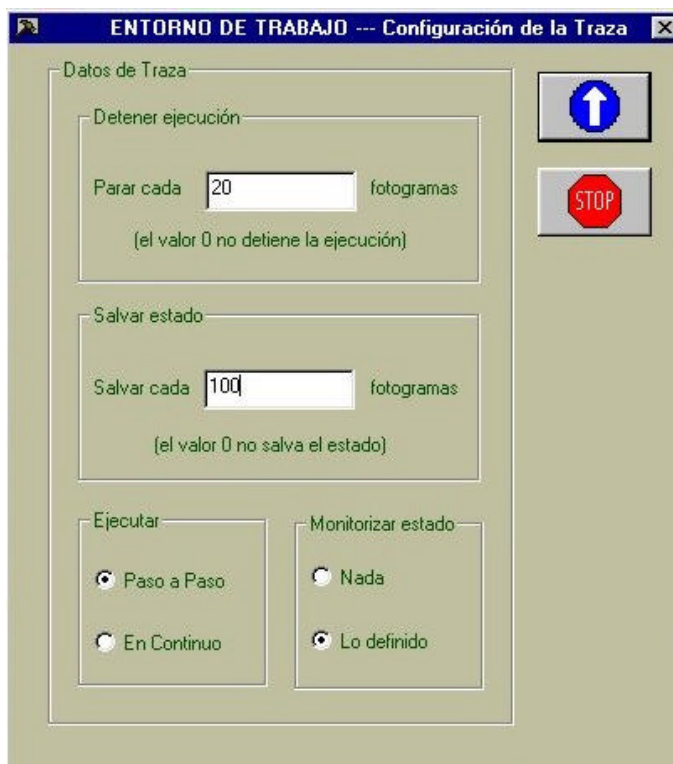


Figura 4.118. ENTORNO DE TRABAJO --- Configuración de la Traza

4.3.3.7. Menú Herramientas

Cuando seleccionemos este menú, observamos la pantalla de la figura 4.119.

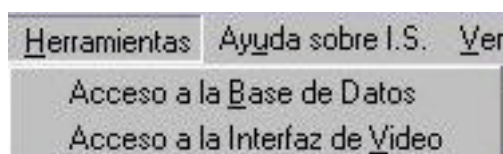


Figura 4.119. Menú Herramientas

Desde este menú podemos lanzar aplicaciones auxiliares directamente relacionadas con Intelligent SILHFinder. En estos momentos podemos ir a: **Acceso a la Base de Datos** o **Acceso a la Interfaz de Vídeo**.

Opción Acceso a la Base de Datos

Esta opción permite acceder a la **Base de Datos SkeleTton** y realizar allí cualquier operación. Aquí presentamos la carátula de la Base de Datos.

Si desea saber más acerca del funcionamiento y manejo de la **Base de Datos SkeleTTon**, le remitimos al capítulo 4.2. Vista del usuario: Base de Datos SkeleTTon.

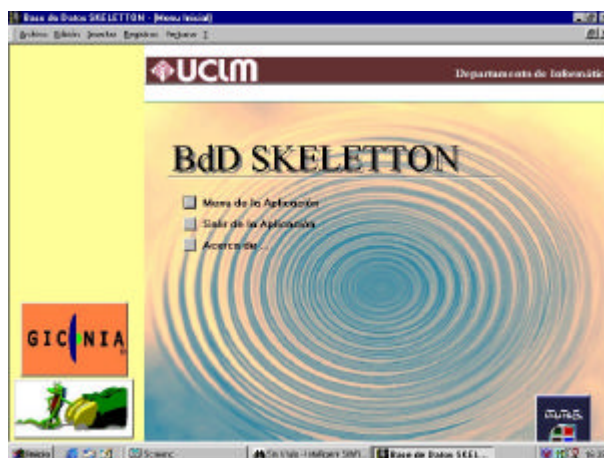


Figura 4.120. Acceso a la Base de Datos SkeleTTon

Opción Acceso a la Interfaz de Vídeo

Esta opción permite acceder a la **Interfaz de Vídeo** y realizar desde esa aplicación cualquier control OFF-LINE sobre el magnetoscopio conectado al hardware del sistema Intelligent SILHFinder.

Al entrar en la aplicación de **Interfaz de Vídeo** aparece, en primer lugar, la siguiente ventana:

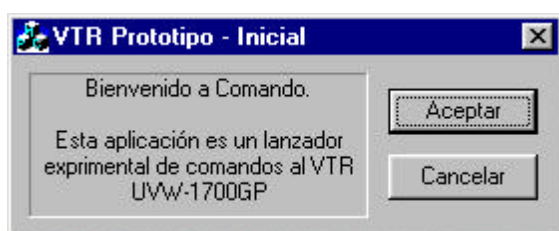


Figura 4.121. VTR Prototipo - Inicial

En este momento puede entrar en la aplicación auxiliar pulsando **Aceptar** o volver a Intelligent SILHFinder pulsando **Cancelar**. A continuación aparece la pantalla principal de la figura 4.122.

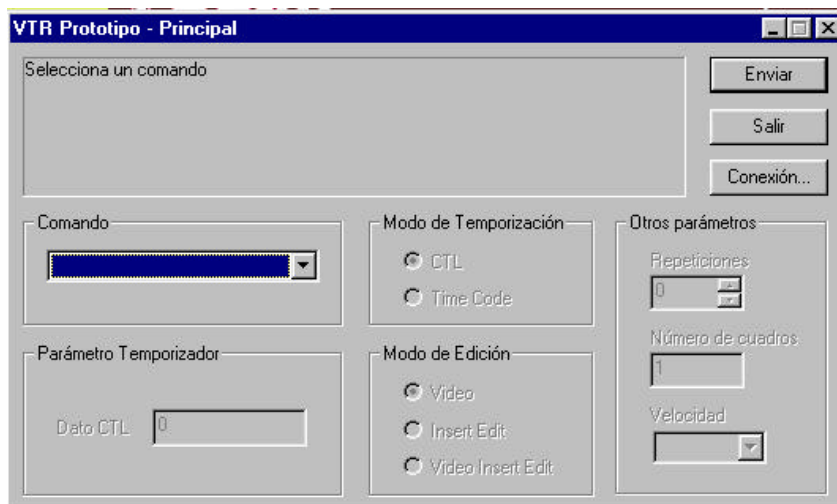


Figura 4.122. VTR Prototipo - Principal

En estos momentos iniciales tendremos que conectar el magnetoscopio. Ello se realiza a través del botón **Conexión**, obteniéndose la ventana de la figura 4.123., donde podemos consultar los siguientes datos de conexión: Puerto, Baudios, Paridad, Bits de datos, Bits de control y Estado del puerto.

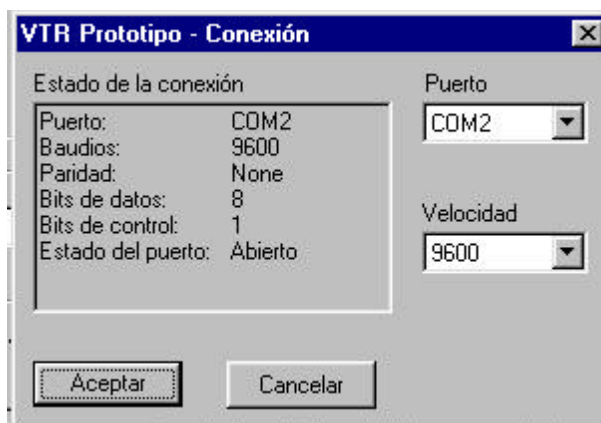


Figura 4.123. VTR Prototipo - Conexión

En el diálogo que aparece puede seleccionarse el puerto al que está conectado el magnetoscopio, así como la velocidad de intercambio de información.

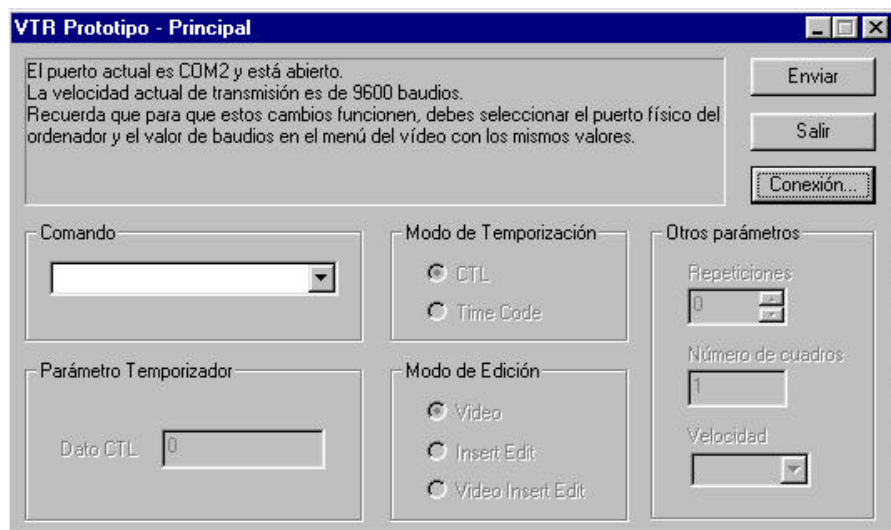


Figura 4.124. VTR Prototipo – Principal. Puerto abierto

A partir de este momento podemos seleccionar un comando y enviarlo mediante el botón **Enviar**. En la ventana de la figura 4.125., puede verse un subconjunto de todos los comandos disponibles para el manejo del magnetoscopio dispensador de imágenes.

Si desea saber más acerca del funcionamiento y manejo de la **Interfaz de Vídeo**, le remitimos al Manual de Usuario de dicha aplicación.

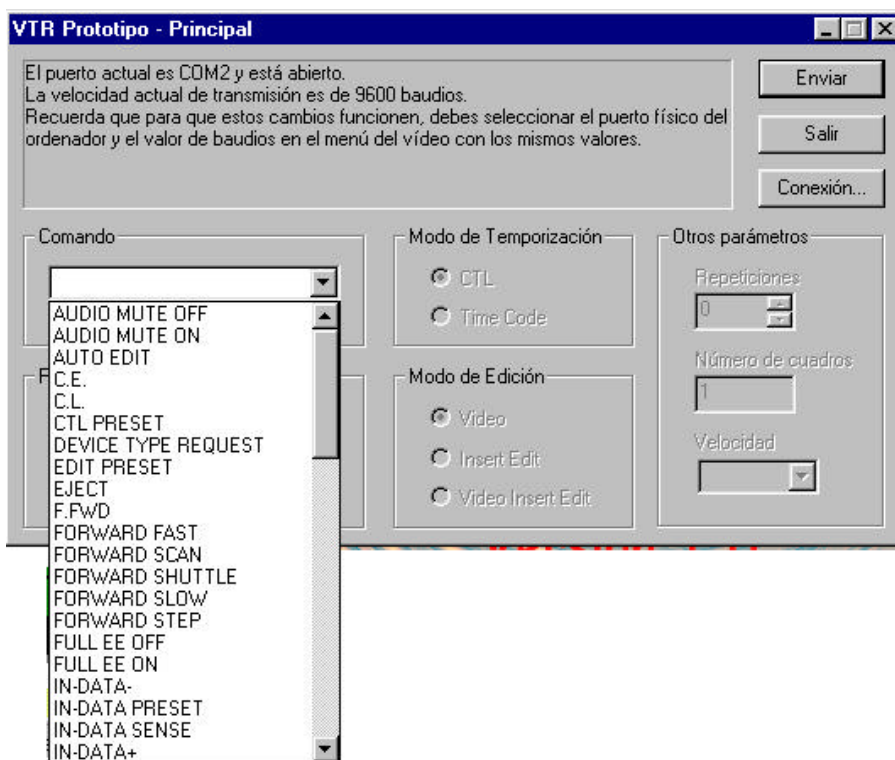


Figura 4.125. VTR Prototipo - Comandos

4.3.3.8. Menú Ayuda sobre Intelligent SILHFinder

Cuando seleccionamos este último menú, observamos la pantalla reflejada en la figura 4.126.

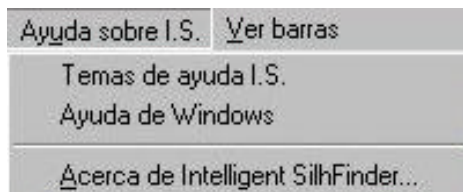


Figura 4.126. Menú Ayuda.

Observamos en la pantalla tres posibles opciones:

?? **Temas de Ayuda Intelligent SilhFinder**

?? **Ayuda de Windows**

?? **Acerca de Intelligent SILHFinder**

que se pasan a detallar a continuación.

Opción Temas de Ayuda Intelligent SilhFinder

Esta opción nos lleva a la ayuda del sistema Intelligent SILHFinder. Se trata de una ayuda on-line que da todo detalla de información acerca de la aplicación Intelligent SilhFinder en formato HTML.

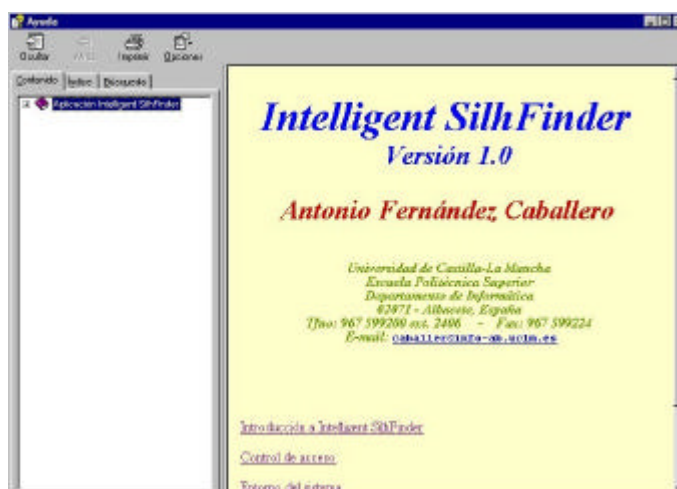


Figura 4.127. AYUDA SOBRE I.S. --- Temas de Ayuda Intelligent SilhFinder

Opción Ayuda de Windows

Esta opción nos lleva a la típica ayuda de Windows incorporada a toda aplicación escrita bajo Visual C++.

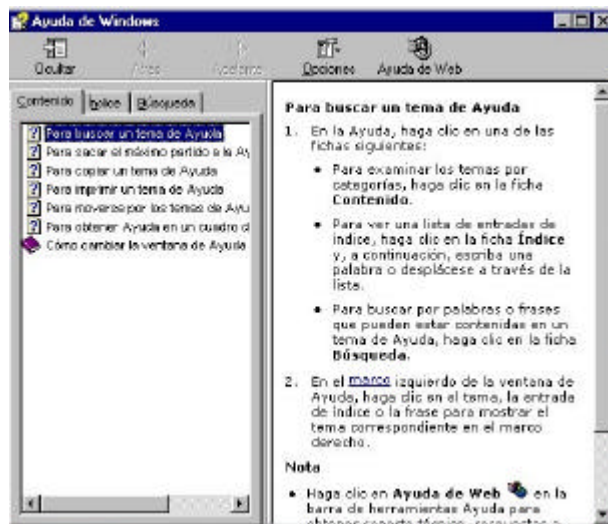


Figura 4.128. AYUDA SOBRE I.S. --- Ayuda de Windows

Opción Acerca de Intelligent SILHFinder

La opción Acerca de Intelligent SILHFinder ofrece la información suficiente acerca de la versión que se está manejando de la aplicación, así como del grupo de desarrollo de la misma.



Figura 4.129. AYUDA SOBRE I.S. --- Acerca de Intelligent SILHFinder

4.3.3.9. Menú Ver barras

Al desplegar el menú **Ver barras** observaremos la pantalla de la figura 4.130.

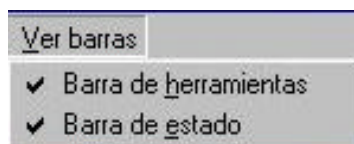


Figura 4.130. Menú Ver barras

La barra de herramientas se encuentra debajo de la barra de menús, en la parte superior de la pantalla. Esta barra puede mostrarse si está seleccionada en la opción **Barra de herramientas** del menú **Ver barras**, o puede ocultarse si no está seleccionada, como se muestra en la figura 4.131.

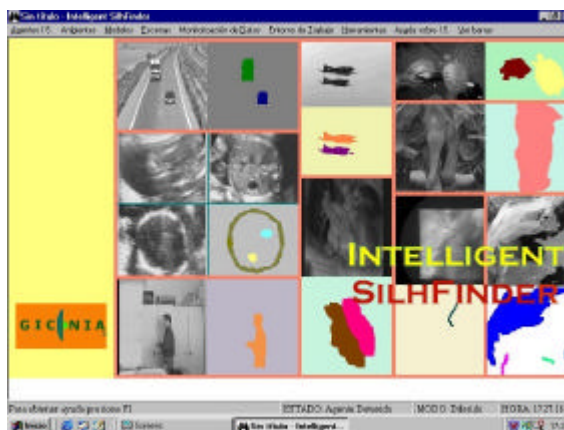


Figura 4.131. Marco principal de Intelligent SILHFinder sin barra de herramientas

La barra de estado se encuentra en la parte inferior de la pantalla debajo de la ventana de trabajo. Para mostrar la barra de estado, seleccione el menú **Ver barras** y active la casilla **Barra de estado**. Esta barra de estado puede quedar oculta si no se activa (ver figura 4.132.).

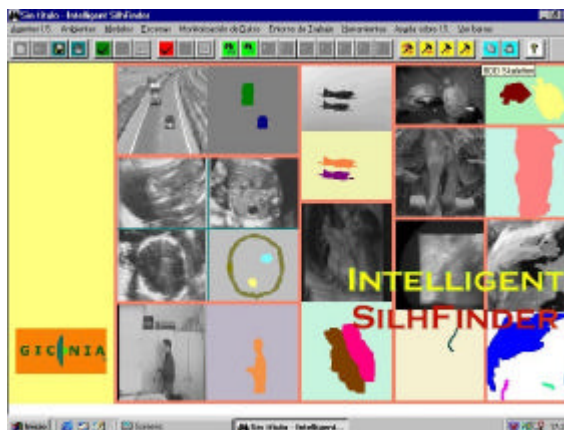


Figura 4.132. Marco principal de Intelligent SILHFinder sin barra de estado

CAPITULO 5. Datos y resultados

5.1. Metodología empleada

Se presenta en este capítulo la metodología seguida para probar el modelo propuesto en esta tesis doctoral, los datos usados, así como los resultados más relevantes obtenidos en dichas pruebas. Para la validación del método se ha decidido trabajar en dos etapas claramente diferenciadas.

En primer lugar, se ha trabajado con el objetivo de probar el modelo de Interacción Lateral en Computación Acumulativa bajo la óptica exclusiva de obtener las siluetas de todos los elementos móviles presentes en una secuencia indefinida de imágenes. Los datos y resultados de este primer grupo de pruebas se muestran en el párrafo 5.2.1.1. De algún modo, podría decirse que esta primera fase tiene como objetivo claro la segmentación a bajo nivel de las imágenes de entrada al modelo.

Durante esta fase, no obstante, han surgido dos ideas complementarias que también se han evaluado con un resultado más que satisfactorio. Como complemento sencillo a la obtención de las siluetas, se ha trabajado en la obtención de los contornos de las siluetas halladas. Para ello, véase el apartado 5.2.1.2.

También en esta primera fase, se ha constatado que, cambiando la luminosidad en una misma imagen en vez de tomar una secuencia de imágenes, se obtienen resultados similares por parte del modelo de Interacción Lateral en Computación Acumulativa. De ahí que se ofrezcan los datos y resultados del apartado 5.2.1.3.

Ahora bien, una vez vista esta primera fase, ha sido especialmente satisfactorio el trabajo desarrollado en la utilización del modelo en aplicaciones de alto nivel reales. Se presenta en el apartado 5.3. toda una serie de pruebas en entornos reales tan dispares como las imágenes médicas, las escenas de tráfico, las escenas de vigilancia o las imágenes satélite. La bondad de los resultados obtenidos permite vaticinar un futuro prometedor tanto al modelo en sí, como a la herramienta software que lo sustenta.

Evidentemente, los casos expuestos en este capítulo son iguales a las escenas descritas en los capítulos anteriores. Recordemos que están actualmente en vigencia:

- **Aplicaciones Generales:** Aquí irán aquellos Agentes que no están asociados a un tipo de aplicación claro y específico. Por ello, se ha optado por realizar la siguiente diferenciación:
 - **Segmentación:** La tarea que se acomete a este nivel es la obtención (sin mayores especificaciones de alto nivel) de: (a) **Siluetas:** Este tipo de Escena obtiene las siluetas de todos los objetos móviles presentes en una secuencia indefinida de imágenes. (b) **Contornos:** Este tipo de Escena aporta un paso más respecto de la obtención de las siluetas, a saber, la obtención de los contornos de todas las siluetas obtenidas.
 - **Luminosidad:** Se ha llegado a la conclusión que ciertos modelos, entre los que se encuentra el Modelo de Interacción Lateral en Computación Acumulativa, permiten obtener las siluetas de los objetos presentes en una sola imagen a partir del cambio forzado de la iluminación de la imagen. Ello se engloba en la opción: **Cambios de Luz.**
- **Aplicaciones Específicas:** En este apartado se incluyen algunas aplicaciones de alto nivel, en las que nos hemos adentrado en el dominio de conocimiento, y hemos inyectado este conocimiento al Agente. Se ofrecen en esta versión de Intelligent SILHFinder algunas aplicaciones de interés, como puede ser el manejo de:
 - **Imágenes Médicas. Angiografías:** Esta es una aproximación volumétrica a las imágenes vasculares, que permite ver las estructuras vasculares, buscando posibles lesiones.
 - **Imágenes Satélite. Meteosat:** La aplicación elegida en este caso es el seguimiento de la nubosidad a partir de imágenes tomadas por el satélite Meteosat.
 - **Escenas de Tráfico.** (a) **Autopistas:** Aplicación de clasificación y conteo de vehículos en una autopista. (b) **Cruces de Calles:** La misma

aplicación en cruces de calles, pero más enfocada a conocer la dirección que toman los vehículos.

- **Escenas de Vigilancia. Personas:** Esta aplicación clasifica las personas en base a su velocidad de paso.

Cabe decir, antes de pasar a los datos y resultados obtenidos, que las imágenes usadas en esta sección han sido tomadas de algunos benchmarks ofrecidos por ciertas comunidades científicas, así como otras tomadas por una cámara de vídeo convencional.

5.2. Segmentación a bajo nivel

5.2.1. Segmentación a partir del movimiento

En las pruebas de segmentación a bajo nivel a partir del movimiento se han usado tanto objetos rígidos como objetos deformables en movimiento. Los objetos rígidos pueden considerarse como una simplificación de los objetos deformables. El siguiente ejemplo¹ formado por dos elementos (una tetera y un juguete) podría ser segmentado a partir del movimiento de los mismos:



Figura 5.1. Objetos rígidos

Cualquier objeto deformable² puede segmentarse igualmente a partir de su movimiento.



Figura 5.2. Objetos deformables

¹ Data for this research were partially provided by the Calibrated Imaging Laboratory at Carnegie Mellon University, supported by ARPA, NSF, and NASA.

² Por cortesía de Institut National de Recherche en Informatique et Automatique, iMAGIS Project.

5.2.1.1. Segmentación a bajo nivel – Debido al movimiento - Siluetas

Se ofrecen a continuación los resultados de la segmentación de objetos deformables. Son cuatro los ejemplos mostrados.

Comenzamos con la obtención de la silueta de un hombre cruzando una habitación. Véase que se ofrece la salida de la misma secuencia de imágenes para los dos posibles casos de umbralización. El ejemplo primero muestra el resultado de aplicar una umbralización global simple en 8 bandas de nivel de gris, mientras que el segundo ejemplo muestra una sensible mejora al usar la umbralización uniforme con solapamiento. El solapamiento ha sido en este caso del 20% para una umbralización en 8 bandas de nivel de gris, igualmente.

Los valores de los diferentes parámetros en cada una de las capas del modelo de Interacción Lateral en Computación Acumulativa han sido en ambos casos los de la tabla 5.1.

Segmentación por Bandas de Nivel de Gris (Capa 0)	
<i>Número de Bandas de Nivel de Gris (n)</i>	8
Interacción Lateral para la Computación Acumulativa (Capa 1)	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	200
<i>Valor de Recarga por Vecindad (v_{rv})</i>	150
<i>Número de Vecinos para la Interacción Lateral (l_{per})</i>	64
<i>Valor Mínimo de Mancha por Banda de Nivel de Gris (θ_{per})</i>	100
Interacción Lateral para el Reparto de Carga (Capa 2)	
<i>Número de Vecinos para el Reparto de Carga por BNG (l_{car})</i>	256
<i>Valor Mínimo de Mancha para la Fusión en Objetos (θ_{car})</i>	110
Interacción Lateral para la Fusión de Objetos (Capa 3)	
<i>Número de Vecinos para el Reparto de Carga por Objeto (l_{obj})</i>	256
<i>Valor Umbral Mínimo para la Detección de Objetos (θ_{obj})</i>	110

Tabla 5.1. Valores de los parámetros del modelo para la obtención de la silueta de una persona en movimiento

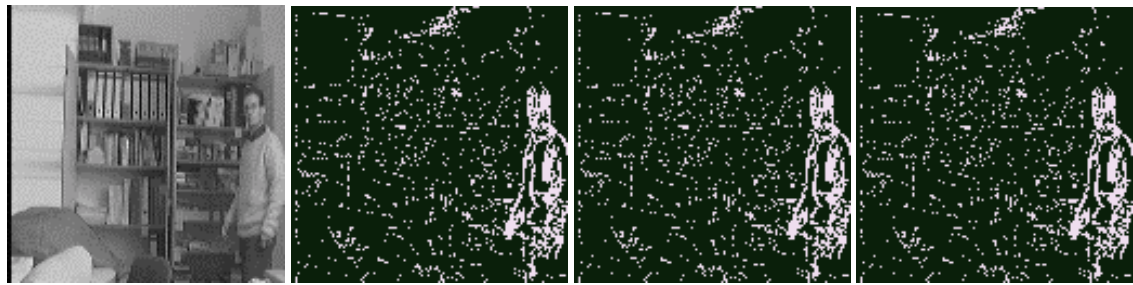


Imagen en t=1

Salida Capa 1

Salida Capa 2

Salida Capa 3

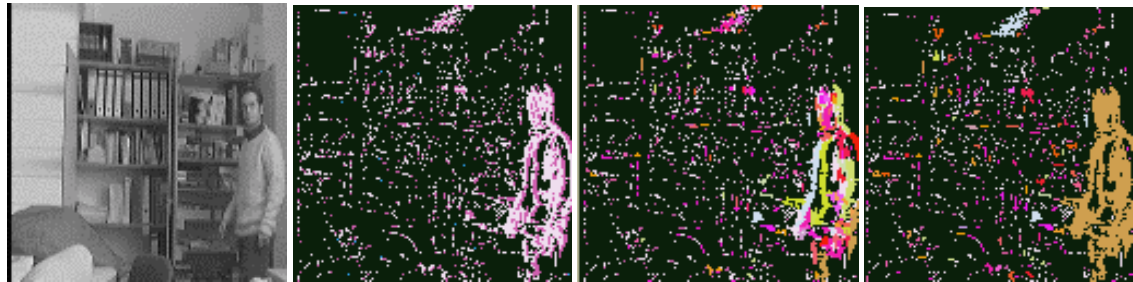


Imagen en t=2

Salida Capa 1

Salida Capa 2

Salida Capa 3

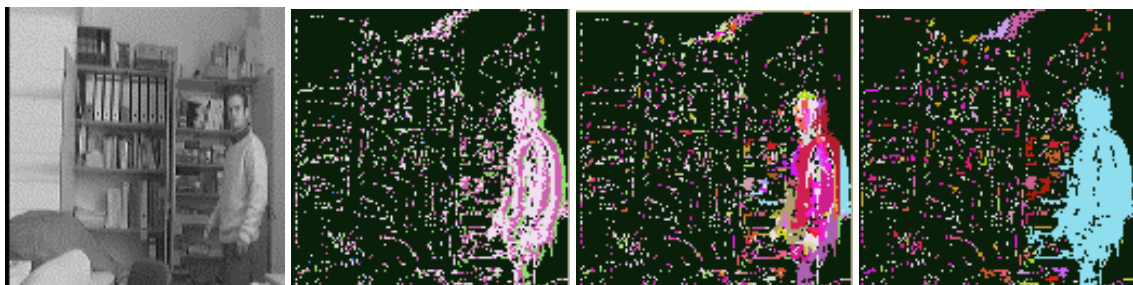


Imagen en t=5

Salida Capa 1

Salida Capa 2

Salida Capa 3

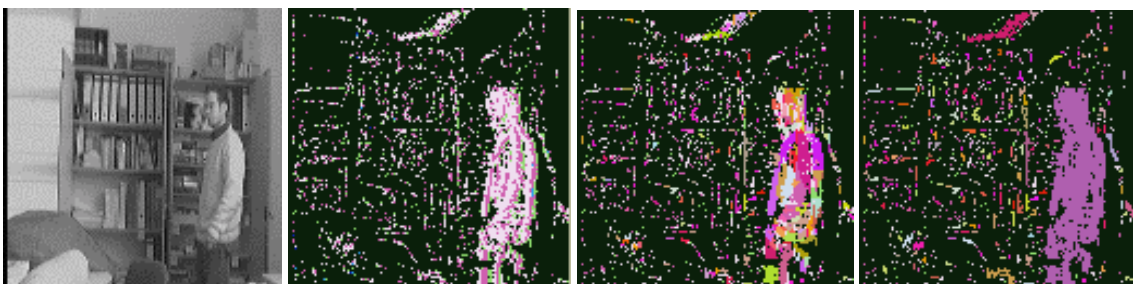


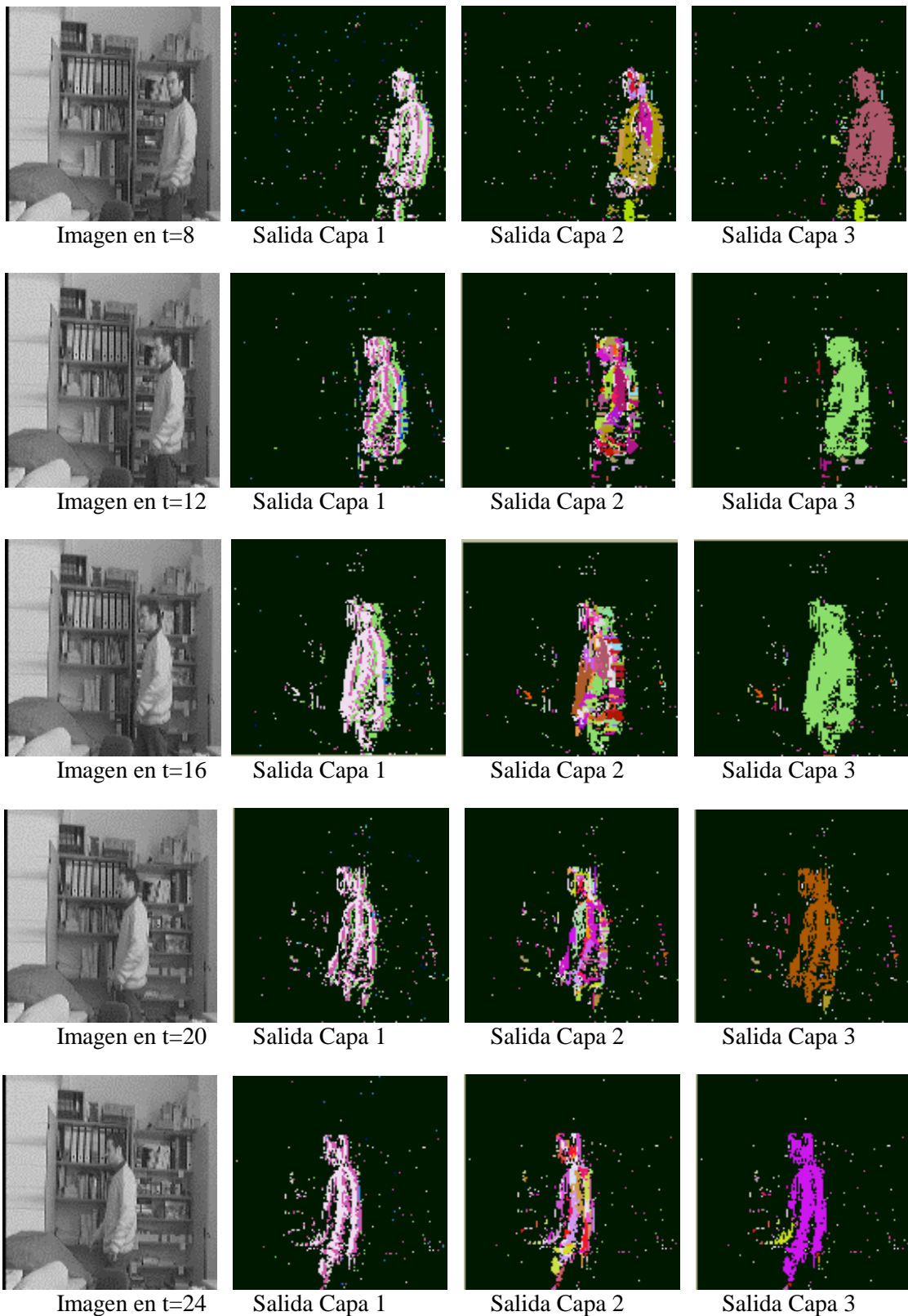
Imagen en t=10

Salida Capa 1

Salida Capa 2

Salida Capa 3

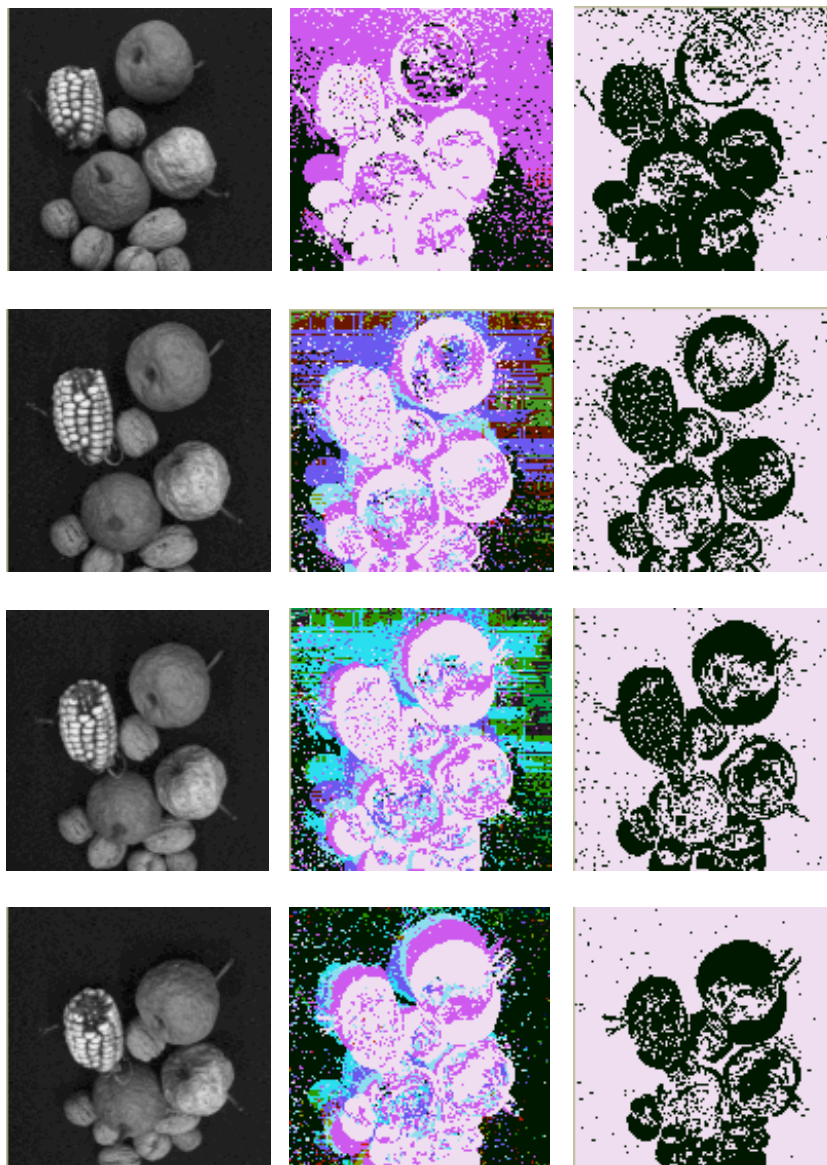
Ejemplo 5.1. Obtención de la silueta de una persona en movimiento usando la umbralización global simple



Ejemplo 5.2. Obtención de la silueta de una persona en movimiento usando la umbralización uniforme con solapamiento

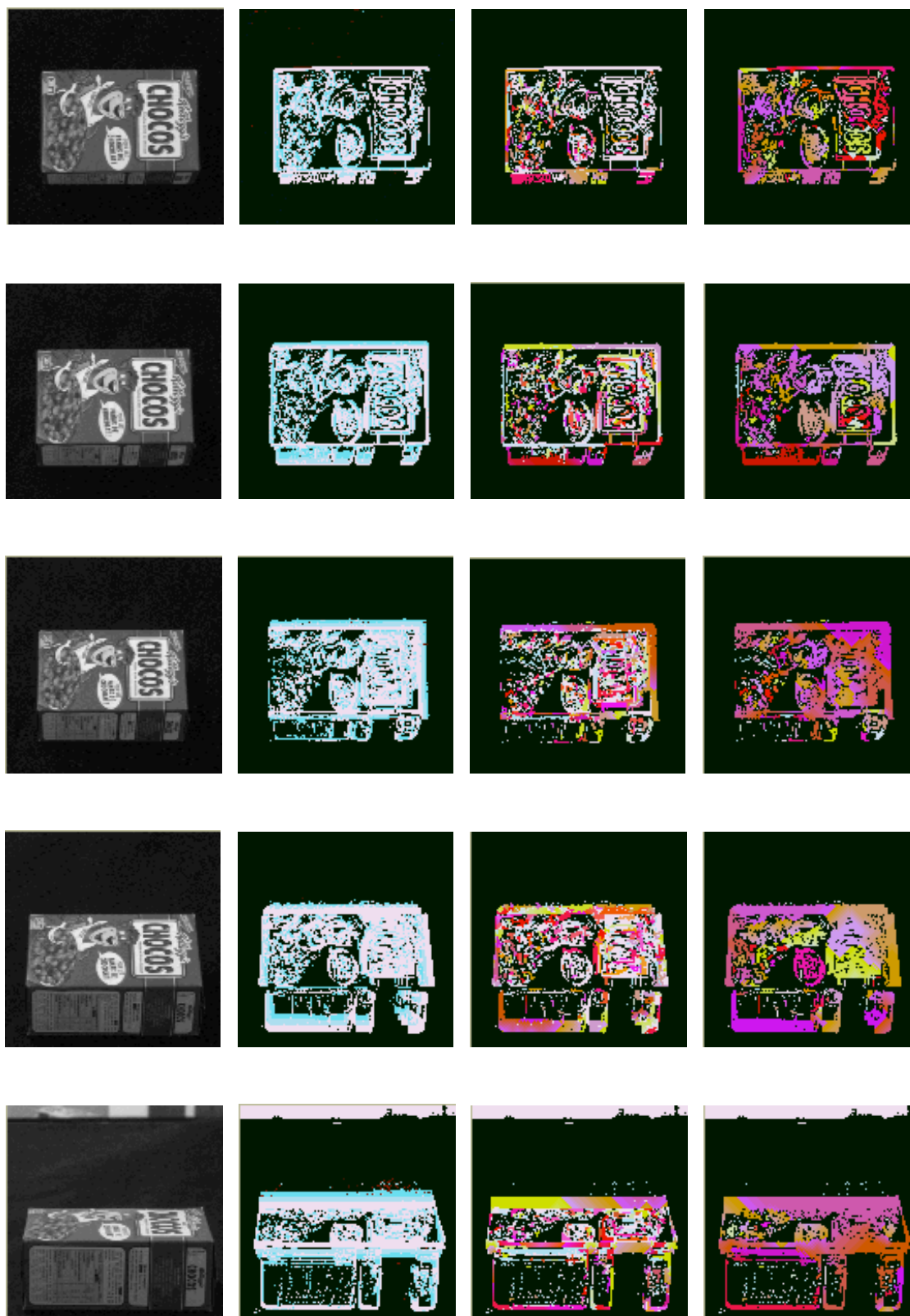
Tras comprobar el buen funcionamiento de la segmentación a bajo nivel mediante el modelo de Interacción Lateral en Computación Acumulativa para la obtención de una silueta en una secuencia de imágenes, se ofrecen los resultados obtenidos para un benchmark propuesto en el INRIA³.

Los tres ejemplos ofrecidos a continuación muestran la versatilidad del modelo propuesto, permitiendo segmentar todo tipo de objetos hasta el grado de precisión deseado.

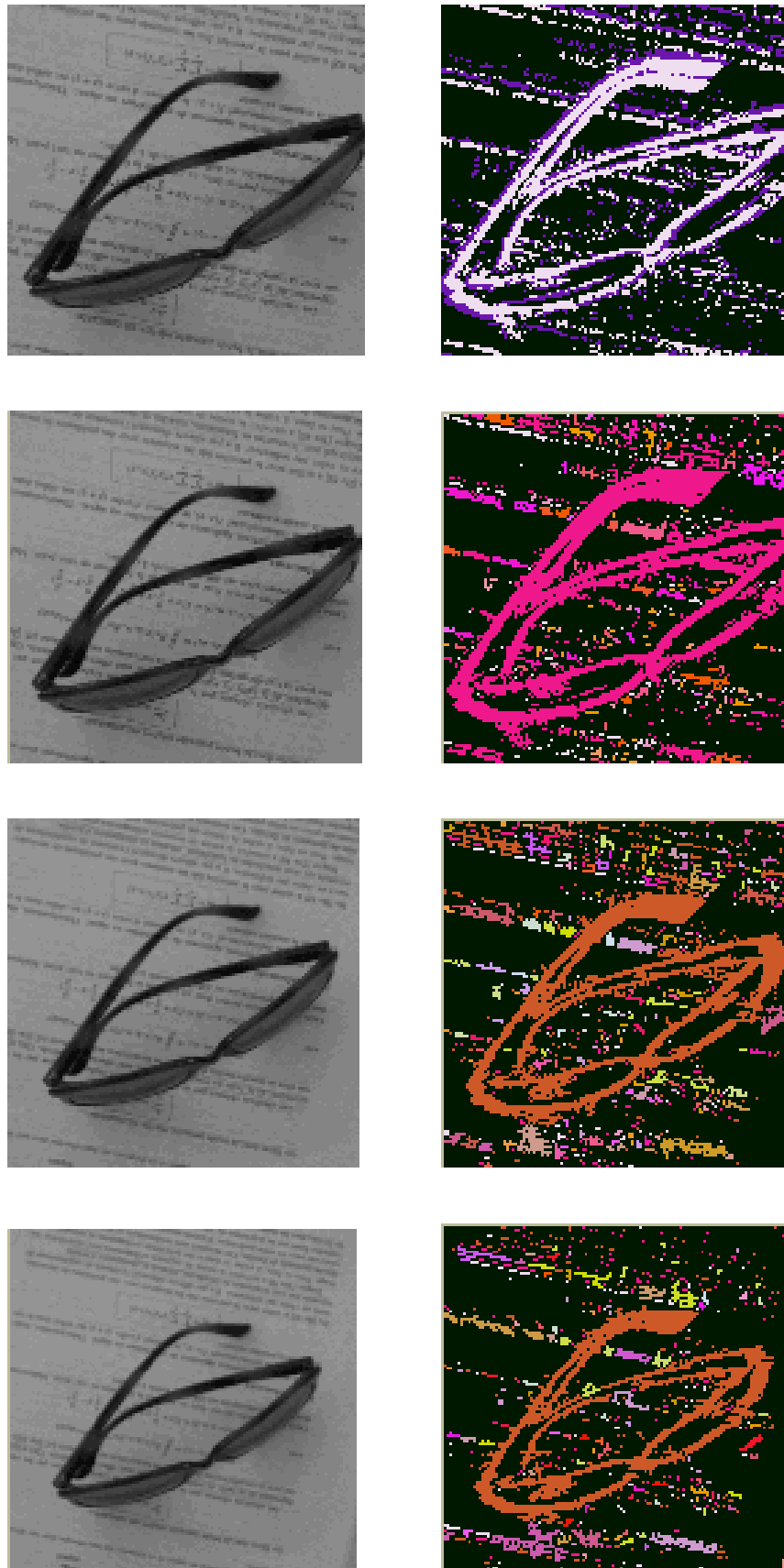


Ejemplo 5.3. Aplicación del modelo al movimiento de una fuente de frutos secos

³ Institut National de Recherche en Informatique et Automatique



Ejemplo 5.4. Aplicación del modelo a la rotación de una caja de cereales



Ejemplo 5.5. Aplicación del modelo al alejamiento de unas gafas de sol sobre una hoja escrita

5.2.1.2. Segmentación a bajo nivel – Debido al movimiento - Contornos

Se ofrecen a continuación los resultados de la obtención del contorno del mismo hombre segmentado en el ejemplo anterior.

La obtención de los contornos se realiza sencillamente aplicando un filtro de bordes una vez obtenidas las siluetas.



Imagen en t=7



Salida Obtención del Contorno



Imagen en t=12



Salida Obtención del Contorno

Ejemplo 5.6. Obtención del contorno de una persona

5.2.2. Segmentación a partir de la luminosidad

5.2.2.1. Segmentación a bajo nivel – Debido a la luminosidad – Cambios de

Luz

Se ofrecen a continuación los resultados de la obtención de las siluetas de todos los objetos presentes en una escena a partir del cambio de luminosidad de la misma. Los valores de los parámetros en cada una de las capas del modelo han sido:

Segmentación por Bandas de Nivel de Gris (Capa 0)	
<i>Número de Bandas de Nivel de Gris (n)</i>	8
Interacción Lateral para la Computación Acumulativa (Capa 1)	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	127
<i>Valor de Recarga por Vecindad (v_{rv})</i>	31
<i>Número de Vecinos para la Interacción Lateral (l_{per})</i>	16
<i>Valor Mínimo de Mancha por Banda de Nivel de Gris (θ_{per})</i>	100
Interacción Lateral para el Reparto de Carga (Capa 2)	
<i>Número de Vecinos para el Reparto de Carga por BNG (l_{car})</i>	128
<i>Valor Mínimo de Mancha para la Fusión en Objetos (θ_{car})</i>	100
Interacción Lateral para la Fusión de Objetos (Capa 3)	
<i>Número de Vecinos para el Reparto de Carga por Objeto (l_{obj})</i>	128
<i>Valor Umbral Mínimo para la Detección de Objetos (θ_{obj})</i>	120

Tabla 5.2. Valores de los parámetros del modelo para la obtención de siluetas a partir del cambio de luminosidad



Ejemplo 5.7. Obtención de siluetas a partir del cambio de luminosidad

5.3. Aplicaciones de alto nivel

5.3.1. Imágenes Médicas

Cada año se producen millones de imágenes médicas tridimensionales en el mundo para establecer un diagnóstico o para controlar una acción terapéutica. Estas imágenes provienen fundamentalmente del tomodensitómetro de rayos X o escáner, de la resonancia magnética, del ultrasonido o ecografía, o bien de la medicina nuclear. Estas imágenes médicas proveen información acerca de la forma y el funcionamiento de los órganos del cuerpo humano. El examen se realiza normalmente por observación en un soporte analógico (una película) de una secuencia de cortes bidimensionales.

La presente propuesta podría, en algunas aplicaciones médicas, optimizar la explotación de las imágenes. Según Ayache [Aya98], el análisis automatizado de las imágenes médicas puede ofrecer todo un abanico de herramientas nuevas de ayuda al diagnóstico. Entre estas herramientas, un buen sistema podría contribuir a tareas genéricas como:

1. la extracción de parámetros cuantitativos objetivos acerca de las formas y sus texturas, aplicables en estructuras anatómicas o patológicas en tres dimensiones;
2. la detección de cambios cuantitativos entre dos imágenes, permitiendo establecer un diagnóstico precoz o evaluar un tratamiento terapéutico;
3. la comparación de imágenes de pacientes diferentes, con la intención de estudiar la naturaleza de patologías similares;
4. la construcción de atlas anatómicos y funcionales probabilísticos, llevando a la construcción de representaciones estadísticas de las formas y de sus intensidades en las imágenes;
5. la medida del movimiento de los órganos dinámicos y de las articulaciones a partir de una secuencia temporal de imágenes 3D; en este caso, los datos son 4D (tres dimensiones espaciales y una dimensión temporal)

Aunque son muchos los científicos interesados en este tema en la actualidad, y existen ya numerosos prototipos que permiten demostrar los progresos realizados en los últimos años en este sentido, la mayoría de los problemas científicos abordados permanecen abiertos. Su resolución efectiva movilizará durante muchos años aún a equipos de investigadores en

matemáticas aplicadas, informática, física y medicina. Sus resultados revolucionarán la práctica médica del mañana.

5.3.1.1. Angiografías

La escena denominada Angiografías, como se recordará, es la escena de obtención del resultado de aplicar el modelo a imágenes de angiografías. Una angiografía es una aproximación volumétrica a las imágenes vasculares. Se requiere de la administración de un agente de contraste intravascular que recorre los vasos.

Se ofrecen en la siguiente tabla los valores dados a los parámetros de las tres capas del modelo.

Segmentación por Bandas de Nivel de Gris (Capa 0)	
<i>Número de Bandas de Nivel de Gris (n)</i>	12
Interacción Lateral para la Computación Acumulativa (Capa 1)	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	63
<i>Valor de Recarga por Vecindad (v_{rv})</i>	31
<i>Número de Vecinos para la Interacción Lateral (l_{per})</i>	16
<i>Valor Mínimo de Mancha por Banda de Nivel de Gris (θ_{per})</i>	100
Interacción Lateral para el Reparto de Carga (Capa 2)	
<i>Número de Vecinos para el Reparto de Carga por BNG (l_{car})</i>	256
<i>Valor Mínimo de Mancha para la Fusión en Objetos (θ_{car})</i>	100
Interacción Lateral para la Fusión de Objetos (Capa 3)	
<i>Número de Vecinos para el Reparto de Carga por Objeto (l_{obj})</i>	256
<i>Valor Umbral Mínimo para la Detección de Objetos (θ_{obj})</i>	100

Tabla 5.3. Valores de los parámetros del modelo para la obtención del resultado de una angiografía

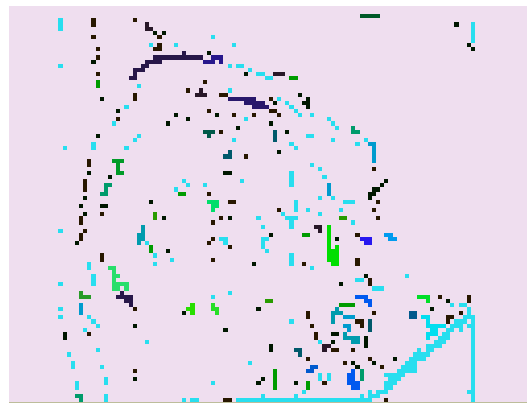
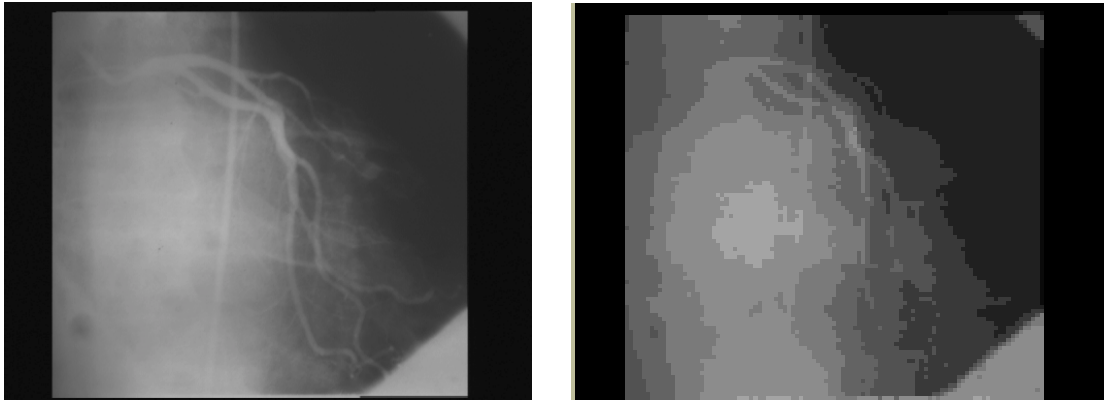


Imagen en $t=3$

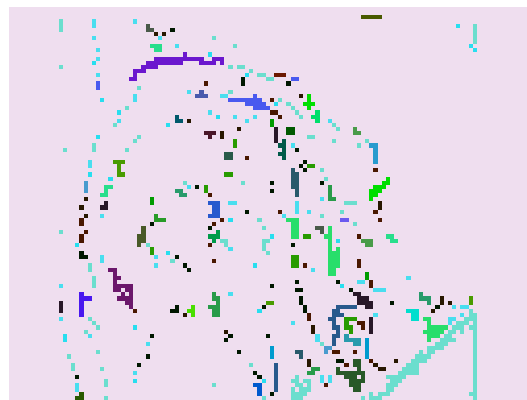
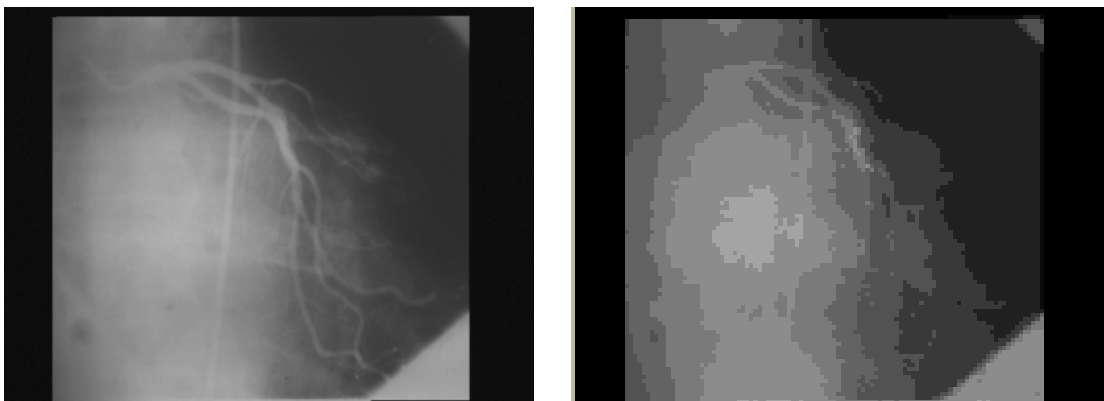


Imagen en $t=4$

Ejemplo 5.8. Aplicación del modelo a angiografías

Se puede observar en las imágenes resultado cómo el modelo de interacción lateral en computación acumulativa es capaz de suministrar la diferencia entre unas imágenes y otras, es decir, en este caso, el líquido contraste que recorre la anatomía vascular. Se ofrece: (a) la imagen original, (b) la imagen segmentada en bandas de nivel de gris, y, (c) el resultado de la aplicación del modelo.

5.3.1.2. Ecografías fetales

Por medio del control prenatal es posible vigilar la evolución de los embarazos. Los objetivos generales del control prenatal son: (a) identificar los factores e riesgo, (b) diagnosticar la edad de gestación, (c) diagnosticar el estado del feto, y (d) diagnosticar el estado de la madre.

Los elementos usados para el cálculo de la edad de gestación son el tiempo de amenorrea, comenzando desde el primer día de la última menstruación, y el tamaño uterino. El ignorar la edad de gestación constituye por sí mismo un factor de riesgo. Por ello es tan importante disponer de recursos de ultrasonido.

A su vez, los elementos cénicos que permiten evaluar el estado del feto son: (a) los latidos del corazón fetal, (b) los movimientos fetales, (c) el tamaño del útero, (d) la estimación clínica del peso fetal, y, (e) la estimación clínica del volumen del líquido amniótico. Los latidos de corazón pueden identificarse mediante ultrasonido a partir de la décima semana de embarazo. El tamaño del útero puede obtenerse utilizando una cinta métrica y permite estimar el tamaño del feto en cada control prenatal. La estimación clínica del peso fetal es probablemente uno de los parámetros más difíciles de obtener en un control prenatal, ya que requiere de una gran experiencia para su determinación. El error medio en la estimación del peso del feto en embarazos durante el tercer trimestre es aproximadamente de un diez por ciento.

La necesidad de hallar métodos rápidos y sencillos para estimar el peso de los fetos en el útero ha sido claramente manifestado por un gran número de profesionales. Las estimaciones por palpado abdominal y la producción de hormonas fetales han demostrado ser de una utilidad limitada [War77]. Solamente medidas muy exactas del feto permiten fechar el embarazo y efectuar un seguimiento del crecimiento del feto a partir de medidas tomadas a lo largo del tiempo.

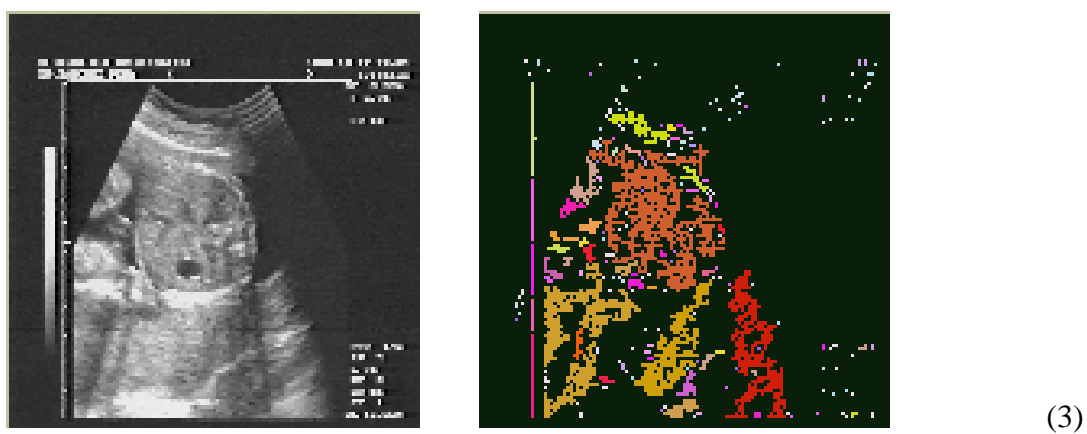
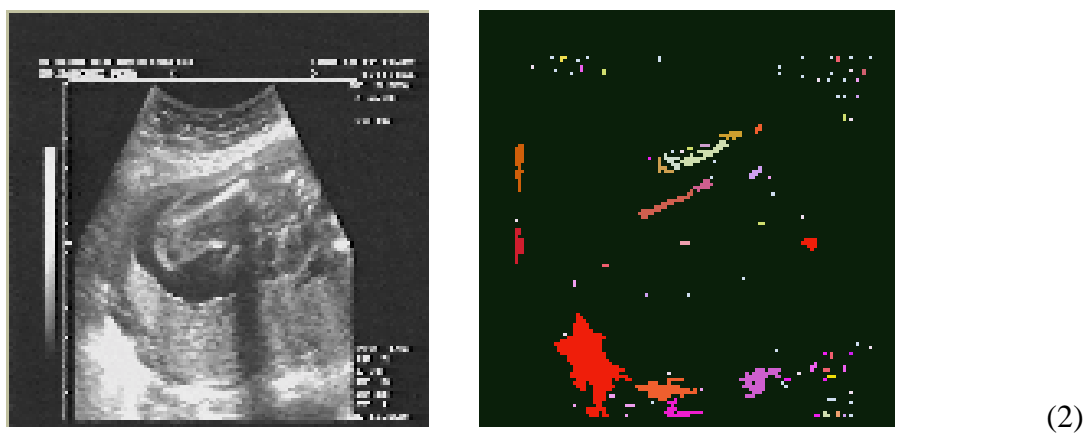
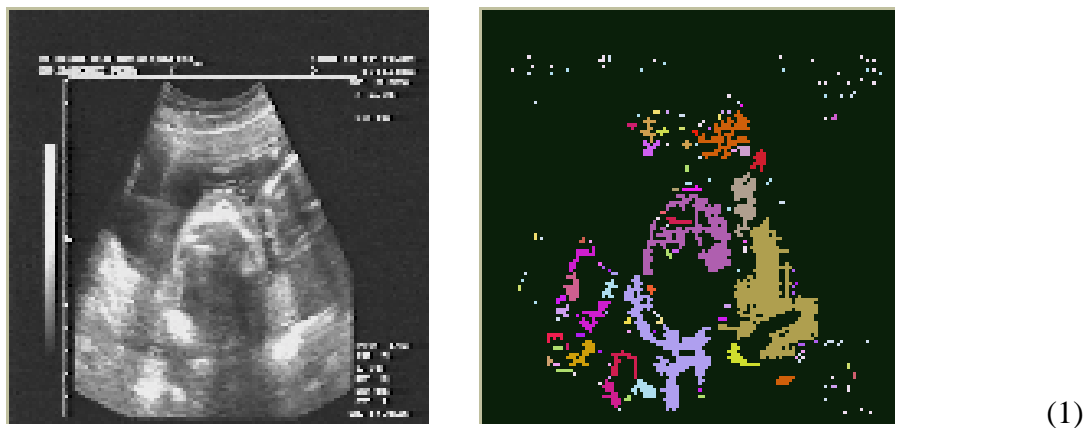
En los años 80 el estudio del crecimiento intrauterino utilizando parámetros ultrasónicos fue motivo de muchos artículos de investigación. El objetivo de predecir el peso fetal a partir de ecuaciones obtenidas en computadoras produjo tablas normalizadas para cada una de las partes

medibles del feto [She82] [Woo00]. La llegada de los scanners en tiempo real han añadido un mayor ímpetu a las técnicas de ultrasonido y han llevado al ultrasonido a ser una de las modalidades más importantes en obstetricia y ginecología.

La llegada del ultrasonido a la obstetricia ha creado una nueva especialidad llamada diagnóstico prenatal que se ha desarrollado grandemente desde su concepción. Las medidas del cuerpo fetal reflejan la edad de gestación del feto. Se realizan usualmente las siguientes medidas:

1. El diámetro biparietal (DBP) y la circunferencia de la cabeza. El diámetro biparietal es el diámetro entre los dos lados de la cabeza. Se mide después de las 13 semanas de gestación.
2. La longitud femoral (LF). Esta es la medida del hueso más largo del cuerpo y refleja el crecimiento longitudinal del feto.
3. La circunferencia abdominal (CA). Esta es la medida más importante a realizar en la última etapa del embarazo. Refleja más del tamaño fetal y de su peso que de su edad.

El ejemplo 5.9. muestra la aplicación del modelo de interacción lateral en computación acumulativa a la obtención (segmentación) del cráneo, del fémur y del abdomen. A partir de la discriminación de estas partes del cuerpo del feto, es posible, acercándose a tablas normalizadas, obtener de modo fiable el peso y el tamaño del mismo. Véase que la segmentación de imágenes médicas no es un problema trivial, ya que no solamente está en movimiento aquella parte del cuerpo que interesa destacar, sino que se está moviendo todo el entorno.



Ejemplo 5.9. Aplicación del modelo a ecografías fetales. (1) Cráneo. (2) Fémur. (3) Abdomen

5.3.2. Escenas de Tráfico

La monitorización del tráfico por carretera a partir de la visión por ordenador supone la recolección de datos que describan las características de los vehículos y su movimiento en la red de carreteras. Algunos ejemplos de datos de utilidad son el número de vehículos, la velocidad de un vehículo, la densidad de los vehículos, la longitud de un vehículo o la clase de vehículo (coche, furgoneta, camión, autobús). Esta información sirve para alguno de los propósitos descritos a continuación. Como herramienta de apoyo a las fuerzas de orden público, resulta interesante poder detectar infracciones al límite de velocidad, situaciones de conducción peligrosa o la utilización indebida de los carriles bus, por ejemplo. También es posible detectar congestiones e incidentes tras la localización de colas de tráfico o accidentes. Por último, sería de gran utilidad poder incrementar la capacidad del tráfico de la red viaria a partir de la información general de la red de carreteras, orientando el tráfico hacia rutas menos congestionadas.

Un típico sistema de visión para la monitorización del tráfico consta de una cámara CCD que capta imágenes de vídeo en directo que son digitalizadas y almacenadas en un ordenador. En este ordenador, los algoritmos pertinentes de visión realizan la detección de los vehículos, su seguimiento y su clasificación.

Un gran número de sistemas de monitorización de tráfico está basado en la detección de movimiento para segmentar las regiones de la imagen. Si las regiones contienen características adecuadas, son consideradas vehículos y pueden ser contados o seguidos según se desee. Existen algunas técnicas bien definidas para la detección de movimiento, de las cuales las más frecuentemente usadas en los sistemas de monitorización de tráfico son la diferencia entre tramas y la detección de movimiento basada en características.

La diferencia entre tramas [Son93] se basa en el hecho de que la diferencia entre dos tramas capturadas en instantes de tiempo distintos revelan regiones en movimiento. Una imagen diferencia I se genera calculando la diferencia absoluta entre dos tramas I_1 e I_2 y umbralizando el resultado.

$$I(x, y) = \begin{cases} 0, & \text{si } |I_1(x, y) - I_2(x, y)| \leq \theta \\ 1, & \text{en caso contrario} \end{cases}$$

siendo θ un umbral adecuado.

En el caso de la monitorización de tráfico es usual que I_1 sea la trama entrante e I_2 sea una trama de referencia (fondo). La trama de referencia es una imagen de la escena sin vehículos. El propósito del umbral es el de reducir los efectos del ruido y los cambios en la iluminación de la

escena. Esto último es un gran problema en visión por ordenador y suele ser necesario emplear métodos de actualización dinámica de la trama de referencia para adaptarse a la iluminación de la escena.

La detección de movimiento basada en características [Son93] funciona siguiendo las características importantes de trama a trama. El primer paso consiste en identificar las características adecuadas. Una vez identificadas en las tramas, se ha de usar un procedimiento de matching para encontrar las correspondencias entre estos puntos en tramas consecutivas.

El sistema TRIP (Traffic Research using Image Processing) [Ini85] [Dic89] fue un sistema diseñado para contar vehículos corriendo en una autopista bidireccional. La cámara estaba montada sobre la carretera, mirando verticalmente sobre la escena. Con la utilización de una imagen de referencia, las imágenes de entrada eran diferenciadas y el resultado umbralizado. Ello lleva a una imagen binaria donde deben aparecer únicamente los vehículos en movimiento. Otro sistema [Tak82] utiliza puntos de muestreo capaces de detectar la presencia de un vehículo con el fin de contar el número de vehículos. La velocidad de los vehículos puede medirse poniendo dos puntos de muestreo y calculando el tiempo entre la detección del vehículo en los dos puntos. En vez de usar puntos de muestreo, se han utilizado también líneas de detección [Abr84] que permiten contar los vehículos y obtener la velocidad aproximada de los vehículos. El sistema WADS (Wide Area Detection System) es capaz de detectar, contar y medir la velocidad de los vehículos en movimiento [Ini89]. Este sistema trabaja en una zona reducida de la imagen, véase una línea horizontal sobre un máximo de tres carriles para la detección de los vehículos. La velocidad se obtiene trazando una treintena de líneas en intervalos regulares de la carretera. La detección de movimiento por diferenciación de tramas es el núcleo de un sistema capaz de contar vehículos, medir sus velocidades y efectuar su seguimiento en complejos cruces de carretera [Ali92] [Bul93]. El sistema comercial CCATS (Camera and Computer Aided Traffic Sensor) usa una cámara CCD montada sobre una carretera, mirando verticalmente, y dando una visibilidad de unos veinticinco metros [Ver89] [Bri92]. Se digitaliza la totalidad de la imagen, pero se procesan únicamente unos veinte píxeles por carril. Este sistema es capaz de trabajar tanto de día como de noche. IMPACTS [Hoo89] [Kel92] opera a un nivel macroscópico al dar una descripción cualitativa de la distribución espacial de tráfico en movimiento y estacionario en la escena. El sistema parte la imagen en celdas, siendo cada celda del ancho aproximado de un carril y longitud de un coche. Cada celda puede estar en uno de los siguientes estados: sin vehículo, vehículo estacionario o vehículo en movimiento. Otro sistema, capaz de medir parámetros de colas de tráfico [Fat95], opera en pequeñas regiones de la imagen. La presencia de vehículos junto con la falta de movimiento significa una cola. Esta

condición se evalúa en todos los intervalos para encontrar la longitud de la cola. Un sistema capaz de detectar y seguir múltiples vehículos y dar al mismo tiempo información de la forma que puede ser usada en clasificación utiliza filtros Kalman [Mal95] [Kol94a] [Kol94b]. El mismo sistema se ha desarrollado incorporando una restricción de movimiento común [Bey97]. Utilizando igualmente la técnica de diferencia entre tramas para segmentar los vehículos en movimiento, se ha desarrollado un método que genera automáticamente una imagen de referencia [Cor96], creando una región de interés que reduce la porción de la imagen donde buscar los vehículos, reduciendo la carga computacional.

5.3.2.1. Autopistas

Segmentación por Bandas de Nivel de Gris (Capa 0)	
<i>Número de Bandas de Nivel de Gris (n)</i>	8
Interacción Lateral para la Computación Acumulativa (Capa 1)	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	127
<i>Valor de Recarga por Vecindad (v_{rv})</i>	63
<i>Número de Vecinos para la Interacción Lateral (l_{per})</i>	64
<i>Valor Mínimo de Mancha por Banda de Nivel de Gris (θ_{per})</i>	100
Interacción Lateral para el Reparto de Carga (Capa 2)	
<i>Número de Vecinos para el Reparto de Carga por BNG (l_{car})</i>	256
<i>Valor Mínimo de Mancha para la Fusión en Objetos (θ_{car})</i>	100
Interacción Lateral para la Fusión de Objetos (Capa 3)	
<i>Número de Vecinos para el Reparto de Carga por Objeto (l_{obj})</i>	256
<i>Valor Umbral Mínimo para la Detección de Objetos (θ_{obj})</i>	110

Tabla 5.4. Valores de los parámetros del modelo para la obtención de las siluetas de los vehículos en movimiento

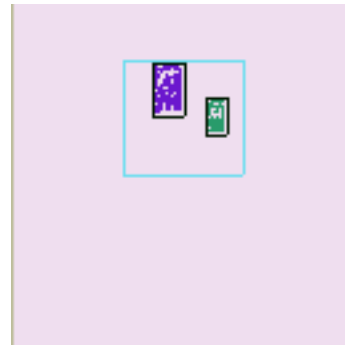
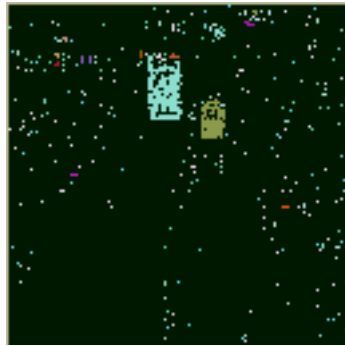


Imagen en t=10

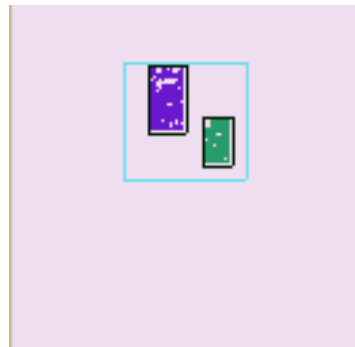


Imagen en t=15



Imagen en t=20

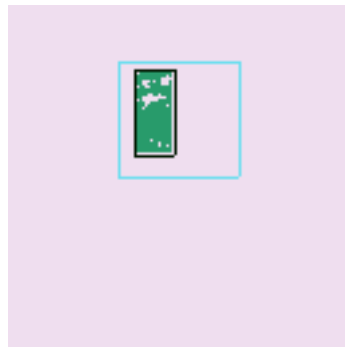


Imagen en t=25

Ejemplo 5.10. Aplicación del modelo a autopistas

La escena denominada Autopistas, como se recordará, es la escena de obtención de los vehículos presentes en la escena de tráfico en autopistas. Está destinada a la clasificación y conteo de los distintos tipos de vehículos presentes en la escena a lo largo del tiempo en función de su tamaño. La columna primera del ejemplo muestra la escena de tráfico original, la columna segunda el resultado de la aplicación del modelo de interacción lateral en computación acumulativa, y, finalmente, en la columna tercera, el resultado de la aplicación de alto nivel. Véase que se clasifican y cuentan los vehículos a su paso por la ventana dibujada en color celeste.

5.3.2.2. Cruces de Calles

Se ha trabajado en esta escena con la siguiente secuencia de imágenes¹, que forma parte de un benchmark internacional, y que es típica de un sistema de vigilancia de cruces de calles:



Figura 5.3. Sistema de vigilancia en cruces de calles

La aplicación ha sido diseñada para permitir elegir (activando o desactivando la opción correspondiente) el (los) tipo(s) de aplicación que se desea lanzar en el contexto de la vigilancia en cruces de calles. Las posibilidades ofrecidas son:

- **Conteo de Vehículos:** Aplicación de conteo de los vehículos.
- **Situaciones de Tráfico:** Aplicación que ofrece la situación del tráfico en un momento dado, pudiéndose hablar de tráfico fluido, tráfico denso y tráfico atascado.

¹ Copyright © 1998 by H.-H. Nagel. Institut für Algorithmen und Kognitive Systeme.
Fakultät für Informatik, Universität Karlsruhe (TH). Postfach 6980, D-76128 Karlsruhe, Germany

Por otra parte, decir que la escena de tráfico dedicada a los cruces de calles incorpora la posibilidad de definir zonas de vigilancia específicas en la secuencia de imágenes. En el ejemplo que nos ocupa se han definido tres zonas de vigilancia diferentes, permitiendo a alto nivel evaluar el paso de vehículos desde unas zonas a otras.

Se ofrecen en la siguiente tabla los valores dados a los parámetros de las cuatro capas del modelo.

Segmentación por Bandas de Nivel de Gris (Capa 0)	
<i>Número de Bandas de Nivel de Gris (n)</i>	8
Interacción Lateral para la Computación Acumulativa (Capa 1)	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	63
<i>Valor de Recarga por Vecindad (v_{rv})</i>	31
<i>Número de Vecinos para la Interacción Lateral (l_{per})</i>	32
<i>Valor Mínimo de Mancha por Banda de Nivel de Gris (θ_{per})</i>	70
Interacción Lateral para el Reparto de Carga (Capa 2)	
<i>Número de Vecinos para el Reparto de Carga por BNG (l_{car})</i>	70
<i>Valor Mínimo de Mancha para la Fusión en Objetos (θ_{car})</i>	64
Interacción Lateral para la Fusión de Objetos (Capa 3)	
<i>Número de Vecinos para el Reparto de Carga por Objeto (l_{obj})</i>	256
<i>Valor Umbral Mínimo para la Detección de Objetos (θ_{obj})</i>	100

Tabla 5.5. Valores de los parámetros del modelo para la obtención de las siluetas de los vehículos en cruces de calles

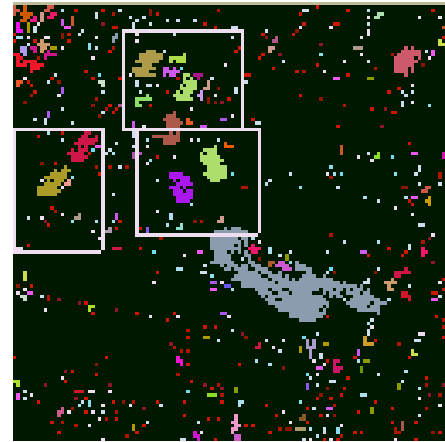


Imagen en $t=5$



Imagen en $t=15$

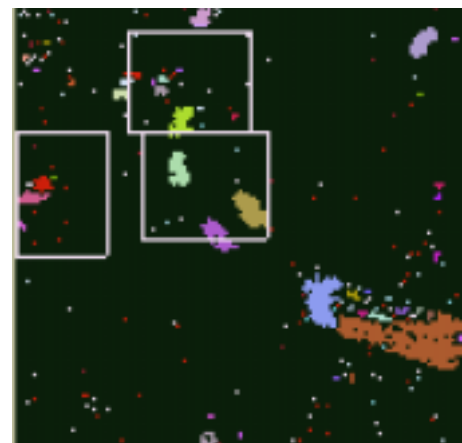


Imagen en $t=35$

Ejemplo 5.11. Aplicación del modelo a cruces de calles

5.3.3. Escenas de Vigilancia

En los últimos años, el empleo de métodos visuales en sistemas de vigilancia automatizada ha gozado de una creciente popularidad, aportando algunas veces de soluciones capaces de competir con aquellas basadas en sensores tradicionales como las células fotográficas o los infrarrojos. En algunos campos de aplicación, los resultados de la investigación son ya suficientemente maduros como para ser explotados comercialmente. En otros, tenemos el control del entorno, el conteo de personas [Ros94], el control del tráfico [Hua94a] o la monitorización de las masas humanas [Boz93]. Se han obtenido bastante buenos resultados en situaciones reales y suficientemente generales.

Por ello, en la última década se ha dado un incremento espectacular en el uso de las cámaras de vigilancia visual. Estas se suelen instalar en cruces de calles, en áreas de aparcamientos de vehículos en lugares de alta seguridad, con el fin de monitorizar el flujo de los peatones y los vehículos, redundando en un mayor análisis de los datos y de la seguridad. La labor de monitorización de secuencias de imágenes se asigna normalmente a un operador humano que está a la espera de que ocurran eventos de importancia. Los operadores, sin embargo, se cansan o se aburren con facilidad, de modo que pierden la concentración necesaria. De ahí que resulte esencial poner en funcionamiento sistemas de vigilancia autónomos que puedan interpretar las imágenes y alertar al operador humano solamente cuando ocurren hechos sospechosos.

Los problemas, candentes en la investigación actual en el campo de la visión por ordenador, relacionados con la vigilancia visual autónoma son principalmente: (1) la adaptación, es decir, la capacidad que tienen los sistemas sensores de ajustarse autónomamente para tratar con las condiciones siempre cambiantes del entorno; (2) la localización, el seguimiento y el reconocimiento de objetos, en presencia de oclusiones parciales o sombras; (3) el reconocimiento de eventos, especialmente en o que concierne a su estructura temporal; (4) la explotación de información a priori bien acerca de la estructura de la escena o de la interacción entre las personas y los objetos.

A lo largo de los diez últimos años, los investigadores en el campo de la visión artificial han ofrecido una serie de soluciones al problema de la vigilancia visual. El artículo sobre la interpretación semántica [Nag88] es un trabajo importante sobre el tema. También se han introducido redes bayesianas [Bux94] [Bux95], basadas en los trabajos de Pearl [Pea88] que detectan los focos de atención de una escena dinámica y que ofrecen interpretaciones sobre las situaciones de tráfico. También se ha propuesto una aproximación probabilística para la vigilancia visual automática de las autopistas californianas [Hua94a]. Otro trabajo de

importancia trata de la vigilancia en estaciones metropolitanas para monitorizar el vandalismo [Bog96]. Más recientemente, se ha propuesto una lógica difusa temporal para descripciones verbales de alto nivel de escenas de tráfico [Haa98]. También es posible leer sobre una aproximación novedosa para la descripción de las actividades humanas [Bob96]. Se ha descrito igualmente un modelo de cadenas ocultas acopladas de Markov para interpretar las interacciones humanas en vigilancia visual [Bra97a] [Bra97b].

5.3.3.1. Personas

La vigilancia visual implica principalmente la interpretación de las secuencias de imágenes. La vigilancia visual avanzada de entornos complejos va más allá y automatiza la detección de situaciones predefinidas de alarma en un contexto dado [Cor91] [Nic93] [Uka93] [Ara93] [Aza96]. El papel de la computación automática en este tipo de sistemas es ayudar al operador humano a realizar diferentes tareas, tales como la observación y la detección, la interpretación y la comprensión, dar alarmas y diagnosticar falsas alarmas.

El propósito de la vigilancia visual avanzada, en muchos casos, es ayudar a los operadores humanos a monitorizar la seguridad de ciertas áreas. Como ejemplos, tenemos las estaciones de metro, las infraestructuras de las redes de ferrocarriles o los aparcamientos de vehículos. En estos lugares es importante reforzar las posibles situaciones de alarma o los comportamientos anómalos o peligrosos de las personas.

En este contexto, también ha habido muchas aproximaciones al análisis de secuencias de imágenes. Estas van desde la visión activa [Ber92] hasta los métodos de seguimiento basados en modelos [Fer98c], desde los procedimientos de los contornos activos [Ley93] hasta la integración de diferentes características (tanto numéricas como semánticas) para el conocimiento de la escena [Dar98].

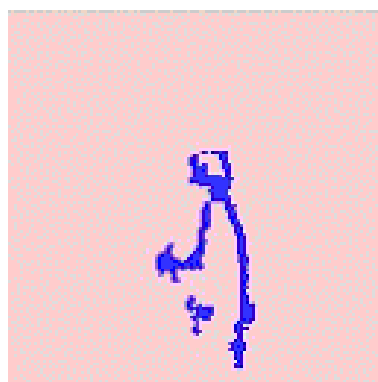
La aplicación, en este caso, ha sido diseñada para permitir elegir (activando o desactivando la opción correspondiente) las situaciones de alarma que se desea lanzar en el contexto de la vigilancia de personas. Las posibilidades ofrecidas son:

- Situación de Riesgo: Aplicación de detección de situaciones anómalas en el comportamiento humano en cuanto a su velocidad de desplazamiento, pudiéndose hablar de persona paseando, persona caminando y persona corriendo.

Se ofrecen en la siguiente tabla los valores dados a los parámetros de las cuatro capas del modelo.

Segmentación por Bandas de Nivel de Gris (Capa 0)	
<i>Número de Bandas de Nivel de Gris (n)</i>	
Interacción Lateral para la Computación Acumulativa (Capa 1)	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	
<i>Valor de Recarga por Vecindad (v_{rv})</i>	
<i>Número de Vecinos para la Interacción Lateral (l_{per})</i>	
<i>Valor Mínimo de Mancha por Banda de Nivel de Gris (θ_{per})</i>	
Interacción Lateral para el Reparto de Carga (Capa 2)	
<i>Número de Vecinos para el Reparto de Carga por BNG (l_{car})</i>	
<i>Valor Mínimo de Mancha para la Fusión en Objetos (θ_{car})</i>	
Interacción Lateral para la Fusión de Objetos (Capa 3)	
<i>Número de Vecinos para el Reparto de Carga por Objeto (l_{obj})</i>	
<i>Valor Umbral Mínimo para la Detección de Objetos (θ_{obj})</i>	

Tabla 5.6. Valores de los parámetros del modelo para la obtención de las siluetas de las personas en un sistema de vigilancia



Ejemplo 5.12. Aplicación del modelo a la vigilancia de personas

5.3.4. Imágenes Satélite

En gran auge se encuentran también los proyectos de estudio de las imágenes dinámicas de observación de la tierra, de los aires y de la atmósfera obtenidas por teledetección. Las principales actividades de investigación en este campo pueden agruparse en dos grandes categorías:

1. La modelización de las estructuras correspondientes a las observaciones aire / tierra / mar obtenidas vía teledetección.
2. El estudio de la evolución temporal de fenómenos naturales.

Un ejemplo de imagen tomada por satélite es la siguiente²:



Figura 5.4. Imagen por satélite

² Por cortesía de la NASA.

5.3.4.1. Meteosat

Los datos climatológicos analizados en los últimos tiempos revelan cambios y variaciones que resultan de un interés muy significativos. Así, por ejemplo, desde comienzos del siglo veinte, ha quedado patente que la temperatura media del globo terrestre se ha incrementado en $0,4^{\circ}$ C, habiéndose dado un incremento mayor durante los treinta últimos años [Kar95].

El clima en la tierra y sus variaciones están fuertemente ligadas a la presencia de sustancia acuosa (en forma de vapor, líquido y sólido) en la atmósfera, así como la relación de ésta con el agua en la superficie, la temperatura atmosférica, la radiación y su circulación. La interacción entre estos procesos constituye, hoy en día, un área poco conocida en las ciencias atmosféricas, y por lo tanto, no existe un conocimiento cuantitativo completo del sistema global climatológico. Todo ello implica que está muy limitada la exactitud de las predicciones de los cambios en las variables climáticas en diferentes instantes de tiempo. Dos de los parámetros de tiempo y clima más importantes y peor medidos son el vapor de agua y la precipitación.

Es conocido que la distribución de la abundancia relativa de diferentes tipos de nubes está íntimamente relacionada con la estructura de patrones meteorológicos en diferentes partes del globo [Lau94]. La evaporación que se produce en los océanos mundiales pone vapor de agua en la atmósfera. Este vapor puede entonces volver a condensarse y aportar energía para que las tormentas produzcan la lluvia necesaria para sostener la vida humana, animal y vegetal. Los estudios recientes muestran que los ciclos hidrológicos en EEUU están cambiando, con un incremento en la cobertura de las nubes [Kar90] y de las precipitaciones [Gro94a] [Gro94b] y un decremento en la evaporación [Pet95].

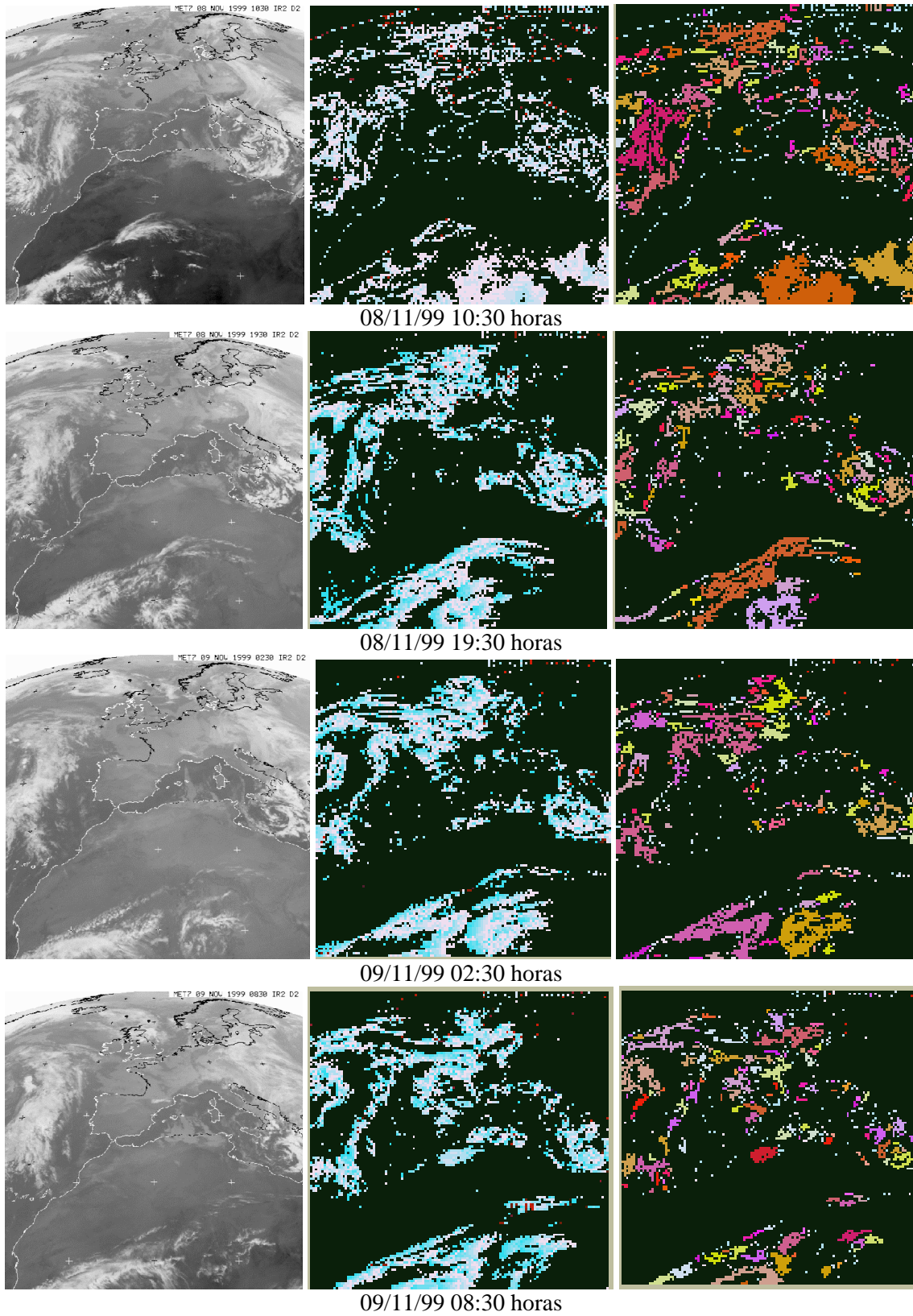
Una nube es un agregado visible de agua líquida o partículas de hielo suspendidas en el aire. Una nube puede estar formada enteramente por agua líquida, enteramente por cristales de hielo, o por una mezcla de ambos. Las nubes se forman cuando el aire se satura, es decir, cuando la humedad relativa alcanza el 100%. Esta saturación se da fundamentalmente cuando la temperatura del aire se enfría. Las nubes pueden clasificarse generalmente de acuerdo a ciertas características como la altitud, la apariencia o el origen [PSC]. En cuanto a la altitud, la distinción es en función de su pertenencia a alguna de las capas de la atmósfera, obteniéndose: (a) nubes altas, (b) nubes medias, (c) nubes bajas, (d) nieblas, (e) nubes multicapa y (f) nubes orográficas. Según su apariencia, las nubes pueden ser gordas o delgadas, tener los bordes bien definidos o difusos, y pueden ser asociadas con buen tiempo o precipitaciones.

Las observaciones por satélite pueden usarse para describir cuantitativamente la distribución espacial y la evolución temporal del campo de vapor acuoso [Car94] [Sod94] [Sod96]. Permiten igualmente conducir investigaciones sobre el sistema climático y utilizar los datos para evaluar y mejorar cualquier modelo de simulación climático. Así, entre otros objetivos, pueden fijarse los siguientes [JCS]: (a) la documentación de la climatología de las distribuciones nubosas y su variabilidad zonal a partir del análisis de los datos de imágenes tomadas desde satélite a escala regional, (b) la obtención de la climatología de nubes por satélite a partir del análisis de los datos de satélite a escala global, (c) el estudio de los procesos elementales de formación de nubes, (d) el estudio de las interacciones entre conjuntos de nubes, y, (e) el estudio de movimientos a gran escala de las nubes.

Las imágenes utilizadas en este último ejemplo han sido tomadas de la dirección <http://www.eumetsat.de/en/> bajo Copyright © 1999 EUMETSAT.

Segmentación por Bandas de Nivel de Gris (Capa 0)	
<i>Número de Bandas de Nivel de Gris (n)</i>	8
Interacción Lateral para la Computación Acumulativa (Capa 1)	
<i>Valor Mínimo de Permanencia (v_{des})</i>	0
<i>Valor Máximo de Permanencia (v_{sat})</i>	255
<i>Valor de Descarga por Detección de Movimiento (v_{dm})</i>	127
<i>Valor de Recarga por Vecindad (v_{rv})</i>	63
<i>Número de Vecinos para la Interacción Lateral (l_{per})</i>	32
<i>Valor Mínimo de Mancha por Banda de Nivel de Gris (θ_{per})</i>	100
Interacción Lateral para el Reparto de Carga (Capa 2)	
<i>Número de Vecinos para el Reparto de Carga por BNG (l_{car})</i>	256
<i>Valor Mínimo de Mancha para la Fusión en Objetos (θ_{car})</i>	100
Interacción Lateral para la Fusión de Objetos (Capa 3)	
<i>Número de Vecinos para el Reparto de Carga por Objeto (l_{obj})</i>	512
<i>Valor Umbral Mínimo para la Detección de Objetos (θ_{obj})</i>	120

Tabla 5.7. Valores de los parámetros del modelo para la obtención de las siluetas de las nubes en movimiento



Ejemplo 5.13. Aplicación del modelo al seguimiento de nubes

Conclusiones y principales aportaciones

El trabajo objeto de esta memoria ha consistido en el estudio, diseño y realización práctica de un sistema de computación acumulativa con interacción lateral, destinado a la segmentación de los objetos deformables en movimiento en secuencias de imágenes.

Los mecanismos de computación utilizados están inspirados en la biología. La arquitectura diseñada, modular, distribuida y parcialmente autoprogramable, consigue realizar tareas complejas. En este caso, la extracción de siluetas deformables en secuencias de imagen. En este caso, esta operación se realiza mediante módulos de computación simples inspirados en el efecto acumulativo que se produce en las conexiones sinápticas. Estos son los módulos de computación acumulativa, a los que se ha añadido, como aportación principal de esta tesis, una interacción lateral. Puede verse, a tenor de los resultados obtenidos, que mejora la capacidad computacional de forma significativa, permitiendo abordar aplicaciones que son enormemente costosas si se acometen con algoritmos de estructura menos biológica. Se ha conseguido abordar problemas complejos mediante estructuras simples.

El sistema desarrollado solventa el problema de extracción de siluetas en secuencias de imagen continuas de forma más que satisfactoria. De hecho, se consigue segmentar las distintas siluetas que aparecen en la escena, separándolas del fondo de forma independiente, a pesar de que éstas estén compuestas de varias zonas con distintos niveles de gris. Los pasos más significativos seguidos para la consecución de la solución ofrecida son: (a) preprocesamiento de las imágenes de entrada, (b) segmentación a partir de las bandas de nivel de gris, (c) computación acumulativa para la identificación de manchas móviles, (d) eliminación del fondo de la imagen mediante interacción lateral, (e) fusión de manchas en objetos mediante interacción lateral.

Como principales aportaciones de esta tesis podemos destacar, por tanto:

- Un exhaustivo análisis de la problemática asociada a la extracción de siluetas en secuencias de imágenes.
- La utilización de interacción lateral en computación acumulativa.

- El diseño de un sistema inspirado en la biología, modular, distribuido y parcialmente autoprogramable (neuronal), con capacidad de segmentar siluetas deformables, trabajando sobre una secuencia indefinida de imágenes.
- Un entorno de dispensación de imágenes capaz de nutrir el sistema con secuencias de imágenes sintéticas y reales.
- Una aplicación destinada a la diferenciación de los diversos elementos móviles que componen una escena representada mediante una secuencia indefinida de imágenes.
- Un análisis de las aplicaciones para las que el sistema puede resultar de interés (control de tráfico en carreteras y urbano, vigilancia, imágenes médicas, imágenes satélite, ...).

Las líneas de trabajo futuro habrían de englobarse bajo:

- Un diseño de capas superiores capaces de inyectar conocimiento de más alto nivel a la información que ofrecen las siluetas extraídas. Esto permitiría pasar del nivel de procesado de imágenes al de análisis de escenas.
- Una búsqueda y discriminación de aplicaciones en las que los resultados obtenidos en la extracción de elementos móviles resulten más eficaces.

Bibliografia

- [Abr84] Abramczuk T. (1984). **A microcomputer based TV detector for road traffic**. Symposium on Road Research Program, Tokyo, Japan.
- [Ade82] Adelson E.H. & Movshon J.A. (1982). **Phenomenal coherence of moving visual patterns**. *Nature*, 300, 523-525.
- [Ade85] Adelson E.H. & Bergen J.R. (1985). **Spatiotemporal energy models for the perception of motion**. *Journal of the Optical Society of America A*, 2, 284-299.
- [Agg88] Aggarwal J.K. & Nandhakumar N. (1988). **On the computation of motion from sequences of images – A review**. *Proceedings of the IEEE*, 917-935.
- [Ali92] Ali A.T. & Dagless E.L. (1992). **A parallel processing model for real-time computer vision-aided road traffic monitoring**. *Parallel Processing Letters*, 2(2), 257-264.
- [Ami91] Amit Y., Grenander U. & Piccioni M. (1991). **Structural image restoration through deformable templates**. *Journal of the American Statistical Association*, 86 (414), 376-387.
- [And77] Anderson J.A. (1977). **Neural models with cognitive implications**. En: La Berge D. & Samuels S.J., eds. *Basic Perception and Comprehension Models*, 27-90, Erlbaum, Hillsdale, NJ.
- [Ara93] Aranda J., Amat J. & Frigola M. (1993). **A multitracking system for trajectory analysis of people in a restricted area**. *Proceedings of the 4th International Workshop on Time-Varying Image Processing and Moving Object Recognition*, Elsevier Science.
- [Aya98] Ayache N. (1998). **L'analyse automatique des images médicales. Etat de l'art et perspectives**. Rapport de Recherche 3364, Febrero 1998, INRIA, Francia.
- [Aza96] Azam M., Potlapalli H., Janet J. & Luo R.C. (1996). **Outdoor landmark recognition using segmentation, fractal model and neural networks**. *Proceedings of the DARPA Image Understanding Workshop*, 189-203.
- [Bal82] Ballard D.H. & Brown C.M. (1982). *Computer Vision*. Prentice-Hall, New Jersey.
- [Bar65] Barlow H.B. & Levick R.W. (1965). **The mechanism of directionally selective units in the rabbit's retina**. *Journal of Physiology*, 178, 477-504.
- [Bar94] Barron J.L., Fleet D.J. & Beauchemin S.S. (1994). **Performance of optical flow techniques**. *International Journal of Computer Vision*, 12 (1), 43-77.
- [Bas94] Bascle B., Bouthemy P., Deriche N. & Meyer F. (1994). **Tracking complex primitives in an image sequence**. *Proceedings of the 12th International Conference on Pattern Recognition*, Jerusalem, 426-431.
- [Bat82] Bathe K. (1982). *Finite Element Procedures in Engineering*. Prentice-Hall.

- [Ber92] Bergen J.R., Burt P.J., Hingorani R. & Peleg S. (1992). **A three-frame algorithm for estimating two component image motion.** IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-14 (9), 886-896.
- [Bey97] Beymer B., McLauchlan P., Coifman B. & Malik J. (1997). **A real-time computer vision system for measuring traffic parameters.** CVPR.
- [Bla84] Blasdel G.G. & Fitzpatrick D. (1984). **Physiological organization of layer 4 in macaque striate cortex.** Journal of Neuroscience, 46, 880-895.
- [Bla93] Blake A., Curwen R. & Zisserman A. (1993). **A framework for spatiotemporal control in the tracking of visual contours.** International Journal of Computer Vision, 11 (2), 127-145.
- [Bla96] Black M.J. & Anandan P. (1996). **The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields.** Computer Vision and Image Understanding, 63 (1), 75-104.
- [Bob96] Bobick A.F. (1996). **Computers seeing action.** Proceedings of the British Machine Vision Conference, vol. 1, 13-22.
- [Bog96] Bogaert M., Chleq N., Cornez P., Regazzoni C.S., Teschioni A. & Thonnat M. (1996). **The PASSWORDS project.** Proceedings of the International Conference on Image Processing, 675-678.
- [Boo91] Bookstein F. (1991). *Morphometric Tools for Landmark Data.* Cambridge University Press.
- [Boz93] Bozzoli A., Barbo R., Caprile B. & Rossi M. (1993). **Crowding evaluation: A visual reflex-based approach.** En: Impedovo S., ed., *Progress in Image Analysis and Processing III – ICIAP Proceedings*, 83-86.
- [Bra97a] Brand M. (1997). **The inverse Hollywood problem: From video clips to storyboards via causal analysis.** Proceedings of the American Association of Artificial Intelligence, Providence, RI.
- [Bra97b] Brand M., Oliver N. & Pentland A. (1997). **Coupled hidden Markov models for complex action recognition.** Proceedings of Computer Vision and Pattern Recognition, San Juan, Puerto Rico.
- [Bri92] Briquet P. (1992). **Video processing applied to road and urban traffic monitoring.** 6th International Conference on Road Traffic Monitoring, 153-157, London, IEE.
- [Bul80] Bullier J. & Henry G.H. (1980). **Ordinal position and afferent input of neurons in monkey striate cortex.** Journal of Comparative Neurology, 193, 913-935.
- [Bul93] Bulas-Cruz J., Ali A.T. & Dagless E.L. (1993). **Real-time motion detection and tracking.** 8th Scandinavian Conference on Image Analysis.
- [Bül89] Bühlhoff H.H., Little J. & Poggio T. (1989). **A parallel algorithm for real-time computation of optical flow.** Nature, 337, 549-553.
- [Bux94] Buxton H. & Howarth R. (1994). **Situation description from image sequences.** AAAI Workshop on Integrating of Natural Language and Vision Processing.
- [Bux95] Buxton H. & Gong S. (1995). **Visual surveillance in a dynamic and uncertain world.** Artificial Intelligence, 78 (1-2): 431-459.

- [Car94] Carleton A.M., Jelinski D.E., Travis D., Arnold D., Brinegar R. & Easterling D.R.. (1994). **Climatic-scale vegetation-cloud interactions during drought using satellite data**. International Journal of Climatology, 593-623.
- [Che92] Chen Z. & Lee H.J. (1992). **Knowledge-guided visual perception of 3D human gait from a single image sequence**. IEEE Transactions on Systems, Man and Cybernetics, 22 (2), 336-342.
- [Che95] Chen E.J., Adler R.S., Carson P.L. & O'Brien W.D. (1995). **Ultrasound tissue displacement imaging with application to breast-cancer**. Ultrasound in Medicine and Biology, 21 (9), 1153-1162.
- [Coh91] Cohen L. (1991). **On active contour models and balloons**. Computer Vision, Graphics, and Image Processing: Image Understanding, 53 (2), 211-218.
- [Col92] Colet C., Quinquis A. & Boucher J.M. (1992). **Cloudy sky velocity estimation based on optical flow estimation leading with an entropy criterion**. ICPR-C, 160-163.
- [Coo92a] Cootes T.F., Taylor C.J., Cooper D.H. & Graham J. (1992). **Training models of shape from sets of examples**. British Machine Vision Conference, 9-18.
- [Coo92b] Cootes T.F. & Taylor C.J. (1992). **Active shape models – ‘smart snakes’**. British Machine Vision Conference, 276-285.
- [Coo93] Cootes T.F., Taylor C.J., Lanitis A., Cooper D.H. & Graham J. (1993). **Building and using flexible models incorporating grey-level information**. International Conference on Computer Vision, 242-246.
- [Coo94a] Cootes T.F., Hill A., Taylor C.J. & Haslam J. (1994). **Use of active shape models for locating structures in medical images**. Image and Vision Computing Journal, 12 (6), 355-365.
- [Coo94b] Cootes T.F. & Taylor C.J. (1994). **Combining point distribution models with shape models based on finite element analysis**. British Machine Vision Conference, vol. 2, 419-428.
- [Coo94c] Cootes T.F. & Taylor C.J. (1994). **Modelling object appearance using the grey-level surface**. En: British Machine Vision Conference, vol. 2, 423-433.
- [Coo94d] Cootes T.F., Taylor C.J. & Lanitis A. (1994). **Active shape models: Evaluation of multi-resolution method for improving image search**. British Machine Vision Conference, vol. 1, 327-336.
- [Cor91] Corral D. (1991). **VIEW: Computer vision for surveillance applications**. IEE Colloquium on Active and Passive Techniques for 3D Vision (Digest 045), IEE, Londres.
- [Cor96] Cornish M.T. & Wakefield J.P. (1996). **Automatically locating an area of interest and maintaining a reference image to aid the real-time tracking of objects**. Proceedings of the British Machine Vision Conference, 475-484.
- [Cur92] Curwen R. & Blake A. (1992). **Dynamic contours: Real-time active splines**. En: Blake A. & Yuille A.L., eds., *Active Vision*, cap. 3, 39-57. The MIT Press.
- [Dar98] Darrell T., Gordon G., Woodfill J. Baker H. & Harville M. (1998). **Robust, real-time people tracking in open environments using integrated stereo, color and face detection**. Proceedings of the IEEE Workshop on Visual Surveillance, 26-33.

- [Der84] Derrington A.M. & Lennie P. (1984). **Spatial and temporal contrast sensitivities of neurones in lateral geniculate nucleus of macaque.** *Journal of Physiology*, 357, 219-240.
- [Der90] Deriche R. & Faugeras O.D. (1990). **Tracking line segments.** *Image and Vision Computing*, 8 (4), 261-270.
- [Dic89] Dickinson K.W. & Wan C.L. (1989). **Road traffic monitoring using the TRIP II system.** 2nd International Conference on Road Traffic Monitoring, 56-60, London, IEE.
- [Dud73] Duda O.R. & Hart P.E. (1973). *Pattern Classification and Scene Analysis.* John Wiley and Sons, New-York.
- [Eme81] Emerson R.C. & Coleman L. (1981). **Does image movement have a special nature for neurons in the cat's striate cortex?** *Investigative Ophthalmology and Visual Science*, 20, 766-783.
- [Fah81] Fahle M. & Poggio T. (1981). **Visual hyperacuity: Spatiotemporal interpolation in human vision.** *Proceedings of the Royal Society of London B*, 213, 451-477.
- [Fat95] Fathy M. & Siyal M.Y. (1995). **A real-time image processing approach to measure traffic queue parameters.** *IEE Proceedings. Vision, image and signal processing*, 142(5), 297-303.
- [Fau87] Faugeras O.D., Lustman F. & Toscani G. (1987). **Motion and structure from motion from point and line matches.** *Proceedings of the 1st International Conference on Computer Vision*, 25-34.
- [Fau93] Faugeras O.D. (1993). *Three-dimensional Computer Vision: A Geometric Viewpoint.* The MIT Press, Cambridge, MA.
- [Fen79] Fennema C.L. & Thompson W.B. (1979). **Velocity determination in scenes containing several multiple moving objects.** *Computer Graphics and Image Processing*, 9, 301-315.
- [Fer92] Fernández M.A. & Mira J. (1992). **Permanence memory: A system for real time motion analysis in image sequences.** *MVA'92 IAPR Workshop on Machine Vision Applications*, 249-252. Tokyo.
- [Fer93] Fernández M.A. (1993). **Análisis de movimiento en secuencias de imagen.** En: *Notas de Visión y Apuntes sobre la Ingeniería del Software*, III Curso de Verano de Informática, Colección Estudios, 24, Universidad de Castilla-La Mancha, 99-110.
- [Fer95a] Fernández M.A., Mira J., López M.T., Alvarez J.R., Manjarrés A. & Barro S. (1995). **Local accumulation of persistent activity at synaptic level: Application to motion analysis.** En: Mira J. & Sandolval S., eds. *From Natural to Artificial Neural Computation*, IWANN'95, 137-143. Springer-Verlag, Germany.
- [Fer95b] Fernández M.A., Alonso M., Gómez F.J., Fernández Caballero A. (1995). **Técnicas y herramientas en VA: Redes Neuronales.** En: Fernández M.A. et al., eds., *Vida Artificial*, Colección Ciencia y Técnica, 10, Universidad de Castilla-La Mancha, 147-171.
- [Fer97] Fernández M.A. (1997). **Una arquitectura modular de inspiración biológica con capacidad de aprendizaje para el análisis de movimiento en secuencias de imagen en tiempo real.** Tesis doctoral. Tesis Doctorales, 48. Universidad de Castilla-La Mancha.
- [Fer98a] Fernández Caballero A., Lozano M.D. & Pons A. (1998). **La inspiración biológica de los modelos**

- computacionales de análisis en movimiento de imágenes.** En: Ensayos, 12, Universidad de Castilla-La Mancha, 293-303.
- [Fer98b] Fernández M.A., Fernández Caballero A., Moreno J & Sebastian G. (1998). **Tirano system with learning capacity for the detection of moving targets.** Proceedings of the International ICSC / IFAC Symposium on Neural Computaton NC'98, 311-317.
- [Fer98c] Ferryman J.M., Maybank S.J. & Worrall A.D. (1998). **Visual surveillance for moving vehicles.** Proceedings of the IEEE Workshop on Visual Surveillance, 73-80.
- [Fer99a] Fernández Caballero A., Montoya M., Fernández M.A. & Gómez F.J. (1999). **Base de Datos SkeleTTon (Manual de Usuario).** Technical Report GICONIA VisLab TR-002-99, Grupo de Investigación en Computación Neuronal e Inteligencia Artificial, Universidad de Castilla-La Mancha, Julio 1999.
- [Fer99b] Fernández Caballero A., Fernández M.A., Gómez F.J. & Montoya M. (1999). **SkeleTTon: Entorno de procesamiento de imágenes en tiempo real (Manual de Usuario).** Technical Report GICONIA VisLab TR-003-99, Grupo de Investigación en Computación Neuronal e Inteligencia Artificial, Universidad de Castilla-La Mancha, Noviembre 1999.
- [For94] Ford R.M., Stickland R.N. & Thomas B.A. (1994). **Image models for 2D flow visualization and compression.** CVGIP: Graphical Models and Image Processing, 56 (1), 75-93.
- [Fun96] Funkalea G. & Gupta A. (1996). **The use of hybrid models to recover cardiac wall-motion from MR-images.** Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 625-630.
- [Gel74] Gelb A. (1974). *Applied Optimal Estimation.* The MIT Press.
- [Gow75] Gower J.C. (1975). **Generalized procrustes analysis.** Psychometrika, 40, 33-51.
- [Gre91] Grenander U., Chow Y. & Keenan D.M. (1991). *Hands. A Pattern Theoretic Study of Biological Shapes.* Springer Verlag.
- [Gri81] Grimson W.E.L. (1981). *From Images to Surfaces: A Computational Study of the Human Early Visual System.* The MIT Press, Cambridge.
- [Gro94a] Groisman P.Y. & Easterling D.R. (1994). **Variability and trends of precipitation and snowfall over the United States and Canada.** Journal of Climate, 186-205.
- [Gro94b] Groisman P.Y. & Easterling D.R. (1994). **Precipitation changes over the Northern Hemisphere extratropics during the last one hundred years.** *Global Precipitation and Climate Change*, NATO ASI Series, V. I26, 119-133.
- [Guo94] Guo Y., Xu G. & Tsuji S. (1994). **Tracking of human body motion based on a stick figure model.** Journal of Visual Communication and Image Representation, 5 (1), 1-9.
- [Haa98] Haag M. & Nagel H.H. (1998). **Incremental recognition of traffic sequences.** Proceedings of the Workshop on Conceptual Description of Images, 1-20.
- [Has56] Hassenstein B. & Reichardt W.E. (1956). **Functional structure of a mechanism of perception of optical movement.** Proceedings of the 1st International Congress of Cybernetics in Namar, 797-801.

- [Has94] Haslam J., Taylor C.J. & Cootes T.F. (1994). **A probabilistic fitness measure for deformable template models.** British Machine Vision Conference, vol. 1, 33-42.
- [Haw84] Hawken M.J. & Parker A.J. (1984). **Contrast sensitivity and orientation selectivity in lamina IV of the striate cortex of old world monkeys.** Experimental Brain Research, 54, 367-372.
- [Heb49] Hebb D.O. (1949). *The Organization of Behavior.* Wiley, New York.
- [Hee87] Heeger D.J. (1987). **Model for the extraction of image flow.** Journal of the Optical Society of America A, 4 (8), 1455-1471.
- [Her92] Herlin I. & Ayache N. (1992). **Features extraction and analysis method for sequences of ultrasound images.** Image and Vision Computing Journal, 10 (10), 673-682.
- [Hil84] Hildreth E.C. (1984). *The Measurement of Visual Motion.* Cambridge. MIT Press.
- [Hil94] Hill A. & Taylor C.J. (1994). **Automatic landmark generation for point distribution models.** British Machine Vision Conference, vol. 2, 429-438.
- [Hog83] Hogg D. (1983). **Model-based vision: A program to see a walking person.** Image and Vision Computing, 1 (1), 5-20.
- [Hoo89] Hoose N. (1989). **Queue detection using computer image processing.** 2nd International Conference on Road Traffic Monitoring, 94-98, London, IEE.
- [Hop82] Hopfield J.J. (1982). **Neural networks and physical systems with emergent collective computational abilities.** Proceedings of the National Academy of Sciences of the USA, 79, 2254-2558.
- [Hor81] Horn B.K.P. & Schunck B.G. (1981). **Determining optical flow.** Artificial Intelligence 17, 185-203.
- [Hor86] Horn B.K.P. (1986). *Robot Vision.* The MIT Press, Cambridge, MA.
- [Hor91] Horowitz B. & Pentland A. (1991). **Recovery of non-rigid motion and structure.** Proceedings of the Computer Vision and Pattern Recognition Conference, Hawaii, 325-330.
- [Hua94a] Huang T., Koller D., Malik J., Ogasawara G., Rao B., Russel S. & Weber J. (1994). **Automatic symbolic traffic scene analysis using belief networks.** Proceedings of the 12th National Conference on Artificial Intelligence, 966-972.
- [Hua94b] Huang T.S. & Netravali A.N. (1994). **Motion and structure from feature correspondences: A review.** Proceedings of the IEEE, 82, 252-269.
- [Hub77] Hubel D.H. & Wiesel T.N. (1977). **Functional architecture of macaque monkey visual cortex.** Proceedings of the Royal Society of London, B, 198, 1-59.
- [Idr97] Idris F. & Panchanathan S. (1997). **Review of image and video indexing techniques.** Journal of Visual Communication and Image Representation, 8 (2), 146-166.
- [Ini85] Inigo R.M. (1985). **Traffic monitoring and control using machine vision: A survey.** IEEE Transactions on Industrial Electronics, 32(3), 177-185.
- [Ini89] Inigo R.M. (1989). **Application of machine vision to traffic monitoring and control.** IEEE Transactions on Vehicular Technology, 38(3), 112-122.

- [Ira93] Irani M. & Peleg S. (1993). **Motion analysis for image enhancement: Resolution, occlusion and transparency.** Journal of Visual Communication and Image Representation, 4, 324-335.
- [Jai79] Jain R., Martin W.N. & Aggarwal J.K. (1979). **Segmentation through the detection of changes due to motion.** Computer Graphics and Image Processing, 11, 13-34.
- [Jai81a] Jain J.R. & Jain A.K. (1981). **Displacement measurement and its application in interframe image coding.** IEEE Transactions on Communications, 29 (12), 1799-1808.
- [Jai81b] Jain A.K. (1981). **Image data compression: A review.** Proceedings of the IEEE, 69, 349-389.
- [Jai89] Jain A.K. (1989). *Fundamentals of Digital Image Processing.* Prentice-Hall.
- [JCS99] Japanese Cloud-Climate Study, <http://mriim1.mri-jma.go.jp/Proj/JACCS/about.html>.
- [Kar90] Karl T.R. & Steurer P.M. (1990). **Increased cloudiness in the United States during the first half of the twentieth century: Fact or fiction?** Geophysical Research Letters, 17, 1925-1928.
- [Kar95] Karl T.R., Knight R.W., Easterling D.R. & Quayle R.G. (1995). **Trends in U.S. climate in the twentieth century.** Consequences, V. 1, 2-12.
- [Kas87] Kass M., Witkin A. & Terzopoulos D. (1987). **Snakes: Active contour models.** Proceedings of the 1st International Conference on Computer Vision, 259-268, London. IEEE Computer Society Press.
- [Kas88] Kass M., Witkin A. & Terzopoulos D. (1988). **Snakes: Active contour models.** International Journal of Computer Vision, 1 (4), 321-331.
- [Kel92] Kelly D.M. (1992). **Results of field trial of the IMPACTS image processing system for traffic monitoring.** 6th Interbational Conference on Road Traffic Monitoring, 137-142, London, IEE.
- [Ker94a] Kervrann C. & Heitz F. (1994). **A hierarchical framework for the segmentation of deformable objects in image sequences.** Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, 724-728.
- [Ker94b] Kervrann C. & Heitz F. (1994). **Robust tracking of stochastic deformable models in long image sequences.** IEEE International Conference on Image Processing, vol. 3, 88-92, IEEE Computer Society Press.
- [Koc86] Koch C., Marroquin J. & Yuille A. (1986). **Analog "neuronal" networks in early vision.** Proceedings of the National Academy of Sciences USA 83, 4263-4267.
- [Koh77] Kohonen T. (1977). *Associative Memory: A System-Theoretical Approach.* Springer-Verlag.
- [Kol94a] Koller D., Weber J. & Malik J. (1994). **Robust multiple car tracking with occlusion reasoning.** European Conference on Computer Vision.
- [Kol94b] Koller D., Weber J., Huang T., Malik J., Ogasawara G., Rao B. & Russel S. (1994). **Towards robust automatic traffic scene analysis in real-time.** International Conference on Pattern Recognition, 126-131, Jerusalem, Israel.
- [Kua88] Kuan D., Phipps G. & Hsueh A.C. (1988). **Autonomous robotic vehicle road following.** IEEE Transactions on Pattern Analysis and machine Intelligence, 10 (5), 648-658.

- [Lan94] Lanitis A., Taylor C.J. & Cootes T.F. (1994). **An automatic face identification system using flexible appearance models.** En: British Machine Vision Conference, vol. 1, 65-74.
- [Lap82] Lappin J.S. & Fuqua M. (1982). **Non-linear recruitment in the visual detection of moving patterns.** Investigative Ophthalmology and Visual Science Supplement 22, 123.
- [Lar97] Large K. (1997). **Sets in space.** Television Buyer, 32-37, Septiembre 1997.
- [Lau94] Lau N-C. & Crane M. (1994). **A satellite view of the synoptic-scale organization of cloud cover in midlatitude and tropical circulation systems.** Monthly Weather Review, 123, 1984-2006.
- [Law89] Lawton T.B. (1989). **Outputs of paired Gabor filters summed across the background frame of reference predict the direction of movement.** IEEE Transactions of Biomedical Engineering, 36, 130-139.
- [LeG91] Le Gall D. (1991). **MPEG: A video compression standard for multimedia applications.** ACM Communications, 34 (4).
- [Lew93] Lewis F., Abdallah C. & Dawson D. (1993). *Control of Robot Manipulators.* Macmillan.
- [Ley93] Leymarie F. & Levine M.D. (1993). **Tracking deformable objects in the plane using an active contour model.** IEEE Transactions on Pattern Analysis and Machine Intelligence, 15 (6), 617-634.
- [Li93] Li H.B., Roivainen P. & Forchheimer R. (1993). **3-D motion estimation in model-based facial image coding.** IEEE Transactions on Pattern Analysis and Machine Intelligence, 15 (6), 545-555.
- [Li94] Li H.B., Lundmark A. & Forchheimer R. (1994). **Image sequence coding at very-low bitrates – A review.** IEEE Transactions on Image Processing, 3 (5), 589-609.
- [Lim75] Limb J.O. & Murphy J.A. (1975). **Estimating the velocity of moving images in television signals.** Computer Graphics and Image Processing, 4, 311-327.
- [LiQ92] Li-Qun X., Young D. & Hogg D. (1992). **Building a model of a road junction using vehicle information.** British Machine Vision Conference, 443-452.
- [Liu88] Liu Y. & Huang T.S. (1988). **A linear algorithm for determining motion and structure from line correspondences.** CVGIP, 44 (1), 35-57.
- [Liu89] Liu Y. & Huang T.S. (1989). **Motion estimation from corner correspondences.** Proceedings of the International Conference on Image Processing ICIP'89, 785-790.
- [Liv88] Livingstone M. & Hubel D. (1988). **Segregation of form, color, movement and depth: anatomy, physiology and perception.** Science, 240, 740-749.
- [Low91] Lowe D.G. (1991). **Fitting parameterized three dimensional models to images.** IEEE Transactions on Pattern Analysis and Machine Intelligence, 441-450.
- [Mai89] Mailloux G.E., Langlois F., Simard P.L. & Bertrand M. (1989). **Restoration of the velocity field of the heart from two-dimensional echocardiograms.** IEEE Transactions on Medical Images, 8 (2), 143-153.
- [Mal95] Malik J., Weber J., Luong Q.T. & Koller D. (1995). **Smart cars and smart roads.** Proceedings of the 6th British Machine Vision Conference, 367-381, Birmingham, England.

- [Mar78] Marr D. & Nishihara H.K. (1978). **Representation and recognition of the spatial organisation of three-dimensional shapes.** Proceedings of the Royal Society of London, B, 269-294.
- [Mar81] Marr D. & Ullman S. (1981). **Directional selectivity and its use in early visual processing.** Proceedings of the Royal Society of London, B, 211, 151-180.
- [Mar82] Marr D. (1982). *Vision.* Freeman & Co., New York.
- [Mau95] Maurizot M., Boutheimy P., Delyon B., Iouditski A. & Odobez J.M. (1995). **Determination of singular points in 2D deformable flow fields.** Proceedings of the 2nd IEEE International Conference on Image Processing, Washington DC.
- [McC74] McCann J.J., Savoy R.L., Hall J.A. & Scarpetti J.J. (1974). **Visibility of continuous luminance gradients.** Vision Research, 14, 917-927.
- [McC86] McClelland J.L. & Rumelhart D.E. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 2.* The MIT Press, Cambridge, MA.
- [McK85] McKee S.P. & Welch L. (1985). **Sequential recruitment in the discrimination of velocity.** Journal of the Optical Society of America A, 2, 243-251.
- [Met94] Metier-Di Nunzio, C. (1994). **Un robot qui voit comme une mouche.** Cybernétique, Science & Vie, 918, 62-67.
- [Mey94] Meyer F.G. & Boutheimy P. (1994). **Region-based tracking using affine motion models in long image sequences.** Computer Vision, Graphics and Image Processing: Image Understanding, 60 (2), 119-140.
- [Mir82] Mira J. & Moreno-Diaz R. (1982). **Un marco teórico para interpretar la función neuronal a alto nivel.** I Reunión Nacional de Biocibernética, 151-179, Madrid.
- [Mir93a] Mira J., Delgado A.E., Alvarez J.R., De Madrid A.P. & Santos M. (1993). **Towards more realistic self contained models of neurons: High- order, recurrence and local learning.** En Mira J, ed. *New Trends in Neural Computation*, 55-62, London.
- [Mir93b] Mira J. (1993). **Computación neural en el camino visual.** En: *Notas de Visión y Apuntes sobre Ingeniería del Software*, Colección Estudios, 24, UCLM, Albacete, 175-197.
- [Mir95] Mira J., Delgado A.E., Boticario J.G. & Díez F.J. (1995). *Aspectos básicos de la Inteligencia Artificial.* Editorial Sanz y Torres, S.L. Madrid.
- [Mir96] Mira J., Delgado A.E., Manjarres A., Ros S. & Alvarez J.R. (1996). **Cooperative processes at the symbolic level in cerebral dynamics: Reliability and fault tolerance.** En: R. Moreno-Diaz and J. Mira, eds. *Brain Processes, Theories and Models*, The MIT Press, Cambridge, MA, 244-255.
- [Mis91] Mishra S.K., Goldgof D.B. & Huang T.S. (1991). **Motion analysis and epicardial deformation estimation from angiography data.** Proceedings of the Conference on Computer Vision and Pattern Recognition, 331-336.
- [Mit96] Mitiche A. & Boutheimy P. (1996). **Computation and analysis of image motion: A synopsis of current problems and methods.** International Journal of Computer Vision, 19 (1), 29-55.
- [Mon99a] Montoya M., Fernández Caballero A., Fernández M.A. & Gómez F.J. (1999). **Base de Datos**

- SkeleTTON para la gestión de secuencias de imágenes.** Technical Report GICONIA VisLab TR-001-99, Grupo de Investigación en Computación Neuronal e Inteligencia Artificial, Universidad de Castilla-La Mancha, Abril 1999.
- [Mon99b] Montoya M., Fernández M.A., Fernández Caballero A., & López M.T. & Gómez F.J. (1999). **Módulo de manejo del magnetoscopio SONY BETACAM-SP modelo UVW-1700GP (Descripción General).** Technical Report GICONIA VisLab TR-004-99, Grupo de Investigación en Computación Neuronal e Inteligencia Artificial, Universidad de Castilla-La Mancha, Noviembre 1999.
- [Mon99c] Montoya M., Fernández M.A., Fernández Caballero A., & López M.T. & Gómez F.J. (1999). **MAGNETO: Módulo de manejo del magnetoscopio SONY BETACAM-SP modelo UVW-1700GP (Análisis).** Technical Report GICONIA VisLab TR-005-99, Grupo de Investigación en Computación Neuronal e Inteligencia Artificial, Universidad de Castilla-La Mancha, Noviembre 1999.
- [Mon99d] Montoya M., Fernández M.A., Fernández Caballero A., & López M.T. & Gómez F.J. (1999). **MAGNETO: Módulo de manejo del magnetoscopio SONY BETACAM-SP modelo UVW-1700GP (Diseño).** Technical Report GICONIA VisLab TR-006-99, Grupo de Investigación en Computación Neuronal e Inteligencia Artificial, Universidad de Castilla-La Mancha, Diciembre 1999.
- [Mon00a] Montoya M., Fernández M.A., Fernández Caballero A., Gómez F.J. & López M.T. (2000). **Especificaciones técnicas del magnetoscopio SONY BETACAM-SP modelo UVW-1700GP.** Technical Report GICONIA VisLab TR-002-2000, Grupo de Investigación en Computación Neuronal e Inteligencia Artificial, Universidad de Castilla-La Mancha, Enero 2000.
- [Mon00b] Montoya M., Fernández Caballero A., Fernández M.A., Gómez F.J. & López M.T. (2000). **Módulo de Dispensación de Imágenes (Descripción General).** Technical Report GICONIA VisLab TR-003-2000, Grupo de Investigación en Computación Neuronal e Inteligencia Artificial, Universidad de Castilla-La Mancha, Enero 2000.
- [Mon00c] Montoya M., Fernández Caballero A., Fernández M.A., Gómez F.J. & López M.T. (2000). **Módulo de Dispensación de Imágenes (Definición del Problema).** Technical Report GICONIA VisLab TR-004-2000, Grupo de Investigación en Computación Neuronal e Inteligencia Artificial, Universidad de Castilla-La Mancha, Enero 2000.
- [Mon00d] Montoya M., Fernández M.A., Fernández Caballero A., Gómez F.J. & López M.T. (2000). **Módulo de Dispensación de Imágenes (Especificaciones Funcionales).** Technical Report GICONIA VisLab TR-005-2000, Grupo de Investigación en Computación Neuronal e Inteligencia Artificial, Universidad de Castilla-La Mancha, Febrero 2000.
- [Mon00e] Montoya M., Fernández M.A., Fernández Caballero A., Gómez F.J. & López M.T. (2000). **Módulo de Dispensación de Imágenes (Diseño).** Technical Report GICONIA VisLab TR-006-2000, Grupo de Investigación en Computación Neuronal e Inteligencia Artificial, Universidad de Castilla-La Mancha, Febrero 2000.
- [Mor69] Moreno-Diaz R., Rubio F. & Mira J. (1969). **Aplicación de las transformaciones integrales al proceso de datos en la retina.** Revista de Automática, 5, 7-17.

- [Mur93] Murphy N., Byrne N. & O'Leary K. (1993). **Long sequence analysis of human motion using eigenvector decomposition.** Proceedings of SPIE, Septiembre 1993.
- [Nag88] Nagel H.H. (1988). **From image sequences towards conceptual descriptions.** Image and Vision Computing, 6 (2): 59-74.
- [Nai98] Nair D. & Aggarwal J.K. (1998). **Moving object detection from a navigating robot.** IEEE Transactions on Robotics and Automation, 14 (3), 404-416.
- [Nak83] Nakayama K. & Silverman G.H. (1983). **Perception of moving sinusoidal lines.** Journal of the Optical Society of America A, 72.
- [Nak84] Nakayama K. & Silverman G.H. (1984). **Temporal and spatial characteristics of the upper displacement limit for motion in random dots.** Vision Research 24, 293-299.
- [Nas94a] Nastar C. (1994). **Vibration modes for nonrigid analysis in 3D images.** European Conference on Computer Vision, vol. 1, 231-238.
- [Nas94b] Nastar C. & Ayache N. (1994). **Spatio-temporal analysis of nonrigid motion from 4D data.** IEEE Workshop on Motion of Non-rigid and Articulated Objects, 146-151, IEEE Computer Society Press.
- [Nic93] Nicchiotti G. & Ottaviani E. (1993). **Automatic vehicle counting from image sequences.** Proceedings of the 4th International Workshop on Time-Varying Image Processing and Moving Object Recognition, Elsevier Science.
- [Niy94] Niyogi S. & Adelson E. (1994). **Analyzing gait with spatiotemporal surfaces.** En: IEEE Workshop on Motion of Non-rigid and Articulated Objects, 64-69. IEEE Computer Society Press.
- [Pea88] Pearl J. (1988). *Probabilistic Reasoning Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann.
- [Pea95] Pearson D.E. (1995). **Developments in model-based video coding.** Proceedings of the IEEE, 83 (6), 892-906.
- [Pen90] Pentland A.P. **Automatic extraction of deformable part models.** International Journal of Computer Vision, 4, 107-126.
- [Pen91a] Pentland A.P. & Sclaroff S. (1991). **Closed-form solutions for physically-based shape modeling and recognition.** IEEE Transactions on Pattern Analysis and Machine Intelligence, 13 (7), 715-729.
- [Pen91b] Pentland A.P. & Horowitz B. (1991). **Recovery of non-rigid motion and structure.** IEEE Transactions on Pattern Analysis and Machine Intelligence, 13 (7), 730-742.
- [Pet95] Peterson T.C., Golubev V.S. & Groisman P.Y. (1995). **Evaporation losing its strength.** Nature, 377, 687-688.
- [Pog85] Poggio T., Torre V. & Koch C. (1985). **Computational vision and regularisation theory.** Nature, 317, 314-319.
- [Pon98a] Pons Penedés A., Fernández M.A., Fernández Caballero A., Gómez F.J. & Sebastian G. (1998). **BRONTO: Interfaz de usuario para un sistema de detección de blancos móviles (Descripción).** Technical Report GICONIA VisLab TR-002-98, Grupo de Investigación en Computación Neuronal e

Inteligencia Artificial, Universidad de Castilla-La Mancha, Marzo 1998.

- [Pon98b] Pons Penedés A., Fernández M.A., Fernández Caballero A., Gómez F.J. & Sebastian G. (1998). **BRONTO: Interfaz de usuario para un sistema de detección de blancos móviles (Diseño)**. Technical Report GICONIA VisLab TR-003-98, Grupo de Investigación en Computación Neuronal e Inteligencia Artificial, Universidad de Castilla-La Mancha, Mayo 1998.
- [Pon98c] Pons Penedés A., Fernández M.A., Fernández Caballero A., Gómez F.J. & Sebastian G. (1998). **BRONTO: Interfaz de usuario para un sistema de detección de blancos móviles (Manual de Usuario)**. Technical Report GICONIA VisLab TR-004-98, Grupo de Investigación en Computación Neuronal e Inteligencia Artificial, Universidad de Castilla-La Mancha, Junio 1998.
- [Pra79] Prager J.M. (1979). **Segmentation of static and dynamic scenes**. COINS Technical Report 79-7. University of Massachusetts.
- [Pra83] Prager J.M. & Arbib M.A. (1983). **Computing the optic flow: The MATCH algorithm and prediction**. Computer Vision, Graphics and Image Processing, 24, 271-304.
- [Pri92] Prince J.L. & McVeigh E.R. (1992). **Motion estimation from tagged MR image sequences**. IEEE Transactions on Medical Imaging, 11, 238-249.
- [PSC99] PSC Meteorology Program Cloud Boutique, <http://vortex.plymouth.edu/cloud.html>.
- [Qui97] Quistgaard J.U. (1997). **Signal acquisition and processing in medical diagnostic ultrasound**. IEEE Transactions on Medical Imaging, 11, 238-249.
- [Rao92] Rao A.R. & Jain J.C. (1992). **Computerized flow field analysis: Oriented texture fields**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14 (7), 693-709.
- [Rat96] Ratib O., Didier D., Chretien A., Rosset A., Magnin A. & Ligier Y. (1996). **Temporal analysis of regional wall-motion from cine cardiac MRI**. Proceedings of the SPIE, 2710, 71-80.
- [Roh93] Rohr K. (1993). **Incremental recognition of pedestrians from image sequence**. Computer Vision and Pattern Recognition, 8-13.
- [Roh94] Rohr K. (1994). **Toward model-based recognition of human movements in image sequences**. Computer Vision, Graphics, and Image Processing: Image Understanding, 59 (1), 94-115.
- [Ros58] Rosenblatt F. (1958). **The perceptron: A probabilistic model for information storage and organization in the brain**. Psychological Review, 65, 386-408.
- [Ros94] Rossi M. & Bozzoli A. (1994). **Tracking and counting moving people**. ICIP Proceedings, vol. III, 212-216.
- [Row95] Rowe S. & Blake A. (1995). **Statistical background modelling for tracking with a virtual camera**. En: British Machine Vision Conference, vol. 2, 423-433.
- [Rum86] Rumelhart D.E. & McClelland J.L. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1*. The MIT Press, Cambridge, MA.
- [Ryd87] Rydfalk M. (1987). **CANDIDE: A parameterised face**. Technical report, Department of Electrical Engineering, Linköping, Suecia, Octubre 1987.

- [Sc193] Sclaroff S. & Pentland A. (1993). **A modal framework for correspondence and description.** Proceedings of the 4th International Conference on Computer Vision, 308-313.
- [Sc194] Sclaroff S. & Pentland A. (1994). **On modal modeling for medical images: Underconstrained shape description and data compression.** Proceedings of the IEEE Workshop on Biomedical Image Analysis.
- [Ser89] Sereno M.I. (1989). **Learning the solution to the aperture problem for pattern motion with a Hebb rule.** Advances in Neural Information Processing Systems, 2, Morgan-Kaufman.
- [Ser93] Sereno M.E. (1993). *Neural Computation of Pattern Motion.* The MIT Press, Cambridge, MA.
- [She82] Shephard M.J., Richards V.A., Berkowitz R.L. & Hobbins, J.C. (1982). **An evaluation of two equations for predicting fetal weight by ultrasound.** American Journal of Obstetrics and Gynecology, 142-147.
- [Sil77] Sillito A.M. (1977). **Inhibitory processes underlying the directional specificity of simple, complex and hypercomplex cells in the cat's visual cortex.** Journal of Physiology, 271, 699-720.
- [Sod94] Soden B.J. & Donner L.J. (1994). **Evaluation of a GCM cirrus parameterization using satellite observations.** Journal of Geophysical Research, 99, 14401-14413.
- [Sod96] Soden B.J. & Fu R. (1996). **A satellite analysis of deep convection, upper tropospheric humidity and the greenhouse effect.** Journal of Climate, 8(10), 2333-2351.
- [Son93] Sonka M., Hlavac V. & Boyle R. (1993). *Image Processing, Analysis and Machine Vision.* Chapman & Hall.
- [Ste85] Sternberg S.R. (1985). **Vision guided robot for automotive assembly.** En: Proceedings of the IEEE International Conference on Robotics and Automation, 686-690.
- [Tak82] Takaba S, et. al. (1982). **Measurement of traffic flow using real-time processing of moving pictures.** 32nd Conference on Vehicular Technology, 488-494, San Diego, CA.
- [Tek95] Tekalp A.M. (1995). *Digital Video Processing.* Prentice Hall.
- [Ter87a] Terzopoulos D., Platt J., Barr A. & Fleischer K. (1987). **Elastically deformable models.** ACM Computer Graphics, 4 (21), 205-214.
- [Ter87b] Terzopoulos D., Witkin A. & Kass M. (1987). **Symmetry-seeking models for 3-D object reconstruction.** International Journal of Computer Vision, 1 (3), 211-221.
- [Ter88] Terzopoulos D. & Witkin A. (1988). **Deformable models. Physically based models with rigid and deformable components.** IEEE Computer Graphics and Applications, 8, 41-51.
- [Ter91] Terzopoulos D. & Metaxas D. (1991). **Dynamic 3D models with local and global deformations: Deformable superquadrics.** IEEE Transactions on Pattern Analysis and machine Intelligence, 13 (7), 703-714.
- [Ter92] Terzopoulos D. & Szeliski R. (1992). **Tracking with Kalman snakes.** En: Blake A. & Yuille A.L., eds., *Active Vision*, 3-20. The MIT Press.

- [Tho81] Thompson W.B. & Barnard S.T. (1981). **Lower-level estimation and interpretation of visual motion.** IEEE Computer 14, 20-28.
- [Tik77] Tikhonov A.N. & Arsenin V.Y. (1977). *Solutions of ill-posed problems.* Washington D.C. Ed. Winston.
- [Tru98] Trucco E. & Verri A. (1998). *Introductory Techniques for 3-D Computer Vision.* Prentice Hall.
- [Tur91] Turk M.A. & Pentland A. (1991). **Face recognition using eigenfaces.** Proceedings of CVPR, 586-591.
- [Uka93] Ukai M. (1993). **Extracting the rail vanishing point from camera view control.** Quarterly Report of the Railway Technical Research Institute, Tokio, 34 (1), 72-77.
- [Val88] de Valois R.L. & de Valois K.K. (1988). *Spatial Vision.* New York. Ed. Oxford University Press.
- [Ver89] Versavel J., Lemaire F. & van der Stede D. (1989). **Camera and computer-aided traffic sensor.** 2nd International Conference on Road Traffic Monitoring, 66-70, London, IEE.
- [Vla96] Vlachos T. & Thomas G. (1996). **Motion estimation for the correction of twin-lens telecine flicker.** Proceedings of the IEEE Conference on Image Processing, vol. 1, Lausana, Suiza.
- [Wal76] Wallach H. (1976). **On perceived identity: 1. The direction of motion of straight lines.** In H.Wallach, ed. *On perception.* New York. Ed. Quadrangle.
- [War77] Warsof S.L., Gohari P., Berkowitz R.L. & Hobbins, J.C. (1977). **The estimation of fetal weight by computer-assisted analysis.** American Journal of Obstetrics and Gynecology, 128 (8), 881-892.
- [Wat85] Watson A.B. & Ahumada A.J. (1985). **Model of human visual motion sensing.** Journal of the Optical Society of America A, 2 (2), 322-341.
- [Wen92] Weng J., Ahuja N. & Huang T.S. (1992). **Motion and structure from line correspondences: Closed-form solution, uniqueness and optimization.** IEEE Transactions on Pattern Analysis and Machine Intelligence, 14 (3), 318-336.
- [Wet97] Wettergreen D., Thomas H. & Bualat M. (1997). **Initial results from vision-based control of the Ames Marsokhod rover.** Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Grenoble, Francia.
- [Wid60] Widrow B. & Hoff M.E. (1960). **Adaptive switching circuits.** 1960 IRE WESCON Convention Record, 96-104, New York.
- [Woo00] Woo J.S.K. (2000). **Obstetric ultrasound: A comprehensive guide to ultrasound scans in pregnancy.** <http://www.ultrasound.net/>
- [Yui88] Yuille A.L. & Grzywacz N. (1988). **A computational theory for the perception of coherent visual motion.** Nature 333, 71-74.
- [Yui92] Yuille A.L. (1992). **Feature extraction from faces using deformable templates.** International Journal of Computer Vision, 8 (2), 90-111.