

Doctoral Dissertation

A Model for Normality Analysis of Events and Behaviours in Monitored Environments: Application to Video Surveillance



University of Castilla-La Mancha

1 December 2009

Author

Javier Alonso Albusac Jiménez
Computer Engineer

Supervisors

Dr. José Jesús Castro Sánchez - Dr. Luis Jiménez Linares

Tesis Doctoral

Modelo para el Análisis de la Normalidad de Eventos y Conductas en Entornos Monitorizados: Aplicación a la Vídeo Vigilancia.



Universidad de Castilla-La Mancha

1 Diciembre 2009

Autor

Javier Alonso Albusac Jiménez

Ingeniero en Informática

Directores

Dr. José Jesús Castro Sánchez - Dr. Luis Jiménez Linares

Autor:
JAVIER ALONSO ALBUSAC JIMÉNEZ

Directores:
DR. JOSÉ JESÚS CASTRO SÁNCHEZ
DR. LUIS JIMÉNEZ LINARES

LICENCIA:
©Javier Alonso Albusac Jiménez. Se permite la copia y distribución de la totalidad o parte de este documento sin ánimo de lucro. Toda copia total o parcial deberá citar expresamente el nombre del autor, de la Universidad de Castilla-La Mancha y deberá incluir esta misma licencia, añadiendo, si es copia literal, la mención "*copia literal*".

Se autoriza la modificación y traducción de la obra sin ánimo de lucro siempre que se haga constar en la obra resultante de la modificación el nombre de la obra originaria, el autor de la obra originaria y el nombre de la Universidad de Castilla-La Mancha. La obra resultante también deberá ser libremente reproducida, distribuida, comunicada al público y transformada en términos similares a los expresados en esta licencia.

Este documento fue maquetado con $\text{\LaTeX}$. Imágenes generadas con Blender/Yafray, compuestas con Gimp y OpenOffice.

Abstract

Since their first developments, traditional video surveillance systems have been designed to monitor environments. However, these systems have several limitations to automatically understand events and behaviours without human collaboration. In order to overcome this problem, intelligent surveillance systems arise as a possible solution. This kind of systems are not affected by negative factors such as fatigue or tiredness and they can be more effective than people when recognising certain kinds of events, such as the detection of suspicious or unattended objects.

Intelligent surveillance refers to using Artificial Intelligence and Computer Vision techniques in order to improve traditional surveillance and process semantic information, obtained from low-level security devices. Normally these systems consist of a set of independent analysis modules that deal with particular problems, such as the analysis of trajectories in parking lots, speed estimation of vehicles, gait or facial recognition, etc. However, most of them present a common problem: lack of flexibility and scalability to include new kinds of analysis and combine all of them in order to obtain a global interpretation.

In this thesis, a formal model to define normal events and behaviours in monitored environments and to build scalable surveillance systems is presented. This model is based on the use of “normality components”, which are independent and reusable for environments with different characteristics and different kinds of objects. Each component specifies how an object should ideally behave according to a surveillance aspect, such as trajectory or speed. The model also includes the fusion mechanisms required for combining the particular analysis made by each component. Finally, when a new component is designed making use of the proposed model, the system increases its capability to detect new kind of abnormal events, and the normality of an object depends on a higher number of factors.

Resumen

A lo largo de los últimos años, los sistemas de vigilancia han sido utilizados como herramientas para monitorizar y mantener la seguridad en entornos vigilados. El principal problema de los sistemas tradicionales es la incapacidad para interpretar por sí mismos los eventos que se producen y los comportamientos llevados a cabo por los individuos que realizan acciones en el entorno.

Como solución a este problema surgen los sistemas de seguridad inteligentes, que utilizan técnicas pertenecientes al área de la Inteligencia Artificial y la Visión por Computador para interpretar lo que sucede en un entorno monitorizado a partir de la información procedente de los sensores. La mayoría de estos sistemas se caracterizan por estar formados por diferentes módulos de análisis diseñados de forma independiente sin seguir un patrón común. Estos sistemas pueden entenderse como un conjunto de piezas que actúan por separado para intentar detectar tipos específicos de eventos en entornos muy concretos, sin que estos análisis se relacionen entre sí para obtener valoraciones globales. Por tanto, surge la necesidad de proponer mecanismos para la construcción de sistemas de monitorización flexibles y escalables, que faciliten la inclusión de nuevos módulos de análisis sin que implique modificaciones en los existentes y, que a su vez, pueda ser fácilmente combinado con el resto para incrementar la capacidad de análisis global del propio sistema.

Ante esta necesidad, en esta Tesis se propone un modelo formal para la construcción de sistemas de monitorización flexibles y escalables. El modelo está basado en el uso de “componentes de normalidad”, independientes y reutilizables para entornos con diferentes características. Cada componente de normalidad especifica como debe comportarse un objeto de acuerdo a un aspecto a vigilar, como por ejemplo, las trayectorias o las velocidades con las que circulan los objetos detectados por el sistema. El modelo incluye además mecanismos de fusión para combinar los análisis realizados por cada uno de los componentes que han sido diseñados mediante el modelo propuesto, obteniendo así un análisis global del estado actual del entorno monitorizado.

Acknowledgements

Agradecimientos

Acknowledgements

I would like to thank all the people who have helped me during my stay at School of Computing in Leeds (England). I would like to express my sincere gratitude to David Hogg for his supervision during my stay in Leeds. He has been always there with stimulation discussions and new suggestions. Special thanks also to all members of the computer vision group for their comments and advices at the journal club, which have improved the quality of this work. Finally, I would like to thank my officemates and colleagues there; Dima Damen, John Greenall, Sandeep Reddy, and Ardhendu Behera for giving me their encouragement, enthusiasm, and friendship. They made my stay in Leeds a wonderful experience. I won't never forget it!

Agradecimientos

En primer lugar me gustaría agradecer a mis directores José Jesús Castro Sánchez y Luis Jiménez Linares su esfuerzo y dedicación a este trabajo. Mil gracias por guiarme durante estos años, por tantas horas de debates realmente gratificantes y, sobre todo, por animarme siempre que lo he necesitado en momentos de dudas. Gracias por vuestra amistad.

Gracias a la Universidad de Castilla-La Mancha por la financiación de mi estancia de investigación en la Universidad de Leeds, la cual ha servido para mejorar la calidad de este trabajo. Gracias a la Escuela Superior de Informática de Ciudad Real y el Departamento de Tecnologías y Sistemas de Información por proporcionar todos los recursos necesarios para el desarrollo de este trabajo. Este trabajo también ha sido posible gracias a la financiación de los proyectos PBC-06-0064, PAC-06-0141 y Hesperia: Homeland Security: tecnologías para la seguridad integral en espacios públicos e infraestructuras (proyecto CENIT).

En estos agradecimientos no pueden faltar mis compañeros del grupo de investigación Oreto a los que respeto y admiro. Pepe, mil gracias por cada uno de los segundos dedicados a este trabajo y perdidos con ¡Samuelillo!. Muchas gracias “Luisji” por tu confianza en mí desde el primer día (¡a veces demasiada!), por saber exigir y valorar el trabajo al mismo tiempo. David Vallejo con el que comencé este camino y juntos también lo acabaremos. Ha sido una suerte que nuestros trabajos estuvieran relacionados y espero que este trabajo sólo sea el comienzo. Muchas gracias también por tus revisiones y tu ayuda en la traducción a inglés. A Morcy, gracias por abrirme las puertas del grupo Oreto. Sin tu confianza hace cuatro años, quizás hoy no estaría escribiendo estas líneas de agradecimiento. Gracias también por contagiarme tus ganas de trabajar y entusiasmo en todo lo que haces (¡y por la plantilla y alguna imagencilla!). Muchas gracias “Luisro” por facilitarme la adaptación a la Escuela Politécnica de Almadén donde actualmente disfruto impartiendo clase y por esas mañanas de bicicleta que me han permitido desconectar un poco. Y al resto de miembros: Nines, Cayetano, Manuel, Raúl, Jose Carlos, Trinchier, Ester, Carmen y los que ya no están “Lonren”, blankito ... a todos gracias!.

Mil gracias serían pocas para mis cinco padres: Mamá, Papá, Manuel, Paco e Isabel, porque ellos se alegran más que yo de las metas que consigo. Gracias por vuestra confianza, amor y apoyo. En especial me gustaría dar las gracias a mi madre por contagiarme sus ganas de aprender y por cada uno de los segundos que me dedicó de su tiempo en mi infancia cuidándome y educándome, aún

sacrificando su carrera profesional. Soy lo que soy en gran parte gracias a ella, GRACIAS MAMA.

No puedo finalizar de otra forma que no sea dando las gracias a Vanesa, a quién está dedicada esta Tesis. Mi compañera de viaje con la que crezco desde los 16 años y descubro nuevas cosas día a día. ¡Qué rápido han pasado estos 13 años, quiero muchos más!. Gracias por estar a mi lado siempre, sin ti me faltaría el aire. Gracias por haberme animado siempre a realizar este trabajo a pesar de sufrir mi ausencia en muchos momentos. Espero que cumplas todos tus deseos.

- *Los caballos no bailan claquet* -

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	5
1.3. Contenidos	7
2. Estado del Conocimiento	11
2.1. Evolución de los sistemas de vigilancia	12
2.2. Medios técnicos de seguridad	17
2.3. Capas de un Sistema de Vigilancia Inteligente	20
2.4. Modelado de entornos reales	23
2.4.1. Ontologías	24
2.4.2. Relaciones espaciales	27
2.4.3. Relaciones temporales entre eventos	31
2.5. Adquisición de conocimiento en sistemas de vigilancia inteligente	34
2.6. Proceso de segmentación	38
2.7. Clasificación de objetos	42
2.8. Seguimiento de objetos mediante técnicas de tracking	44
2.9. Interpretación de eventos y conductas	46
2.9.1. Dynamic time warping (DTW)	48
2.9.2. Máquinas de estados finitos	49
2.9.3. Gramáticas libres de contexto y lógicas inductivas	50
2.9.4. Algoritmos de <i>Clustering</i>	52
2.9.5. Redes Neuronales	53
2.9.6. Redes Bayesianas	54
2.9.7. Modelos Ocultos de Markov	56
2.9.8. Modelos difusos	57
2.9.9. Otros métodos	59
2.10 Fusión de Información	63
2.11 Ámbitos de aplicación	65
2.12 Proyectos de investigación	67
3. Modelo Teórico Formal	75
3.1. Esquema del diseño propuesto	76
3.2. Normalidad en entornos monitorizados	81
3.2.1. Problema de la vigilancia: divide y vencerás	82
3.2.2. Entornos monitorizados	83
3.2.3. Vigilancia basada en el análisis de conceptos	85
3.2.4. Restricción de normalidad	85
3.2.5. Instancia de un concepto	87
3.2.6. Instancia de una restricción de normalidad	88
3.2.7. Grado de normalidad asociado a una instancia	88
3.2.8. Grado de normalidad asociado a un concepto	90
3.2.9. Comportamiento normal en un entorno en base a un concepto	94

3.3. Análisis global de normalidad	98
3.4. Adquisición de conocimiento	103
3.4.1. Algoritmo de aprendizaje automático inductivo	103
3.5. Arquitectura multi-agente	112
4. Componente para el análisis de trayectorias	121
4.1. Concepto de trayectoria normal	122
4.2. Variables para la definición y el análisis de trayectorias	125
4.3. Módulos de preprocesamiento	128
4.3.1. Información espacial de un objeto: segmentación y <i>tracking</i>	128
4.3.2. Estimación de la localización de un objeto	130
4.3.3. Clasificación de objetos	133
4.4. Definición de restricciones	143
4.5. Particularizaciones del concepto trayectoria	147
4.6. Consideraciones finales	150
5. Componente para el análisis de velocidades	157
5.1. Concepto de velocidad normal	158
5.2. Variables para la representación de velocidades	162
5.3. Módulos de preprocesamiento	164
5.3.1. Desplazamiento de un objeto	165
5.4. Definición de restricciones	166
5.5. Particularizaciones del concepto velocidad normal	170
5.6. Consideraciones finales	173
6. Resultados	179
6.1. Descripción de los experimentos	179
6.1.1. Evaluación de los métodos propuestos	185
6.2. Discusión de los resultados	187
6.2.1. Clasificación de objetos	187
6.2.2. Estimación de zonas	189
6.2.3. Análisis de trayectorias	190
6.2.4. Análisis de velocidades	192
6.2.5. Análisis global	193
6.2.6. Tiempos	196
6.3. Consideraciones finales	197
6.4. Publicaciones	199
7. Conclusions and Future Works	205
7.1. Conclusions	205
7.2. Future works	208
A. Herramientas Desarrolladas	231
A.1. Herramientas de adquisición de conocimiento	231
A.1.1. Herramienta para la caracterización de entornos	232
A.1.2. Herramienta de aprendizaje automático	234
A.2. Herramientas de Monitorización	237
A.2.1. Herramienta web de monitorización	238
A.2.2. Herramienta para la generación de vídeos analizados	240

Índice general	IX

B. A Scalable Approach based on Normality Components	247

Índice de figuras

2.1. Áreas de conocimiento relacionadas con la vigilancia inteligente.	12
2.2. Medios físicos y electromecánicos empleados en recintos vigilados.	19
2.3. Posible secuencia de etapas que podrían formar parte de la arquitectura de un sistema de vigilancia inteligente.	21
2.4. Relaciones básicas del esquema RCC8.	30
2.5. Ejemplo de la representación de eventos simples y sus relaciones mediante un grafo [14].	33
2.6. Ejemplo de propagación de restricciones tras añadir $e_1 < e_4$ y $e_4 < e_2$, [14].	33
2.7. Relaciones temporales entre periodos de tiempo definidas en el álgebra de Allen [14].	35
2.8. Captura de las herramientas pertenecientes al sistema presentado en <i>Automatic detection of dangerous events for underground surveillance</i> , escrito por Spirito et al. [170]. La imagen de la izquierda muestra la herramienta que permite dibujar las zonas de peligro a las que los peatones no deben acercarse; en este caso las vías del tren. La imagen de la derecha muestra como el sistema segmenta y detecta a las personas que se mueven en la escena y emite mensajes de alarmas cuando los objetos están demasiado cerca de las zonas de peligro.	36
2.9. Segmentación de objetos en vídeo a partir de una imagen modelo. La Figura (a) muestra la imagen modelo de un escenario virtual. La Figura (b) muestra el frame actual captado por la cámara de vigilancia. La Figura (c) muestra la diferencia entre la imagen modelo y el frame actual. Finalmente, la Figura (d), muestra tintados de color negro aquellos píxeles que son considerados diferentes con respecto a la imagen modelo.	39
2.10 Ejemplo de relación entre frames de tipo I, P y B	41
2.11 Ejemplo del resultado ofrecido por un proceso de <i>tracking</i> de un vehículo en una secuencia de seis frames. Las imágenes fueron tomadas desde una cámara situada en el grupo de investigación Oreto de la Universidad de Castilla-La Mancha.	45
2.12 Situaciones en estaciones subterráneas de metro detectadas en [56].	50
2.13 Sistema cognitivo capaz de asimilar protocolos de actuación a partir de percepciones visuales [124, 135]. La Figura (a) corresponde con el esquema general del sistema propuesto en [124], en la Figura (b) se puede apreciar la interacción entre agentes humanos y virtuales, y finalmente, la Figura (c) muestra un par de ejemplos de percepciones visuales a partir de las cuales el sistema aprende los protocolos de actuación.	51
2.14 Interpretación de trayectorias mediante árboles cuyos nodos representan los <i>clusters</i> [146].	52

2.15	Análisis de trayectorias para la detección de movimientos inusuales realizados por peatones, propuesto en [142].	54
2.16	Representación de trayectorias en [105].	60
2.17	Detección automática de las líneas de una carretera para determinar la perspectiva de la cámara y posterior generación de vistas aéreas de las zonas relevantes para la estimación de velocidades [41].	62
2.18	Métodos de calibración de cámaras propuesto en [217]. En la imágenes se puede apreciar como se marcan las zonas en común observadas desde cámaras diferentes.	65
2.19	Estudio de poses en el movimiento de las personas llevado a cabo en el sistema VSAM [52].	68
2.20	Arquitectura del sistema VSAM [52].	69
2.21	Arquitectura del sistema W^4 [92].	70
2.22	Análisis del comportamiento de grupos de personas y multitudes en estaciones de metro subterráneas, por el sistema AD-VISOR [165].	70
3.1.	Esquema general en el que se refleja la construcción de sistemas de vigilancia basados en componentes de normalidad. La capa intermedia ofrece una visión general del modelo propuesto en este trabajo para el análisis de eventos y comportamientos.	79
3.2.	Escenario virtual monitorizado mediante el uso de tres cámaras de seguridad. Cada una de las cámaras observa parte del entorno y cada una de estas percepciones E_1, E_2 y E_3 son consideradas a su vez como un subescenario E_i . En las Figuras (a), (b) y (c) es posible apreciar como diferentes cámaras pueden observar zonas comunes en un escenario global, por lo que la parte del sistema encargada de analizar los eventos puede recibir información sobre un mismo lugar procedente de múltiples fuentes.	83
3.3.	Función escalonada de salida para la clasificación de una situación como <i>normal/anormal</i> en base a un grado de normalidad.	95
3.4.	Dominio de definición para $N_{c_i}^j$. Cada uno de los conjuntos difusos representa un posible grado de normalidad. Los límites definidos para cada uno de los conjuntos difusos pueden variar en función del nivel de seguridad que se pretenda alcanzar; cuanto mayores sean los conjuntos CAA y CPA más estricto será el sistema.	96
3.5.	Comparación visual entre el rango de valores obtenidos por operadores t-normas, t-conormas y operadores OWA	99
3.6.	Ejemplos de asociaciones entre objetos, conceptos e instancias de conceptos.	102
3.7.	Dominios de definición para las variables v_1 y v_2	106
3.8.	Representación gráfica de los subconjuntos ZD_1 y ZD_2 incluidos en la regla difusa R_1	107

3.9. Representación gráfica de los valores que toma la variable tamaño para cada una de las reglas R_i, R_j, R_k y $R_{i'}$. La variable distancia con respecto a la cámara no se ha incluido en la imagen debido a que su valor no varía en ninguna de las reglas anteriores.	109
3.10. Relaciones existentes entre dos conjuntos difusos L_i, L_j para la estimación de la distancia a la que se encuentran.	111
3.11. Propuesta de arquitectura para sistemas de vigilancia inteligente.	114
3.12. Visión general del flujo de información entre los principales componentes de la arquitectura de vigilancia.	116
4.1. Descripción, con un alto nivel de abstracción, de las entradas y salidas del componente de normalidad encargado de analizar las trayectorias en un entorno vigilando por un conjunto de cámaras.	121
4.2. (a) Grupo de trayectorias realizadas por vehículos y marcadas mediante líneas continuas sobre la escena. (b) Distinción del grupo de trayectorias sin el uso de la imagen de la escena de fondo. (c) Ejemplo de secuencia de zonas que engloban al grupo de trayectorias mostrado en la Figura (b).	124
4.3. (a) Escena captada desde una cámara de vigilancia. (b) Ejemplo de división de la escena mostrada en (a) en diferentes zonas. . .	131
4.4. Zonas de apoyo de los objetos marcada con un rectángulo sombreado. En la parte derecha de la imagen se puede apreciar como un objeto puede estar situado en varias zonas al mismo tiempo.	131
4.5. Ejemplo de puntos de la matriz de apoyo de un objeto, que son empleados para estimar la zona/as en las que se encuentra un objeto. El algoritmo estudia si cada uno de los píxeles marcados con una cruz en la matriz de la derecha se encuentran en el interior de una determinada zona.	132
4.6. Estudio de las propiedades de los objetos con capacidad de movimiento en la escena que se muestra en la Figura 4.3	135
4.7. Ejemplo de división vertical y horizontal de un escena. El significado de las etiquetas lingüísticas utilizadas para la división horizontal es el siguiente: VL: very left, L: left, C: center, R: right, VR: very right. Por otro lado, las etiquetas lingüísticas utilizadas en la división vertical tienen el siguiente significado: VU: very up, U: up, C: center, D: down, VD: very down. Cada uno de los círculos representa zonas de la imagen en la que la pertenencia a los conjuntos difusos es máxima.	139

4.8. Ejemplo de clasificación de un objeto en el que se tiene en cuenta la evolución temporal, es decir, el grado de pertenencia actual a cada clase, obtenido mediante la activación de las reglas y los grados de pertenencia obtenidos en instantes anteriores. La primera fila de la Figura representa el tiempo, cada instante situado a la derecha representa un instante posterior a su antecesor. En la fila intermedia se muestra la forma y el tamaño de la elipse que envuelve al objeto en cada uno de los instantes t_i . Finalmente, en la última fila se muestra la clasificación del objeto. Si la imagen que representa la clase tiene un valor numérico asociado a la derecha, quiere decir que el objeto ha sido clasificado en ese instante t_i con ese valor de pertenencia. Sin embargo, el valor definitivo de pertenencia se calcula como la media aritmética de los valores de pertenencia anteriores y el actual.	144
4.9. División en zonas de la escena elegida para la particularización de trayectorias normales	147
4.10. Secuencia de frames en los que el sistema de vigilancia analiza las posibles trayectorias que está siguiendo un vehículo.	150
4.11. Comportamiento anómalo de un vehículo debido a que no sigue ninguna de las trayectorias normales definidas en el sistema.	152
5.1. Descripción, con un alto nivel de abstracción, de las entradas y salidas del componente de normalidad encargado de analizar las velocidades en un entorno vigilando por un conjunto de cámaras.	157
5.2. División de una escena teniendo en cuenta las distancias con respecto a la cámara. La parte derecha muestra como pueden variar notablemente las dimensiones de los objetos dependiendo de si se encuentra en una zona cercana o lejana a la cámara.	160
5.3. Conexión entre los preprocesadores y el componente de normalidad encargado de analizar las velocidades de los objetos.	167
5.4. Ejemplo del cálculo del grado de satisfacción la restricción de localización horizontal de un objeto (μ_{22}), para $HP = 130$ y $\ell_{HP} = \{L, C, R\}$	168
5.5. Secuencia de pasos para obtener el conjunto de reglas difusas que clasifican la velocidad de los objetos como normales o anómalas.	175
6.1. Captura de cada una de las escenas empleadas en las pruebas donde es posible observar las condiciones de iluminación.	188

6.2. Conjunto de histogramas que representan los valores de brillo y contraste en cada una de las escenas empleadas en las pruebas. Cuanto más altos sean los valores alcanzados en el eje Y, mayor es la iluminación de la escena. Las seis primeras escenas corresponden a un día nublado donde los objetos carecen de sombras acentuadas, mientras que en las cuatro últimas se puede apreciar que el grado de iluminación es mucho mayor.	188
6.3. Las tres primeras gráficas representan espacios ROC donde se muestran las relaciones entre la razón de verdaderos positivos (VPR) y la razón de falsos positivos (FPR) para el primer y segundo componente, y el análisis global de normalidad. La última gráfica muestra la relación entre VP, VN, FP y FN en el análisis global para cada una de las escenas empleadas en los experimentos.	196
6.4. Relaciones entre las situaciones anómalas que suceden en los componentes de normalidad y el análisis global. Los verdaderos negativos (VN) representan situaciones anómalas detectadas, mientras que los falsos positivos (FP) representan situaciones anómalas interpretadas como normales.	197
A.1. Captura de la herramienta de adquisición de conocimiento donde se muestra la parte de la interfaz dedicada a la definición de zonas y especificación de la topología de la escena.	233
A.2. Fragmentos de código XML, que representan parte del conocimiento de una escena, generados con la herramienta de adquisición de conocimiento	234
A.3. Parte de la interfaz gráfica de la herramienta de aprendizaje automático donde se establecen los parámetros de configuración	237
A.4. Parte de la interfaz gráfica de la herramienta de aprendizaje automático donde se muestran las reglas obtenidas tras el proceso de aprendizaje.	238
A.5. Captura de la interfaz del sistema de monitorización OCULUS. .	240
A.6. Parte derecha de la herramienta de monitorización donde se muestra la información de salida de los componentes de normalidad y los módulos de preprocesamiento.	241
A.7. Ejemplo de un <i>frame</i> perteneciente a uno de los vídeos generados con la herramienta descrita en esta sección, en el que se incluye información relativa al análisis de normalidad.	242
A.8. Ejemplo de un <i>frame</i> perteneciente a uno de los vídeos generados con la herramienta descrita en esta sección, en donde la situación global del entorno no es normal.	243

Índice de tablas

2.1. Evolución de los sistemas de vigilancia.	17
3.1. Ejemplos de intersecciones difusas (t-normas) [111]. T_{min} representa la intersección drástica, es decir, $T_{min}(a, b)$ es a , cuando $b = 1$; b cuando $a = 1$ y 0 en caso contrario. Además siempre se cumple la siguiente relación: $T_{min}(a, b) \leq \max(0, a + b - 1) \leq ab \leq \min(a, b)$	91
3.2. Ejemplos de uniones difusas [111](t-conormas). S_{max} representa la unión drástica, es decir, $S_{max}(a, b)$ es a , cuando $b = 0$; b cuando $a = 0$ y 1 en caso contrario. Además siempre se cumple la siguiente relación: $\max(a, b) \leq a + b - ab \leq \min(1, a + b) \leq S_{max}(a, b)$	93
3.3. Configuración de un operador OWA para que su comportamiento sea el mismo que el de la t-norma del mínimo	100
3.4. Configuración de un operador OWA para que su comportamiento sea el mismo que el de la t-conorma del máximo	100
3.5. Operador OWA en el que los pesos se asignan de forma ponderada	100
4.1. Conjunto A de áreas que representan las zonas de principal actividad en un entorno observado.	126
4.2. Variables utilizadas junto con el conjunto de áreas A para representar la ubicación de los objetos en la escena.	126
4.3. Pertenencia de un objeto obj a una clase c	127
4.4. Conjunto de variables temporales asociadas a las trayectorias.	127
4.5. Variables Υ , Ψ y $\ell_{\Psi}(obj)$ para la definición de restricciones de rol y espaciales, respectivamente.	128
4.6. Dominios de definición de las variables elegidas para la construcción de reglas difusas que permitan clasificar objetos en una escena observada por cámaras de seguridad.	138
4.7. Algunos ejemplos de elementos que forman parte del conjunto de entrenamiento empleado para la generación de reglas difusas, que permiten la clasificación de los objetos detectados por el proceso de segmentación.	140
4.8. Etiquetado de los ejemplos de la Tabla 4.7, donde los valores numéricos son sustituidos por etiquetas lingüísticas.	141
4.9. Agrupación de los ejemplos que forman parte del conjunto de entrenamiento.	142
4.10. Ordenación de los ejemplos de la Tabla 4.9 según el número de instancias o apariciones.	142
4.11. Ejemplos de reglas difusas para la clasificación de objetos.	143
4.12. Relaciones temporales entre instantes e intervalos de tiempo	146
4.13. Clases o categorías a las que pertenece cada zona definida en la escena	148
4.14. Conjunto de instancias del concepto trayectoria normal para los vehículos.	148

4.15. Conjunto de instancias del concepto trayectoria normal para peatones. HU es el identificador del intervalo de tiempo en el que la universidad se encuentra abierta al público.	149
4.16. Grados de satisfacción de las restricciones y grados de normalidad correspondiente a la secuencia mostrada en la Figura 4.10. En la primera columna, c_1 representa la clase <i>vehículo</i> , mientras que c_2 la clase <i>peatón</i> . Los valores numéricos marcados en negrita en la última columna representan el máximo grado de normalidad, es decir $N_{c_1}^1(obj)$	151
4.17. Ejemplo de definición de trayectoria normal mediante el modelo formal	152
5.1. Conjunto de variables utilizado para determinar la localización de un objeto. Los valores que definen los conjuntos difusos pueden variar dependiendo de la escena y la resolución de la cámara empleada. En este caso, los valores corresponden con la escena descrita en 4.5.	163
5.2. Pertenencia de un objeto obj a una clase c	164
5.3. Conjunto de variables utilizado para determinar el desplazamiento de un objeto. Al igual que en la Tabla 5.1 los valores numéricos que establecen los límites de los conjuntos difusos corresponden con la escena utilizada en las pruebas y que ha sido descrita previamente en 4.5. En esta escena, el estudio de los desplazamientos de los objetos y la clasificación de la velocidad como normal o anómala se ha realizado cada 25 <i>frames</i> . En otro tipo de escenas en los que los vehículos pueden circular a velocidades mucho mayores, el análisis se debe realizar con mayor frecuencia. Cuanto menor sea el intervalo de tiempo entre análisis, mayor será la carga computacional para el sistema.	165
5.4. Listado del resto de variables que se emplean para definir las restricciones espaciales del concepto velocidad normal.	166
5.5. Ejemplo de instancia de velocidad normal en base a las restricciones definidas en la sección.	171
5.6. Algunos ejemplos de elementos que forman parte del conjunto de entrenamiento empleado para la generación de reglas difusas que permiten clasificar la velocidad de los objetos detectados. En este caso sólo se emplean dos clases, peatón y vehículo, cuyos identificadores son 0 y 1 respectivamente.	173
5.7. Conjunto de ejemplos de la Tabla 5.6 etiquetados. Cada uno de ellos representa reglas iniciales a partir de las cuales se construyen reglas generales.	173
5.8. Ejemplos de reglas difusas para la clasificación de velocidades.	174
5.9. Ejemplo de instancias del concepto velocidad normal aprendidas mediante el algoritmo de aprendizaje automático, para la escena descrita en la Sección 4.5.	174
6.1. Situaciones relevantes en cada una de las escenas elegidas para probar el sistema.	185

6.2. Resultados obtenidos en el proceso de clasificación de objetos como peatones y vehículos.	189
6.3. Resultados obtenidos en la estimación de la zona/as en las que se encuentran cada uno de los objetos detectados en el entorno monitorizado	190
6.4. Resultados obtenidos por el componente de normalidad encargado de analizar las trayectorias de los objetos detectados. Las etiquetas C1, C2 y C3 se refieren a las causas de los fallos y su significado es el siguiente: C1: Clasificación incorrecta de objetos, C2: Posición incorrecta de la elipse, C3: Cambio de elipses entre objetos.	191
6.5. Relaciones existentes entre Verdaderos/Falsos positivos y Verdaderos/Falsos negativos en el componente de normalidad encargado de analizar las trayectorias.	192
6.6. Resultados obtenidos por el segundo componente de normalidad encargado de analizar las velocidades que alcanzan los objetos detectados. Las etiquetas C1, C2, C3 y C4 hacen referencia a las causas de error en el componente y sus significados son los siguientes: C1: clasificación incorrecta de objetos, C2: Cambios bruscos en la posición y el tamaño de la elipse, C3: Activación incorrecta de las reglas que clasifican las velocidades como normal o anómalas, C4: Cambio de elipse entre objetos de distintas clases	193
6.7. Relaciones existentes entre Verdaderos/Falsos positivos y Verdaderos/Falsos negativos en el componente de normalidad encargado de analizar las velocidades de los objetos detectados. .	194
6.8. Resultados obtenidos en el análisis global como combinación de los análisis realizados por el primer y segundo componente de normalidad. Las etiquetas C1 y C2 significan errores procedentes del primer componente y del segundo, respectivamente. .	194
6.9. Relaciones existentes entre Verdaderos/Falsos positivos y Verdaderos/ Falsos negativos en el análisis global de normalidad. .	195
6.10. Tiempos invertidos en cada uno de los procesos que constituyen el sistema (medidos en milisegundos). El significado de las etiquetas empleadas en la cabecera de la tabla es el siguiente: Esc: escena, Elem: elementos a analizar, T: tiempo total invertido, M: tiempo medio.	197
6.11. Tiempos invertidos en el proceso de aprendizaje de reglas difusas para la clasificación de objetos y velocidades.	197



*“ Un comienzo no
desaparece nunca,
ni siquiera con
un final ”*

Harry Mulisch



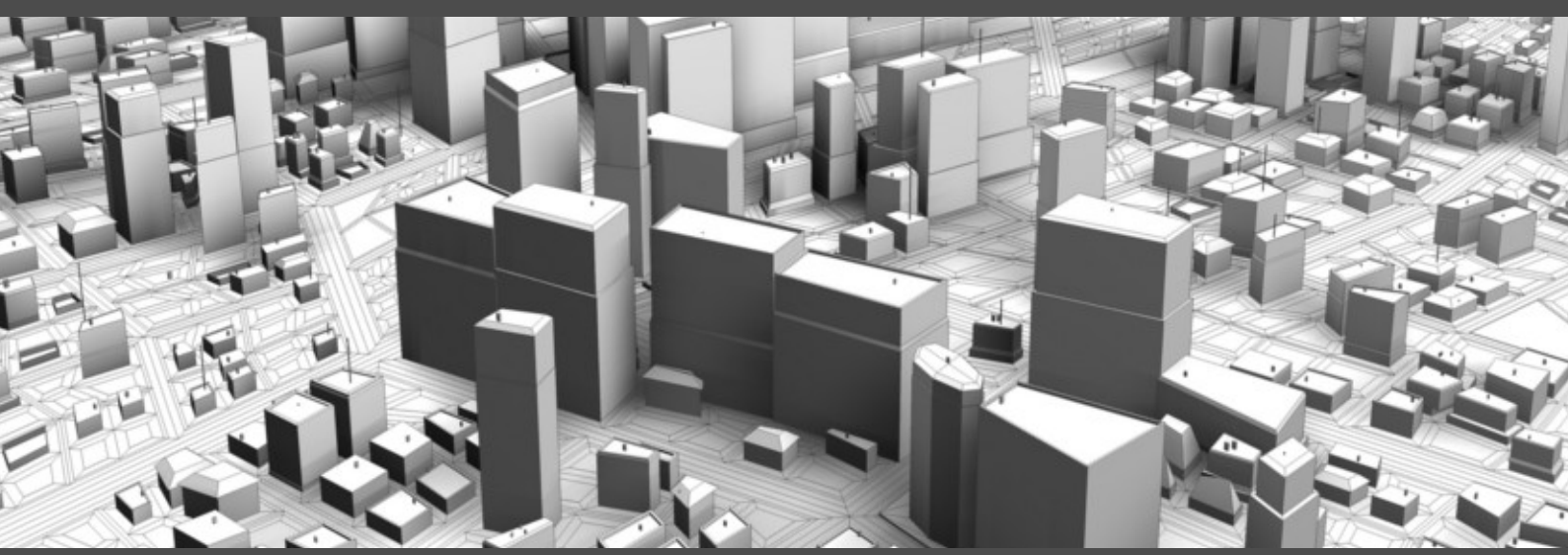
Capítulo

Introducción

1

1

Capítulo



1

Introducción

En este Capítulo se plantea de forma breve la necesidad actual de utilizar sistemas de vigilancia inteligentes, así como los principales problemas y dificultades que presentan. Precisamente estos problemas constituyen la principal motivación del presente trabajo y da pie a la definición de los objetivos a alcanzar. El Capítulo concluye con una breve descripción sobre la forma en la que se estructura el documento.

1.1. Motivación

Actualmente, la seguridad de los ciudadanos e infraestructuras en espacios públicos y privados es uno de los principales temas de preocupación para la mayor parte de gobiernos e instituciones. Como ejemplo citar la ciudad de Londres donde se han implantado más de cuatro millones de cámaras que envían la señal a centros de control de seguridad, donde operadores humanos supervisan las imágenes recibidas para detectar situaciones anómalas que puedan conllevar peligro para los ciudadanos [63].

En los últimos años, las nuevas tecnologías han influido notoriamente en el diseño de los sistemas de seguridad presentes en los centros de control, otorgando mayor robustez a los procesos de vigilancia y evitando situaciones desagradables o peligrosas. La Inteligencia Artificial está jugando un papel clave en la evolución de los sistemas de seguridad. Un buen ejemplo de ello se encuentra en el área del análisis de contenido de vídeo, también llamado vídeo cognitivo, que permite el análisis de las conductas o comportamientos de los objetos en movimiento detectados por el sistema. Esta técnica está cada vez más presente en los sistemas de vídeo-seguridad de última generación.

Los primeros sistemas de vídeo vigilancia fueron los sistemas CCTV

(circuito cerrado de televisión), los cuales están constituidos por una red de cámaras que envían la señal a una sede central, donde el personal de seguridad puede observar lo que ocurre a través de un panel formado por varios monitores. El principal problema de estos sistemas es la dependencia absoluta de la actividad humana para detectar sucesos anómalos. En cuestiones de vigilancia, existen actividades que los seres humanos realizan a la perfección como, por ejemplo, la detección de trayectorias anómalas o inapropiadas; sin embargo, existen otros tipos de actividades donde las máquinas son mucho más eficientes, como es el caso de la detección de objetos abandonados y posiblemente sospechosos [63], que para los agentes humanos pueden pasar fácilmente desapercibidos. Además, existen estudios experimentales que demuestran que el rendimiento de una persona observando un monitor disminuye notablemente tras 20 minutos de vigilancia [166], aumentando así la probabilidad de que las situaciones anómalas no sean detectadas cuando éstas ocurran. Por tanto, la vídeo vigilancia tradicional tiene ciertas dificultades para cumplir con las, cada vez mayores, demandas del sector [153]. Es decir, los sistemas de seguridad deberían dejar de ser simples y reactivos para convertirse en sistemas inteligentes y proactivos.

Surge así la necesidad de crear sistemas de seguridad inteligentes para mejorar la efectividad de los sistemas tradicionales y aliviar la carga de los operadores humanos [36]. En las dos últimas décadas la comunidad científica ha propuesto diversos algoritmos, técnicas y modelos que han dado lugar a sistemas de segunda y tercera generación (aún en fase de experimentación) [186]. Los sistemas de segunda generación combinan las tecnologías de los sistemas CCTV y vigilancia IP con algoritmos de Visión por Computador e Inteligencia Artificial, mientras que los sistemas de tercera generación se caracterizan por ser altamente distribuidos y conservar las propiedades de los sistemas pertenecientes a la segunda generación. El fin que se pretende conseguir con este tipo de sistemas es que se proporcione una buena interpretación de la escena a estudiar y que sólo en los momentos necesarios se llame la atención del operador humano en tiempo real.

En la actualidad, la vídeo vigilancia inteligente es una línea de investigación en continuo crecimiento. Más de una docena de contratos de investigación han sido otorgados a laboratorios y universidades con el objetivo de acelerar los procesos de investigación en las técnicas de análisis de vídeo, como es el caso de los grupos de visión, inteligencia artificial y robótica de las universidades de Leeds y Kingston en Inglaterra y el Grupo de Investigación Nacional Inria en Francia. En concreto, el presente trabajo está enmarcado dentro del proyecto Hesperia: "Homeland Security Technologies for the Security in Public Spaces and Infrastructures", proyecto CENIT I+D 2006-2009 en el que participan numerosas empresas y universidades nacionales, entre ellas la Universidad de Castilla-La Mancha con la participación de los grupos de investigación ARCO y ORETO, cuyo objetivo principal es el desarrollo de tecnologías que permitan la creación de sistemas punteros de seguridad, vídeo vigilancia y control de operaciones de infraestructuras en

espacios públicos. En la literatura existen numerosos sistemas de vigilancia que emplean técnicas de procesamiento de imágenes e Inteligencia Artificial, entre los que cabe destacar, VSAM [52], W^4 [92], y los sistemas presentados en [25, 26, 4, 50, 99, 109, 165, 191], entre otros. Además, se han celebrado numerosos *Workshops* y Conferencias donde se incluyen trabajos sobre vigilancia cognitiva (CVPR, ECCV, ICCV, PETS, VISAPP), y existen revistas especializadas en esta línea de investigación, donde se pueden encontrar también trabajos relevantes, como por ejemplo *Image and Vision Computing*, *Signal Processing*, *International Journal of Pattern Recognition and Artificial Intelligence* o *Multimedia Systems Journal*. Algunas de estas revistas incluso dedican números especiales a la vídeo vigilancia inteligente como es el caso de “Visual Analysis and Understanding for Surveillance Applications” en la revista *International Journal of Pattern Recognition and Artificial Intelligence* o “Multimedia Surveillance Systems” en la revista *Multimedia Systems Journal*.

Aunque los sistemas de segunda y tercera generación ofrecen importantes ventajas con respecto a los sistemas tradicionales, estos presentan todavía una serie de carencias que deben ser resueltas para ofrecer ciertas garantías y reemplazar o ayudar a los agentes de seguridad humanos en algunas tareas. Dos de los principales problemas de la mayoría de los sistemas de seguridad actuales son su falta de flexibilidad y escalabilidad. Existe una falta de escalabilidad porque normalmente estos sistemas se centran en el estudio de un aspecto a vigilar (velocidad, trayectorias, control de accesos, etc) de una forma muy específica para entornos muy concretos. Por otro lado, aquellos sistemas que tienen la capacidad de analizar más de un aspecto al mismo tiempo, suelen estar formados por módulos independientes, que no han sido diseñados siguiendo un patrón común y cuyas salidas no suelen ser combinadas con el objetivo de obtener una interpretación global, tal como hacen los agentes humanos al observar un entorno vigilado. Para que los sistemas de vigilancia sigan evolucionando es necesario crear sistemas escalables que se adapten correctamente a diferentes tipos de entornos e incrementen su capacidad de interpretación a medida que se diseñan nuevos módulos de análisis. La combinación de estos análisis es necesaria para obtener una interpretación global de la situación actual del entorno. Por ejemplo, en un sistema donde se analice únicamente las trayectorias el comportamiento de cualquier objeto será normal siempre y cuando éste siga una trayectoria permitida. Sin embargo, si se incluye un nuevo módulo para analizar las velocidades, no tiene sentido analizar de forma independiente las trayectorias y las velocidades, sino que la normalidad del comportamiento de los objetos dependerá de si éstos realizan trayectorias correctas y lo hacen con una velocidad apropiada.

Con respecto a la flexibilidad, los análisis incluidos en la mayoría de sistemas se suelen aplicar de la misma forma a cualquier objeto de manera constante, independientemente de la clase a la que pertenezca y el lugar donde desarrolle su actividad. Por esta razón, sería interesante proponer modelos y arquitecturas flexibles que faciliten la inclusión de

nuevos estudios sobre amenazas o aspectos a vigilar no considerados y que permitan establecer diferentes configuraciones para el análisis acorde al tipo de objeto. Es decir, las amenazas o aspectos a vigilar para cada objeto en la escena dependerán de la clase a la que pertenezca y el entorno en el que se encuentre. Por ejemplo, en ciertos entornos exteriores podría ser interesante estudiar la velocidad de los vehículos, pero no la de las personas. Sin embargo, en edificios o cualquier otro entorno interior sí que podría ser interesante el estudio de la velocidad de las personas para detectar situaciones anómalas como, por ejemplo, personas huyendo ante posibles incendios, explosiones, etc. El uso de unos módulos u otros en el análisis de un tipo de objeto dependerá de las características del entorno en el que se implanta el sistema y el nivel de seguridad exigido. Un sistema que goce de flexibilidad en este sentido podrá mejorar su eficiencia reduciendo considerablemente su carga computacional, ya que evita realizar tipos de análisis innecesarios.

Un tercer problema es la dificultad para detectar situaciones anómalas desconocidas o que no han sucedido previamente. El objetivo final de cualquier sistema de vigilancia avanzado es la detección automática de anomalías y, para ello, existen diferentes enfoques. El primero de ellos, se basa en la definición única de situaciones anómalas y la búsqueda de similitudes con las situaciones que ocurren en el entorno vigilado. El problema de los sistemas que siguen este enfoque es la incapacidad para detectar situaciones anómalas que no hayan sido definidas previamente y, que al no ser capaces de ofrecer una respuesta cuando este tipo de situaciones ocurren, se convierten en sistemas débiles alejados de su objetivo principal. Por este motivo, resulta más apropiado diseñar sistemas que permitan definir o aprender las situaciones normales de cualquier entorno junto con las situaciones anómalas más comunes. De esta forma, la peor situación aparece cuando una situación anómala sucede y no ha sido definida previamente, pero al menos el sistema es consciente de que dicha situación no es normal. Además la mayoría de estos sistemas otorgan casi toda la importancia a la creación y prueba de algoritmos de procesamiento de imágenes, con el fin de realizar detección y seguimiento de objetos. Son más escasos los sistemas que intentan incorporar inteligencia dándole mayor énfasis a la etapa de análisis del comportamiento de objetos. Sin embargo, esta etapa es vital en el proceso de vigilancia, ya que los resultados de la misma pueden servir al operador como herramienta de ayuda en su trabajo. Los sistemas de análisis de contenido mediante técnicas de inteligencia artificial ofrecen detección automática en tiempo real, identificación y análisis de amenazas potenciales, permitiendo que estos tipos de sistemas sean mucho más potentes.

El cuarto problema está directamente relacionado con la construcción de las bases de conocimiento donde se modela el entorno observado. Normalmente los expertos en el dominio son los encargados de llevar a cabo esta tarea tediosa y nada trivial. El inconveniente es que desde que se implanta el sistema hasta que éste ofrece una fiabilidad alta, los expertos

deben realizar numerosas modificaciones sobre la base de conocimiento original. Por tanto, para poder resolver este inconveniente es necesario proporcionar herramientas de adquisición de conocimiento, o bien, dotar al sistema de capacidad para ampliar y modificar su base de conocimiento por sí mismo mediante el uso de algoritmos de aprendizaje automático.

Finalmente, los sistemas de seguridad deben ofrecer respuestas en tiempo cercano al real con el objetivo de detectar las situaciones anómalas lo antes posible. En la literatura existen, por un lado, algoritmos que ofrecen buenos resultados en el proceso de interpretación de escenas reales, sin embargo, invierten demasiado tiempo en la obtención de los resultados. Por otro lado, también existen algoritmos que ofrecen peores resultados pero, en cambio, ofrecen sus respuestas en menor tiempo. Por tanto, en la construcción de los sistemas expertos de monitorización existe la necesidad de encontrar un equilibrio entre fiabilidad y tiempo de respuesta.

1.2. Objetivos

Para ofrecer soluciones a los problemas planteados en la sección anterior, surge la necesidad de alcanzar los siguientes objetivos:

1. **Estudio y análisis del estado del arte** de los sistemas de seguridad actuales, así como de las técnicas empleadas para el análisis de comportamientos a partir de la información procedente de los sensores. También se estudiarán técnicas y algoritmos de aprendizaje automático para la construcción de bases de conocimiento.
2. **Propuesta de diseño y desarrollo de sistemas de seguridad escalables basados en componentes generales instanciables** para problemas concretos y **definición de un modelo teórico formal** para llevarlo a cabo. El modelo debe ser flexible, escalable e independiente de cualquier entorno, y que permita representar e identificar las situaciones normales, así como las situaciones anómalas más comunes. Dicho modelo estaría ubicado en un nivel intermedio entre el nivel donde están situados los sensores y el nivel encargado de tomar las decisiones. Por tanto, el modelo tendría como entrada el flujo de información procedente del nivel sensorial y ofrecería como salida el análisis de situaciones y eventos que se realiza en cada momento. La consecución de este objetivo implica alcanzar una serie de hitos:
 - a) **Uso de métodos formales para el tratamiento de la incertidumbre y vaguedad.** El uso de la lógica difusa y técnicas de *softcomputing* permitirán el tratamiento de la incertidumbre procedente de los dispositivos hardware encargados de percibir el entorno. Además, el propio entorno y sus condiciones ambientales pueden incluir incertidumbre en las percepciones del sistema. Por todo ello,

se diseñarán técnicas que permitan al sistema manejar y razonar con este tipo de información.

- b) **Interpretabilidad de los resultados.** Otro objetivo, no menos importante, es la representación del conocimiento y las conclusiones obtenidas en el proceso de inferencia con la mayor interpretabilidad posible. De esta forma los usuarios expertos que hagan uso del modelo, no sólo podrán apreciar que el sistema ha determinado que una situación no es normal, sino que además, podrán comprender los motivos por los cuales el sistema proporcionó dicha salida.
- c) **Definición de componentes de normalidad.** El modelo formal se basará en la definición de componentes que analizan la normalidad de conceptos generales que son instanciables para entornos concretos. Un componente de normalidad determina como debería ser el comportamiento correcto de un tipo de objeto de acuerdo a un concepto y siempre tratando la incertidumbre y la imprecisión procedente de las capas inferiores. Ejemplos de conceptos son, trayectorias, velocidad, relaciones de proximidad, objetos abandonados, etc. Cada uno de los componentes será independiente y permitirá realizar un estudio específico del comportamiento de un objeto. De esta forma, se solventará el problema de la escalabilidad mencionado anteriormente, ya que el modelo debe soportar la inclusión de nuevos componentes sin afectar al resto, dotando así al sistema de mayor capacidad de análisis.
- d) **Diseño de mecanismos y técnicas de fusión para la combinación de interpretaciones procedentes de los componentes de normalidad.** El análisis global de normalidad se realizará a partir de la combinación de los análisis individuales realizados por cada uno de los componentes de normalidad.
- e) **Asociación de técnicas de adquisición de conocimiento y aprendizaje automático a cada uno de los componentes.** Cada uno de los componentes de normalidad necesitará su propia base de conocimiento para realizar el análisis y obtener conclusiones. La complejidad y extensión de dicha base de conocimiento dependerá del concepto que se analice en el componente. A la hora de instanciar¹ los componentes de normalidad generales, se hace necesario construir las bases de conocimiento particulares para el problema concreto donde se quiera desplegar el sistema de vigilancia. Por este motivo, se estudiarán y aplicarán diferentes técnicas de adquisición de conocimiento y aprendizaje automático que faciliten la construcción de cada base de conocimiento asociada a un concepto. Cada componente tendrá asociado sus propias herramientas para facilitar el proceso de instanciación.

¹Particularización de un componente de normalidad en un entorno concreto. Por ejemplo, si un componente de normalidad estudiara las trayectorias, la particularización de éste en un entorno, consistiría en la definición de las trayectorias normales de dicho entorno.

Una vez establecido el marco teórico y las bases formales, se implementará un prototipo de sistema inteligente para analizar situaciones y eventos a partir de la información procedente de cámaras de seguridad. En dicho prototipo se materializan las ideas y desarrollos propuestos en la Tesis. Para ello, se llevarán a cabo las siguientes tareas:

- Implementación del modelo teórico. Para la implementación del sistema se llevarán a cabo técnicas de Ingeniería del Software como es el diseño multicapa, así como la aplicación de patrones de diseño. De esta forma se realizará una diferenciación clara entre las capas de diseño, lógica de programa y persistencia, de tal forma que la modificación en una de ellas no afecte al resto.
- Evaluación del sistema. Para cada uno de los componentes o módulos del sistema implementados se llevarán a cabo diferentes pruebas o tests que permitirán determinar su validez y robustez.
- Etapa de integración. Una vez que los componentes o módulos hayan superado las pruebas se integrarán en el sistema final.

1.3. Contenidos

El documento está organizado en siete Capítulos y dos Anexos. En el primer Capítulo se han expuesto las principales motivaciones del presente trabajo, se ha establecido un marco de trabajo y se han definido los objetivos a alcanzar.

En el **Capítulo 2** se elabora un estado del conocimiento en el que se analiza la evolución de los sistemas de vigilancia, fases relevantes en este tipo de sistemas y su función principal, estudio de relaciones espacio-temporales, técnicas para la interpretación de eventos, representación y adquisición del conocimiento, fusión de información procedente de múltiples fuentes heterogéneas, ámbitos de aplicación y análisis de las principales características de los sistemas existentes. En definitiva, todo tipo de trabajos y propuestas en el campo de la vigilancia cognitiva que ayuden a detectar las principales necesidades de los sistemas actuales para la propuesta de posibles mejoras.

En el **Capítulo 3** se presenta un modelo formal para la definición de la normalidad de eventos y conductas en entornos monitorizados, así como para el desarrollo de sistemas de seguridad escalables. El modelo permite definir nuevos componentes de normalidad instanciables para problemas concretos. Estos componentes se basan principalmente en el análisis de la normalidad para detectar situaciones anómalas. Dichos componentes necesitan conocimiento experto para desarrollar su actividad y la adquisición de este conocimiento no siempre es una tarea trivial, por esta razón, se

propone el uso de un algoritmo de aprendizaje automático y herramientas de adquisición de conocimiento. El Capítulo concluye con una breve discusión sobre cómo las arquitecturas multi-agente podrían ofrecer soporte al modelo propuesto en el capítulo.

En los **Capítulos 4 y 5** se aplica el modelo propuesto en el Capítulo 3, para la definición de dos componentes de normalidad. El primero de ellos permite en análisis de las trayectorias que siguen los objetos detectados por el sistema, mientras que el segundo se encarga del estudio de las velocidades. Tanto el modelo como los componentes creados a partir de él, son evaluados en el **Capítulo 6** donde se muestran y discuten los resultados obtenidos.

Finalmente, en el **Capítulo 7** se exponen las conclusiones y las principales líneas de investigación que han quedado abiertas para su desarrollo en trabajos futuros. La memoria concluye con las referencias bibliográficas y dos anexos. En el **Anexo A** se describen las herramientas de monitorización y adquisición de conocimiento desarrolladas en este trabajo. Por último, en el segundo anexo o **Anexo B** se resume en inglés parte del trabajo desarrollado en esta Tesis.



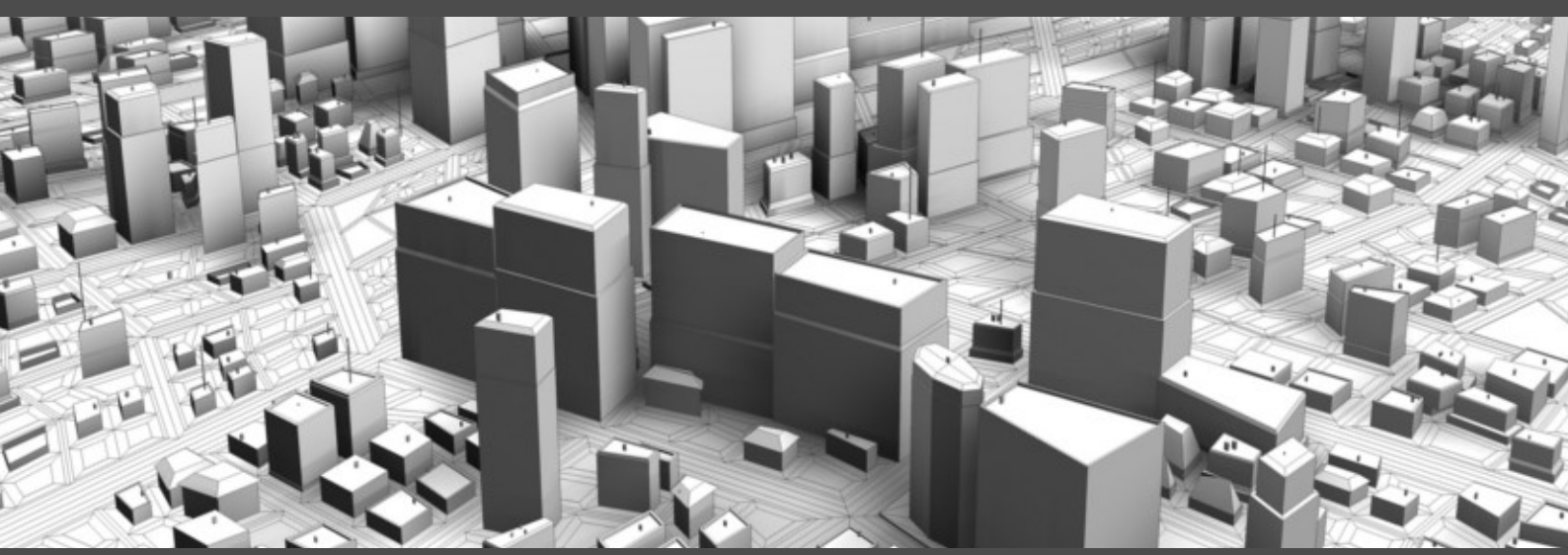
Capítulo

2

Estado del Conocimiento

2

Capítulo



2

Estado del Conocimiento

La vigilancia inteligente tiene como principal objetivo el diseño de sistemas que desempeñen una actividad similar a la que llevan acabo los agentes humanos en tareas de vigilancia. La construcción de sistemas expertos e inteligentes aplicados a la vigilancia es un problema complejo en el que intervienen diferentes áreas de conocimiento, entre las que podemos destacar dos de ellas: Visión por Computador e Inteligencia Artificial.

Las diferentes técnicas y métodos pertenecientes al área de la Visión por Computador permiten a cualquier sistema artificial captar un entorno real de una forma similar a como lo hacen los agentes humanos mediante el sentido de la vista. Por otro lado, las técnicas pertenecientes al área de la Inteligencia Artificial permiten representar y adquirir el conocimiento que los procesos de razonamiento necesitan para identificar los eventos que se producen en el entorno vigilado y clasificarlos como normales o anómalos; todo ello para una posterior activación de alarmas si fuera necesario.

La Figura 2 muestra de forma esquematizada las principales áreas de conocimiento que intervienen en la construcción de sistemas de vigilancia inteligentes y cómo están relacionadas entre sí. El presente Capítulo está estructurado de acuerdo a este esquema y se discutirán algunos de los trabajos más relevantes en cada una de las áreas de conocimiento. Dicho estudio ha servido para detectar las principales carencias que existen actualmente entorno a los sistemas de vigilancia inteligentes y, a su vez, estas carencias se han convertido en la principal motivación de este trabajo que ha dado lugar al planteamiento de los objetivos descritos en el Capítulo anterior.

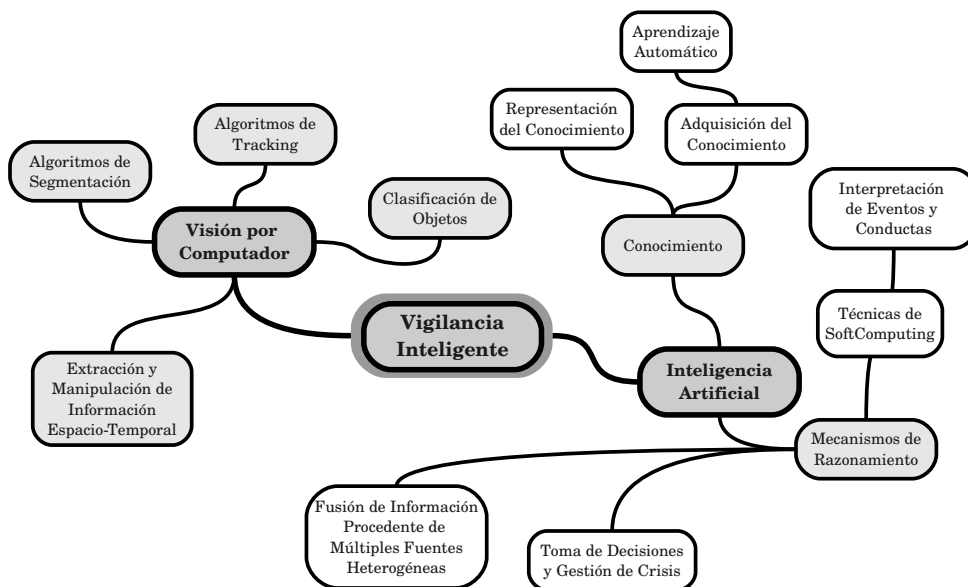


Figura 2.1: Áreas de conocimiento relacionadas con la vigilancia inteligente.

2.1. Evolución de los sistemas de vigilancia

En esta sección se pretende mostrar la evolución de los sistemas de vigilancia [36, 116, 128, 158, 186], que engloba desde los primeros sistemas analógicos de circuito cerrado CCTV, hasta soluciones distribuidas más complejas con capacidad para interpretar eventos y comportamientos del mundo real.

Valera y Velastin [186] clasifican los sistemas de vigilancia en tres generaciones de acuerdo a las tecnologías que emplean, así como las ventajas y problemas que presentan. La primera generación está formada por los sistemas de circuito cerrado de televisión, cuyo acrónimo es CCTV (*Closed Circuit Television*). Se le denomina circuito cerrado porque todos los componentes que forman el sistema están enlazados unos con otros. Estos tipos de sistemas están formados por un conjunto de cámaras distribuidas a lo largo del entorno vigilado y conectadas a un conjunto de monitores que suelen estar ubicados en una sede central.

Entre los problemas que presentan los sistemas CCTV se podrían destacar dos de ellos como los más significativos: el primero es el uso de técnicas analógicas para la distribución y el almacenamiento de imágenes (realizado con grabadores de vídeo en cinta VCR). Este último hecho dificulta en gran medida el mantenimiento intensivo del sistema, así como la posibilidad de acceso remoto o la integración con otros sistemas. El segundo

problema nace a partir de la dependencia de la actividad humana para detectar situaciones anómalas capturadas por las cámaras de vigilancia. La observación prolongada de los monitores causa fatiga en los vigilantes y, como consecuencia, falta de atención [166]; lo que da lugar a que la probabilidad de que una situación anómala no sea detectada aumente considerablemente. A pesar de las deficiencias comentadas anteriormente, los sistemas de vídeo vigilancia CCTV son ampliamente utilizados en todo el mundo, sobre todo en ámbitos comerciales e industriales. Los sistemas CCTV más modernos solventan algunas de estas deficiencias con algunas mejoras, entre las que se incluyen las tecnologías digitales. Algunos ejemplos de mejoras son el control remoto de las cámaras desde una sala de control, que incluye el ajuste de parámetros como la inclinación o el nivel de zoom, o la visión nocturna y detección de movimiento, que permite al sistema cambiar a un estado de alerta ante posibles intrusiones. Los sistemas de vigilancia digitales, más conocidos como sistemas de vigilancia-IP, transmiten la información en una red donde existen componentes heterogéneos conectados (entre ellos ordenadores que procesan las imágenes) y almacenan las imágenes con grabadores de vídeo digital (DVR). Las ventajas que presentan las tecnologías digitales frente a las analógicas en los sistemas de seguridad son múltiples, entre ellas, mayor facilidad de uso y mantenimiento, capacidades avanzadas de búsqueda, procesamiento de imágenes, accesibilidad remota, fácil integración con otros sistemas, posibilidad de comprimir las imágenes, etc.

Por otra parte, los sistemas de vigilancia de segunda generación combinan las tecnologías de los sistemas CCTV y vigilancia-IP con algoritmos de Visión por Computador e Inteligencia Artificial. Es decir, los sistemas de segunda generación intentan reducir la participación del hombre en las tareas de vigilancia, interpretando en la medida de lo posible los eventos y comportamientos que se producen en el entorno monitorizado. Actualmente, la interpretación de sucesos en entornos reales no es un problema totalmente resuelto y existe un gran número de líneas de investigación abiertas en este campo. Ni siquiera existe un consenso lo suficientemente claro sobre las tecnologías y metodologías más adecuadas para ofrecer soluciones óptimas al problema. En cuanto a los principales retos que se plantean actualmente en este tipo de sistemas podríamos destacar tres de ellos: el primero es la representación del conocimiento de cualquier dominio del mundo real para poder interpretar las situaciones que ocurren en él. Es decir, el sistema artificial debe conocer los elementos que participan en la escena y qué relaciones existen entre ellos. Normalmente, estos dominios suelen ser bastante complejos y la representación del conocimiento no es una tarea trivial. A lo largo de los últimos años, numerosos autores han propuesto herramientas y métodos de adquisición de conocimiento para facilitar al experto la definición ontológica de los elementos del dominio. Sin embargo, el uso de este tipo de herramientas no siempre es suficiente, sobre todo en dominios donde el número de elementos y acciones a definir es demasiado elevado. Por este motivo, el uso de algoritmos de aprendizaje semi-automáticos y automáticos es necesario para complementar la base

de conocimiento que el sistema utilizará para identificar los elementos del entorno e interpretar las acciones que llevan a cabo.

El segundo reto es la interpretación de eventos/sucesos y el tratamiento adecuado de la incertidumbre, imprecisión y vaguedad de la información generada a partir de los datos que proceden de los sensores. Para un sistema artificial es prácticamente imposible afirmar, con total certeza, qué ocurre en un entorno determinado en cualquier instante de tiempo. De hecho, no sólo es complejo para un sistema artificial; si se realizase un experimento en el cual se mostrara a cinco personas un vídeo y se les pidiera que interpretaran lo que ocurre en él, seguramente obtendríamos algunas interpretaciones comunes y otras diferentes. Por tanto, un tratamiento adecuado de la incertidumbre, imprecisión y vaguedad en un entorno monitorizado, es vital para la obtención de buenos resultados en el proceso de razonamiento e interpretación de eventos y acciones.

El tercer reto es el diseño de algoritmos eficientes que proporcionen resultados en un tiempo cercano al real. Este aspecto es vital sobre todo en los sistemas de seguridad, ya que una respuesta temprana podría evitar males mayores. Es decir, los sistemas de seguridad de segunda generación mejorarían las prestaciones de los de la primera, siempre y cuando tuvieran la capacidad de alertar al personal de seguridad sobre lo que ocurre, sin demorarse demasiado tiempo.

Finalmente, los sistemas de vídeo vigilancia de tercera generación se caracterizan sobre todo por ser altamente distribuidos. Estos sistemas utilizan los avances de las dos generaciones anteriores y están formados por un amplio repertorio de sensores distribuidos por todo el entorno observado, transmitiendo información de forma simultánea en tiempo real. La naturaleza distribuida de estos sistemas supone un gran avance para los sistemas de seguridad por varias razones. Una de ellas es que la carga de procesamiento no se encuentra centralizada y, por tanto, el sistema ofrece mayores garantías de responder en un tiempo cercano al real. Una segunda razón sería la ganancia en robustez, es decir, el sistema puede seguir trabajando perfectamente a pesar de que algunos componentes sean dañados.

En cuanto a los principales problemas a los que se enfrentan los sistemas de tercera generación, cabe destacar la dificultad de combinar múltiples dispositivos heterogéneos en una misma red. En este caso el uso de *middlewares* es apropiado para tal propósito. Y, por otra parte, la necesidad de relacionar la información procedente de diversos sensores para fortalecer el proceso de razonamiento e interpretación. Es decir, cómo encontrar la forma de indicar al sistema que parte de la información que proporciona un sensor A y un sensor B, se refiere a un mismo objeto. Por ejemplo, supóngase un vehículo en movimiento en una zona de aparcamiento, donde se encuentran instaladas varias cámaras de seguridad. Supóngase también que la primera cámara tiene dificultades para capturar el vehículo debido a la

2.1. Evolución de los sistemas de vigilancia

| 15 |

ubicación de algunos objetos en el ambiente que dificultan la visión. A partir de las imágenes capturadas por esta cámara, un núcleo de procesamiento podría determinar que el objeto pertenece a la clase vehículo con un valor de 0.2. Supóngase que una segunda cámara no tiene tantos problemas como la primera y puede determinar que el valor de pertenencia del objeto a la clase vehículo es de 0.8. Si el sistema es consciente de que el objeto que observan ambas cámaras es el mismo, podría mejorar el valor de pertenencia ofrecido por la primera cámara y, por tanto, realizar una interpretación mucho más fiel a la realidad.

La Tabla 2.1 resume las principales ventajas y problemas pertenecientes a cada una de las generaciones de sistemas de videovigilancia [186].

1ª Generación	
Descripción	Sistemas de videovigilancia analógicos de circuito cerrado CCTV.
Ventajas	■ Buen funcionamiento del sistema con baja tasa de errores. ■ A pesar de sus limitaciones, son sistemas ampliamente utilizados, debido en gran parte a la madurez de la tecnología que emplean.
Problemas	■ Utilizan técnicas analógicas para el tratamiento, la distribución y el almacenamiento de las imágenes. ■ Dependen demasiado de la actividad humana para detectar las anomalías que suceden en un entorno.
Investigación actual	■ Tratamiento, distribución y almacenamiento de imágenes mediante técnicas digitales. ■ Algoritmos de compresión de vídeo. ■ Recuperación eficiente de datos multimedia.
2ª Generación	
Descripción	Interpretación automática de escenas reales.

Ventajas	■ Incremento de la eficiencia de los sistemas de seguridad. Observación y análisis de un mayor número de situaciones al mismo tiempo. ■ Reducción de la dependencia de la actividad humana para detectar situaciones anómalas.
Problemas	■ Actualmente no existe una solución que permita realizar un razonamiento general sobre cualquier situación. Existen soluciones parciales, para razonar e interpretar sobre situaciones muy concretas (análisis de velocidades, trayectorias seguidas por objetos observados, etc). ■ Algunas situaciones anómalas imprevisibles podrían ser no detectadas. ■ Soluciones no demasiado robustas que dan lugar a un número elevado de falsas alarmas.
Investigación actual	■ Necesidad de diseñar algoritmos eficientes que ofrezcan respuestas en un tiempo cercano al real. ■ Representación de los elementos físicos de un entorno real y la relación que existe entre ellos. ■ Reconocimiento de eventos y actividades. ■ Distinción entre situaciones normales y anormales. ■ Algoritmos de aprendizaje que amplían el conocimiento que tiene el sistema sobre el entorno. ■ Anticipación a posibles acciones que podrían dañar el entorno. ■ Apoyo a la toma de decisiones y gestión de crisis en situaciones caóticas.

3ª Generación	
Descripción	Sistemas avanzados de vigilancia, constituidos por una amplia gama de sensores heterogéneos distribuidos a lo largo de un recinto vigilado. En realidad, son sistemas de segunda generación altamente distribuidos.
Ventajas	■ Además de las ventajas que proporcionan los sistemas de vigilancia de segunda generación, hay que añadir la mejora en el proceso de interpretación, ya que cuenta con información procedente de un mayor tipo de sensores. ■ Mayor consistencia ante posibles ataques, debido a la naturaleza distribuida que posee. ■ Carga de trabajo descentralizada.
Problemas	■ Distribución de la información. ■ Comunicación entre dispositivos heterogéneos. ■ Relación e interpretación de la información que procede de múltiples fuentes.
Investigación actual	■ Los temas de investigación expuestos en la segunda generación también son apropiados para la tercera generación. ■ Cada uno de los problemas que presentan los sistemas de tercera generación da lugar a una línea de investigación.

Tabla 2.1: Evolución de los sistemas de vigilancia.

2.2. Medios técnicos de seguridad

Como se comentó en la sección 2.1, los sistemas de vigilancia de tercera generación se caracterizan principalmente por estar constituidos por una

red distribuida de sensores heterogéneos, donde cada uno de ellos ofrece una funcionalidad diferente. Actualmente, existe un gama muy amplia de sensores para construir la infraestructura necesaria en aquellos lugares donde se desee llevar a cabo tareas de vigilancia. La elección de estos sensores dependerá de qué situaciones se desean detectar y la forma de hacerlo. En esta sección, con el objetivo de ofrecer una visión general, se propone una clasificación y breve descripción de los medios físicos utilizados con mayor frecuencia en los sistemas de vigilancia actuales.

Acreditación para el control de acceso a recintos y edificios. Las corporaciones que regulan el acceso a sus instalaciones disponen de un sistema de control de las entradas y las salidas. Algunos de los medios que se utilizan para la identificación de individuos son los siguientes:

- *Tarjetas de identificación.* Identifican a las personas en el sistema de control de accesos. Existen diferentes tecnologías empleadas en las credenciales como por ejemplo: código de barras visible y oculto, banda magnética de alta y baja coercitividad, tarjetas de ferrita barium, weigand, proximidad, tarjetas de lectura y escritura por contacto, tarjetas de lectura y escritura sin contacto.
- *Lectores de tarjetas.* Leen la información de las tarjetas identificativas para enviarla a los controladores. Si el proceso de identificación es correcto, los controladores envían las señales oportunas para habilitar los accesos que impiden la entrada del individuo. En algunos casos los lectores disponen de un teclado para la introducción de contraseñas de manera manual por parte del usuario.
- *Controladores de acceso.* Los controladores de acceso se encargan de procesar la información que proviene de los lectores y dispositivos de entrada para tomar decisiones, registrar y transmitir a los niveles superiores los eventos y alarmas que se producen.
- *Uso de etiquetas y tecnología RFID.* En este caso la identificación es automática y se realiza mediante radiofrecuencia. Los individuos llevan consigo unas etiquetas donde se almacena toda la información necesaria para el control de acceso. Estas etiquetas no necesitan alimentación eléctrica y disponen de un emisor-receptor RFID para responder a las peticiones por radiofrecuencia. El sistema obtiene por sí mismo la información necesaria para que el individuo pueda acceder al recinto sin que éste tenga que realizar alguna acción.

Medios físicos y electromecánicos.

- *Vallado con cable sensor.* El objetivo principal de este tipo de sensor es detectar intentos de escalada o penetración en el recinto de un intruso. Cuando la valla se mueve, las pequeñas perturbaciones se transmiten al cable el cual convierte estas vibraciones en señales eléctricas.

- *Portalones monitorizados y barreras con brazos abatibles.* Este tipo de dispositivos se suele emplear para controlar el acceso de los vehículos a un recinto. Para la apertura de los portalones y las barreras abatibles se utilizan credenciales y lectores, o bien, pueden ser activadas desde el centro de control.
- *Portillos monitorizados y tornos.* Sirven para regular el acceso de las personas a un edificio y normalmente suelen utilizarse con presencia de personal de seguridad. La apertura de estos dispositivos también se realiza mediante tarjetas identificativas y el uso de lectores.
- *Tornos de altura completa.* Este tipo de tornos se suele utilizar para controlar el acceso a instalaciones sensibles a amenazas especiales.
- *Puertas con cerradero eléctrico.* Normalmente, se utiliza para el control de acceso de personas sin presencia del personal de seguridad o a zonas de acceso restringido.



Barreras abatibles



Portalón motorizado



Portillos motorizados



Torno



Torno de altura completa

Figura 2.2: Medios físicos y electromecánicos empleados en recintos vigilados.

Medios electrónicos para la detección de presencia.

- *Barreras infrarrojas.* Este tipo de barreras proyectan haces de luz infrarroja desde un emisor hasta un receptor. Un sistema de seguridad emite una alarma de intrusión cuando se interrumpe el haz de luces.
- *Barreras de microondas.* Al igual que en las barreras infrarrojas se emplean emisores y receptores. En este caso se establece un campo electromagnético y la modificación de dicho campo por la presencia de un objeto produce la alarma.

- *Sensores de estado de puertas.* Permite al sistema conocer si una puerta está abierta o cerrada.
- *Sensores volumétricos de movimiento.* Estos tipos de sensores se pueden clasificar en dos clases de acuerdo a la tecnología que emplean: detección con infrarrojos (alarmas PIR) y microondas por efecto doppler. El primero de ellos registra un impulso cuando se recibe radiación infrarroja producto del calor de los cuerpos. Por otro lado, la tecnología de microondas emite una señal electromagnética en la banda de las microondas y ante el movimiento de un objeto se produce una variación en la señal recibida por el efecto doppler.

Sistemas de detección de incendios. Este tipo sensores está orientado a la detección precoz de la presencia de humos, llamas y el incremento anómalo de la temperatura ambiente. Algunas de las técnicas más utilizadas para la detección de incendios son:

- *Detección óptica con microcámaras instaladas en el sensor.*
- *Sistemas láser de detección lineal.*
- *Detectores térmico-termovelocímetros.*

Sensores de vídeo y audio.

- *Cámaras de seguridad.* Las cámaras de vigilancia instaladas en un entorno envían un flujo constante de vídeo que será procesado por el sistema central de seguridad. La información visual se emplea para la detección y seguimiento de objetos, así como para la identificación de situaciones anómalas que puedan implicar cierto peligro [163].
- *Micrófonos.* Los micrófonos, al igual que las cámaras de seguridad, envían de manera continua señales de audio a la sede central donde se encuentra el personal de seguridad. Los sistemas de seguridad utilizan este tipo de señales para detectar eventos tales como disparos, explosiones, rotura de cristales, apertura de puertas, etc.

2.3. Principales capas de la arquitectura de un sistema de vigilancia inteligente

Normalmente, los sistemas de vigilancia de segunda y tercera generación disponen de una arquitectura multicapa. Cada capa de la arquitectura desempeña una función bien definida y genera una serie de resultados que sirven como flujo de entrada para las capas superiores. En [186, 194] se realizan dos propuestas interesantes sobre las etapas o fases de las que

2.3. Capas de un Sistema de Vigilancia Inteligente

| 21 |

debería constar un sistema de vigilancia inteligente. La Figura 2.3 unifica en un único esquema las fases propuestas por los diferentes autores.

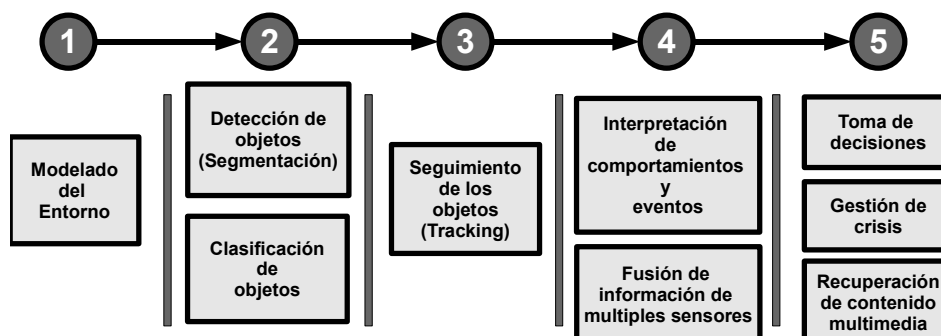


Figura 2.3: Posible secuencia de etapas que podrían formar parte de la arquitectura de un sistema de vigilancia inteligente.

La primera fase, o fase de modelización [194], consiste en definir los elementos o actores que pueden participar en el entorno, así como su propiedades y las principales relaciones que existen entre éstos y el entorno. Las máquinas, al igual que las personas, necesitan adquirir conocimiento a partir de experiencias previas para poder interpretar la información procedente de los sentidos (red de sensores en el caso de las máquinas). El modelado de entornos es una manera efectiva de interpretar y predecir los comportamientos en escenas estáticas [62]. Sin embargo, en escenas donde los actores o las condiciones ambientales pueden variar con facilidad, el modelo diseñado inicialmente podría dejar de ofrecer buenos resultados ante tales cambios. Por este motivo, las herramientas de adquisición de conocimiento que se utilizan en esta fase para modelar el entorno, deberían ser complementadas con algoritmos de aprendizaje cuando las escenas son complejas y dinámicas. De esta forma, el modelo inicial varía y se adapta a los cambios que se producen en el tiempo.

Una vez que la formalización del conocimiento del dominio observado ha sido realizada en la etapa anterior, el siguiente paso es identificar los elementos y las acciones que ocurren en cada momento en dicho dominio. Normalmente, los sistemas de seguridad se centran en el estudio de los comportamientos de personas y vehículos, que suelen ser los elementos con capacidad de movimiento que representan, salvo excepciones, una mayor amenaza para el entorno. La forma en la que los objetos móviles son detectados puede variar en gran medida en función del tipo de sensores instalados; la gama es muy amplia, desde los sensores de presencia (volumétricos, infrarrojos, etc.) hasta red de micrófonos y cámaras de vídeo vigilancia. Son éstas últimas las que proporcionan información más interesante al sistema para clasificar un objeto, como por ejemplo la forma, el tamaño o la posición. La clasificación correcta de los objetos móviles es primordial para cualquier sistema de seguridad avanzado, ya que la definición y el análisis de comportamientos para un tipo y otro

puede variar considerablemente. Por ejemplo, el comportamiento que debe tener un vehículo en una zona ajardinada puede ser totalmente distinto al comportamiento que debe tener una persona. Por tanto, es muy probable que el sistema elabore un juicio equivocado sobre el comportamiento de un objeto cuando éste realice una clasificación incorrecta. La clasificación de objetos en un entorno vigilado es considerado como un problema clásico de reconocimiento de patrones, en donde se estudia la pertenencia de un objeto a una clase a partir de un conjunto de características, que son proporcionadas en gran parte por los sensores de vídeo y audio.

El siguiente paso a la identificación y clasificación de un objeto, es el seguimiento del mismo hasta que éste deja de ser percibido por la red de sensores instalada en el entorno vigilado. El proceso en el cual se determina la posición en la que se encuentra un objeto en cada instante de tiempo se denomina *tracking*. Este proceso es un requisito fundamental para la siguiente capa, ya que sin la evolución temporal de los objetos, ésta sería incapaz de interpretar comportamientos complejos. Es decir, sin la evolución temporal de los objetos tan sólo es posible obtener conclusiones referentes a eventos simples que se producen en instantes concretos de tiempo.

A partir de la información obtenida en las etapas anteriores, el sistema de vigilancia debe intentar reconocer los comportamientos y eventos que suceden en el entorno observado. El comportamiento de un objeto viene dado por una simple acción o evento que se produce en un instante concreto, o bien por una secuencia de acciones simples a lo largo del tiempo. Si el comportamiento es complejo se representa mediante una secuencia de acciones, las cuales siguen un orden determinado y cumplen una serie de restricciones temporales. Por tanto, para que un sistema artificial pueda reconocer comportamientos complejos, es necesario que éste identifique las acciones simples cuando suceden y estudiar las relaciones temporales que existen entre ellas. La principal problemática de esta etapa es la fuerte dependencia que existe con las etapas anteriores. Una clasificación errónea de un objeto o una reproducción equivocada de su trayectoria derivaría, casi con toda seguridad, en una interpretación inapropiada de su conducta. Si estableciéramos un símil con las personas, es como si un individuo no pudiera interpretar fielmente lo que sucede a su alrededor porque ha dejado de ver u oír correctamente. Precisamente, esta dependencia con las capas anteriores puede ser uno de los principales motivos por los cuales el número de propuestas realizadas en la literatura en esta capa es bastante inferior a las anteriores, donde las tecnologías y algoritmos empleados alcanzan un mayor nivel de madurez.

Opcionalmente, la capa de razonamiento e interpretación de comportamientos puede incluir la posibilidad de fusionar la información que proviene de múltiples sensores. Esta fusión puede proporcionar grandes beneficios como por ejemplo la eliminación de ruidos y distorsiones, tratamiento del problema de la oclusión, o incluso reforzar las creencias que se mane-

jan en los procesos de esta etapa¹.

Por último, un sistema de vigilancia avanzado debería tener la capacidad de tomar decisiones y elaborar un plan de emergencia en el caso de que suceda una situación crítica. La capa encargada de esta tarea se enfrenta a una problemática similar a la de la capa anterior, es decir, existe una fuerte dependencia con los resultados que ofrece la capa de razonamiento e interpretación de comportamientos. Una interpretación equivocada puede llevar a tomar decisiones equivocadas. La toma de decisiones en una situación de crisis es un aspecto muy delicado que puede afectar a la integridad de las personas que habitan en el entorno observado. Por esta razón y debido a que las propuestas en esta etapa no han alcanzado un nivel de madurez suficiente, la amplia mayoría de instituciones que disponen de un sistema de seguridad prefieren que las decisiones las tome un experto humano en colaboración con el sistema.

Además, en esta última etapa podríamos encuadrar una actividad demandada frecuentemente tanto por el personal de seguridad de la institución donde está implantado el sistema, así como por las fuerzas de seguridad del estado: *recuperación de datos multimedia para reproducir los sucesos pasados*, los cuales han servido como prueba válida en multitud de juicios. Para la recuperación de dichos datos se formularían consultas al sistema del tipo “*obtener todos los vídeos donde una persona entra por el acceso A*”, o “*Recuperar todos los sonidos de explosiones en el mes de junio*”, o también “*Recuperar los vídeos del día 3 de Mayo en el que un coche haya circulado por el parking*”. La principal dificultad de esta tarea es la elaboración de los modelos de datos adecuados que serán utilizados tanto para la representación de los elementos que aparecen en los medios multimedia, como para la recuperación de contenidos en base a estos elementos y las activaciones de las alarmas.

2.4. Modelado de entornos reales

La construcción de sistemas basados en conocimiento implica la creación de un modelo de dominio particular [49] (por ejemplo, seguridad en plantas nucleares, aeropuertos, estaciones de tren, etc). Además modelar un dominio implica decidir cuáles son las entidades que queremos distinguir y las relaciones entre éstas. Por tanto, un modelo es una abstracción del dominio que estamos considerando. Buxton et al. [38] consideran que para interpretar correctamente las acciones y los eventos que se producen en un dominio, es necesario identificar correctamente *qué* elementos producen los eventos a partir de las acciones que realizan, *dónde* se producen y *cuándo* se llevan a cabo. Es decir, para una correcta interpretación de los sucesos

¹Una oclusión se produce cuando un objeto no es visible total o parcialmente debido a que otro objeto se interpone entre éste y la trayectoria de la cámara.

de un dominio es necesario la clasificación de los objetos a partir de sus características, así como el estudio de las relaciones espaciales y temporales.

H. Dee y D. Hogg [62] consideran que el modelado de un entorno es una manera eficaz de interpretar los eventos que suceden en un entorno, pero además, proponen modelar la psicología de los agentes² con el objetivo de anticiparse a sus acciones. Para ello, construyen un modelo donde se definen un conjunto de objetivos y sub-objetivos, y se estudia como cada agente se acerca a la consecución de estos objetivos. Estos modelos se pueden representar manualmente mediante herramientas de adquisición de conocimiento o mediante métodos automáticos tales como los que proponen Wang et al. en [196]. Wang considera que la estructura de un entorno condiciona la forma de comportarse de las entidades que participan en él, por tanto, mediante el estudio del comportamiento de las entidades es posible extraer automáticamente las principales características del entorno: zonas de entrada y salida, trayectorias habituales, velocidades a las que circulan los objetos, etc.

En las sucesivas secciones se describirán diferentes formas de representar y modelar el conocimiento de un dominio del mundo real, haciendo mayor hincapié en la representación ontológica de los elementos de un dominio y el modelado de relaciones espaciales y temporales necesarias para interpretar los eventos que suceden en el dominio.

2.4.1. Ontologías

La conceptualización de un dominio permite a un sistema artificial conocer las entidades que existen y cómo éstas se relacionan entre sí. Además, para poder comparar las conceptualizaciones es necesario que éstas se hagan explícitas. El uso de ontologías surge por la necesidad de dar explicaciones racionales, y no mitológicas, a los sucesos que ocurren en el mundo real. Según la Real Academia de la Lengua, una ontología es la *parte de la metafísica que trata del ser en general y de sus propiedades transcendentales*. Gruber [88, 89] también se refirió al término ontología como una *especificación explícita de una conceptualización*, entendiendo conceptualización como una representación abstracta y simplificada de la porción de la realidad que nos interesa. Si tenemos en cuenta las definiciones anteriores, podemos afirmar que una ontología está formada por las definiciones formales de los conceptos junto con sus propiedades y las relaciones que existen con el resto de conceptos del dominio.

El diseño de ontologías proporciona importantes beneficios a los expertos en diferentes áreas de conocimiento. Algunos de estos beneficios son los siguientes.

²En el presente trabajo se está utilizando el término *objeto* para referirse a los elementos con capacidad de acción, H. Dee y D. Hogg utilizan en [62] el término *agente* para referirse al mismo tipo de elementos.

- Posibilidad de analizar el conocimiento de un dominio.
- Compartición del conocimiento entre expertos y agentes software.
- Reutilización del conocimiento.
- Facilidad de mantenimiento.
- Separación entre la representación del conocimiento y la implementación.

Según Chandrasekaran et al. [42] existen una serie de aspectos que son comunes en cualquier ontología.

- Existen objetos en el mundo real.
- Los objetos tienen propiedades o atributos que pueden tomar valores.
- Pueden existir distintas relaciones donde participe un mismo objeto.
- Las propiedades y las relaciones pueden cambiar en el tiempo.
- Existen eventos que ocurren en distintos instantes de tiempo.
- Hay procesos que transcurren en el tiempo y en los que participan objetos.
- El mundo y sus objetos pueden estar en diferentes estados.
- Los eventos pueden causar otros eventos, cambios de estados, efectos, etc.
- Los objetos pueden estar formados por otros objetos, o un objeto puede ser parte de otro.

Para el diseño de una nueva ontología es posible seguir diferentes estrategias. Una de estas estrategias es el uso de metodologías que permiten construir las ontologías de forma sistemática, como por ejemplo, las metodologías CYC [5] y METHONTOLOGY [75]. En concreto, CYC es un proyecto iniciado en 1984 por Doug Lenat, cuyo objetivo principal es la integración de ontologías en bases de conocimiento para realizar razonamientos de tipo humano. En definitiva, se trata de dotar a un sistema artificial de “sentido común”. Además, tal como se especifica en [5], CYC tiene multitud de aplicaciones; algunas de ellas son las siguientes:

- Comportamiento inteligente de personajes en videojuegos.
- Realidad aumentada virtual.
- Ofrece mejoras para la traducción de textos de un idioma a otro.

- Mejoras para el reconocimiento del habla.
- Data Mining Semántico.
- Servicios de asistencia en comercio electrónico.

Por otra parte, METHONTOLOGY sirve como guía para la construcción de ontologías e imita el ciclo de vida del software, propuesto en el estándar IEEE 1074 y cuyas etapas principales son las siguientes:

- **Planificación** de los hitos y los objetivos que se pretenden alcanzar con el diseño y uso de la ontología.
- **Especificación**. En esta etapa se detallan las metas, los requisitos y el alcance de la ontologías.
- **Conceptualización**, es decir, la ontología como tal. El conocimiento queda organizado y estructurado mediante el uso de lenguajes de representación (tablas, UML, jerarquías, etc). El uso de estos lenguajes permite separar la representación del conocimiento de su implementación final.
- **Implementación**. Formalización del modelo conceptual con herramientas informáticas, por ejemplo, lenguajes formales (RDF, OWL, Ontolingua).

Otra alternativa a las metodologías son los lenguajes diseñados exclusivamente para la creación de ontologías. El hecho de que constituyan una segunda alternativa no quiere decir que sea excluyente con la anterior, es decir, se pueden utilizar este tipo de lenguajes y tener en cuenta las metodologías anteriores para diseñar una nueva ontología. Entre los lenguajes más representativos podemos destacar por una parte Ontolingua [88] y CycL [114], y por otra parte los lenguajes más orientados al diseño de ontologías para la web semántica como RDF, DAM+OIL o OWL.

Ontolingua [9] es un lenguaje basado en KIF (Knowledge Interchange Format) y en la *Frame Ontology* (ontología de representación del conocimiento para el modelado del mismo mediante el uso de *marcos* o *frames*). Además, *Ontolingua* es utilizado en *Ontolingua Server* un software creado por la Universidad de Stanford, ampliamente utilizado, y que sirve para la construcción de ontologías de forma colaborativa [85]. KIF es un formato de intercambio de conocimiento que tiene el suficiente poder expresivo como para representar el conocimiento declarativo contenido en multitud de bases de conocimiento de sistemas expertos [85]. Por otro lado, CycL es un lenguaje declarativo con gran expresividad (similar al cálculo de predicados de primer orden) y fue creado con el objetivo de representar y organizar grandes cantidades de conocimiento. Además, se trata de un lenguaje con una sintaxis similar a la de LISP y está basado en marcos con herencia múltiple.

Por otro lado, OWL (*Web Ontology Language*), es un lenguaje para representar el significado de los términos y las relaciones de un área de conocimiento en una ontología. En contraposición con las aplicaciones que únicamente presentan los datos al usuario, OWL permite procesar la información contenida en los documentos. Comunidades específicas (generalmente la científica y las dedicadas al desarrollo de aplicaciones para el comercio electrónico) han utilizado lenguajes anteriores a OWL para desarrollar ontologías, pero éstos no son especialmente compatibles con la web semántica y generalmente con la World Wide Web. En realidad, OWL es una revisión del lenguaje de ontologías web DAML+OIL y ofrece mayores facilidades que XML, RDF y RDF-S para expresar el significado de un término. OWL proporciona tres sublenguajes diseñados para ser utilizados por comunidades específicas de desarrolladores y usuarios. Estos tres lenguajes son: OWL Lite, OWL DL y OWL Full. El grado de expresividad en estos lenguajes es incremental, siendo OWL Full el que proporciona un mayor nivel de detalle. La elección de alguno de estos lenguajes dependerá del nivel que se precise por parte de los desarrolladores o usuarios.

A pesar de que existen lenguajes específicos para la construcción de ontologías, no quiere decir que éstos sean el único medio para hacerlo. En las siguientes secciones veremos como cualquier mecanismo, modelo o herramienta que permita definir los elementos de un dominio, sus propiedades y las relaciones que existen entre estos y el dominio, pueden ser una forma válida para construir una ontología.

2.4.2. Relaciones espaciales

La representación de los elementos de un entorno a partir de sus características físicas y las relaciones espaciales que existen entre éstos y el propio entorno, son fundamentales para que un sistema artificial que interactúa con una porción del mundo real pueda interpretar lo que sucede a su alrededor. Normalmente, la descripción de entidades en el espacio se suele realizar mediante puntos, intervalos, áreas o volúmenes. Coenen et al. [49] en su diseño de una ontología general para el razonamiento espacial proponen una serie de propiedades que debe tener cualquier objeto:

- *Identificador*. Cada objeto debe ser identificado de forma unívoca.
- *Clase* o tipo del objeto.
- *Localización* del objeto en el espacio (valores de las coordenadas x, y, z).
- *Forma* del objeto (silueta, contorno, etc).
- *Orientación* del objeto en el espacio.
- *Tamaño*. En este atributo se puede especificar el tamaño actual y el máximo y mínimo que puede alcanzar.

- *Conectividad*. Un elemento no tiene por qué ser continuo en el espacio, ya que este puede estar dividido en diferentes porciones y cada una de ellas situadas en diferentes lugares.

Buxton [38] y Coenen [49] coinciden en la necesidad de disponer de cierta información visual para poder estudiar las relaciones espaciales en un entorno determinado:

- El sistema artificial debe disponer de un modelo donde se describa cada uno de los elementos del entorno: zonas de entrada/salida, diferenciación entre las zonas habilitadas para la circulación de vehículos y peatones, zonas de aparcamiento, etc.
- El sistema debe conocer en cada momento el espacio que ocupa cada objeto en el entorno, así como sus tamaños y el lugar en el que se encuentran.
- Organización espacial de los objetos en el entorno en un momento dado.
- Información acerca de los movimientos realizados por los objetos.

Por otro lado, existen dos familias principales de relaciones espaciales [172], las basadas en los modelos de intersección creados por Egenhofer [70], y los esquemas basados en el cálculo de regiones conectadas desarrollados por Cohn [51]. Los modelos de intersección analizan la relación que existe entre dos regiones A y B teniendo en cuenta el área de estas dos regiones y sus límites (puntos que la delimitan). En concreto el modelo 4-intersección mide como A se relaciona con B teniendo en cuenta las posibles intersecciones.

- Intersección de los límites de A con los límites de B
- Intersección entre las áreas de A y B.
- Intersección entre los límites de A y el área de B.
- Intersección entre los límites de B y el área de A.

El modelo 9-intersección [70] es una extensión del anterior, en donde se tienen en cuenta también las intersecciones con los complementos de los límites y áreas de las regiones. Entendemos complemento del área de A, como todas las zonas que no pertenecen al área de A. En el caso del límite, correspondería a todos los elementos del dominio que quedan fuera de los límites de A. Existen varias extensiones del modelo 9-intersección como, por ejemplo, las propuestas por Clementini y Di Felice en [48] y Zhan en [214] que introduce el concepto de regiones difusas. Dichas extensiones, dotan al modelo 9-intersección con capacidad para tratar regiones de vaguedad.

La teoría del cálculo de regiones conectadas (RCC) parte de la idea de que el mundo espacial consta de regiones y relaciones entre regiones. Las regiones no son independientes, sino que, guardan relaciones unas con otras. Por ejemplo, una región puede ser disjunta a otra o bien una región puede ser parte de otra. En RCC las relaciones *parte-de* y *complemento* forman un pilar básico, ya que las relaciones entre dos regiones cualesquiera se determinan analizando los valores de verdad de la 3-tupla $\langle A \text{ es parte de } B \rangle$, $\langle A \text{ es parte del complemento de } B \rangle$, $\langle B \text{ es parte de } A \rangle$, de esta forma, cuando la 3-tupla tome los valores (verdadero, falso, verdadero) A y B serán iguales, debido a que A forma parte de B y B forma parte de A. La principal diferencia entre RCC y los modelos de intersección es que en los modelos de intersección las regiones son consideradas como un conjunto de puntos con una estructura topológica, mientras que los esquemas RCC están basados en la axiomatización de las regiones como primitivas.

Los esquemas RCC han sido extendidos en varias ocasiones con el fin de permitir distinciones más sofisticadas sobre cómo las regiones se relacionan unas con otras. Descripciones de RCC5 y RCC8 pueden encontrarse en [51]. Los esquemas RCC también han sido extendidos a sistemas RCC15 y RCC23 [51], los cuales tienen en cuenta las envolturas convexas de las regiones implicadas. Düntsch et al. [68] demuestran cómo un análisis de RCC5 y RCC8, desde el punto de vista de álgebras de relación (consisten en un conjunto de relaciones binarias cerrado bajo una serie de operaciones y que contiene unas relaciones particulares: unión, intersección, simétrico y complementario), da lugar a extensiones como RCC7 y RCC10. En el modelo RCC existen una serie de relaciones base: (C) Conexión, (DR) Discreto, (O) Solapamiento, (P) Parte de, (PP) Parte propia, (DC) Desconexión, (EC) Conexión externa, (PO) Solapamiento parcial, (EQ) Igualdad, (TPP) Parte propia tangencial y (NTPP) Parte propia no tangencial.

RCC5 al igual que el resto de esquemas RCC se basa en los valores de verdad de la 3-tupla (A es parte de B, A es parte del complemento de B, B es parte de A). Las posibles combinaciones que se pueden dar son 16, ya que la 3-tupla toma tres valores de verdad y estos a su vez sólo pueden ser verdadero o falso ($2^3 = 16$ posibilidades). De las 16 posibilidades, RCC5 sólo tiene en cuenta cinco de ellas, de ahí su nombre. Si las regiones A y B satisfacen los axiomas del cálculo de regiones conectadas (reflexividad y simetría), las relaciones posibles entre A y B en el esquema RCC5 son las siguientes:

- A es disjunta de B
- A y B se superponen.
- A está dentro de B.
- A contiene a B.
- A y B son iguales.

El esquema RCC8, al igual que RCC5, es un esquema basado en RCC. A diferencia del anterior, el conjunto de relaciones base entre superficies está formado por ocho tipos de relaciones: DC, EC, PO, EQ, TPP, NTPP, TPPi, NTPPi. La Figura 2.4 ilustra gráficamente estas relaciones.

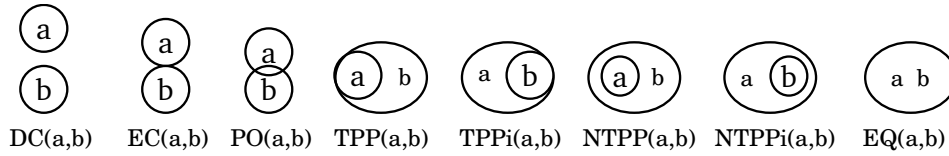


Figura 2.4: Relaciones básicas del esquema RCC8.

En la mayoría de los sistemas de visión artificial, los objetos y las zonas de un entorno se representan mediante áreas o polígonos formados por un conjunto de puntos. El estudio de la distancia y el grado de intersección que existe entre las regiones, proporciona el tipo de relación espacial que existe entre ellas (relaciones definidas en los modelos de intersección de Egenhofer o en los esquemas basados en el cálculo de regiones conectadas). Uno de los métodos utilizados con mayor frecuencia para determinar la distancia entre dos regiones definidas mediante un conjunto de puntos es la distancia de Hausdorff [62]:

$$h(X, Y) = \max_{x \in X} \{ \min_{y \in Y} \{ ||x - y|| \} \} \quad (2.1)$$

La distancia de Hausdorff es una medida asimétrica que se define como la máxima distancia de un conjunto de puntos X al punto más cercano de un conjunto Y . Por otro lado, para determinar el grado de intersección entre una región A y una región B es necesario calcular la cantidad de puntos de A que pertenecen al área de B . El algoritmo creado por W. Randolph Franklin [201], basado en un algoritmo anterior [164], determina si un punto en el plano $p = (x, y)$ se encuentra en los límites o el interior de un polígono definido mediante un conjunto de puntos. A continuación se muestra la implementación de dicho algoritmo en el lenguaje de programación C:

Algoritmo npoly

```

1  int npoly(int nvert, float *vertx, float *verty, float testx,
2          float testy)
3  {
4      int i, j, c = 0;
5      for (i = 0, j = nvert-1; i < nvert; j = i++) {
6          if ( ((verty[i]>testy) != (verty[j]>testy)) &&
7              (testx < (vertx[j]-vertx[i]) * (testy-verty[i])
8                / (verty[j]-verty[i]) + vertx[i]) )
9              c = !c;
10     }
11     return c;
12 }
13 
```

Donde *nvert* es el número de vértices del polígono, *vertx* y *verty* los arrays que contienen las coordenadas *x*, *y* de los vértices del polígono y, finalmente, *testx* y *testy* representan las coordenadas del punto *p*. La función devuelve el valor 1 si el punto *p* se encuentra en el interior del polígono, y 0 en caso contrario. Para calcular la intersección entre una región *A* y otra *B*, se utiliza la función anterior con todos los puntos de *A*. El grado de intersección de la región *A* con la región *B*, será el número de puntos de *A* que pertenecen a *B* entre el número total de puntos de *A*.

2.4.3. Relaciones temporales entre eventos

Al igual que sucede con las relaciones espaciales, la representación del tiempo y las relaciones temporales entre los eventos también son necesarias para que un sistema artificial pueda interpretar lo que sucede a su alrededor cuando interactúa con una parte del mundo real. La representación acerca de los aspectos dinámicos del mundo real es un problema de interés para múltiples disciplinas.

James F. Allen propuso en 1994 un álgebra de intervalos para la representación de acciones y eventos. Dicha representación se basa en la lógica temporal de intervalos, más expresiva y natural que la mayoría de propuestas realizadas previamente en el área de la Inteligencia Artificial. El trabajo propuesto por Allen para la representación del tiempo está ampliamente extendido en la comunidad científica y son muchos los autores que lo utilizan para representar y, posteriormente, reconocer los eventos y acciones en escenas reales. Según Allen et al. [15] la representación de acciones y eventos facilita a un agente inteligente la realización de las siguientes tareas:

- *Predicción*. Dada la descripción del escenario, incluyendo las acciones y eventos ¿qué puede suceder?.
- *Planificación*. Elaboración de planes para alcanzar los objetivos propuestos.
- *Explicación*. Dada una serie de observaciones, encontrar la mejor explicación a lo sucedido.

Además, Allen también expone en [15] una serie de propiedades que considera esenciales para la representación general de las acciones y los eventos:

- Las acciones y los eventos transcurren a lo largo del tiempo. Mientras que algunos eventos pueden ocurrir en instantes de tiempo concretos, la mayoría de ellos ocurren en intervalos de tiempo.

- La relación entre acciones/eventos y sus efectos es compleja. Algunos efectos sólo se convierten en verdaderos al final de la acción, por ejemplo “poner un libro en la mesa”, no es verdadero hasta que el libro está totalmente encima de la mesa.
- Las acciones y los eventos pueden interactuar de forma compleja cuando coinciden en el tiempo.
- En el mundo real pueden ocurrir eventos externos que no están previstos y por tanto pueden afectar a la secuencia de acciones planeada por un agente artificial.
- El conocimiento del mundo es incompleto y en muchas ocasiones impredecible, por esta razón las predicciones sólo se pueden realizar en base a las asunciones ciertas.

J. F. Allen, antes de describir el álgebra basada en la lógica temporal de intervalos, expone otras formas de representar el tiempo que también son válidas, incluso apropiadas para sistemas que satisfagan ciertas condiciones [14]. Algunas de ellas son (i) representación del tiempo mediante fechas absolutas, (ii) pseudo-fechas y, finalmente, (iii) representación basada en grafos con propagación de restricciones.

La primera de ellas se basa en la representación de eventos instantáneos con el uso de fechas absolutas, siempre y cuando sea posible obtenerlas y el sistema lo permita. Una fecha absoluta consta del año, día del año, hora, minuto y segundo en el que se produjo el evento. Por ejemplo, (2008, 110, 10, 4, 50) equivaldría al día 110 del año 2008 a las 10:04 a.m y 50 segundos [14]. La gran ventaja de este método es que proporciona algoritmos eficientes para el tratamiento de fechas, ya que tan sólo es necesario una sencilla comparación entre valores numéricos. Sin embargo, este método sólo es aplicable en sistemas en donde siempre se pueda obtener la fecha absoluta en la que se ha producido un evento.

Para los casos en los que no es posible obtener las fechas absolutas, Allen propone un segundo método en el que el tiempo se representa mediante pseudo-fechas. En este caso, se conoce el intervalo de tiempo aproximado en el cual el evento ha sucedido. Dicho evento está formado por el instante más temprano (e_1) en el que el evento ha podido comenzar y el instante más tardío (e_2) en el que puede finalizar. Normalmente, para calcular el instante de tiempo en el que ha sucedido el evento se opta por la aproximación del punto medio: $\frac{time(e_1) + time(e_2)}{2}$. El gran inconveniente de este método es que se pierde la información real del momento en el que transcurren los eventos y sus duraciones.

El tercer método, al igual que muchos de los trabajos propuestos en Inteligencia Artificial para definir sistemas de razonamiento temporal, utiliza técnicas de propagación de restricciones. Estos sistemas utilizan representación basada en grafos, donde cada nodo se corresponde con un

instante de tiempo y todos ellos están relacionados mediante los arcos que los unen (las relaciones entre un nodo y otro se especifican en el arco que los une mediante etiquetas) [14]. La Figura 2.5 muestra un ejemplo de un grafo en el que se representan tres eventos simples y las restricciones que existen entre éstos. La Figura 2.6 muestra una ejemplo de cómo varía el grafo anterior cuando se añaden nuevos eventos y nuevas restricciones.

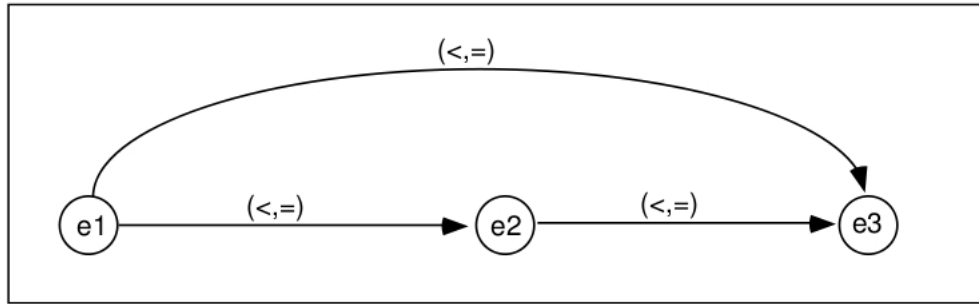


Figura 2.5: Ejemplo de la representación de eventos simples y sus relaciones mediante un grafo [14].

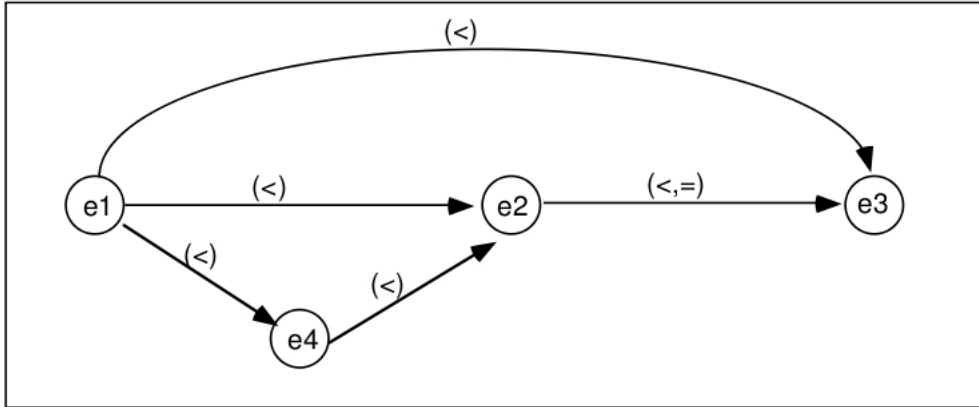


Figura 2.6: Ejemplo de propagación de restricciones tras añadir $e_1 < e_4$ y $e_4 < e_2$, [14].

Finalmente, el álgebra de intervalos de Allen esta basada, como bien indica su nombre, en la representación del tiempo mediante intervalos y las relaciones que existen entre estos. La teoría temporal comienza con un objeto primitivo, el *periodo de tiempo*, y una relación primitiva: *Meets* [15]. Intuitivamente, un periodo de tiempo es el tiempo asociado a un evento que ocurre en el mundo. Además, dos periodos m y n se encuentran en el tiempo si m precede a n (comienza antes), no existe tiempo entre ambos, y m no finaliza durante el transcurso de n . A partir de la definición de periodo de tiempo y la relación primitiva *meets* se establecen los siguientes axiomas:

- (1) $\forall i, \exists j, k. Meets(j, i) \wedge Meets(i, k)$

- (2) $\forall i, j, k, l. Meets(i, j) \wedge Meets(j, k) \wedge Meets(k, l) \supset \exists m. Meets(i, m) \wedge Meets(m, l)$
- (3) $\forall i, j, k, l. Meets(i, j) \wedge Meets(i, k) \wedge Meets(l, j) \supset Meets(l, k)$
- (4) $\forall i, j, k, l. Meets(k, i) \wedge Meets(k, j) \wedge Meets(i, l) \wedge Meets(j, l) \supset i = j$
- (5) $\forall i, j, k, l. (Meets(i, j) \wedge Meets(k, l)) \subset Meets(i, l) \otimes (\exists m. Meets(k, m) \wedge Meets(m, j)) \otimes (\exists m. Meets(i, m) \vee Meets(m, l))$. Donde $\otimes$ representa “o - exclusivo”.

A partir de estos axiomas se define un rango completo de relaciones entre periodos de tiempo (ver Figura 2.7). Precisamente este rango de relaciones temporales es una de las claves del éxito del trabajo de Allen, ya que no sólo ofrecen un medio para representar las relaciones entre periodos de tiempo sino que también permiten establecer restricciones temporales, muy importante en los sistemas de seguridad. Es decir, con este tipo de relaciones es posible definir reglas o restricciones temporales que indiquen cuándo se deben producir los eventos en un entorno vigilado y que relaciones deben existir entre ellos para ser considerados eventos normales.

2.5. Adquisición de conocimiento en sistemas de vigilancia inteligente

En la Sección 2.4 se hizo especial hincapié en la necesidad de disponer de un modelo donde se definieran los diferentes elementos que forman parte del entorno, los objetos con capacidad de acción y sus principales características, así como los eventos que se pueden producir como consecuencia de las acciones. Este modelo incluye el conocimiento necesario para poder interpretar la información procedente de los sensores y puede ser definido directamente por un experto en el dominio, o bien, mediante el uso de técnicas de adquisición de conocimiento. El modelado de escenas proporciona una manera efectiva de detectar comportamientos anómalos en entornos estáticos donde las condiciones ambientales apenas varían [62]. Sin embargo, en escenas dinámicas donde las condiciones ambientales pueden variar en el tiempo con cierta frecuencia, el modelo inicial puede dejar de tener validez, dando lugar a continuas interpretaciones erróneas por parte del sistema. En este caso es cuando las técnicas de adquisición de conocimiento y algoritmos de aprendizaje automático cobran especial importancia, ya que permiten una redefinición del modelo para adaptarse las nuevas condiciones.

Además, el proceso de aprendizaje se puede llevar a cabo en tiempo real, o bien, se puede lanzar a partir de un histórico de datos obtenidos durante un periodo prolongado de tiempo. En los sistemas de vigilancia avanzados, los sensores captan y envía datos de manera continua, obteniendo de esta forma

2.5. Adquisición de conocimiento en sistemas de vigilancia inteligente | 35 |

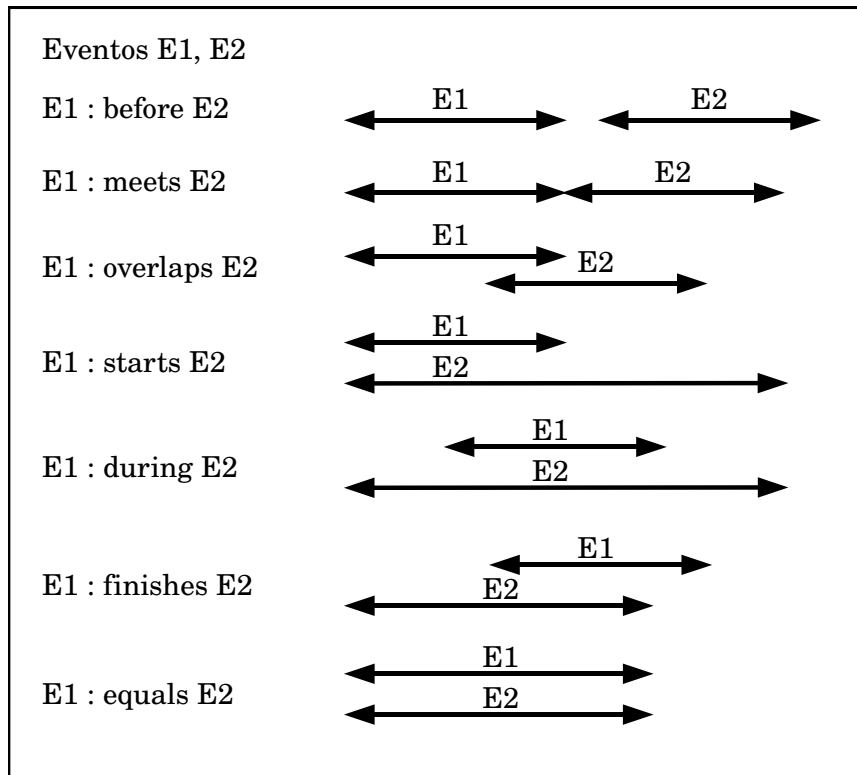


Figura 2.7: Relaciones temporales entre periodos de tiempo definidas en el álgebra de Allen [14].

grandes volúmenes de información a lo largo del día. Según P. Angelov [16] el proceso de aprendizaje a partir de tales volúmenes de datos se convierte en impracticable. Por este motivo, los algoritmos de aprendizaje en tiempo real que no varían demasiado la estructura del conocimiento adquirido ante nuevas muestras son, quizás, más apropiados para este tipo de sistemas.

Las técnicas de aprendizaje y adquisición de conocimiento actuales para el modelado de escenas se pueden clasificar en tres grandes grupos, de acuerdo a la participación del experto:

- Modelado manual de escenas mediante el uso de herramientas de adquisición de conocimiento.
- Uso de algoritmos de aprendizaje con la supervisión de un experto.
- Algoritmos de aprendizaje automático.

Si el modelado se hace de forma manual, normalmente se suelen emplear herramientas que faciliten esta tarea. Un ejemplo es el sistema propuesto por

Spirito et al. [170], que permite estudiar eventos en una estación subterránea de metro. En concreto, el sistema tiene la capacidad de detectar personas situadas en una posición muy cercana a las vías o bien caminando por ellas. El sistema dispone de una herramienta que permite al usuario dibujar las zonas de peligro cuando las cámaras han sido instaladas. La Figura 2.8 muestra unas capturas de las herramientas descritas en [170]. Otro ejemplo de modelado manual de escenas se puede encontrar en [216], en este caso mediante el uso de modelos ocultos de Markov (HMM). El sistema dispone principalmente de dos capas, una primera capa de bajo nivel donde se modelan las acciones básicas que pueden suceder en un *parking* externo, y una segunda capa donde se emplea una variación de los modelos de Markov para detectar acciones complejas, en función del análisis y la relación temporal de las acciones simples detectadas.

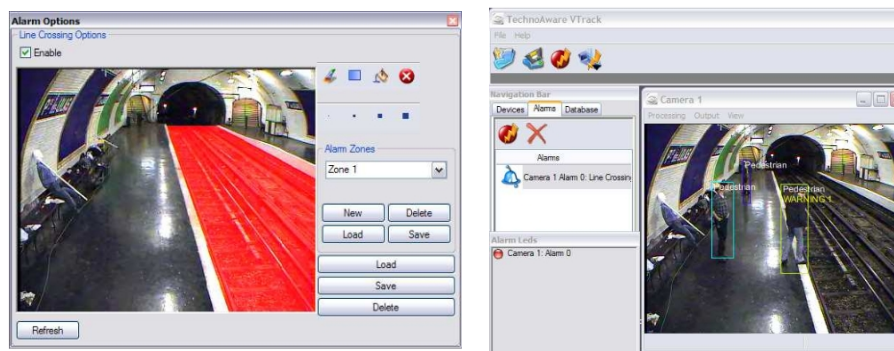


Figura 2.8: Captura de las herramientas pertenecientes al sistema presentado en *Automatic detection of dangerous events for underground surveillance*, escrito por Spirito et al. [170]. La imagen de la izquierda muestra la herramienta que permite dibujar las zonas de peligro a las que los peatones no deben acercarse; en este caso las vías del tren. La imagen de la derecha muestra como el sistema segmenta y detecta a las personas que se mueven en la escena y emite mensajes de alarmas cuando los objetos están demasiado cerca de las zonas de peligro.

Por otro lado, las características de la escena o el entorno donde se implanta el sistema de seguridad pueden determinar en gran medida el tipo de técnicas de aprendizaje que se deben emplear. Existen escenas en las que es complicado obtener muestras de algunas de las clases, por ejemplo, escenas donde es fácil obtener situaciones normales pero no situaciones anómalas. En este caso los algoritmos de tipo *one-class* son adecuados para resolver este problema [181, 180]. También existen escenas en las que obtener muestras de cada clase es una tarea complicada y los algoritmos a emplear deben aprender a partir de un conjunto reducido de ejemplos [215, 219].

Otra posible clasificación de las técnicas de aprendizaje podría ser de acuerdo a las partes del modelo de la escena que se desean aprender:

- Algoritmos que aprenden y clasifican los objetos de acuerdo a sus

2.5. Adquisición de conocimiento en sistemas de vigilancia inteligente [37]

características físicas (peatón, grupo de personas, coche, camión, etc).

- Algoritmos que realizan un estudio espacio-temporal de los eventos y cambios de estado que suceden en el entorno, con el objetivo de clasificar los comportamientos y las diferentes situaciones.

En cuanto al primer tipo de algoritmos existen numerosos trabajos como por ejemplo [53], donde se propone un algoritmo de aprendizaje que estudia la forma o silueta de los objetos a partir de características espaciales para generar un modelo de cada clase. El principal problema del método propuesto es la baja interpretabilidad de los resultados, que necesitan la interacción humana para el posterior etiquetado. O. Brdiczka et al. [210] propone un algoritmo para adquirir el contexto de un entorno a partir de la información procedente de las cámaras. El algoritmo genera un modelo llamado modelo de situación, donde se definen las entidades, clases de objetos, situaciones y relaciones entre las situaciones. H. Noor [139] también propone un algoritmo de aprendizaje para la generación de modelos de objetos a partir de imágenes en dos dimensiones. Dicho algoritmo aprende la estructura topológica de una imagen a partir de un conjunto de datos medibles y la forma de los objetos. Posteriormente, el conocimiento adquirido es aplicado sobre secuencias de vídeo para el reconocimiento de objetos. Dos de las grandes ventajas del algoritmo propuesto es que no necesita la calibración previa de la cámara y, además, los modelos generados siguen siendo válidos aunque las imágenes sean rotadas o escaladas.

Con respecto al segundo tipo, una de las tareas más comunes es el estudio de las trayectorias que siguen los objetos. D. Makris y T. Ellis [126] propusieron un algoritmo para el aprendizaje de las trayectorias realizadas por peatones en entornos exteriores monitorizados. El algoritmo observa y analiza la frecuencia de las trayectorias llevadas a cabo por los transeúntes con el objetivo de elaborar un modelo espacial, en el cual se representan las trayectorias en base a las regiones visitadas con mayor frecuencia. Además, dicho modelo incluye las zonas de entrada, salida y puntos intermedios entre ambos tipos de zonas. N. Johnson y D. Hogg [104] realizaron una propuesta similar para el aprendizaje de la distribución de trayectorias mediante el uso de redes neuronales. Una vez que la red neuronal había sido entrenada, se utilizaba para la identificación de trayectorias y la predicción de movimientos atípicos. El principal inconveniente de este trabajo, a diferencia del desarrollado por Makris et al. [126], es que las trayectorias aprendidas no son etiquetadas y, por tanto, los resultados del algoritmo de aprendizaje son poco interpretables. David Hogg defiende en sus trabajos la necesidad de estudiar las trayectorias de los objetos con capacidad de movimiento para identificar comportamientos complejos, ya que las personas siempre se mueven por objetivos y para cumplirlos necesitan recorrer un camino. En [61] y [104] propone un modelo estadístico para el aprendizaje de objetivos y subobjetivos. Los objetivos suelen coincidir con las zonas de salida de la escena, mientras que los subobjetivos son los puntos intermedios que las personas deben recorrer para alcanzar sus objetivos.

Además, el algoritmo extrae automáticamente la topología de la escena donde se incluyen las zonas de entrada/salida y los posibles obstáculos. D. Hogg expone que las personas para cumplir sus objetivos suelen realizar el recorrido más corto y simple, con el menor número de giros o cambios de sentido posibles. Por tanto, los obstáculos existentes en una escena se pueden detectar mediante el estudio de la desviación entre la trayectorias realizadas por la personas y las trayectorias ideales.

La información visual procedente de las cámaras de seguridad no sólo se puede utilizar para determinar las trayectorias de los objetos, existen trabajos, como por ejemplo [178, 179, 204], donde también se aprende la forma de andar específica de ciertos individuos, acciones cotidianas que se pueden llevar a cabo en un oficina [21], o acciones básicas que se pueden realizar en zonas de aparcamiento [216], como por ejemplo, la entrada de una persona a un vehículo, persona que se aleja de un coche, rodear un vehículo, etc.

Un inconveniente común en gran parte de los trabajos citados anteriormente, es la baja interpretabilidad del conocimiento adquirido. Los sistemas de seguridad de última generación no sólo deberían tener la capacidad de aprender a partir de ciertas observaciones, sino que también, deberían tener la capacidad de justificar cada una de las interpretaciones que realizan de la forma más concisa y clara posible. De esta forma, un operador humano podría comprobar fácilmente si el sistema está analizando correctamente el estado actual del entorno observado y realizar modificaciones directamente sobre la base de conocimiento si fuera necesario.

2.6. Proceso de segmentación

El objetivo principal del proceso de segmentación en tareas de vigilancia es identificar aquellos objetos que son ajenos a la escena, es decir, aquellos que no forman parte del entorno como tal. La mayoría de métodos de segmentación en vídeo utilizan información espacio-temporal y están basados principalmente en tres aproximaciones:

- Background subtraction
- Temporal differencing
- Optical flow

La técnica *background subtraction* consiste en la diferenciación de un elemento en movimiento con respecto a una imagen modelo del entorno. Este método calcula las regiones en movimiento a partir de la comparación píxel a píxel entre una imagen de referencia y una imagen actual capturada por una

cámara. El gran inconveniente de este método es que es demasiado sensible a cambios en el entorno como por ejemplo las condiciones de iluminación. Por tanto, se debe utilizar un modelo de escena lo más representativo posible para evitar el mayor número de errores. Normalmente, este método de segmentación no suele ser utilizado en espacios abiertos por los problemas mencionados anteriormente, sin embargo ofrecen buenos resultados en espacios cerrados donde las condiciones de iluminación no suelen variar con frecuencia, como por ejemplo en estaciones subterráneas de metro y ferrocarril. Algunos trabajos destacados en este campo son [92, 129, 171]. La Figura 2.9 muestra un ejemplo de los resultados obtenidos con la utilización de este método sobre un entorno virtual.

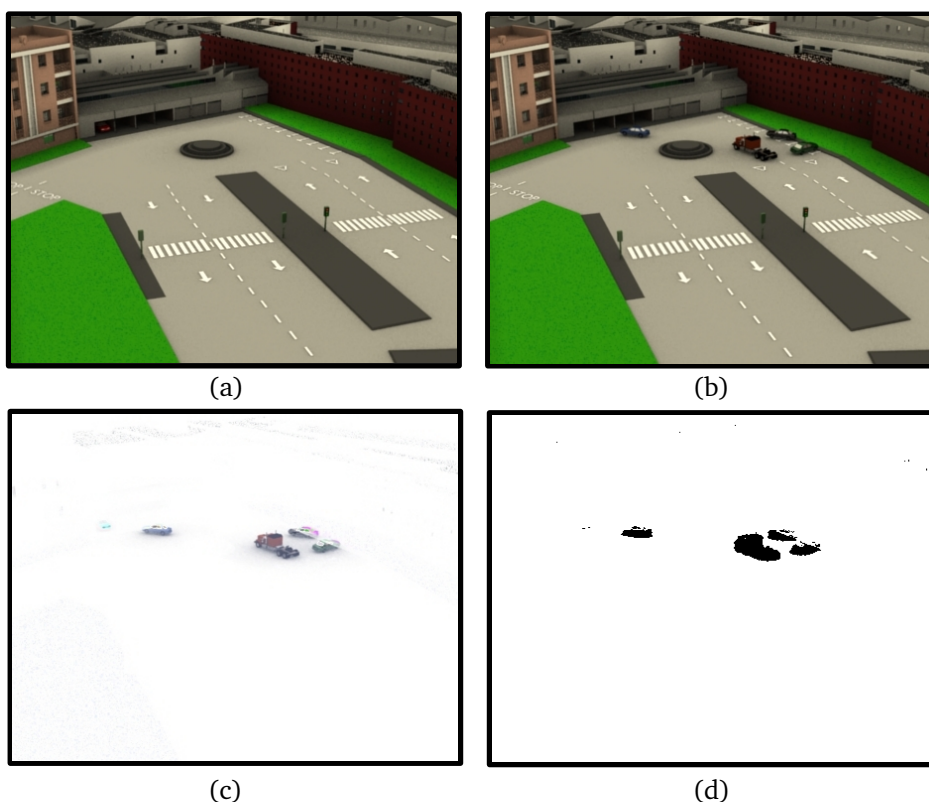


Figura 2.9: Segmentación de objetos en vídeo a partir de una imagen modelo. La Figura (a) muestra la imagen modelo de un escenario virtual. La Figura (b) muestra el frame actual captado por la cámara de vigilancia. La Figura (c) muestra la diferencia entre la imagen modelo y el frame actual. Finalmente, la Figura (d), muestra tintados de color negro aquellos píxeles que son considerados diferentes con respecto a la imagen modelo.

Por otro lado, la técnica de diferenciación temporal o *Temporal differencing* extrae las regiones en movimiento mediante el análisis de una secuencia de imágenes y el estudio de la evolución de los píxeles a lo largo del

tiempo. A diferencia del método anterior, éste funciona mejor en entornos dinámicos debido a sus capacidades adaptativas [117], sin embargo, tiene un rendimiento más pobre en la extracción de los píxeles relevantes pertenecientes a objetos en movimiento.

El método de flujo óptico u *Optical flow* extrae las regiones en movimiento a partir de las características que ofrecen los vectores de movimiento de los objetos a lo largo del tiempo [20, 73]. Los algoritmos de este tipo de segmentación ofrecen como gran ventaja la detección de objetos incluso con cámaras en movimiento. Sin embargo, la mayoría de ellos implican un alto coste computacional [194] y difícilmente pueden ser aplicados directamente sobre flujo de vídeo en tiempo real. En [20] se discute con mayor profundidad los métodos de flujo óptico para la segmentación de objetos en escenas de vídeo.

Normalmente, el proceso de segmentación consiste en el procesamiento de imágenes extraídas directamente del flujo de vídeo procedente de una cámara. Sin embargo, en la literatura existen algunas propuestas que permiten trabajar directamente sobre vídeo comprimido con datos de bajo nivel y en tiempo real. Un ejemplo es el algoritmo propuesto por L. Rodríguez et al. en [155], donde se aplican técnicas de softcomputing directamente sobre vídeo comprimido en formato MPEG. Dicho algoritmo de segmentación se basa en la fuzzificación de los datos de movimiento como son los macrobloques, blobs y vectores de movimiento. Estos términos son ampliamente utilizados y reconocidos en Visión por Computador. A continuación, se realiza una breve descripción de cada uno de ellos:

- **Macrobloque:** un macrobloque es la unidad básica en la que se dividen las imágenes para su codificación. Contiene la información correspondiente a una región de 16x16 píxeles.
- **Blob:** un blob es un grupo o colección de macrobloques que determinan la silueta de un objeto detectado en el proceso de segmentación.
- **Vectores de movimiento:** este tipo de vectores almacena información relativa al movimiento de un conjunto de píxeles o macrobloques. Cada macrobloque almacena sus propios vectores de movimiento, definen una distancia y una dirección, y poseen dos magnitudes *right_x* y *down_y*.

Además, el formato de codificación MPEG presenta tres tipos de cuadros o *frames*:

- **Frames de tipo I:** para la codificación sólo se tiene en cuenta dicho frame.
- **Frames de tipo P:** en este caso, para la codificación se utiliza como referencia un frame anterior de tipo I y se emplea estimación y

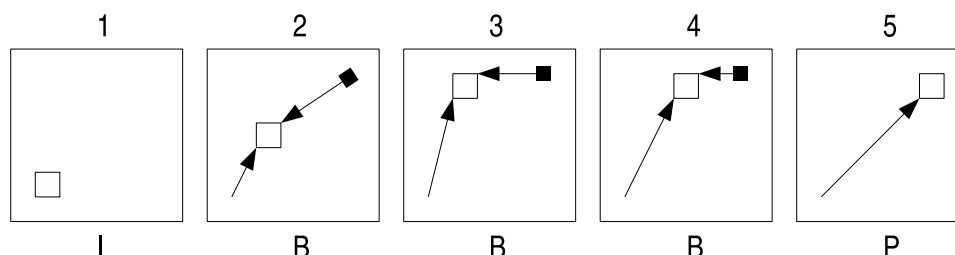


Figura 2.10: Ejemplo de relación entre frames de tipo I, P y B

compensación de movimiento. Es decir, el codificador se centra en aquellas partes de la imagen que hayan sufrido mayores cambios con respecto al frame anterior de tipo I.

- **Frames de tipo B:** al igual que en los frames de tipo P, también se utiliza estimación y compensación de movimiento, y se hace referencia a otros frames, tanto pasados (frame de tipo I) como futuros (frame de tipo I o P).

La Figura 2.10 muestra un ejemplo de relación entre frames de tipo I, P y B. El primer frame es de tipo I y en él se puede apreciar donde está situado originalmente el macrobloque (marcado con un recuadro). El último frame, el cual es de tipo P, muestra la posición final del macrobloque, así como el vector de movimiento que determina cómo se ha desplazado el macrobloque desde el frame 1 hasta el 5. Para calcular dicho vector, el codificador utiliza como referencia el último frame de tipo I, en este caso, el primer frame. Los frames intermedios 2,3,4 son de tipo B; en cada uno de ellos el macrobloque dispone de dos vectores de movimiento. El primero (en la parte inferior del macrobloque) se obtiene a partir de la referencia al último frame de tipo I, y el segundo (situado en la parte superior del macrobloque) a partir de la referencia a un frame futuro de tipo I o P (frame 5).

Una alternativa al formato de vídeo MPEG, es el estándar de codificación H.264/AVC. Dicho estándar fue desarrollado conjuntamente por el ITU-T Video Coding Experts Group (VCEG) y el ISO/IEC Moving Picture Experts Group (MPEG) en el año 2003. En realidad H.264 es la evolución del formato MPEG, de hecho es equivalente a la versión MPEG-4 Part 10 o MPEG-4 AVC (codificación avanzada de vídeo). Este nuevo formato de codificación de vídeo presenta numerosas mejoras con respecto a las versiones anteriores de MPEG. Algunas de las más destacadas son las siguientes:

- Incluye dos nuevos tipos de frames: SP (Switching P) y SI (Switching I), ambos utilizados para codificar la transición entre flujos de vídeo.
- Mejora notable en la compensación de movimiento. Los macrobloques de 16x16 píxeles pueden ser, a su vez, divididos en subgrupos de 16x8,

8x16 o 8x8 píxeles.

- Uso de múltiples vectores de movimiento por macrobloque. En las versiones anteriores de MPEG, el número máximo de vectores de movimiento por macrobloque es de dos y sólo en frames de tipo B. Por el contrario, en el estándar H.264 un macrobloque puede almacenar hasta 32 vectores de movimiento, obteniendo así, una información mucho más detallada sobre el movimiento del macrobloque que forma parte de un objeto.
- Mejora en el grado de compresión y la calidad visual final del vídeo.

Teniendo en cuenta estas mejoras, C. Solana et al. proponen en [169] un algoritmo de segmentación en tiempo real sobre vídeo comprimido H.264/AVC, que mejora los resultados ofrecidos por el algoritmo original en el que se basa [155]. Al igual que en [155], el nuevo algoritmo emplea técnicas de softcomputing para detectar objetos en movimientos en entornos monitorizados.

2.7. Clasificación de objetos

Los sistemas de vigilancia avanzados no sólo deben tener la capacidad de identificar los objetos que participan en la escena (proceso de segmentación) y reconocer las acciones que llevan acabo, sino que también deben ser capaces de diferenciar si estas acciones implican comportamientos normales o anómalos. Es decir, el objetivo prioritario de un sistema de estas características es alertar al personal de seguridad cuando suceda alguna situación peligrosa y, para ello, el sistema debe ser capaz de discernir entre lo que es normal y anómalo.

Para realizar esta diferenciación, la capa encargada de analizar las situaciones que suceden en el entorno necesita la clase a la que pertenece cada objeto. Cada clase determina la forma correcta en la que debe comportarse cada objeto o agente con capacidad para realizar acciones. Es decir, cada una de ellas engloba un conjunto de reglas o normas, tal que la satisfacción de las mismas implica un comportamiento normal o correcto. Por el contrario, si alguna de las normas se infringe, el comportamiento debería ser considerado como anómalo. Por ejemplo, si un sistema de vigilancia detecta un objeto en movimiento en un jardín, sería normal siempre y cuando ese objeto fuera una persona o grupo de personas. Sin embargo, si fuera un vehículo, el comportamiento no sería normal ya que, normalmente, no está permitido que los vehículos circulen por zonas ajardinadas.

En el proceso de clasificación, el caso ideal sucede cuando el sistema puede determinar con absoluta certeza la clase a la que pertenece un objeto.

Sin embargo, en la mayoría de los casos el sistema maneja varios valores de pertenencia del objeto a diferentes clases. En este caso, un sistema artificial con capacidad para tratar la incertidumbre tendría criterio para elegir la clase más adecuada o, incluso, manejar varias alternativas al mismo tiempo. Además, es importante tener en cuenta que el valor de pertenencia de un objeto a una clase puede variar a lo largo del tiempo y, por tanto, el sistema también debería estar preparado ante tales cambios. Es decir, cualquier objeto, mientras permanezca en el radio de alcance de los sensores, puede realizar movimientos, acciones o bien que se produzcan cambios en sus características físicas que ofrezcan al sistema motivos suficientes para variar los grados de pertenencia que fueron establecidos inicialmente.

Según Wang et al. [194] actualmente existen dos categorías principales en la que se engloban la mayoría de los métodos para la clasificación de objetos en movimiento:

- **Clasificación basada en formas.** Una de las posibles formas que existen para clasificar un objeto es a partir de su silueta [77, 82, 129, 218]. En esta categoría se establecen mecanismos de comparación entre los patrones de formas definidos previamente para cada una de las clases y la silueta actual del objeto. Estos mecanismos proporcionan un valor numérico que indica el grado de pertenencia de un objeto a una clase. Finalmente el sistema optará por aquella clase cuyo valor de pertenencia sea mayor. Por ejemplo, el sistema VSAM [52] tiene en cuenta la información de bajo nivel que proporciona un flujo de vídeo MPEG (grado de dispersión, área, etc.) para clasificar los objetos en cuatro clases: persona, vehículo, grupo de personas y, finalmente, aglomeración de formas no identificadas. D. Damen y D. Hogg [60] proponen un método para detectar individuos que transportan objetos a partir del estudio de sus siluetas. El principal inconveniente de este método es la incapacidad para detectar objetos de tamaño reducido u ocultos tras la ropa del individuo. Cómo se verá a continuación en la siguiente categoría, este problema puede ser solventado con el análisis de los movimientos.
- **Clasificación basada en el movimiento.** Una alternativa al método anterior es la clasificación de objetos en función de los movimientos que éstos realicen [58]. Para distinguir entre personas y vehículos, la mayoría de autores parte de la premisa de que las personas varían con cierta facilidad su forma (objetos no rígidos o deformables) y realizan un movimiento periódico en su desplazamiento. En cambio, con los vehículos sucede todo lo contrario, no cambian de forma con frecuencia (a menos que realizan algún giro) y no tiene un movimiento cíclico. En [178, 179, 204] se estudia el movimiento que realizan las personas al andar, no sólo para diferenciarlas de otros tipos de elementos, sino también para identificar a una persona en particular. Es decir, cada persona tiene una forma característica de andar y, a partir de estas características, se diseña un patrón que

permite identificar a cada persona. Además, los autores de los trabajos citados anteriormente demuestran que las personas tienen una forma característica de andar mientras cargan con uno o varios objetos. Este tipo de algoritmos se utiliza en vigilancia para detectar el abandono de objetos, que se produce cuando una persona que carga con un objeto varía la forma de andar después de abandonarlo.

Bremond [32] añade una categoría más a las anteriores, la cual clasifica objetos en movimiento basándose en características como la iluminación, los colores o las texturas. Oren et al. [141] hacen uso de este tipo de características para detectar peatones en movimiento en entornos urbanos.

2.8. Seguimiento de objetos mediante técnicas de tracking

Normalmente, el proceso de *tracking* es un proceso posterior a la identificación y clasificación de objetos³. Dicho proceso se basa en la identificación y el seguimiento de los objetos a lo largo del tiempo, hasta que éstos dejan de ser percibidos por la red de sensores instalada en el entorno vigilado. Además, es un requisito fundamental para la capa encargada de interpretar los eventos y comportamientos, ya que sin la evolución temporal de los objetos, ésta sería incapaz de interpretar comportamientos complejos. Es decir, sin la evolución temporal de los objetos tan sólo es posible obtener conclusiones referentes a eventos simples que se producen en instantes concretos de tiempo. La Figura 2.11 muestra un ejemplo del proceso de *tracking* en el que se ha realizado el seguimiento de un vehículo en una secuencia de seis frames. Tal como se puede apreciar en la figura, el vehículo es identificado y marcado mediante una elipse en cada uno de los frames. La conexión entre estas elipses determina el camino que ha recorrido el vehículo a lo largo del tiempo. Wang et al. [194] y Bremond [32] clasifican los algoritmos de *tracking* en tres categorías: *tracking* basado en regiones, *tracking* basado en contornos activos y, finalmente, *tracking* basado en modelos. El hecho de que existan varias categorías no implica que pueda existir soluciones mixtas, como bien se especifica en [102].

- **Tracking basado en regiones.** El *tracking* basado en regiones quizás sea la técnica más popular para el seguimiento de objetos a lo largo del tiempo. La mayoría de algoritmos pertenecientes a esta categoría están basados en los filtros de Kalman [106] o derivados, los cuales son utilizados para eliminar el ruido o interferencias que se producen debido en gran parte a las condiciones de iluminación.

³En realidad la clasificación de un objeto puede depender de los movimientos o caminos que éste siga. Por tanto, el proceso de clasificación, en algunos casos, puede ser posterior al proceso de tracking.

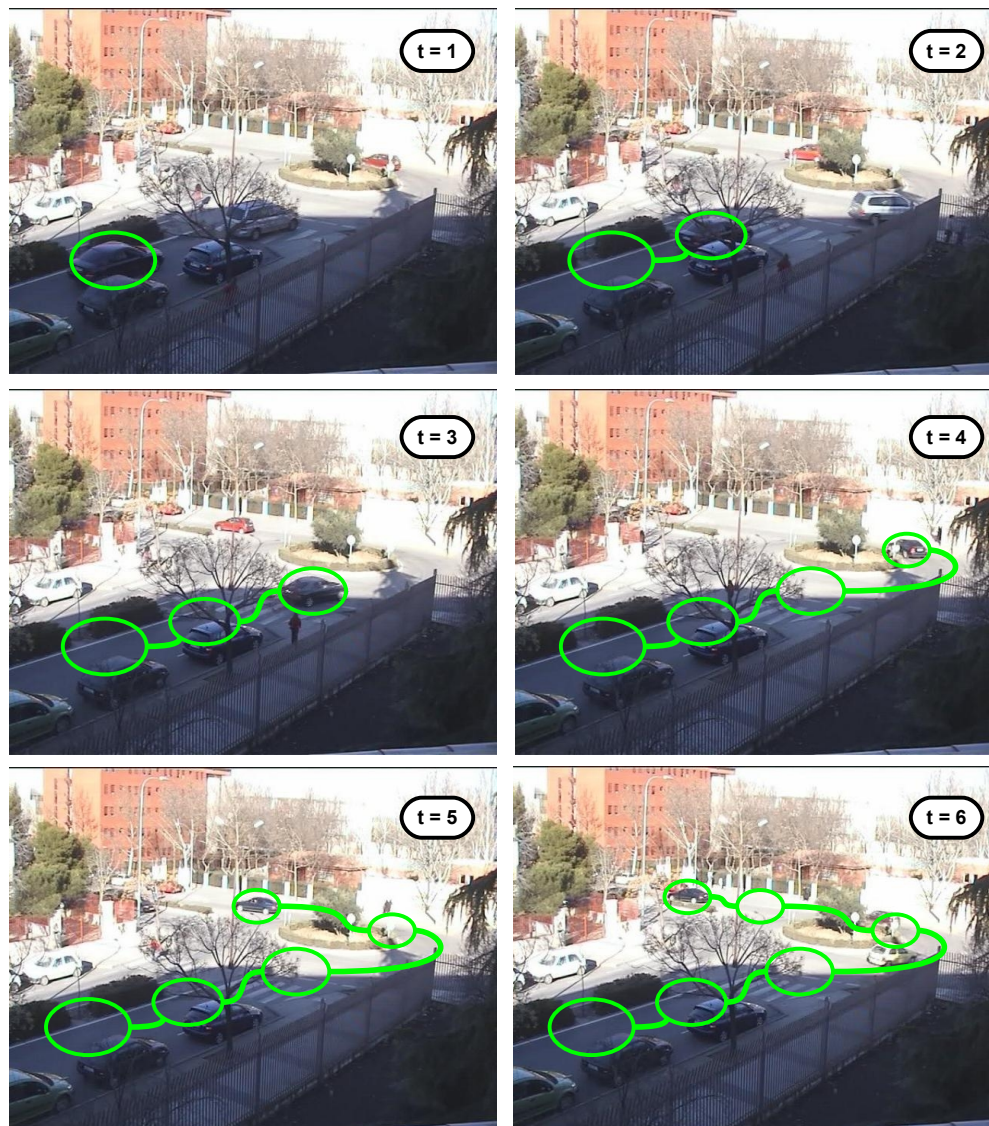


Figura 2.11: Ejemplo del resultado ofrecido por un proceso de *tracking* de un vehículo en una secuencia de seis frames. Las imágenes fueron tomadas desde una cámara situada en el grupo de investigación Oreto de la Universidad de Castilla-La Mancha.

Básicamente, esta técnica [108, 110] realiza el seguimiento de los objetos teniendo en cuenta la variación de ciertas regiones de la imagen, las cuales corresponden con los objetos en movimiento. Para ello, este tipo de algoritmos dispone de un modelo que corresponde con el entorno capturado por una cámara, y que es utilizado en cada *frame* para substraer los objetos en movimiento. En realidad, este método es equivalente a realizar el proceso de segmentación (*background subtraction*) en cada uno de los *frames*.

- **Tracking basado en contornos activos.** Esta técnica se basa en la elaboración de los contornos de los objetos móviles y de su actualización en cada uno de los *frames* [22, 132]. Los algoritmos pertenecientes a esta categoría están enfocados principalmente a la extracción de la forma de los sujetos. Dicha extracción proporciona una descripción mucho más detallada de los objetos que la que se obtiene de los métodos pertenecientes a la categoría anterior.
- **Tracking basado en modelos.** En esta categoría se elaboran a priori modelos bidimensionales y tridimensionales de los objetos que pueden actuar en una escena, mediante herramientas CAD y técnicas avanzadas de visión por computador. El siguiente paso es establecer un mecanismo de correspondencia entre los objetos que se encuentran en movimiento actualmente en la escena y estos modelos. De esta forma es posible identificar los objetos móviles y realizar el seguimiento de los mismos. Algunos de los trabajos más destacados en esta categoría son [107, 138, 156].

El principal problema al que se enfrentan los algoritmos de *tracking* es la **oclusión**. Se produce oclusión cuando un objeto es ocultado parcial o totalmente por otro elemento de la escena. Cuando se produce este hecho, la mayoría de sistemas que realizan *tracking* de objetos tienen serias dificultades para reproducir totalmente la trayectoria seguida por un objeto, incluso para no perder su referencia (en muchas ocasiones, después de producirse una oclusión, un objeto puede ser considerado por el sistema como un objeto nuevo). Actualmente, existen multitud de propuestas que ofrecen el tratamiento de oclusiones parciales y totales en escenas de vídeo, la mayoría de ellas se basan en el uso de múltiples cámaras [65, 184, 185].

2.9. Interpretación de eventos y conductas

Uno de los mayores retos en el área de la Inteligencia Artificial es conseguir que los sistemas que interactúan con entornos del mundo real sean capaces de interpretar lo que sucede a su alrededor, a partir de la información procedente de los dispositivos que perciben el entorno. En el caso de los sistemas de vigilancia, no es suficiente con la capacidad de interpretar lo que sucede, sino que además deben diferenciar entre

2.9. Interpretación de eventos y conductas

|47|

comportamientos normales y anómalos para la activación de alarmas, y reducir así la dependencia de la actividad humana en la tarea de la vigilancia.

Existen diferentes criterios para la clasificación de los eventos que se pueden producir en un determinado entorno. Si tenemos en cuenta las características temporales, los eventos se pueden clasificar de la siguiente manera:

- **Eventos simples:** determinan la situación o el estado de un objeto observado en un instante concreto de tiempo.
- **Eventos compuestos:** secuencia de eventos simples relacionados temporalmente. El estudio de las relaciones temporales entre los eventos simples asociados a un objeto particular, determinan su comportamiento.

Por el contrario, si tenemos en cuenta las normas o reglas establecidas en un determinado entorno, los eventos se pueden clasificar de la siguiente manera [78]:

- Eventos normales.
- Eventos sospechosos.
- Eventos anómalos.

Los eventos normales que se pueden producir en un entorno, a diferencia de los anómalos, son bien conocidos. Muchas de las situaciones anómalas que pueden suceder en un determinado entorno son imprevisibles incluso para los mayores conocedores de dicho entorno. En la literatura existen numerosas propuestas para la interpretación de eventos, muchas de ellas se basan únicamente en la definición y detección de situaciones anómalas, otras en la definición de situaciones normales y, finalmente, propuestas que combinan ambos tipos de eventos. Los sistemas que se basan únicamente en la definición de situaciones anómalas no son capaces de detectar aquellas situaciones anómalas que no han sido definidas previamente, aspecto que es crítico en un sistema de seguridad. Por otro lado, los sistemas que definen únicamente las situaciones normales son capaces de determinar que una situación no normal ha sucedido, pero no saben de cuál se trata, por tanto, presentan serias dificultades a la hora de tomar decisiones. Ante los inconvenientes que presentan las dos aproximaciones anteriores, la mejor opción es definir las situaciones normales de un entorno y las situaciones anómalas conocidas. De esta forma, cuando una situación anómala sucede y no ha sido definida previamente, el sistema considerará la situación como sospechosa y mantendrá al sistema en alerta.

El estudio de los comportamientos y eventos que se producen en un determinado entorno depende en gran medida del tipo de objetos o entidades

sobre los que se desee prestar especial atención. La mayoría de autores en la literatura coincide en acotar el conjunto de tipos de objetos o entidades para reducir la complejidad del problema. Este conjunto suele estar formado por los siguientes elementos:

Objetos con capacidad de acción:

- Personas.
- Grupo de personas y multitudes.
- Vehículos.

Entidades sin capacidad de acción que pueden variar su estado:

- Lugares (zonas o regiones del entorno).
- Objetos estáticos relevantes: puertas, tornos, papeleras, máquinas, etc.

Modelar el comportamiento y los cambios de estado de cualquiera de los elementos mencionados anteriormente de manera general y que sea válido para cualquier entorno es un problema prácticamente inabordable. Por este motivo, todas las propuestas realizadas en la literatura suelen centrar su estudio en un contexto determinado. A. Dey [64] define contexto como cualquier información que se pueda utilizar para caracterizar la situación de un objeto o entidad. A su vez, define situación como la descripción de los estados de los objetos o entidades relevantes para el sistema artificial.

A continuación se realiza una clasificación de las técnicas más utilizadas para la interpretación de eventos y se describen algunos de los trabajos más relevantes en cada una de ellas. Dichas técnicas han sido empleadas en diferentes áreas como por ejemplo el análisis de comportamientos en base a las trayectorias realizadas por los objetos, análisis de la forma de andar de las personas, control de acceso a edificios mediante reconocimiento facial y de la voz, detección de abandono de objetos, etc. La mayoría de ellas tienen en común la posibilidad de representar la evolución temporal del estado de los elementos observados y el tratamiento de la incertidumbre e imprecisión.

2.9.1. Dynamic time warping (DTW)

DTW es una técnica utilizada para medir la similitud entre dos secuencias que varían en tiempo o velocidad. Esta técnica, que se utiliza normalmente para el reconocimiento de la voz, ha sido adaptada recientemente en el análisis de vídeo para el reconocimiento de movimientos humanos. Según Wang et al. [194], la potente simplicidad conceptual y la facilidad de su implementación son dos de las grandes ventajas que presenta esta técnica,

además permite la comparación de secuencias representadas con diferentes escalas de tiempo.

Algunos de los trabajos destacados con esta técnica son los de Takahashi et al. [176] que utiliza DTW para reconocer acciones realizadas con las manos por personas que son diestras, y el trabajo de Bobick et al. [27] que analiza una secuencia determinista de estados para reconocer gestos faciales. Niennattrakul et al. [137] proponen un método para la recuperación de datos multimedia basado en la combinación del algoritmo de *clustering* k-means y el análisis temporal DTW, con el que es posible realizar análisis forenses. Otro ejemplo es el caso de Tanawongsuwan et al. [177] que proponen el uso de DTW para el análisis de la forma de andar de los humanos estudiando la evolución temporal del movimiento de los huesos que forman parte del esqueleto. Finalmente, Ming Hsiao et al. [96] emplean DTW para realizar una fusión temporal procedente de múltiples sensores y obtener así una valoración global sobre el estado actual del entorno observado.

2.9.2. Máquinas de estados finitos

Las máquinas de estados finitos (*Finite State Machine (FSM)*) constan de un conjunto de estados, acciones y transiciones. En el caso concreto de los sistemas de vigilancia, los estados representan las situaciones en las que se puede encontrar un objeto observado por los sensores ubicados en el entorno. Según las acciones que realice un objeto y si éstas son reconocidas, se producirán unas transiciones de estado u otras. El sistema artificial reconoce que un objeto ha realizado cierto comportamiento cuando éste llega a un estado final a partir de una secuencia de acciones.

Rawlinson et al. [149] utilizan una máquina de estados finitos para controlar las trayectorias de un robot en un entorno observado por una cámara. Esta máquina de estados incorpora tres posibles comportamientos: evitar obstáculos, deambular y, finalmente, aproximarse al objetivo. Cupillard et al. [56] emplean máquinas de estados finitos para representar la secuencia de acciones que implican ciertos comportamientos en estaciones de metro subterráneas. El método propuesto es capaz de detectar situaciones tales como lucha entre dos personas, vandalismo, bloqueo del acceso a la línea de metro en los tornos, aglomeraciones de personas, etc (ver Figura 2.12).

Los mismos autores proponen en [57] el uso de técnicas similares para analizar los comportamientos de grupos de personas a partir de la información procedente de múltiples cámaras. Wilson et al. [200] presentan un método para la clasificación temporal de gestos naturales a partir de imágenes de vídeo con el uso de máquinas de estados finitos (identificación de gestos claves). Un trabajo similar es el descrito en [55], donde se analizan comportamientos humanos a partir de los gestos, para la activación final de alarmas ante comportamientos anómalos. Finalmente, Bremond et al. [33] emplean autómatas finitos para reconocer comportamientos de vehículos



Figura 2.12: Situaciones en estaciones subterráneas de metro detectadas en [56].

observados desde cámaras aéreas.

2.9.3. Gramáticas libres de contexto y lógicas inductivas

Una gramática libre de contexto se define como una 4-tupla formada por los siguientes elementos: $G = (N, T, P, S)$, donde N es el conjunto finito de símbolos no terminales, T es un conjunto finito de símbolos terminales, P es un conjunto finito de producciones, y S es el denominado símbolo inicial de la gramática. Las producciones de una gramática libre de contexto son de la forma $N \rightarrow (T \cup N)^*$. Las gramáticas libres de contexto pueden generar lenguajes libres del contexto y éstos a su vez pueden ser reconocidos por un autómata de pila determinístico o no determinístico.

Uno de los trabajos más destacados con gramáticas libres de contexto para el análisis de comportamientos y ampliamente citado es el de Ivanov et al. [101], en el que se propone un sistema de vigilancia basado en vídeo para vigilar zonas exteriores de aparcamiento. El sistema consta principalmente de un módulo de *tracking*, un generador de eventos y un *parser* de interpretación semántica para reconocer eventos y acciones realizadas por los objetos. Christensen et al. [47] también proponen el uso de gramáticas libres de contexto para la interpretación de comportamientos en sistemas de vídeo vigilancia.

2.9. Interpretación de eventos y conductas

| 51 |

Los trabajos anteriores se caracterizan sobre todo por su poder descriptivo, característica que comparten los trabajos propuestos en [124, 135] donde el conocimiento se representa mediante lógicas inductivas. En concreto, los autores de [124, 135] proponen un sistema de visión cognitivo capaz de aprender un conjunto de protocolos de actuación a partir de una serie de percepciones en escenas dinámicas. Es decir, el agente artificial observa los movimientos y acciones de agentes humanos, por ejemplo, en juegos de mesa. Posteriormente el agente virtual, a partir de los protocolos de actuación adquiridos, es capaz de interactuar con el mundo real de la misma forma que lo hacían los agentes humanos en momentos anteriores.

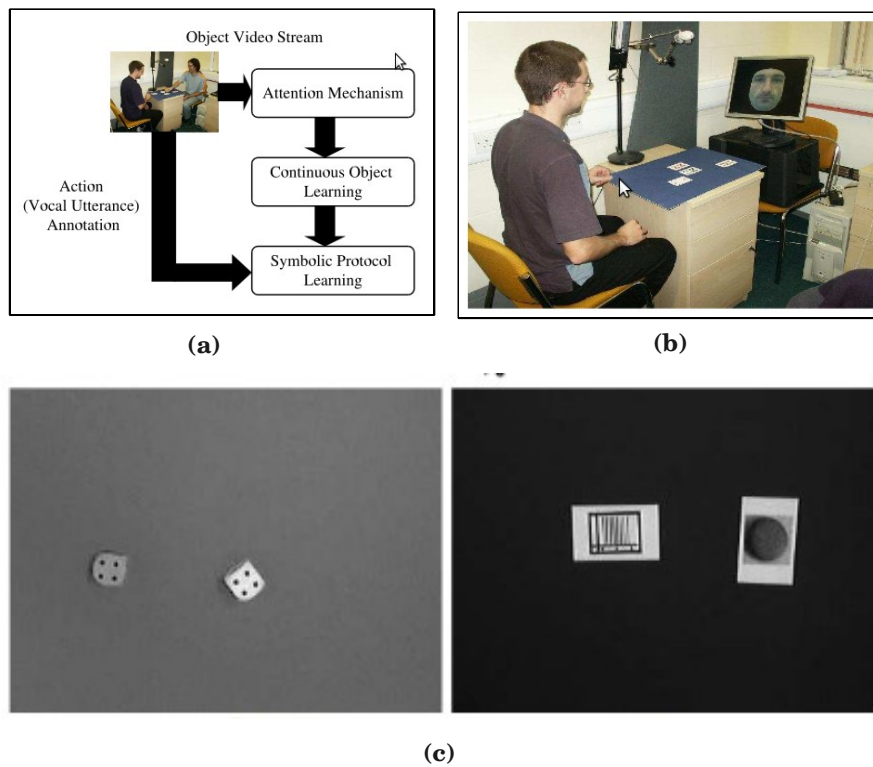


Figura 2.13: Sistema cognitivo capaz de asimilar protocolos de actuación a partir de percepciones visuales [124, 135]. La Figura (a) corresponde con el esquema general del sistema propuesto en [124], en la Figura (b) se puede apreciar la interacción entre agentes humanos y virtuales, y finalmente, la Figura (c) muestra un par de ejemplos de percepciones visuales a partir de las cuales el sistema aprende los protocolos de actuación.

2.9.4. Algoritmos de *Clustering*

Las técnicas de *clustering* también han sido ampliamente utilizadas para la interpretación y clasificación de eventos en entornos monitorizados. Un ejemplo de ello, es el trabajo propuesto por Piciarelli y Foresti [146] para la clasificación de trayectorias y posterior interpretación de comportamientos. Este algoritmo, a diferencia de otros algoritmos de *clustering* propuestos para alcanzar el mismo objetivo [24, 104, 127, 115], crea y actualiza de manera incremental los *clusters* a medida que obtiene la información espacio-temporal procedente de las cámaras de seguridad. Tal como se puede apreciar en la Figura 2.14 una trayectoria se representa mediante una secuencia de *clusters* organizados en estructura de árbol, donde cada uno de las ramas que une los nodos dispone de la probabilidad de transición. Cuando una nueva trayectoria se produce, el algoritmo propuesto crea nuevos nodos y reestructura el árbol. El análisis de comportamientos llevado a cabo considera como trayectorias normales aquellas que se realizan con mayor frecuencia, sin tener en cuenta la clase a la que pertenecen los objetos. Es decir, aunque las trayectorias se representan mediante información temporal y espacial, ésta es insuficiente para los sistemas de vigilancia. Por ejemplo, las trayectorias que realizan habitualmente los vehículos no son normales si las llevan a cabo los peatones.

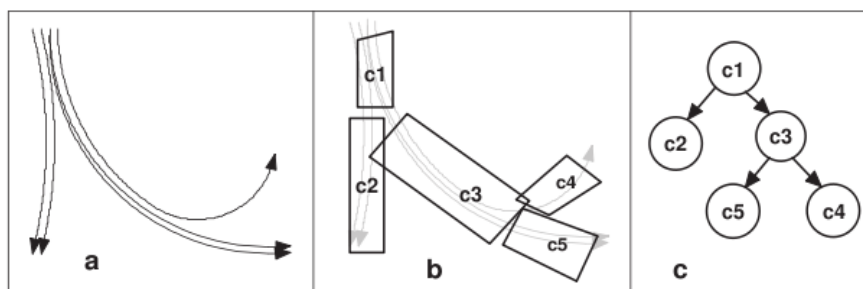


Figura 2.14: Interpretación de trayectorias mediante árboles cuyos nodos representan los *clusters* [146].

Otro método propuesto recientemente para la identificación y clasificación de trayectorias basadas en técnicas de *clustering* es el propuesto por Schreck et al. [160]. Este método realiza un análisis visual interactivo haciendo uso de mapas de Kohonen, y combina con éxito técnicas de análisis automático con la supervisión e interacción de agentes humanos expertos. Otra definición más amplia de trayectoria se presenta en [148], en la que se emplean seis características diferentes, todas ellas relativas a información espacial: posición, distancias, ángulos. En comparación con la propuesta de Piciarelli et al. [146] la información espacial para representar las trayectorias es mucho más rica, sin embargo, carece de información temporal. Anjum et al. [17] también proponen métodos de *clustering* basados en múltiples características para representar y clasificar las trayectorias pero, al igual que

el método anterior, estas características se basan únicamente en información espacial. La mayoría los métodos propuestos para el análisis de trayectorias se centran únicamente en información espacial e identifican con éxito las trayectorias que realizan los objetos detectados en una escena. Sin embargo, existe una carencia importante de otros tipos de información, cuando el análisis se aplica en sistemas de vigilancia.

2.9.5. Redes Neuronales

Una red neuronal es un modelo matemático inspirado en el sistema nervioso de los seres humanos. Cada red esta formada por un conjunto de neuronas o nodos unidos mediante conexiones, y cada conexión posee un peso numérico que tienen gran influencia en el proceso de aprendizaje. Cada una de las neuronas tiene un conjunto de entradas y una sola salida; la salida se calcula en base a tres funciones: función de propagación (normalmente el sumatorio de cada una de las entradas multiplicada por su peso), función de activación (modifica la anterior), y función de transferencia (esta función acota el valor de salida de la función de activación en un determinado intervalo).

Aunque las redes neuronales son utilizadas con mayor frecuencia para la clasificación de objetos a partir de sus características físicas, existen algunas propuestas del uso de este modelo matemático para la interpretación de eventos. Un ejemplo es el trabajo propuesto por Jhonson et al. [104] en el que se utilizan redes neuronales para aprender las trayectorias de los objetos y el reconocimiento de eventos en base a los movimientos realizados por los objetos detectados en la escena. En concreto, describen el movimiento de un objeto en términos de una secuencia de flujo de vectores, cada uno de los cuales consta de cuatro componentes que representan la posición y velocidad del objeto en una imagen plana. Por otro lado, Sumpter et al. [173] proponen el uso de redes neuronales para aprender relaciones espacio-temporales y utilizarlas posteriormente para la predicción de los comportamientos de los objetos, es decir, no sólo proponen un método para interpretar lo que ya ha sucedido, sino que también intentan anticiparse a posibles acciones futuras. De esta forma el sistema podría evitar situaciones dañinas antes de que sucedieran. J. Owens et al. [142] proponen una red neuronal auto-organizada para la detección de movimientos inusuales realizados por peatones en zonas de aparcamiento público. La Figura 2.15 muestran ejemplos de trayectorias analizadas mediante este método, los puntos blancos representan los recorridos considerados como normales, mientras que los de color negro son considerados como inusuales. Las flechas de color blanco representan la dirección del movimiento.

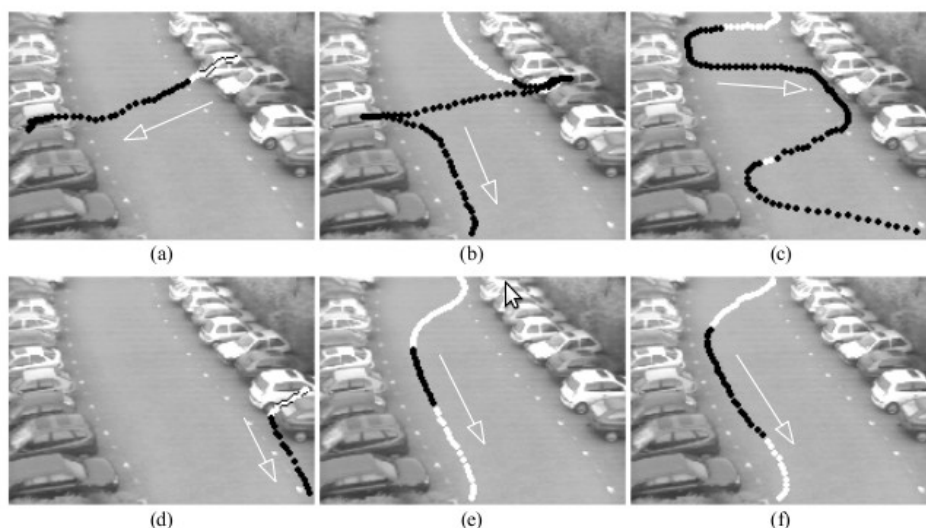


Figura 2.15: Análisis de trayectorias para la detección de movimientos inusuales realizados por peatones, propuesto en [142].

2.9.6. Redes Bayesianas

Las Redes Bayesianas son modelos gráficos que representan relaciones probabilistas entre múltiples variables de interés [93]. Una Red Bayesiana es en realidad un grafo acíclico dirigido cuyos nodos representan las variables y los arcos que conectan los nodos las dependencias que existen entre dichas variables [81]. La creación de una nueva red implica la definición de las variables, la relación entre las variables y una asignación de probabilidades a priori, existiendo para ello técnicas manuales y automáticas. Este modelo ha sido ampliamente utilizado en los últimos años para la construcción de sistemas expertos, los cuales están constituidos por una base de conocimiento y un proceso de inferencia. El modelo de razonamiento o inferencia probabilístico, que se basa en la aplicación del Teorema de Bayes, ofrece dos variantes principales: i) propagación de la evidencia: se calcula la probabilidad a posteriori de variables no observadas dada la evidencia disponible y, ii) abducción total o parcial; en el primer caso se busca la configuración para todas las variables no observadas que tienen máxima probabilidad, mientras que en el segundo caso únicamente se tiene en cuenta un subconjunto de las variables no observadas para la configuración de la probabilidad máxima.

Las Redes Bayesianas han sido ampliamente utilizadas para la interpretación de eventos y conductas. Un trabajo destacado es el propuesto por H. Buxton y S. Gong en [37, 38] para la monitorización e identificación de actividades en el control de tráfico urbano. El sistema diferencia entre comportamientos de bajo y alto nivel según el nivel conceptual; los de bajo nivel

están relacionados directamente con el valor que toman las variables como, por ejemplo, la velocidad o la orientación de los objetos, mientras que los de alto nivel implican una secuencia de eventos relacionados temporalmente, por ejemplo, adelantamiento de un vehículo en una carretera. En un trabajo posterior, Howarth y Buxton [94] realizan descripciones conceptuales de los comportamientos, ofreciendo una representación del conocimiento más cercana al lenguaje natural y mejorando así la interpretabilidad de la salida ofrecida por el sistema de visión artificial.

P. Remagnino et al. [151, 152] propusieron sistemas multi-agentes para la interpretación de comportamientos entre peatones y vehículos. En el sistema propuesto se asigna de forma dinámica un agente a cada objeto detectado por el sistema de visión y sobre el que se está realizando un seguimiento. Cada uno de estos agentes hace uso de Redes Bayesianas donde se modelan los comportamientos, el sistema considerará que el comportamiento de un objeto es aquel que sea más probable. La peculiaridad de este sistema es que no sólo analiza el comportamiento de los objetos individualmente sino que además estudia la interacción entre objetos. Los mismos autores también emplean Redes Bayesianas para la clasificación de objetos en el proyecto VIGILANT [153].

Otros trabajos más recientes que emplean Redes Bayesianas para la interpretación de comportamientos en sistemas de vigilancia inteligentes, son los propuestos por Zajdel et al. [213], Wang et al. [195] y Saleemi et al. [159]. En el primero de ellos se propone un sistema para detectar personas y analizar sus comportamientos haciendo uso de varias cámaras de seguridad de forma simultánea. El sistema está dividido en dos niveles de procesamiento: el de más bajo nivel se encarga de detectar las personas y el de mayor nivel trata de estimar la identidad de los objetos identificados así como la trayectoria que siguen. El sistema es evaluado en escenas interiores, en concreto en una oficina. En el segundo trabajo se proponen métodos de aprendizaje no supervisados para generar automáticamente modelos jerárquicos bayesianos que posteriormente se utilizarán para clasificar y detectar actividades complejas llevadas a cabo por grupos numerosos o multitudes. Finalmente, el tercer trabajo guarda ciertas similitudes con el segundo ya que se proponen métodos para el aprendizaje automático de las actividades que suceden en una escena tras largos periodos de observación. Los métodos propuestos se basan principalmente en la observación de variables espaciales y cómo están relacionadas temporalmente. Las trayectorias se encuentran entre las posibles actividades que el modelo puede aprender, y considera como normales aquellas que suceden con mayor frecuencia, de tal forma que las trayectorias inusuales son clasificadas como anómalas.

Las Redes Bayesianas son modelos apropiados para evaluar los eventos en un instante de tiempo concreto, sin embargo, no ofrecen demasiadas facilidades para trabajar con eventos compuestos [32]. Ante este problema se han propuesto algunas soluciones como por ejemplo las Redes Bayesianas

dinámicas [36], cuyo principal inconveniente es la dependencia del tiempo de segmentación (frame/rate); cuando este tiempo varía, la red debe ser entrenada de nuevo. Una segunda solución es el uso de estados temporales ocultos propuestos por S.Gong y T.Xiang [86], los cuales guardan ciertas similitudes con los Modelos Ocultos de Markov (HMM). Precisamente estos modelos son considerados como Redes Bayesianas Dinámicas y son ampliamente utilizados para el reconocimiento de secuencias temporales, resolviendo así el problema planteado anteriormente.

2.9.7. Modelos Ocultos de Markov

Inicialmente los Modelos Ocultos de Markov (HMM) fueron utilizados principalmente para el reconocimiento de patrones, sobre todo en el reconocimiento automático del habla. Con el paso del tiempo se han extendido hasta multitud de campos, entre ellos el reconocimiento de eventos y conductas en sistemas de visión. Un modelo oculto de Markov es un modelo estadístico que está formado por un conjunto de estados ocultos, un conjunto de salidas observables, las probabilidades de transición y las probabilidades de salida. El objetivo de un HMM es determinar los parámetros *ocultos* en función de los parámetros *observables*.

El uso de los HMMs para el reconocimiento de movimientos se basa principalmente en dos etapas [194]: (i) etapa de entrenamiento y (ii) clasificación. En la etapa de entrenamiento se especifica el número de estados y las transiciones entre estados. Además se optimizan las probabilidades de salida para que los símbolos generados puedan corresponder con las características observadas en imágenes. En la etapa de clasificación, se buscan correspondencias entre el movimiento actual de un objeto y los movimientos definidos en el HMM. Vinod Nair y J. Clark [134] proponen un sistema de vigilancia automático mediante el uso de HMMs. Este sistema se basa principalmente en el reconocimiento de conductas normales en una oficina y considera como anómalas aquellas situaciones cuya semejanza con las situaciones normales sea mínima. Uno de los principales inconvenientes que presenta el método propuesto es su escasa interpretabilidad debido a la forma en la que se representan las actividades definidas en el modelo. Este hecho supone una complejidad añadida para la persona encargada de supervisar y mantener el conocimiento que forma parte del sistema experto. Otro trabajo destacado es el que se lleva a cabo en el proyecto VIGILANT [153] donde, tal como se comentó en la sección anterior, se emplean Redes Bayesianas para la clasificación de objetos y, además, se hace uso de HMM para modelar y reconocer eventos en un entorno vigilado, en concreto, las trayectorias que realizan los objetos con capacidad de movimiento. El trabajo de E. Swears et al. [175], al igual que los dos anteriores, también se centra en el aprendizaje y modelado de las trayectorias normales mediante la observación continua del movimiento de los objetos en la escena. En este caso los autores combinan técnicas de clustering con HMMs y consideran como anómala cualquier trayectoria que

no haya sido definida previamente como normal. Todos los parámetros del HMM, incluido el número de estados, son aprendidos automáticamente.

A pesar de la popularidad de la que gozan los HMM y su uso extendido en el ámbito de la interpretación de eventos en vídeos, Bremond et al. destacan en [32] algunos inconvenientes, como por ejemplo: i) dificultad en la interpretación de eventos en los que participan varios objetos al mismo tiempo, ii) complejidad a la hora de modelar algunas de las relaciones temporales propuestas por J. F. Allen [15], iii) dificultad para tratar varias actividades que se producen de forma concurrente y iv) la fase de aprendizaje es crítica para aprender los parámetros de los HMM, es necesario un número de ejemplos elevado para aprender correctamente la distribución de probabilidades que forman parte del modelo.

Algunos de los problemas citados anteriormente se han resuelto con variaciones de los HMM originales. Brand et al. [31] utilizan una variante de los modelos ocultos de Markov, *Coupled Hidden Markov Model* o *CHMM*, para reconocer acciones complejas en entornos vigilados. Esta variante permite establecer relaciones o emparejamientos entre distintos HMMs y modelar así la interacción entre dos objetos que interactúan por separado, o acciones que suceden de forma concurrente y guardan algún tipo de relación. Otro intento de relacionar los HMMs jerárquicamente para representar comportamientos complejos ha sido mediante el uso de Modelos Ocultos de Markov Jerárquicos (HHMM), un ejemplo se presenta en [136], donde se emplean este tipo de HMMs para el análisis de trayectorias en entornos de interiores. Finalmente, H. Bui et al. [35] proponen Modelos Ocultos de Markov Abstractos (AHMM) para reconocer y predecir la trayectoria de los objetos que están siendo observados por las cámaras de vigilancia. Los AHMM además de ser una variante de los HMM, también son considerados como un tipo especial de Redes Probabilísticas Dinámicas (DPN) y la principal diferencia con respecto a los originales es que la estructura de estados ocultos es remplazada por otra más compleja donde se emplean múltiples capas. Estas variantes de los HMMs originales no son las únicas y existen muchas otras como por ejemplo Modelos Ocultos de Markov que permiten múltiples observaciones (MOHMM) [202] o HMM Transformados (THMM) [97], entre otros.

2.9.8. Modelos difusos

La mayoría de los modelos presentados anteriormente son modelos probabilistas. La principal diferencia entre un modelo probabilista y un modelo difuso o borroso, es que los primeros trabajan en términos de probabilidad, mientras que los segundos trabajan con valores de posibilidad. Para comprender mejor la diferencia entre el concepto de probabilidad y posibilidad supongamos el lanzamiento de un dado; la probabilidad de que salga seis es $1/6$, en cambio la posibilidad de que salga 6 es 1.

Los modelos difusos tienen su origen en la lógica difusa que nació a raíz del artículo [211] publicado por Lotfi A. Zadeh en 1965. Desde entonces hasta hoy, son numerosas las propuestas que se han realizado en la literatura y los sistemas difusos que se han implementado basados en el trabajo de Zadeh. La principal característica de la lógica difusa es que permite cuantificar valores imprecisos de nuestro lenguaje como por ejemplo “mucho”, “poco”, “muy poco”, “demasiado”, etc. Por esta razón, la lógica difusa se adapta al mundo real mejor que la lógica tradicional. Una diferencia importante entre la lógica tradicional y la lógica difusa, es que la primera sólo admite dos valores de verdad (verdadero o falso), mientras que la segunda asigna a una proposición un valor perteneciente al intervalo $[0, 1]$, siendo 0 falso y 1 totalmente cierto.

Por otro lado, el profesor Zadeh destacó la habilidad del cerebro humano para interpretar y resolver situaciones complicadas sin la necesidad de manejar valores numéricos [212]. A partir de este pensamiento, nace su teoría de percepciones y computación con palabras, a la que se adapta perfectamente la lógica difusa con los conceptos de variables lingüísticas y reglas difusas. El modelo planteado por Zadeh pretende trabajar principalmente con percepciones del mundo real como por ejemplo, tamaño, distancia, peso, velocidad, temperatura, formas, dirección, etc. Precisamente, los múltiples sensores que pertenecen a un sistema de seguridad avanzado proporcionan estas percepciones para poder interpretar lo que sucede en el entorno, por tanto y debido a su propia naturaleza, los modelos difusos son una alternativa interesante a tener en cuenta para los sistemas de vigilancia avanzados. Además, este tipo de modelos puede ofrecer a un sistema de vigilancia gran robustez, ya que permite ofrecer soluciones en un tiempo cercano al real con bajo coste y es sensible a pequeñas variaciones. Esta última característica permite crear sistemas mucho más flexibles y reducir el número de falsas alarmas (uno de los principales problemas en los sistemas de seguridad). Por ejemplo, supongamos que existe una norma en la cual se enuncia que un coche no puede circular por un jardín. Quizás, no sería necesario activar la alarma si un coche invade muy poco o poco un jardín. Un sistema difuso sería flexible en este aspecto, es decir, únicamente activaría la alarma si la situación fuera lo suficientemente grave.

Petry et al. [154] hacen uso de la lógica difusa para estudiar las relaciones espaciales entre objetos y, entre objetos y regiones observadas por cámaras de vídeo vigilancia. Aunque inicialmente los autores aplican los algoritmos difusos que proponen a sistemas de información geoespacial, estos podrían ser aplicados a sistemas de seguridad para estudiar las relaciones de proximidad. Zhang et al. [215] emplean Redes de Petri Difusas para el modelado, reconocimiento y descripción con un alto nivel de interpretabilidad de los eventos que suceden en un determinado entorno. Y. Ma y L. Li [122] proponen una variante difusa de la técnica *Support Vector Machine* para la detección de patrones anómalos de comportamiento, en concreto, la detección de trayectorias anómalas. De las Redes Neuronales

también se han propuesto versiones difusas [98], y han sido empleadas para el modelado de actividades mediante una red formada por un conjunto de nodos o neuronas y su uso posterior en procesos de clasificación e identificación de eventos. G. Iovane et al. [100] proponen un sistema de *tracking* multicámara que incluye un sistema de predicción difuso organizado en dos niveles y que es capaz de predecir la posición de los objetos detectados por el sistema, mediante el uso de un conjunto de reglas difusas. Además, el sistema dispone de métodos de etiquetado consistentes que proporciona al sistema interpretabilidad en los resultados. H.M. Feng et al. [74] proponen un controlador difuso para ser incluido en cámaras de vigilancia, que detecta intrusiones en regiones vigiladas, empleando sistemas de reglas. F. El Baf et al. [19] proponen un sistema difuso para clasificar los píxeles de una imagen como píxeles pertenecientes al fondo de la imagen, o bien, píxeles pertenecientes a objetos en movimiento. El sistema trata la incertidumbre existente en algunos píxeles causada por cambios de iluminación o sombras. Xiang et al. [203] emplean técnicas de *clustering* difusas para la generación de un modelo espacio-temporal que permita representar y reconocer diferentes expresiones faciales en humanos. Otros trabajos destacados con modelos difusos son [121] donde se gestiona la información procedente de varios sensores, y [95] donde se combinan modelos difusos con redes neuronales para la detección de incidentes en escenas de tráfico.

2.9.9. Otros métodos

Otros métodos alternativos a los expuestos anteriormente, son aquellos que hacen uso de la geometría para estudiar otro tipo de eventos como la interpretación y el análisis de las velocidades de los objetos. Normalmente, este tipo de métodos requieren de un proceso de calibración en el que se suelen emplear parámetros como el ángulo de inclinación de la cámara o la altura a la que está situada. Dicho proceso permitirá establecer correspondencias entre los objetos y las zonas representadas en una imagen 2D captada por la cámara y el entorno real que está siendo observado. De esta forma es más sencillo estimar la localización de los objetos y sus velocidades a partir del desplazamiento realizado en un intervalo de tiempo.

I. Junejo et al. [105] proponen un nuevo método para analizar las trayectorias en el que se estudia el camino que ha recorrido el objeto, la velocidad con la que lo ha hecho y, finalmente, la curvatura de la trayectoria. De esta forma, el sistema propuesto es capaz de detectar las siguientes situaciones:

- Trayectorias espacialmente diferentes.
- Trayectorias espacialmente similares pero recorridas con velocidades diferentes.

- Trayectorias seguidas en línea recta frente a las que se han realizado con diversos cambios de sentido (mayor curvatura).

El proceso de análisis establece similitudes entre las trayectorias modelo o consideradas como normales y las trayectorias que están siguiendo los objetos en el momento actual. Para ello se llevan a cabo tres fases: i) Análisis de los puntos que forman parte de la trayectoria: si el 90 % de los puntos de la trayectoria actual queda fuera de los límites definidos para la trayectoria modelo, el sistema considera que el objeto no está siguiendo esa trayectoria. Además, se debe cumplir que la distancia de Hausdorff entre los puntos de la trayectoria media y los de la trayectoria a analizar sea menor que la distancia que existe con los puntos que forman parte de las líneas límite. En la Figura 2.16 se puede observar la forma en la que se representan las trayectorias en [105]. La línea de puntos discontinuos representa la distancia media y las continuas los límites de dicha trayectoria.

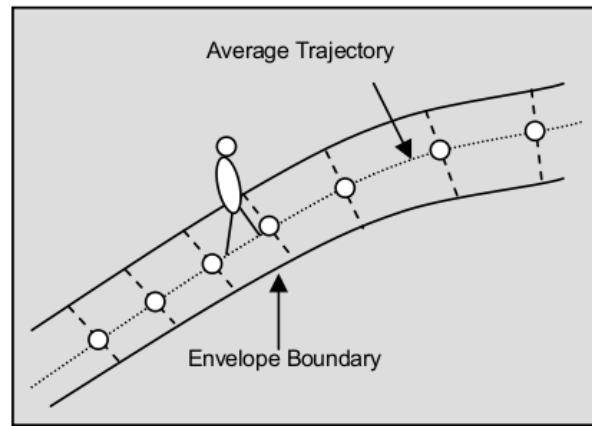


Figura 2.16: Representación de trayectorias en [105].

ii) En segundo lugar se estima la velocidad media del objeto, teniendo en cuenta que la velocidad en un punto se calcula de la siguiente forma:

$$v' = \left(\frac{x_{i+1} - x_i}{t_{i+1} - t_i}, \frac{y_{i+1} - y_i}{t_{i+1} - t_i} \right) \quad (2.2)$$

donde x_i e y_i son las coordenadas del objeto en el instante t_i y, de la misma forma, x_{i+1} e y_{i+1} son las coordenadas del objeto en el instante t_{i+1} . La comparación de velocidades entre la trayectoria modelo y la trayectoria del objeto se calcula mediante la distancia de Mahalanobis.

iii) Finalmente se estudia la curvatura de la línea seguida por el objeto, que es calculada mediante la primera derivada de la velocidad. Para la comparación

entre curvaturas también se emplea la distancia de Mahalanobis, al igual que sucedía con las velocidades. Si un objeto no satisface alguno de los tres puntos anteriores quiere decir que el objeto no está siguiendo correctamente la trayectoria. Aunque el método propuesto está correctamente formalizado y destaca por su originalidad en la forma de definir las trayectorias frente a otros métodos más comunes, en [105] no se exponen resultados que puedan demostrar su validez.

Otra forma extendida de estimar las velocidades es estudiar el desplazamiento de los objetos en imágenes aéreas tomadas perpendicularmente. Quizás, esta sea la forma más sencilla de estimar las velocidades de los objetos ya que se elimina el problema de la perspectiva en gran medida pero, sin embargo, es una de las menos económicas. Un ejemplo es el trabajo propuesto por W. Liu et al. en [118] donde se analizan dos imágenes tomadas de manera consecutiva para establecer correspondencias entre los objetos de una imagen y otra, y estimar sus velocidades en función del desplazamiento realizado. El método propuesto no ha sido evaluado sobre vídeo real, sino que los autores han simulado una secuencia de imágenes. El estudio se centra en vehículos que circulan por la carretera y para ello, un experto define de forma manual las regiones de la imagen que corresponden con las carreteras. De esta forma, el sistema se centra únicamente en aquellos objetos que se encuentren en el interior de las zonas definidas. Los autores también consideran constante la altura desde la que se toman las fotografías y, por este motivo, reducen considerablemente el problema de la estimación de velocidades ya que no es necesario realizar un nuevo proceso de calibrado.

Por otro lado, cuando las cámaras que se emplean están situadas en edificios o a pie de calle, la estimación de la localización de los objetos y sus velocidades se complica debido al problema de la perspectiva de la cámara. En este caso, el desplazamiento del objeto en la imagen 2D tomada por la cámara no tiene por qué corresponder con el desplazamiento real. Algunas soluciones a este problema se basan en el uso de varias cámaras para realizar una reconstrucción tridimensional del entorno [44, 113], o bien, mediante un proceso de calibrado en el que se conocen los parámetros de la cámara y la distancia a la que se encuentran algunas regiones visibles en la imagen capturada por la cámara. F. Cathey et al. [41] presentan un método novedoso que permite estimar la velocidad de los vehículos que circulan por carreteras de alta velocidad situadas en la ciudad de Seattle. Las cámaras están situadas sobre puentes y pueden ser movidas desde un centro de control, así como variar su nivel de zoom. Ante esta situación los autores proponen un método de calibrado rápido que permite al sistema continuar con la estimación de velocidades. Este proceso se lanza cada vez que se selecciona una cámara y se modifican sus parámetros. En realidad se trata de un proceso de aprendizaje en el que el sistema detecta automáticamente las líneas continuas que limitan la carretera por la que circulan los vehículos; precisamente estas líneas marcan la perspectiva de la cámara. En una segunda fase, se generan un conjunto de imágenes con vista aérea de las regiones más importantes de la escena, todo ello mediante

métodos de transformación y proyección. Un ejemplo se puede observar en la Figura 2.17; en la imagen superior se muestra como el sistema detecta automáticamente tres de las líneas continuas de la carretera (según sus autores es necesario como mínimo la detección de tres líneas para llevar a cabo la segunda fase en la que se generan las vistas aéreas) y en la imagen inferior la vista aérea generada (tómese como referencia la sombra de la farola y el puntero del ratón).

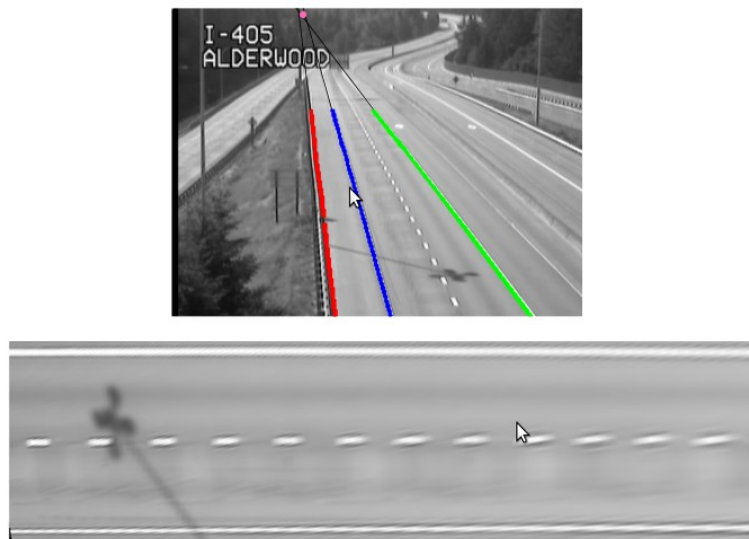


Figura 2.17: Detección automática de las líneas de una carretera para determinar la perspectiva de la cámara y posterior generación de vistas aéreas de las zonas relevantes para la estimación de velocidades [41].

En la tercera y última fase se estudia el desplazamiento de los objetos en un intervalo de tiempo concreto. Otros métodos de estimación de velocidades similares, en el que se emplean métodos de calibración y detección automática de las líneas de las carreteras son [23, 143].

Finalmente, C. Maduro et al. afirman en [123] que el tipo de métodos descritos anteriormente disminuyen su efectividad notablemente cuando se producen oclusiones, los vehículos recorren grandes distancias entre frames (hecho que puede suceder si se utiliza una baja tasa de frames por segundo), o cuando la imagen tiene una baja resolución y el algoritmo de segmentación ofrece peores resultados en la detección de objetos. Por todo ello, proponen un nuevo método para la estimación de velocidades basado en otros dos anteriores [45, 125], sin depender de la efectividad de la detección y seguimiento de objetos en el proceso de *tracking*. Al igual que en los métodos anteriores existe un periodo de calibración en el que se aprenden las líneas continuas que limitan la carretera, para generar posteriormente vistas aéreas de las distintas regiones de la imagen 2D capturada por la cámara de vigilancia. Por cada una de las líneas aprendidas se construye un perfil

denominado “perfil de intensidad”. La idea que defienden los autores es que un perfil de intensidad en un punto x se repetirá en un intervalo de tiempo consecutivo con un desplazamiento diferente, de esta forma, es posible estimar la velocidad aunque los algoritmos de segmentación y *tracking* no hayan detectado correctamente al objeto o hayan perdido su referencia.

Los métodos anteriores se centran en el estudio de la velocidad de una única clase de objetos, bien peatones en zonas habilitadas para ellos o vehículos en carreteras donde se alcanzan grandes velocidades. Sin embargo, existen multitud de escenas en las que conviven diferentes clases de objetos cuyos valores mínimos, medios y máximos de velocidad pueden variar notablemente. Un ejemplo es el caso de los peatones y vehículos; velocidades que pueden ser normales para los vehículos pueden ser excesivas para las personas. Por tanto, cuando la estimación de las velocidades se desarrolla en el contexto de los sistemas de vigilancia es deseable el tratamiento de este problema. Dicho de otro modo, un sistema de vigilancia inteligente debería ser capaz de diferenciar entre varias clases de objetos y conocer cuáles son las velocidades apropiadas para cada uno de ellos. De esta forma, el sistema desarrollado gozará de mayor versatilidad y podrá ser fácilmente adaptado a diferentes tipos de escenas.

2.10. Fusión de información procedente de múltiples fuentes heterogéneas

Como se comentó en la Sección 2.1, los sistemas de vigilancia de segunda y tercera generación están formados por una red de sensores que proporcionan información relativa al entorno y los elementos que participan en él. De esta forma, el sistema puede recibir información sobre un mismo objeto procedente de diferentes sensores. La fusión de esta información permite al sistema determinar la clase a la que pertenece un objeto e interpretar su comportamiento con mayor exactitud que en el caso en el que recibiera información procedente de una única fuente [46, 90, 193]. La etapa de fusión [131] se puede llevar a cabo en las capas inferiores, con la información procedente de los sensores directamente, o bien, en las capas superiores donde se realiza la interpretación de comportamientos. En el primer caso, el sistema dispondría de más información sobre las características físicas de un objeto para determinar su clase, posición y los movimientos que éste realiza con mayor exactitud. Sin embargo, la fusión de diferentes análisis o interpretaciones de comportamientos en las capas superiores permite al sistema obtener una visión general de lo que ocurre en el entorno.

Tal como se indica en [192], la fusión de datos se refiere al proceso de asociación, correlación y combinación de datos procedentes de múltiples fuentes. En concreto, la fusión de información procedente de sensores es un

subcampo de la fusión de datos, en el cual las fuentes de datos e información son dispositivos con capacidad para capturar características del entorno en el que se encuentran. Luo et al. [119] definen la integración y fusión de datos procedentes de múltiples sensores⁴ como la combinación sinérgica de los datos procedentes de múltiples sensores para mejorar la fiabilidad y precisión de la información.

L. Snidaro et al. [167] proponen algoritmos de fusión que combinan la información redundante procedente de los sensores (en este caso, cámaras de vigilancia); redundante en el sentido de que el sistema maneja información sobre un mismo objeto procedente de diversas fuentes. El objetivo de los algoritmos propuestos es mejorar la detección de objetos y la estimación de sus localizaciones, tratando de evitar problemas como las oclusiones parciales y absolutas. Una independencia clara entre las capas que fusionan la información procedente de los sensores y las capas encargadas de realizar la interpretación y análisis de comportamientos, siempre reportará importantes beneficios para cualquier sistema de visión cognitivo. Esta independencia permite que los módulos de análisis reciban información sobre cualquier objeto sin la necesidad de conocer el lugar de donde procede, es decir, el proceso de recolección de datos referente a las características espaciales de cada uno de los objetos es totalmente transparente para las capas intermedias. De esta forma, aunque alguno de los sensores deje de percibir un determinado objeto, el sistema puede seguir analizando su comportamiento. Los mismos autores proponen en otro trabajo [79] métodos de fusión, en esta caso para combinar información procedente de sensores heterogéneos: ópticos, infrarrojos, radares, etc.

Un trabajo similar es el propuesto por Zhang et al. [217] en el que se combina la información procedente de los procesos de *tracking* llevados a cabo en diferentes cámaras. Una gran ventaja a destacar es que cuenta con mecanismos automáticos de calibración que permiten establecer relaciones entre zonas observadas desde diferentes cámaras. En la Figura 2.18 se muestra un ejemplo de este proceso, donde se puede apreciar como existen zonas en común entre ambas cámaras.

S. Sutor et al. [174] también siguen una línea similar a la de los trabajos anteriores y para ello proponen una arquitectura escalable en la que es posible combinar sensores de diferente naturaleza y fusionar la información que proporcionan. N. Cvejic et al. [59] realizan fusión a nivel de píxel combinando la información procedente de cámaras tradicionales e infrarrojas. De esta forma reducen el ruido producido por las condiciones de iluminación y mejoran los resultados obtenidos en el proceso de segmentación y *tracking*. La fusión con imágenes procedentes de cámaras infrarrojas también ha sido tratada recientemente por autores como A. Sekmen [162], con la salvedad de que las cámaras utilizadas son móviles y toman las imágenes desde el aire.

⁴Multi-Sensor Integration and Fusion, MSIF



Figura 2.18: Métodos de calibración de cámaras propuesto en [217]. En la imágenes se puede apreciar como se marcan las zonas en común observadas desde cámaras diferentes.

Finalmente, un ejemplo de fusión en un nivel conceptual mayor se puede encontrar en [54]. M. Cristani et al. proponen un sistema en el que se utiliza como sensores una cámara de vigilancia tradicional y un micrófono para capturar el sonido ambiente en una oficina. De manera individualizada el sistema analiza los eventos de vídeo y audio, y en una etapa posterior se lleva a cabo un proceso de sincronización en el que se establecen correspondencias entre los sonidos y las imágenes capturadas por la cámara. Una vez realizada la sincronización, el sistema utiliza la nueva información generada junto con técnicas de *clustering* para clasificar los eventos que suceden en la oficina.

2.11. Ámbitos de aplicación

Con el transcurso de los años, el rango de entornos en los que se utilizan sistemas de seguridad ha crecido considerablemente debido a la creciente demanda y la aparición de soluciones más sofisticadas. Las nuevas propuestas para la mejora de los sistemas de seguridad provienen tanto del ámbito académico como del comercial [186]. La principal diferencia entre unas y otras, es que las propuestas comerciales están principalmente orientadas a la utilización de hardware específico y a la implantación inmediata con el objetivo de obtener grandes beneficios económicos. La mayoría de estos sistemas no van más allá de la detección de intrusos y el seguimiento de los mismos en la escena. Sin embargo, las propuestas que se realizan en el ámbito académico suelen ser algoritmos más avanzados, que no están ligados a dispositivos hardware concretos y no suelen ser utilizados en el mercado hasta que no trascurren algunos años desde su publicación. Algunos de los ámbitos donde se emplean sistemas de seguridad comerciales y se realizan investigaciones desde entornos académicos son los siguientes:

- **Aeropuertos.** Los aeropuertos son uno de los entornos que demandan

mayores medidas de seguridad, debido en gran parte a que son uno de los principales objetivos de los grupos terroristas. El grupo INRIA - actualmente llamado PULSAR - del Instituto Nacional de Investigación de Francia y liderado por la Dra. Monique Thonnat, es uno de los más destacados en el área de interpretación de escenas de vídeo y seguridad en aeropuertos [29, 182, 76]. Un proyecto reciente es Co-Friend [4], en el que participan diversas universidades, centros de investigación y empresas privadas, y está financiado con fondos europeos. El objetivo principal de este proyecto es el diseño de un sistema de visión cognitivo capaz de interpretar los eventos que suceden en los exteriores del aeropuerto Toulouse-Blagnac en Francia.

- **Entornos marítimos.** Los sistemas de vigilancia también tienen cabida en los entornos marítimos. Pozzobon et al. proponen en [147] un sistema de seguridad para la vigilancia en puertos de carga y descarga de mercancías. También P. Avis detalla en [18] cuales son los sistemas de seguridad que se utilizan en Canada para vigilar las costas. Otro ejemplo, es el proyecto ARGOS [26] implantado en Italia desde el año 2007 para el control del tráfico marítimo.
- **Estaciones de tren.** Seguridad en estaciones de tren y vigilancia en vías de ferrocarril. Lai Lo et al. [120] propone un sistema distribuido para la vigilancia en estaciones de tren el cual combina información de vídeo y audio (fusión de información de múltiples sensores). C. Nwagboso [140] propone diferentes medidas de seguridad que puede ser aplicadas en estaciones de transporte público, entre ellas las estaciones de ferrocarril. Ronetti et al [157], plantean el problema de la seguridad en el transporte público en Italia y justifican la necesidad de implantar sistemas más sofisticados. Normalmente, la mayoría de propuestas para mejorar la seguridad en las estaciones de tren se pueden adaptar fácilmente a las **estaciones de metro**, otro entorno más donde podemos observar frecuentemente cámaras de seguridad instaladas.
- **Vigilancia en tráfico.** Cada año los departamentos de tráfico de los países más desarrollados invierten una cantidad de dinero importante en la instalación de cámaras de seguridad en las carreteras públicas. La principal función de estos sistemas [23, 30, 83, 112] es la detección de congestiones en el tráfico y posibles accidentes que requieran la atención de los servicios sanitarios.
- **Entornos Industriales.** En los entornos industriales, además de la instalación de sistemas clásicos de seguridad para la detección de intrusos, se suelen utilizar sistemas de vídeo para el control de calidad en el proceso de producción [87, 69, 198].
- **Aplicaciones militares.** La vigilancia también es una actividad muy común en el ámbito militar, cualquiera de los trabajos citados anteriormente para la interpretación de eventos a partir de la información visual procedente de las cámaras es válido para este área. Además, también

se han empleado dispositivos y algoritmos de visión que mejorar la efectividad de los soldados y los vehículos de combate [71, 103, 144].

- Vigilancia en lugares públicos como **bancos** [126, 84], **hogares** [34], **casinos**, **grandes almacenes** y **zonas de aparcamiento** [130, 101].

2.12. Proyectos de investigación orientados a la vigilancia inteligente

Como se comentó en la sección anterior, las prestaciones de los sistemas de vigilancia comerciales no suelen ir más allá de la detección, clasificación y el seguimiento de los objetos en un entorno vigilado [3, 6, 7, 8, 10]. Esto es debido a que la interpretación de eventos y el análisis de comportamientos que implican una mayor complejidad no han alcanzado un nivel de madurez suficiente como para ser explotados comercialmente. En el ámbito académico sin embargo, se encuentran las propuestas más novedosas e interesantes que dotan a los sistemas artificiales con habilidades similares a las de los agentes de seguridad humanos, superándoles incluso en ciertas actividades como la detección de objetos abandonados o la identificación de personas de acuerdo a la forma de andar. Estos nuevos avances aparecen con frecuencia en centros de investigación donde uno de los objetivos principales es la generación de nuevos conocimientos sin tener la necesidad de generar un producto final a corto plazo, que pueda ser explotado comercialmente, y con el que se establecen responsabilidades directas con las empresas que lo adquieren. Por este motivo, los sistemas comerciales incluyen únicamente aquellas funcionalidades que han sido correctamente probadas en multitud de entornos y situaciones, y garantizan al cliente la fiabilidad que desean.

Normalmente, las soluciones completas de vigilancia (no métodos concretos para solucionar problemas específicos) nacen a partir de proyectos financiados por organismos públicos o privados, cuya duración es igual o superior a tres años. Uno de los primeros proyectos desarrollados en el ámbito académico fue CROMATICA (*Crowd Monitoring with Telematic and Communication Assistance*) [109]. Se trata de un proyecto financiado con fondos europeos que tiene como principal objetivo mejorar la vigilancia de pasajeros en transportes públicos. En dicho proyecto se combinan tecnologías basadas en el análisis de vídeo y la transmisión inalámbrica de datos [186]. Como continuación de este proyecto surge PRISMATICA (*Pro-active Integrated Systems for SEcurity Management by Technological Institutional and Communication Assistance*) [191], donde se presta más atención a los aspectos sociales, éticos, organizacionales y técnicos de la vigilancia en el transporte público [186]. En cuanto a las novedades técnicas, la nueva versión del sistema de seguridad, a diferencia del primero, es distribuido y permite la comunicación entre dispositivos heterogéneos mediante el uso del *middleware* CORBA. Algunos de los eventos anómalos que el sistema es capaz de detectar son: estacionamiento anómalo de personas en pasillos, intrusiones (acceso a áreas

prohibidas), longitud anómala en colas de espera o persona caminando en sentido contrario. En estos sistemas, el análisis de comportamientos e interpretación de eventos se lleva a cabo siempre en interiores y sólo para personas.

Otro de los sistemas más completos, posterior a CROMATICA y anterior a PRISMATICA, fue VSAM (*Video Surveillance and Monitoring*) [52], el cual fue financiado por el Departamento de Defensa de los Estados Unidos, en concreto por la agencia DARPA (*Defense Advanced Research Projects Agency*). El sistema utiliza como información de entrada el flujo de video proporcionado por cámaras de vigilancia y clasifica los objetos detectados como personas, grupos de personas o vehículos. La capa encargada de identificar los eventos está diseñada exclusivamente para el análisis de trayectorias por una parte y para el análisis del movimiento de los peatones por otra, de tal forma que es difícil incluir nuevos tipos de análisis que contribuyan al crecimiento de capacidades del sistema. A partir del estudio de la forma de andar de las personas, en el que se analiza las poses (ver Figura 2.19), el sistema es capaz de determinar si una persona está andando o corriendo. En los entornos en los que se utiliza el sistema se considera que una persona corriendo no es una situación normal, de ahí el interés por detectar este tipo de eventos.

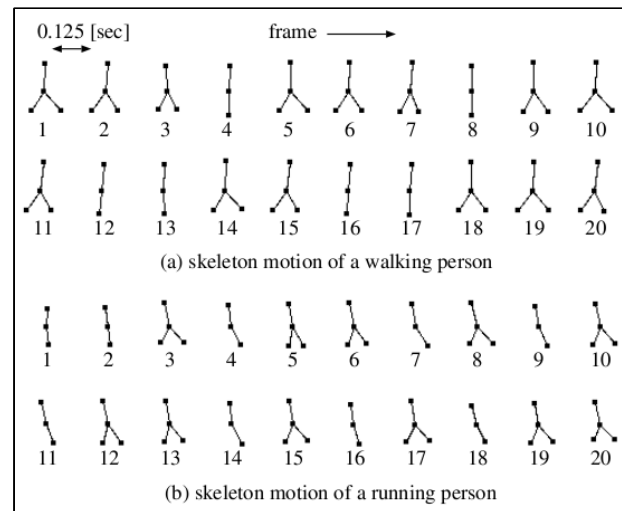


Figura 2.19: Estudio de poses en el movimiento de las personas llevado a cabo en el sistema VSAM [52].

Los resultados presentados en [52] son demasiado escuetos y no permiten determinar con claridad la eficiencia real del sistema en cada uno de sus módulos, los cuales se muestran en la arquitectura de la Figura 2.20

W^4 [92] es un sistema creado por Haritaoglu et al. y, a diferencia del anterior, se centra únicamente en el análisis de los comportamientos de las personas a partir de sus poses en escenas al aire libre, e ignora el

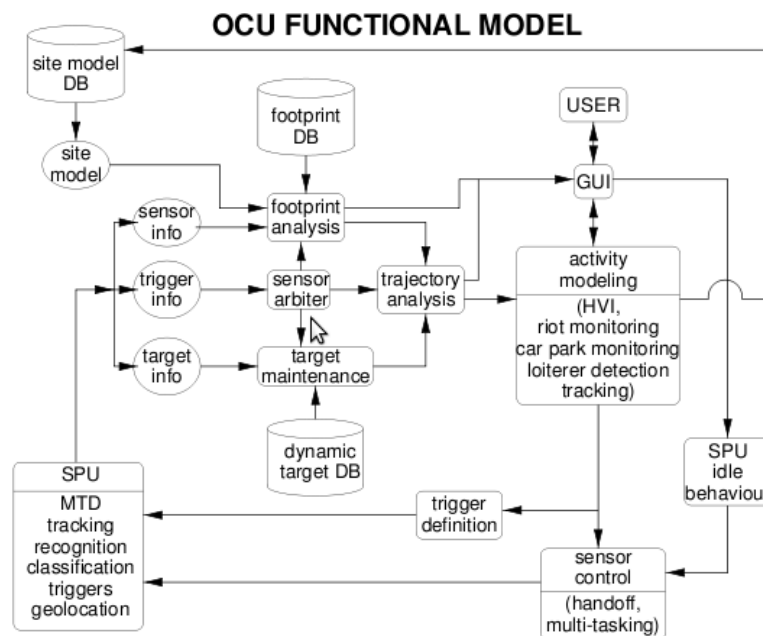


Figura 2.20: Arquitectura del sistema VSAM [52].

comportamiento de otros elementos como por ejemplo los vehículos. El sistema detecta y determina las siluetas de las personas que se encuentran en una escena monitorizada, a partir de las imágenes tomadas por cámaras de vídeo tradicionales y de infrarrojos. Una vez obtenida la silueta, un segundo componente del sistema identifica diferentes partes del cuerpo humano como la cabeza, tronco, brazos o piernas. La posición de estos elementos junto con el contorno de las siluetas sirven para identificar eventos tales como : personas corriendo, andando con mochila, transportando objetos en las manos, personas con las manos levantadas ante una amenaza con arma de fuego, etc. El sistema, aunque ofrece buenos resultados para el propósito por el que fue diseñado, no está preparado para incluir otros tipos de análisis que extiendan la funcionalidad del sistema de vigilancia, ya que el diseño del mismo está centrado exclusivamente en el análisis de poses. Además, los autores se centran en definir situaciones anómalas concretas y establecer correspondencias entre éstas y las situaciones captadas por las cámaras. Como ya se discutió anteriormente, este enfoque puede favorecer la aparición de falsos positivos: situaciones anómalas clasificadas como normales, hecho que es realmente crítico en los sistemas de seguridad.

ADVISOR (*Annotated Digital Video for Intelligent Surveillance and Optimised Retrieval*) [165] es otro sistema financiado con fondos europeos cuyo objetivo principal es el análisis del comportamiento de las personas en estaciones de metro subterráneas. Al igual que PRISMATICA se trata de un sis-

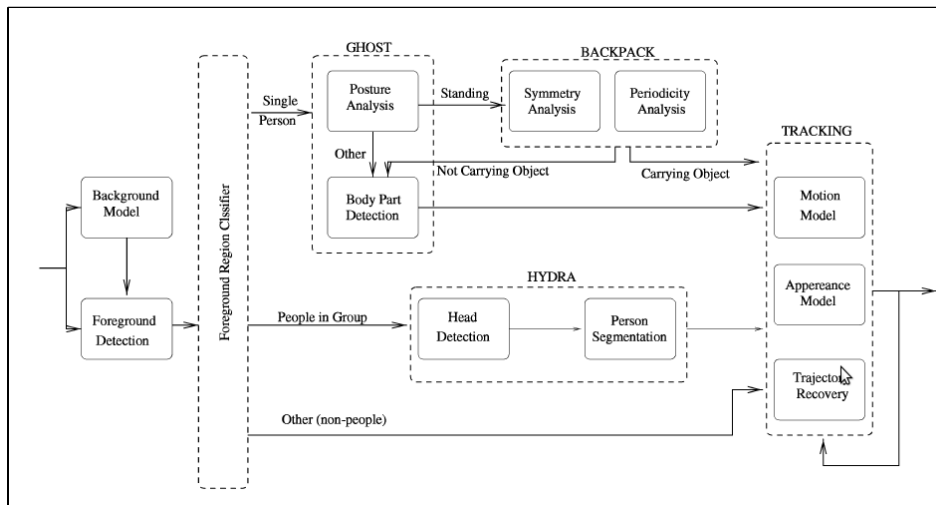


Figura 2.21: Arquitectura del sistema W^4 [92].

tema en el que las cámaras de vigilancia CCTV están distribuidas a lo largo de toda la estación y envían su señal a una estación central donde se realiza el análisis de imágenes. El módulo de análisis registra en una base de datos las secuencias de imágenes en las que ha detectado algún fenómeno anómalo. Cada una de estas secuencias están descritas mediante información textual. Precisamente esta información es utilizada posteriormente por los operadores humanos para la recuperación de los eventos registrados.



Figura 2.22: Análisis del comportamiento de grupos de personas y multitudes en estaciones de metro subterráneas, por el sistema ADVISOR [165].

Uno de los primeros proyectos que incluyen agentes inteligentes distribuidos para el análisis de comportamientos es VIGILANT [151]. El objetivo principal es el análisis de las trayectorias que realizan los peatones en zonas de aparcamientos situadas al aire libre. Para cada una de las personas detectadas se asigna un agente encargado de analizar su comportamiento de acuerdo al camino recorrido. Los agentes se comunican y debaten entre sí

sobre la posibilidad de que el objeto que están siguiendo se trate del mismo. De esta forma varios sensores pueden aportar información sobre un mismo objeto y reforzar así los procesos de *tracking*, clasificación de objetos e interpretación de eventos.

Otros proyectos más recientes son AVITRACK [25, 39, 13], Co-Friend [4] y Hesperia [1]; en todos ellos participan organizaciones públicas y privadas. Los dos primeros están relacionados de la misma forma que CROMATICA y PRISMATICA, ya que el proyecto Co-Friend es una evolución de AVITRACK. Ambos proyectos tienen como objetivo principal el análisis de comportamientos de personas y vehículos en aeropuertos, concretamente en tareas de mantenimiento: carga y descarga de aviones, repostaje de combustible, pequeñas reparaciones, etc.

Finalmente, tal como que describe en [1], el proyecto Hesperia (Homeland Security: tecnologías para la seguridad integral en espacios públicos e infraestructuras) tiene por objeto el desarrollo de tecnologías que permitan la creación de sistemas punteros de seguridad, vídeo vigilancia y control de operaciones de infraestructuras y espacios públicos. El proyecto, que comenzó en enero del año 2006 y finaliza en diciembre del 2009, tiene como participantes numerosas empresas del sector privado, universidades y centros de investigación públicos. El consorcio está integrado por Indra Software Labs, Unión Fenosa, Tecnobit, SAC Control, Technosafe, Visual Tools y Brainstorm Multimedia; la Universidad de Castilla-La Mancha, la Universidad de Granada, la Universidad de Extremadura, la Universidad Politécnica de Madrid, la Universidad de Las Palmas de Gran Canaria, la Universidad Politécnica de Valencia y la Universidad Politécnica de Cataluña. La lista se completa con la colaboración del Consejo Superior de Investigaciones Científicas(CSIC) y el Centro Tecnológico del País Vasco (Ikerlan).

El proyecto está financiado principalmente por el Ministerio de Industria, Turismo y Comercio de España, y por otra parte el sector privado que participa en dicho proyecto. A diferencia de los proyectos citados anteriormente, no se otorga toda la importancia al análisis de vídeo, sino que existen diferentes grupos de trabajo dedicados al diseño de nuevos sistemas de comunicación, cifrado de datos, análisis de audio, creación de nuevos algoritmos de *tracking*, interpretación de eventos a partir de información visual, aplicación de agentes inteligentes, diseño de arquitecturas distribuidas, fusión de información, toma de decisiones y gestión de crisis entre otras tareas.

En este punto se da por finalizado el estudio del estado actual del conocimiento relativo a los sistemas de monitorización, para dar paso al Capítulo 3 donde se propone un modelo formal para la construcción de sistemas de vigilancia flexibles y escalables en los que es posible incluir nuevos tipos de análisis y combinar todos ellos para obtener un análisis global del entorno monitorizado. Dicho modelo será aplicado al análisis de

trayectorias y velocidades, y se demostrará su validez en el Capítulo 6, donde se discuten los resultados obtenidos.



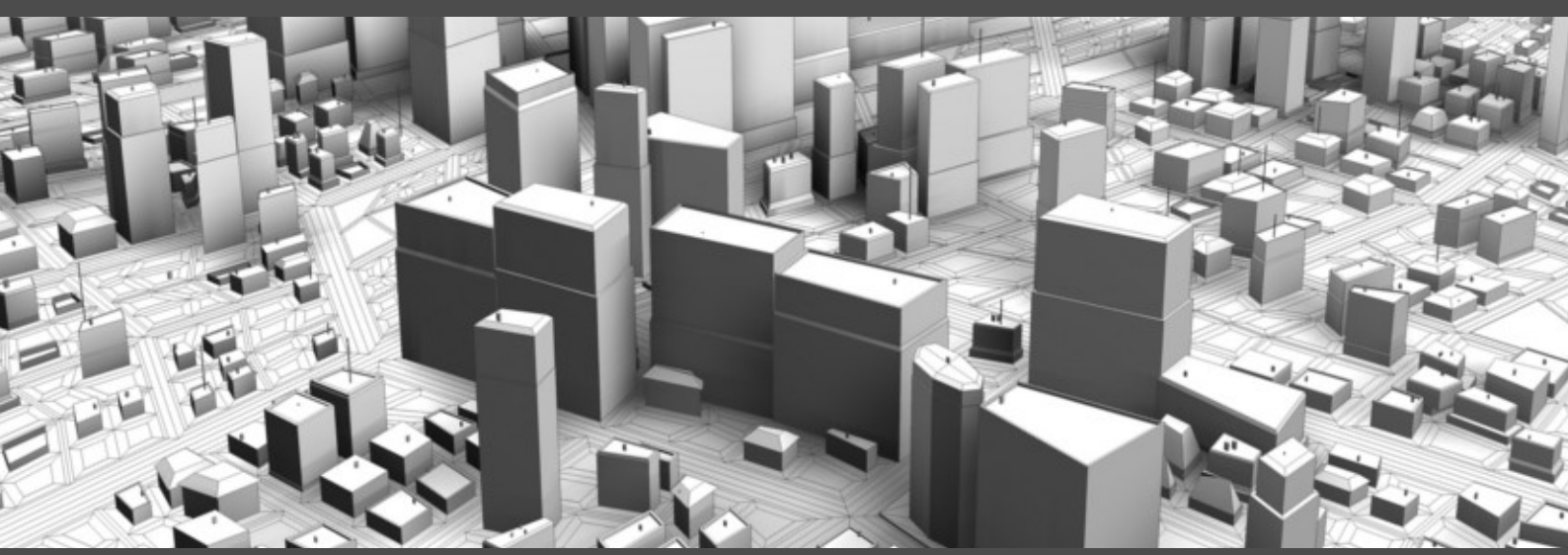
Capítulo

3

Modelo para el Diseño y Desarrollo
de Sistemas de Seguridad Escalables
Basados en Componentes de Normalidad

2

Capítulo



3

Modelo para el Diseño y Desarrollo de Sistemas de Seguridad Escalables Basados en Componentes de Normalidad

En este Capítulo se presenta un modelo para la definición y el análisis de la normalidad en entornos monitorizados, inspirado en el paradigma del desarrollo del software basado en componentes, en el diseño multicapa de sistemas y en la división de problemas en subproblemas, basada en la idea de “divide y vencerás”. El Capítulo comienza con una breve introducción en la se propone un esquema conceptual general para la construcción de sistemas de vigilancia multicapa, donde las tareas de análisis e interpretación de comportamientos se distribuyen entre varios componentes independientes y reutilizables para diferentes entornos. De esta forma, se pretende ofrecer una visión general para facilitar la comprensión futura del modelo. A continuación, se definirá formalmente el modelo empleado para la definición de los componentes que estudian la normalidad del entorno observado en base a un determinado aspecto a vigilar. En la siguiente sección se describen las técnicas empleadas para combinar los distintos análisis llevados a cabo por cada uno de los componentes, con el objetivo de obtener un análisis global que refleje el estado actual del entorno vigilado. En la penúltima sección se discuten diferentes alternativas para la adquisición de conocimiento experto necesario para los componentes que analizan la normalidad, y se presenta una algoritmo de aprendizaje para la instanciación de dichos componentes. Finalmente se describe de forma breve cómo una arquitectura multiagente puede ofrecer soporte al modelo planteado en el Capítulo, la cual ha sido diseñada en una línea de investigación paralela al trabajo actual.

3.1. Esquema del diseño propuesto

Normalmente, los sistemas expertos que interactúan con el mundo real se enfrentan a problemas de gran magnitud, cuyas soluciones se obtienen como consecuencia del trabajo llevado a cabo en diversas etapas o fases. La mayoría de las arquitecturas de estos sistemas se caracterizan por estar divididas jerárquicamente en diferentes capas [186, 194], donde cada una de ellas cumple una función bien definida. Dicha división dota al sistema de una mayor flexibilidad y modularidad, facilitando las tareas de mantenimiento e inclusión de nuevas funcionalidades. Los sistemas de vigilancia inteligentes, son un claro ejemplo de sistemas complejos que pueden beneficiarse de un diseño basado en capas. En este tipo de sistemas las capas inferiores se centran en el procesamiento de señales y la generación de información espacio-temporal relativa a cada uno de los elementos que interactúa en el entorno, para que las capas intermedias puedan interpretar los eventos y comportamientos que suceden alrededor. Finalmente, las capas situadas en los niveles superiores suelen estar enfocadas a la monitorización de los resultados obtenidos en las capas inferiores y a la ayuda para la toma de decisiones en situaciones críticas.

Por otra parte, la complejidad de los sistemas de vigilancia inteligentes actuales lleva a buscar la reutilización de los diseños y desarrollos existentes a la hora de diseñar y desarrollar sistemas que vigilen nuevos entornos. Un entorno vigilado E consta de un conjunto de sensores heterogéneos $S_1, S_2, \dots, S_n$ distribuidos geográficamente en dicho entorno. Cada uno de estos sensores ofrece una visión particular de una región o zona del entorno E y, además, uno o varios sensores pueden aportar información sobre una misma zona del entorno global, la cual puede ser considerada a su vez como un sub-entorno E_i de E . El problema de la vigilancia está enfocado principalmente a la interpretación de los sucesos en cada uno de estos sub-entornos E_i , con respecto a los aspectos que se quieran vigilar. Partiendo de esta idea, en la Figura 3.1 se propone un esquema conceptual general para la construcción de sistemas de vigilancia basado en componentes, donde cada componente analizará el entorno teniendo en cuenta un aspecto a vigilar. Estos componentes podrían estar ya diseñados y desarrollados por terceras partes y disponibles en repositorios públicos o privados para su uso en el diseño de un nuevo sistema de vigilancia. O por el contrario, en caso de no existir, deberían ser diseñados y desarrollados. El uso de esta idea implica que cada componente del repositorio lleve asociado herramientas que permitan su adaptación a un entorno concreto.

El desarrollo de sistemas de vigilancia inteligentes basado en componentes permite reutilizar piezas preelaboradas con diferentes funciones. En líneas generales la construcción de este tipo de sistemas en base a una serie de componentes implica el tratamiento de las siguientes cuestiones:

- Cuáles son los objetos con capacidad de acción o elementos del entorno

a los que se va a prestar especial atención.

- Qué aspectos son los que se quieren vigilar referente a cada uno de estos elementos.
- Qué tipo de sensores se encuentran disponibles en el entorno vigilado, o bien, cuáles son necesarios para proporcionar información útil para realizar la vigilancia deseada.
- Qué mecanismos de preprocesamiento existen para tratar las señales procedentes de los sensores. El objetivo principal de estos preprocesadores es la transformación y adaptación de los datos procedentes de los sensores para construir la información empleada como flujo de entrada en los componentes encargados de realizar la vigilancia.
- Qué componentes existen ya diseñados para realizar tareas de vigilancia. Si existe ya un componente diseñado y desarrollado que realiza la tarea de vigilancia deseada, habría que adaptarlo al problema concreto en el que se quiere aplicar. Si no existe habría que diseñarlo y desarrollarlo.

En particular, en la primera capa o capa sensorial (ver Figura 3.1) se encuentran los sensores encargados de captar cualquier cambio de estado en el entorno vigilado. La información procedente de los sensores puede ser enviada directamente a la capa intermedia (si no necesita procesamiento previo), o bien, ser procesada por diferentes tipos de algoritmos para la generación de información espacio-temporal, como por ejemplo, algoritmos de segmentación, *tracking*, análisis de audio o clasificación de objetos en movimiento. Para que un sistema de vigilancia avanzado pueda interpretar correctamente lo que sucede a su alrededor, debería procesar la información procedente de los sensores para generar, al menos, la siguiente información sobre los objetos detectados en la escena [36]:

Información espacial (Dónde):

- ¿Cuántos objetos hay en la escena?
- ¿Dónde está localizado cada uno de ellos?
- Información acerca del tamaño o las dimensiones de cada uno de los objetos.
- Información relativa al movimiento y la orientación.
- Información relativa a los desplazamientos entre instantes de tiempo consecutivos y velocidad del objeto.

Información temporal (Cuándo): el sistema debe conocer cuál es el momento actual y ofrecer la posibilidad de manejar intervalos y referencias temporales. Existen situaciones que son sólo normales siempre y cuando transcurran en un tiempo inferior a un tiempo máximo establecido, o bien, si suceden en intervalos concretos de

tiempo. Por ejemplo, ciertas acciones que realizan personas en un museo durante el horario de apertura, pueden ser consideradas como anómalas cuando el museo tiene cerrada sus puertas al público. Además, el tratamiento de la información temporal es fundamental para la representación y detección de eventos complejos, los cuales son una secuencia de eventos simples relacionados temporalmente.

Clase a la que pertenece cada objeto (Qué): La normalidad o anomalía de un comportamiento está ligada, en la mayoría de las ocasiones, a la clase a la que pertenece el objeto que realiza la acción. Existen comportamientos que son normales siempre y cuando los lleven a cabo ciertas clases de objetos. Por ejemplo, si el sistema detecta un objeto en movimiento en un jardín, puede ser una situación normal siempre y cuando se trate de una persona o un animal, sin embargo, si el objeto es un vehículo la situación probablemente será anómala. Por este motivo, determinar la clase a la que pertenece cada objeto es vital para que el sistema puede diferenciar entre situaciones normales y anómalas.

Por otro lado, existe un flujo continuo de comunicación entre la capa inferior y la capa intermedia. Ésta última, recibe en cada momento la información espacio-temporal relativa a cada uno de los objetos detectados en la escena, que ha sido procesada en la capa inferior. Gran parte de las propuestas publicadas en la literatura no realizan una división clara entre la capa sensorial y la capa intermedia donde se analizan comportamientos. La mayoría de los autores centran sus trabajos en la capa inferior donde trabajan con datos de bajo nivel y establecen fuertes enlaces entre la forma de procesarlos y el análisis de comportamientos. Sin embargo, una división clara de estas capas permite que el análisis de sucesos sea independiente de los algoritmos empleados para procesar las señales y facilita la reutilización de conocimiento experto. Es decir, el cambio de algoritmos para el procesamiento de señales no tendría por qué influir en la forma de analizar o determinar si una situación es normal o no lo es.

El principal objetivo de la capa o nivel intermedio es interpretar lo que sucede en el entorno monitorizado y clasificar cada una de las situaciones como normales o anómalas para la activación de las alarmas y ofrecer soporte en la toma de decisiones. Para ello, se define la normalidad del entorno y las situaciones anómalas más comunes. La peor situación que puede ocurrir con este enfoque es que suceda una situación anómala que no ha sido definida previamente, pero al menos, el sistema es consciente de que “algo” sospechoso o no normal está sucediendo en el momento actual. Definir la normalidad de manera general en todo el entorno global puede ser una tarea demasiado complicada e incluso inabordable. Por este motivo, en este trabajo se opta por la fragmentación del problema en problemas de menor complejidad: el entorno se divide en entornos más pequeños (zonas específicas del entorno observadas por ciertos sensores) donde el número de posibles situaciones que pueden suceder es acotado, y se realiza la división del análisis de normalidad en análisis específicos mediante el uso

3.1. Esquema del diseño propuesto

| 79 |

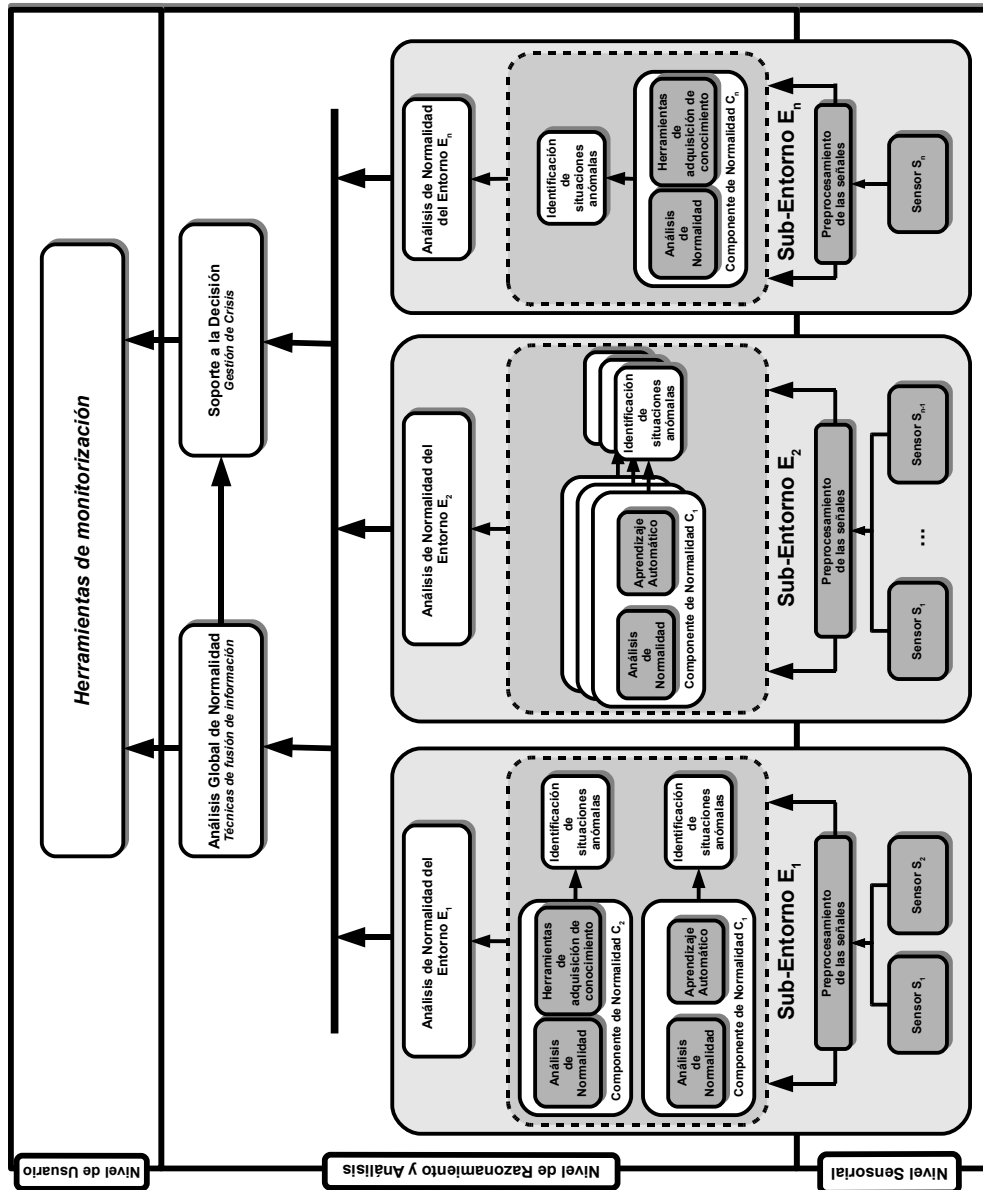


Figura 3.1: Esquema general en el que se refleja la construcción de sistemas de vigilancia basados en componentes de normalidad. La capa intermedia ofrece una visión general del modelo propuesto en este trabajo para el análisis de eventos y comportamientos.

de componentes de normalidad. Cada uno de estos componentes define de manera general cómo es la normalidad en un entorno cualquiera de acuerdo a un concepto¹ particular.

¹Algunos ejemplos de conceptos son: trayectorias, velocidad, normalidad en un paso de

Una vez que un componente ha sido definido de manera general ya puede ser instanciado para entornos particulares. Instanciar un componente implica crear ciertas particularizaciones. Por ejemplo, si se ha diseñado un componente para definir y analizar de manera general las trayectorias normales en cualquier escenario, la instanciación del componente implicaría definir las trayectorias normales específicas de ese entorno tal como indica dicho componente. Tal como muestra la Figura 3.1 se establecen asociaciones entre los componentes y herramientas de adquisición de conocimiento y algoritmos de aprendizaje automático con el objetivo de facilitar la instanciación. El uso de este tipo de herramientas es aconsejable en entornos dinámicos donde las condiciones ambientales pueden variar con cierta facilidad o en entornos donde el número de instancias necesario es demasiado elevado. Por tanto, el diseño de sistemas de vigilancia basado en las ideas antes citadas, a saber: capas, componentes y subentornos, presenta algunas de las siguientes ventajas:

- **Reutilización.** La definición de componentes independientes donde se especifica de forma general como definir la normalidad de un concepto, independientemente del contexto de aplicación, facilita la reutilización de conocimiento experto.
- **Reducción del tiempo del ciclo de desarrollo** como consecuencia de la ventaja anterior.
- **Mejora de la calidad.** La reutilización de experiencias previas evita cometer los mismos errores que se han cometido en el pasado.
- **Fiabilidad.** Si un componente ha sido probado con anterioridad en multitud de entornos y ello ha permitido depurar los posibles errores existentes, la inclusión de dicho componente en un nuevo sistema de seguridad garantiza una alta probabilidad de éxito.
- **Flexibilidad.** La combinación de unos componentes u otros permiten construir diferentes tipos de sistemas de vigilancia. Se podría decir que los componentes son los “ingredientes” de la aplicación y que se combinan de una forma u otra para conseguir un objetivo concreto, en este caso, el análisis de situaciones particulares.

Por otro lado, cada uno de los componentes de normalidad tiene asociado un módulo para la identificación de situaciones anómalas² relacionadas con el concepto a partir del cual ha sido creado el componente. Por ejemplo, si un componente de normalidad estudia las trayectorias normales, el módulo de detección de situaciones anómalas asociado trataría

peatones, relaciones de proximidad entre objetos normales, etc.

²Cómo bien se comentó anteriormente, el análisis de vigilancia más adecuado es el estudio de las situaciones normales y las situaciones anómalas más comunes. El presente trabajo de investigación se centra principalmente en la representación y el análisis de situaciones normales, mientras que la representación y detección de situaciones anómalas particulares serán tratadas en trabajos futuros.

de identificar circulaciones no permitidas. Cuando un componente de normalidad determina que un objeto no se comporta de manera adecuada con respecto a un concepto, activa el módulo de identificación de situaciones anómalas. La combinación de los diferentes análisis de normalidad dados por cada uno de los componentes y las salidas de los módulos de identificación de anomalías constituyen el análisis global de normalidad en un sub-entorno E_i . De la misma forma, el análisis global en el entorno E es el resultado de aplicar diferentes técnicas de fusión sobre los diferentes análisis realizados en cada uno de los sub-entornos. El sistema utilizará esta información para ofrecer ayuda al personal de seguridad en la toma de decisiones ante situaciones críticas. Finalmente, en la capa superior se encuentran las herramientas de monitorización donde el personal de seguridad puede observar los resultados obtenidos en las etapas inferiores.

Actualmente no existen soluciones que ofrezcan las suficientes garantías como para sustituir completamente a los agentes humanos pero, sin embargo, este tipo de sistemas sí que pueden resultar de gran ayuda para reducir la carga de trabajo en operadores humanos y ofrecer consejos o proponer soluciones en situaciones complejas donde se han desencadenado una sucesión de situaciones anómalas.

En la siguiente Sección se presenta un modelo formal para la definición de la normalidad en entornos monitorizados y la construcción de sistemas de vigilancia escalables y flexibles. En los sucesivos capítulos se hará uso del modelo para la creación de dos componentes de normalidad: un componente para el análisis de trayectorias y un segundo componente para el análisis de las velocidades de los objetos.

3.2. Normalidad en entornos monitorizados

El modelo que se presenta a continuación establece las bases para la definición de componentes de normalidad que analizan los comportamientos y eventos en un entorno observado en base a un concepto, y que puede ser empleado en esquemas de diseño como el que se ha presentado. Dicho modelo se basa principalmente en la definición de restricciones difusas de diferente naturaleza (restricciones de roles, espaciales, temporales, etc.) y la manera en la que se satisfacen las mismas: grados de normalidad. Estos grados representan una forma cuantitativa de medir la normalidad del comportamiento de un objeto, en base a un concepto y a partir de un conjunto de observaciones y cambios de estados.

La Lógica Difusa [211] es un modelo matemático ampliamente utilizado para el tratamiento de la incertidumbre y la vaguedad presentes en el mundo real. El concepto de normalidad, dadas sus características, es un concepto inherentemente vago [212]. Existen casos en los que se producen las suficientes evidencias como para considerar claramente que la situación

actual es anómala, sin embargo, existen muchos otros casos en los que la catalogación de la situación actual como normal o anómala no es tan clara, pudiendo pertenecer a ambas clases con diferentes grados de pertenencia. Además, en multitud de ocasiones hay que trabajar con datos proporcionados por las capas inferiores que no son del todo precisos. El objetivo principal de los sistemas de vigilancia construidos con el modelo propuesto en este capítulo es determinar el grado de normalidad del momento actual una vez que hayan sucedido los hechos, es decir, en ningún momento pretenden anticiparse a posibles acciones futuras, sino que interpretan sucesos pasados.

3.2.1. Problema de la vigilancia: divide y vencerás

La aplicación de la técnica de divide y vencerás puede ser muy útil para el problema de la vigilancia y el diseño de sistemas de vigilancia inteligentes. Teniendo en mente esta idea, el problema de la vigilancia se puede definir de la siguiente manera:

Definición. El problema de la vigilancia P , en un entorno E , se define como la interpretación de las múltiples percepciones obtenidas a partir de los sensores instalados en el entorno monitorizado $\{S_1, S_2, \dots, S_n\}$. Cada percepción del entorno global se considera, a su vez, como un entorno E_i . Así, la definición global de normalidad en el entorno E , se realiza mediante la definición particular de normalidad en cada subentorno E_i . De esta forma, se reduce la complejidad a la hora de definir la normalidad, ya que el número de situaciones que pueden suceder en un subentorno E_i es menor que en el entorno global E . Por lo tanto, un problema de vigilancia P , en un entorno E , puede ser visto como varios problemas de monitorización sobre varios subentornos más simples E_i y construidos a partir de las percepciones de los sensores disponibles en el entorno:

$$P = \{E_1, E_2, \dots, E_n\} \quad (3.1)$$

Ejemplo. Supóngase que existe la necesidad de analizar los comportamientos llevados a cabo por personas en un centro comercial, donde se han instalado varias cámaras de seguridad; E representaría el centro comercial, mientras que cada E_i representaría una porción o parte del centro comercial observada desde alguna de las cámaras instaladas en el entorno. El análisis de normalidad global de E se obtiene a partir del análisis individual de cada uno de los subentornos E_i .

Por otra parte, para una misma región E_i pueden existir diferentes tipos de sensores que aportan probablemente la misma información desde distintas perspectivas, los cuales constituyen a su vez distintos subentornos

(ver Figura 3.2). El análisis de normalidad en el entorno E_i dependerá del análisis de normalidad realizado en cada uno de los subentornos que perciben E_i .

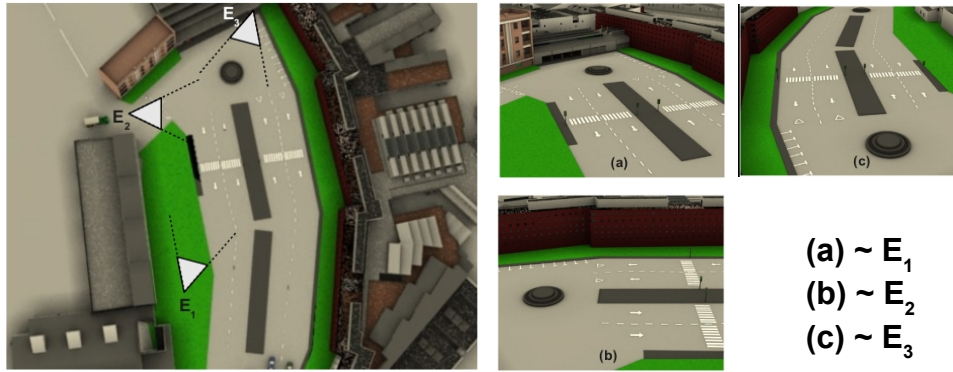


Figura 3.2: Escenario virtual monitorizado mediante el uso de tres cámaras de seguridad. Cada una de las cámaras observa parte del entorno y cada una de estas percepciones E_1 , E_2 y E_3 son consideradas a su vez como un subescenario E_i . En las Figuras (a), (b) y (c) es posible apreciar como diferentes cámaras pueden observar zonas comunes en un escenario global, por lo que la parte del sistema encargada de analizar los eventos puede recibir información sobre un mismo lugar procedente de múltiples fuentes.

3.2.2. Subproblema de la vigilancia: los entornos monitorizados

Cada uno de los entornos obtenidos por medio de la técnica divide y vencerás aplicada al problema de vigilancia, constituye un entorno monitorizado.

Definición. Un entorno monitorizado se define como una 4-tupla, la cual consta de los siguientes elementos:

$$E = \langle V; O; C; O \times C \rangle \quad (3.2)$$

donde cada uno de los elementos representa lo siguiente:

- V es el conjunto de variables de entrada que se utiliza para definir los conceptos, las propiedades de los objetos y el propio entorno. El conjunto de variables y los valores que toman representan el estado actual del entorno; los cambios que se producen en los valores de las variables reflejan los cambios de estado que se producen en el entorno, los cuales deben ser estudiados por el sistema de seguridad

para detectar posibles anomalías. Por otra parte, los valores de estas variables podrían ser obtenidos directamente de los sensores, o bien, a partir de algún proceso que transforme la información obtenida inicialmente. Además es importante recalcar que los valores podrían ser precisos, o bien, datos con imprecisión y vaguedad dependiendo de las características del sensor.

- O representa el conjunto de clases al que pueden pertenecer los objetos que actúan en el entorno y a los que el sistema debe prestar especial atención. Ejemplos de clases podrían ser: persona, grupo de personas, coche, camión, bicicleta, etc.
- C se refiere al conjunto de conceptos que se utilizan en el análisis de normalidad, los cuales vienen determinados por los aspectos que se quieren vigilar en el entorno.
- $O \times C$ determina los conceptos que deben emplearse para analizar la normalidad de cada tipo de objeto. Es decir, dependiendo de la clase a la que pertenezca un objeto, se utilizan unos conceptos u otros para determinar si su comportamiento es normal.

Ejemplo. Sea E_i parte de un entorno exterior observado desde una cámara de vigilancia, donde existen componentes urbanos clásicos tales como jardines, aceras, pasos de peatones, semáforos, carreteras, etc. En primer lugar, se especifican las clases de los objetos a los que el sistema debe prestar especial atención. Por ejemplo, $O = \{peaton, vehiculo\}$. Una vez que se ha tomado la decisión sobre qué tipo de objetos deben ser vigilados, es necesario decidir los aspectos a vigilar. Estos aspectos dan paso a la construcción de los componentes de normalidad. Un ejemplo podría ser: $C = \{trayectorias, velocidad, paso\_de\_peatones\}$. Es decir, se estudiará la normalidad de acuerdo a las trayectorias, velocidades y el comportamiento de los peatones y vehículos en los pasos de peatones.

La elección de los objetos a vigilar y los conceptos de normalidad dan paso a la selección de variables necesarias para cada componente. Ejemplos de estas variables pueden ser: localización de un objeto, clase a la que pertenece, tamaño, tiempo actual, lista de áreas importantes en el escenario, ubicación de elementos relevantes sin capacidad de movimiento, etc. Finalmente, se seleccionan los conceptos que se utilizarán para analizar el comportamiento de cada clase de objeto y se especifica en $O \times C = \{(peatón, trayectorias), (peatón, paso de peatones), (vehículo, trayectorias), (vehículo, velocidad), (vehículo, paso de peatones)\}$. En la configuración anterior, un peatón se comporta de forma normal si realiza trayectorias normales y si se comporta de forma adecuada en los pasos de peatones. En el caso de los vehículos también se analiza la velocidad.

3.2.3. Vigilancia basada en el análisis de conceptos

Como ya se ha comentado previamente, la vigilancia en un entorno se va a realizar en base a conceptos, existirá un concepto por cada uno de los aspectos del entorno que se quieran vigilar.

Definición. Un concepto c_i ($c_i \in C$) se define como una 3-tupla compuesta por los siguientes elementos:

$$c_i = \langle V_i; DDV_i; \Phi_i \rangle \quad (3.3)$$

donde V_i es el conjunto de variables de entrada que se utilizan para definir el concepto c_i , tal que $V_i \subseteq V$. Por otro lado, DDV_i es el conjunto de dominios de definición de las variables que pertenecen a V_i . Por tanto, si $V_i = \{v_{1i}, v_{2i}, \dots, v_{ni}\}$, entonces DDV_i se define como $DDV_i = \{DDV_{1i}, DDV_{2i}, \dots, DDV_{ni}\}$, donde DDV_{ji} es el dominio de definición de la variable v_{ji} . El dominio de definición de una variable especifica los posibles valores que ésta puede tomar. Finalmente, Φ_i es el conjunto de restricciones que se utilizan para completar la definición del concepto c_i , de acuerdo a los elementos de V_i ($\Phi_i = \{\mu_{1i}, \mu_{2i}, \dots, \mu_{ki}\}$). Tal como se comentará más adelante, el análisis de normalidad de un concepto c_i depende de cómo se satisfagan las restricciones definidas para dicho concepto. Algunas clases de restricciones podrían ser las siguientes:

- **Restricciones de rol o clase de un objeto:** son restricciones asociadas a una clase y determinan que tipo de objetos pueden realizar ciertas acciones.
- **Restricciones espaciales:** este tipo de restricciones permiten definir la normalidad de un concepto de acuerdo a una serie de características espaciales, tales como la ubicación de un objeto, su tamaño, distancias entre objetos, etc.
- **Restricciones temporales:** permiten definir la normalidad de un concepto de acuerdo a una serie de características temporales, tales como el tiempo mínimo o máximo en el que se debe realizar una acción, intervalo de tiempo en el que deben transcurrir los eventos, etc.

3.2.4. Restricción de normalidad

A continuación se verá que se entiende por restricción en el modelo propuesto y cómo se define.

Definición. Una restricción de normalidad, asociada a un concepto i (c_i), se define como un conjunto difuso X_i sobre el dominio $\mathcal{P}(V_i)$, con una función

de pertenencia asociada μ_{X_i} :

$$\mu_{X_i} : \mathcal{P}(V_i) \longrightarrow [0, 1] \quad (3.4)$$

donde 1 representa el máximo grado de satisfacción de la restricción y 0 el caso contrario. El resto de valores representan valores intermedios entre el cumplimiento total de la normalidad y el no cumplimiento absoluto.

En algunas ocasiones puede resultar más práctico la definición de algunas restricciones mediante conjuntos crisp³. En tal caso, la función de pertenencia a dicho conjunto sería:

$$\mu_{X_i}(x) = \begin{cases} 1 & \text{if } x \in X_i; \\ 0 & \text{if } x \notin X_i; \end{cases}$$

Operaciones entre restricciones

La definición de nuevas restricciones es posible como combinación de otras más simples. Para ello se propone el uso de operaciones pertenecientes al álgebra clásica de conjuntos difusos, tal que la aplicación de estas operaciones sobre restricciones simples dan como resultado una nueva restricción [111]. Sean A y B dos restricciones de normalidad definidas sobre el dominio $\mathcal{P}(V_i)$:

a) Unión ($A \cup B$) : Un elemento del entorno satisface la restricción $A \cup B$ siempre y cuando satisfaga A o satisfaga B . La función de pertenencia de la nueva restricción se define como $\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x)$

$$\mu_{A \cup B}(x) = \begin{cases} \mu_A(x) & \text{if } \mu_A(x) \geq \mu_B(x) \\ \mu_B(x) & \text{if } \mu_A(x) < \mu_B(x) \end{cases}$$

o, dicho de otra forma, $\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$.

b) Intersección ($A \cap B$) : Un elemento del entorno satisface la restricción $A \cap B$ siempre y cuando satisfaga A y satisfaga B . La función de pertenencia de la nueva restricción se define como $\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x)$

$$\mu_{A \cap B}(x) = \begin{cases} \mu_A(x) & \text{if } \mu_A(x) \leq \mu_B(x) \\ \mu_B(x) & \text{if } \mu_A(x) > \mu_B(x) \end{cases}$$

o, dicho de otra forma, $\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$.

c) Complemento ($\bar{A}$): Un elemento del entorno satisface la restricción $\bar{A}$ siempre y cuando no satisfaga la restricción A , o dicho de otra forma, “la

³Un elemento no puede pertenecer a un conjunto crisp parcialmente, es decir, el objeto pertenece totalmente o no pertenece al conjunto.

situación actual es normal siempre y cuando no se produzca cierto evento”. La función de pertenencia de $\bar{A}$, se definiría como: $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$.

Para crear nuevas restricciones es posible combinar cualquiera de las operaciones anteriores. Por ejemplo, sean A, B, C y D restricciones definidas sobre el dominio $\mathcal{P}(V_i)$. Un elemento del entorno satisface la restricción $A \cup (B \cap C) \cup \bar{D}$ siempre y cuando satisfaga A o, B y C al mismo tiempo o que no suceda D .

Propiedades de las restricciones de normalidad

Las propiedades que cumplen las restricciones de normalidad en un entorno observado son exactamente las mismas que las propiedades existentes para los conjuntos difusos. Sean A, B y C restricciones de normalidad definidas sobre el dominio $\mathcal{P}(V_i)$:

1. **Idempotencia:** $A \cup A = A$; $A \cap A = A$
2. **Conmutativa:** $A \cup B = B \cup A$; $A \cap B = B \cap A$
3. **Asociativa:** $A \cup (B \cup C) = (A \cup B) \cup C$; $A \cap (B \cap C) = (A \cap B) \cap C$
4. **Distributiva:** $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$; $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
5. **Doble negación:** $\neg(\neg A) = A$
6. **Transitividad:** Si $A \subset B \wedge B \subset C \rightarrow A \subset C$
7. **Condiciones límite:** $A \cup \emptyset = A$; $A \cap \emptyset = \emptyset$; $A \cup \mathcal{P}(V_i) = \mathcal{P}(V_i)$; $A \cap \mathcal{P}(V_i) = A$

3.2.5. Instancia de un concepto

Una vez que se ha definido un concepto y sus restricciones de forma general, es necesario realizar particularizaciones de dicho concepto para entornos concretos. Volviendo al ejemplo del concepto *trayectoria normal*, la elección de las variables, la especificación de los dominios de definición y la definición de las funciones de pertenencia de las restricciones, establecerían los mecanismos necesarios para la definición de trayectorias. Estos mecanismos son los que se deben emplear para definir las trayectorias normales particulares de un entorno concreto.

Definición. Una instancia y de un concepto c_i en un entorno E_j ($E_j \in P$), representado como c_{iy}^j , se define como:

$$c_{iy}^j = \langle V_i; DDV_i; \tilde{\Phi}_i = \{\tilde{\mu}_{1i}, \tilde{\mu}_{2i}, \dots, \tilde{\mu}_{zi}\} \rangle \quad (3.5)$$

tal que $\tilde{\Phi}_i$ es un conjunto de restricciones particularizadas del conjunto Φ_i , es decir una restricción empleada en la definición, verificándose que $|\Phi_i| \geq |\tilde{\Phi}_i|$. Cada $\tilde{\mu}_{xi} \in \tilde{\Phi}_i$ representa la particularización de $\mu_{xi} \in \Phi_i$.

Ejemplo. Si c_i es el concepto *trayectoria normal*, una instancia de c_i sería una posible trayectoria normal y, en consecuencia, todas las instancias de c_i representarían todas las trayectorias normales que existen en el entorno E_j .

3.2.6. Instancia de una restricción de normalidad

La particularización o instanciación de una restricción de normalidad, sirve para adaptar las definiciones generales de un tipo de análisis basado en un concepto a un entorno concreto. A continuación se verá qué se entiende por instancia de una restricción de normalidad.

Definición. Una instancia de una restricción de normalidad es un conjunto difuso definido sobre $\mathcal{P}(DDV_i)$ con una función de pertenencia asociada ($\tilde{\mu}_{X_i}$):

$$\tilde{\mu}_{X_i} : \mathcal{P}(DDV_i) \longrightarrow [0, 1] \quad (3.6)$$

tal que si $v_{ki} \in \mathcal{P}(V_i)$ se utiliza para definir μ_{X_i} , entonces los valores de v_{ki} definidos en $DDV_{ki} \in \mathcal{P}(DDV_i)$ se utilizan para crear la instancia $\tilde{\mu}_{X_i}$.

Ejemplo. Si μ_{X_i} representa una restricción espacial, definida de forma general, que comprueba si un objeto está siguiendo de manera correcta una secuencia de áreas, $\tilde{\mu}_{x_i}$ representaría una instancia de la restricción μ_{X_i} , la cual comprueba si un objeto está siguiendo una secuencia particular de áreas en el entorno observado.

3.2.7. Grado de normalidad asociado a una instancia

Cada objeto deberá tener asociado un grado de normalidad que establezca cómo de normal es su comportamiento con respecto a cada concepto asociado a él, y esto se hará a partir de las instancias que definen el concepto en el entorno.

Definición. El grado de normalidad de un objeto obj en un entorno E_j ($E_j \in P$), con respecto a una instancia y del concepto c_i (c_{iy}^j), y representado como $N_{c_{iy}^j}(obj)$, es calculado a partir de los valores obtenidos para cada $\tilde{\mu}_{x_i}$ de la siguiente forma:

$$N_{c_{iy}^j}(obj) = \bigwedge_{x=1}^{|\tilde{\Phi}_i|} \tilde{\mu}_{x_i}^y \quad (3.7)$$

siendo $\bigwedge$ una t-norma, por ejemplo, la t-norma que calcula el valor mínimo.

Ejemplo. Si c_i es el concepto *trayectoria normal* y c_{i1}^j es una trayectoria normal particular definida en el entorno E_j a partir de un conjunto de restricciones μ_{X_i} y particularizaciones de estas restricciones ($\tilde{\mu}_{X_i}^1$), entonces $N_{c_{i1}^j}$ determinaría el grado de normalidad del objeto obj con respecto a la trayectoria c_{i1}^j . Un valor elevado de $N_{c_{i1}^j}$ significaría que el objeto obj está siguiendo de una forma adecuada la trayectoria c_{i1}^j , y un valor bajo representaría el caso contrario, o bien, directamente, que el objeto no está siguiendo esa trayectoria. En otras palabras, un valor bajo del parámetro $N_{c_{i1}^j}$ indica que no se satisfacen una o varias restricciones de la instancia c_{i1}^j .

Los operadores t-norma son una familia de operadores “and” o “y-lógico” de la lógica multivaluada [208] y se pueden representar de la siguiente forma [206]:

$$T : [0, 1] \times [0, 1] \rightarrow [0, 1] \quad (3.8)$$

tal que

1. $T(a, b) = T(b, a)$
2. $T(a, b) \geq T(c, d)$ si $a \geq c$ y $b \geq d$
3. $T(a, T(b, c)) = T(T(a, b), c)$
4. $T(1, a) = a$

siendo a y b valores numéricos pertenecientes al intervalo $[0, 1]$. Algunos ejemplos de t-normas utilizadas con frecuencia y que satisfacen las propiedades anteriores son:

- $T(a, b) = \min(a, b)$
- $T(a, b) = a \cdot b$
- $T(a, b) = \max(0, a + b - 1)$

Una propiedad importante de los operadores t-norma, es que cualquiera de ellos siempre es menor o igual que el valor obtenido por el operador que calcula el valor mínimo [207], es decir:

$$T(a, b) \leq \min(a, b) \quad (3.9)$$

La elección de la t-norma del mínimo es apropiada para los sistemas de vigilancia y, en particular, para el modelo propuesto en este Capítulo. El modelo está basado en el planteamiento de un conjunto de restricciones y

el análisis del cumplimiento de las mismas. En un sistema de seguridad lo que interesa es detectar posibles incumplimientos de estas restricciones o normas, por lo que el valor mínimo puede ofrecer un valor real sobre la situación actual. Es decir, cuando alguna de las restricciones no se satisface, el valor de normalidad $N_{c_{iy}}^j(obj)$ será bajo. Utilizar otras t-normas, implicaría valores más bajos que el mínimo, una vigilancia más estricta y, por tanto, un posible exceso de activaciones de alarmas. Por ejemplo, si se empleara la t-norma $T(a, b) = a \cdot b$ y dos restricciones cualesquiera se satisfacen con grado 0.6, el grado de normalidad $N_{c_{iy}}^j(obj)$ sería $0.6 \times 0.6 = 0.36$, lo cual carecería de sentido ya que la totalidad de las restricciones se han satisfecho con un grado mucho mayor. En cambio si se utilizara la t-norma del mínimo el valor de normalidad sería $\min(0.6, 0.6) = 0.6$, un valor bastante más elevado que 0.36 obtenido con la t-norma anterior y que representa mejor el modo en el que se satisfacen las restricciones.

Finalmente, la Tabla 3.1 muestra algunos ejemplos de esta familia de operadores propuestos por diferentes autores. Bonnisone en [28] y Yager en [206] proponen algunos criterios para elegir el operador más adecuado de acuerdo a las principales características del problema que se pretende resolver.

3.2.8. Grado de normalidad asociado a un concepto

Una vez conocidos los grados de normalidad de un objeto para cada instancia de un determinado concepto, habrá que determinar cómo de normal es su comportamiento según el conjunto de todos ellos. De esta forma se obtiene una idea general del comportamiento del objeto con respecto a un concepto.

Definición. El grado de normalidad de un objeto obj , en un entorno E_j ($E_j \in P$), de acuerdo a un concepto c_i y representado como $N_{c_i}^j(obj)$, es calculado de la siguiente forma:

$$N_{c_i}^j(obj) = \bigvee_{y=1}^w N_{c_{iy}}^j(obj) \quad (3.10)$$

donde w es el número de instancias del concepto c_i y $\bigvee$ es un operador t-conorma, por ejemplo la t-conorma del máximo. Los operadores t-conorma son una familia de operadores “or” o “o-lógico” de la lógica multivaluada [208] y se pueden representar de la siguiente forma [206]:

$$S : [0, 1] \times [0, 1] \rightarrow [0, 1] \quad (3.11)$$

Ref	Formula $T(a, b)$	Generador de Decremento $f(a)$	Rango del parámetro	A medida que el parámetro converge a 0	A medida que el parámetro converge a 1 o -1	A medida que el parámetro converge a ∞ o $-\infty$
[161]	$\{(\max(0, a^p + b^p - 1))^{\frac{1}{p}}\}$	$1 - a^p$	$p \neq 0$	ab	$\max(0, a+b-1)$, cuando $p = 1$; $\frac{ab}{a+b-ab}$, cuando $p = -1$	$T_{\min}(a, b)$ cuando $p \rightarrow \infty$; $\min(a, b)$ cuando $p \rightarrow -\infty$
[91]	$\frac{ab}{r+(1-r)(a+b-ab)}$	$-\ln(\frac{a}{r+(1-r)a})$	$r > 0$	$\frac{ab}{a+b-ab}$	ab cuando $r = 1$	$T_{\min}(a, b)$
[67]	$\frac{ab}{\max(a, b, \alpha)}$		$\alpha \in [0, 1]$	$\min(a, b)$	ab cuando $\alpha = 1$	
[205]	$1 - \min\{1, [(1 - a)^w + (1 - b)^w]^{\frac{1}{w}}\}$	$(1 - a)^w$	$w > 0$	$T_{\min}(a, b)$	$\max(0, a + b - 1)$ cuando $w = 1$	$\min(a, b)$
[197]	$\max(0, \frac{a+b+\lambda ab-1}{1+\lambda})$	$\frac{1}{\lambda} \ln[1 + \lambda(1 - a)]$	$\lambda > -1$	$\max(0, a + b - 1)$	$T_{\min}(a, b)$ cuando $\lambda \rightarrow -1$; $\max[0, (a + b + ab - 1)/2]$ cuando $\lambda = 1$	ab
[209]	$\max[0, (1 + \lambda)(a + b - 1) - \lambda ab]$	$\frac{1}{\lambda} \ln \frac{1+\lambda}{1+\lambda a}$	$\lambda > -1$	$\max(0, a + b - 1)$	ab a medida que $\lambda \rightarrow -1$; $\max[0, 2(a + b) - ab/2 - 1]$ cuando $\lambda = 1$	$T_{\min}(a, b)$
[161]	$1 - [((1 - a)^p + (1 - b)^p - (1 - a)^p(1 - b)^p)^{\frac{1}{p}}]$	$\ln[1 - (1 - a)^p]^{\frac{1}{p}}$	$p > 0$	$T_{\min}(a, b)$	ab cuando $p = 1$	$\min(a, b)$

Tabla 3.1: Ejemplos de intersecciones difusas (t-normas) [111]. $T_{\min}$ representa la intersección drástica, es decir, $T_{\min}(a, b)$ es a , cuando $b = 1$; b cuando $a = 1$ y 0 en caso contrario. Además siempre se cumple la siguiente relación: $T_{\min}(a, b) \leq \max(0, a + b - 1) \leq ab \leq \min(a, b)$

tal que todo operador t-conorma satisface las siguientes propiedades:

1. $S(a, b) = S(b, a)$
2. $S(a, b) \geq S(c, d)$ si $a \geq c$ y $b \geq d$
3. $S(a, S(b, c)) = S(S(a, b), c)$
4. $S(0, a) = a$

siendo a y b dos valores numéricos pertenecientes al intervalo $[0, 1]$. Algunos ejemplos de t-conormas utilizadas con frecuencia y que satisfacen las propiedades anteriores son:

- $S(a, b) = \max(a, b)$
- $S(a, b) = a + b - a \cdot b$

La Tabla 3.2 muestra algunos ejemplos de esta familia de operadores propuestos por diversos autores. Del mismo modo que $T(a, b) = \min(a, b)$ representa el máximo valor que puede ser obtenido mediante la aplicación de una t-norma, $S(a, b) = \max(a, b)$ representa el mínimo valor obtenido con un operador t-conorma, es decir:

$$S(a, b) \geq \max(a, b) \quad (3.12)$$

Por otro lado, si se tiene en cuenta el modo en el que se emplean los operadores *t-norma* y *t-conorma*, es posible afirmar que el análisis de normalidad de un objeto obj , de acuerdo a un concepto c_i en un entorno E_j , es el resultado de aplicar una red difusa “AND-OR” sobre un conjunto de restricciones:

$$\begin{array}{ccccccc}
 c_{i1}^j = & \tilde{\mu}_{1i}^1 \wedge \tilde{\mu}_{2i}^1 \wedge & \dots & \tilde{\mu}_{ni}^1 = & N_{c_{i1}}^j(obj) \\
 & & & \vee & \\
 c_{i2}^j = & \tilde{\mu}_{1i}^2 \wedge \tilde{\mu}_{2i}^2 \wedge & \dots & \tilde{\mu}_{ni}^2 = & N_{c_{i2}}^j(obj) \\
 \cdot & \cdot & \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot & \cdot & \vee \\
 \cdot & \cdot & \cdot & \cdot & \cdot \\
 c_{iw}^j = & \tilde{\mu}_{1i}^w \wedge \tilde{\mu}_{2i}^w \wedge & \dots & \tilde{\mu}_{ni}^w = & N_{c_{iw}}^j(obj)
 \end{array}$$

$$N_{c_i}^j(obj)$$

Ref	Formula $T(a, b)$	Generador de Decremento $f(a)$	Rango del parámetro	A medida que el parámetro converge a 0	A medida que el parámetro converge a 1 o -1	A medida que el parámetro converge a ∞ o $-\infty$
[161]	$1 - \{(\max(0, (1 - a^p) + (1 - b^p) - 1)^{\frac{1}{p}}\}$	$1 - (1 - a)^p$	$p \neq 0$	$a + b - ab$	$\min(1, a + b)$, cuando $p = 1$; $\frac{a+b-2ab}{1-ab}$, cuando $p = -1$	$S_{max}(a, b)$ cuando $p \rightarrow \infty$; $\min(a, b)$ cuando $p \rightarrow -\infty$
[91]	$\frac{a+b+(r-2)ab}{r+(r-1)ab}$	$-\ln(\frac{1-a}{r+(1-r)(1-a)})$	$r > 0$	$\frac{a+b-2ab}{1-ab}$	$a + b - ab$ cuando $r = 1$	$S_{max}(a, b)$
[67]	$1 - \frac{(1-a)(1-b)}{\max((1-a), (1-b), \alpha)}$		$\alpha \in [0, 1]$	$\max(a, b)$	$a + b - ab$ cuando $\alpha = 1$	
[205]	$\min[1, (a^w + b^w)^{\frac{1}{w}}]$	a^w	$w > 0$	$S_{max}(a, b)$	$\min(1, a + b)$ cuando $w = 1$	$\max(a, b)$
[197]	$\min(1, a + b - \frac{\lambda}{1-\lambda}ab)$	$\frac{1}{\lambda} \ln \frac{1+\lambda}{a+\lambda(1-a)}$	$\lambda > -1$	$\min(1, a + b)$	$S_{max}(a, b)$ cuando $\lambda \rightarrow -1$; $\min(1, a + b - ab/2)$ cuando $\lambda = 1$	$a + b - ab$
[209]	$\min(1, a + b + \lambda ab)$	$\frac{1}{\lambda} \ln(1 + \lambda a)$	$\lambda > -1$	$\min(1, a + b)$	$a + b - ab$ a medida que $\lambda \rightarrow -1$; $\min(1, a + b + ab)$ cuando $\lambda = 1$	$S_{max}(a, b)$
[161]	$[a^p + b^p - a^p b^p]^{\frac{1}{p}}$	$\ln[1 - a^p]^{\frac{1}{p}}$	$p > 0$	$S_{max}(a, b)$	$a + b - ab$ cuando $p = 1$	$\max(a, b)$

Tabla 3.2: Ejemplos de uniones difusas [111](t-conormas). S_{max} representa la unión drástica, es decir, $S_{max}(a, b)$ es a , cuando $b = 0$; b cuando $a = 0$ y 1 en caso contrario. Además siempre se cumple la siguiente relación: $\max(a, b) \leq a + b - ab \leq \min(1, a + b) \leq S_{max}(a, b)$

Ejemplo. Si c_i es el concepto *trayectoria normal*, el valor de normalidad $N_{c_i^j}(obj)$ de cada objeto de acuerdo al concepto, se calcula mediante la aplicación de un operador t-conorma sobre los valores $N_{c_{iy}^j}(obj)$ obtenidos para cada una de las instancias, es decir, para cada una de las trayectorias normales definidas mediante la instanciación de variables y restricciones.

El objetivo que se pretende alcanzar con el cálculo del valor de normalidad asociado a un concepto ($N_{c_i^j}(obj)$) es determinar si un objeto satisface las restricciones de alguna de las instancias definidas para el concepto. Por este motivo, la t-norma del máximo es un operador apropiado para el cálculo del valor de normalidad $N_{c_i^j}(obj)$. En el ejemplo siguiente, un objeto satisface notablemente las restricciones definidas para la instancia c_{i2}^j del concepto c_i , ya que el valor para $N_{c_{i2}^j}(obj)$ es 0.8.

$$\begin{array}{rcl}
 c_{i1}^j = & \tilde{\mu}_{1i}^1 \wedge \tilde{\mu}_{2i}^1 \wedge \dots & \tilde{\mu}_{ni}^1 = N_{c_{i1}^j}(obj) = 0,2 \\
 & & \vee \\
 c_{i2}^j = & \tilde{\mu}_{1i}^2 \wedge \tilde{\mu}_{2i}^2 \wedge \dots & \tilde{\mu}_{ni}^2 = N_{c_{i2}^j}(obj) = \mathbf{0.8} \\
 \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots \\
 & & \vee \\
 c_{iw}^j = & \tilde{\mu}_{1i}^w \wedge \tilde{\mu}_{2i}^w \wedge \dots & \tilde{\mu}_{ni}^w = N_{c_{iw}^j}(obj) = 0,0
 \end{array}$$

$$N_{c_i^j}(obj) = 0,8$$

En resumen, el grado de normalidad de un objeto asociado a un concepto, $N_{c_i^j}(obj)$, es un valor numérico perteneciente al intervalo $[0,1]$ y es una muestra representativa del comportamiento de un objeto con respecto a un concepto o aspecto de vigilancia. Valores altos de este parámetro representan situaciones normales en el entorno observado, mientras que valores bajos representan el caso contrario.

3.2.9. Comportamiento normal en un entorno en base a un concepto

Los grados de normalidad $N_{c_i^j}(obj)$, descritos en la sección anterior, permitirán al sistema decidir si un objeto se comporta de forma normal en base a un concepto. Este tipo de decisiones es muy importante para los sistemas de seguridad ya que el objetivo final es activar una serie de alarmas o llamar la atención del personal de seguridad cuando una situación excede los límites de la normalidad. Una forma sencilla de decidir si el comportamiento de un objeto es normal sería mediante el uso de valores umbrales α_{c_i} . Es decir, un objeto obj tiene un comportamiento normal de

acuerdo a la definición del concepto c_i en un entorno E_j siempre y cuando se satisfaga la siguiente condición:

$$N_{c_i}^j(obj) \geq \alpha_{c_i} \quad (3.13)$$

siendo α_{c_i} un valor umbral asociado al concepto c_i y definido por un experto en seguridad que conoce bien el entorno donde se establece el valor del parámetro. Dicho experto tiene en cuenta experiencias previas para asignar el valor adecuado. Normalmente, $\alpha_{c_i} \in [0, 1]$ toma valores comprendidos entre $0,2 < \alpha_{c_i} < 0,5$, donde valores altos implican un análisis estricto de vigilancia, es decir, los valores de normalidad deben ser muy altos para que el sistema considere que un objeto satisface correctamente las restricciones definidas para el concepto c_i , mientras que valores bajos representan el caso contrario.

Sin embargo, este método para decidir si el comportamiento de un objeto es normal a partir de los grados de normalidad $N_{c_i}^j(obj)$ presenta algunos inconvenientes importantes:

- El experto debe especificar un valor umbral α_{c_i} para cada concepto c_i . El ajuste de este parámetro no es una tarea sencilla y depender de un único valor numérico no dota al sistema de demasiada flexibilidad.
- La función de salida que toma como valor de entrada un grado de normalidad $N_{c_i}^j(obj)$ y como salida el valor *normal/anómalo*, es una función escalonada. Tal como puede apreciarse en la Figura 3.3 un valor $N_{c_i}^j(obj)=0.29$ implicaría una situación anómala, sin embargo, $N_{c_i}^j(obj) = 0.3$ implicaría una situación totalmente normal. La diferencia es mínima, tan sólo 0.01, por lo que el cambio de anormal a normal es excesivamente brusco.

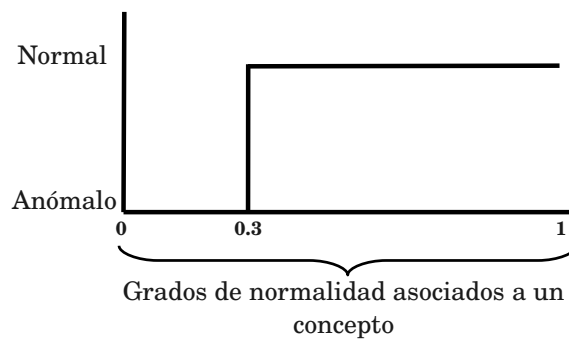


Figura 3.3: Función escalonada de salida para la clasificación de una situación como *normal/anormal* en base a un grado de normalidad.

Para solucionar los problemas planteados en el método anterior, se empleará una partición difusa del dominio $[0,1]$. Cada vez que se calcula

el grado de normalidad del comportamiento de un objeto con respecto a un concepto, se estudia la pertenencia de este valor a los conjuntos difusos mostrados en el dominio de definición de la Figura 3.4.

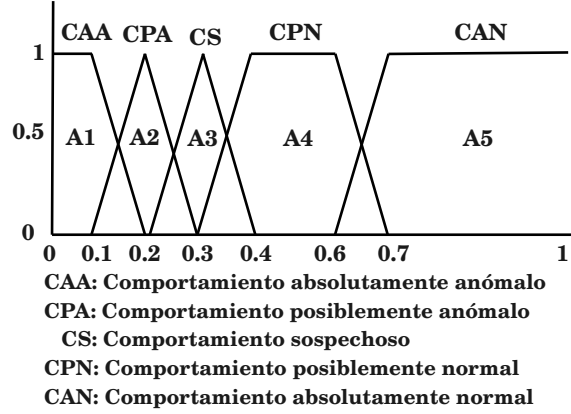


Figura 3.4: Dominio de definición para $N_{c_i}^j$. Cada uno de los conjuntos difusos representa un posible grado de normalidad. Los límites definidos para cada uno de los conjuntos difusos pueden variar en función del nivel de seguridad que se pretenda alcanzar; cuanto mayores sean los conjuntos CAA y CPA más estricto será el sistema.

De esta forma, el comportamiento de un objeto, o el estado actual de un elemento del entorno, puede ser absolutamente anómalo, posiblemente anómalo, sospechoso, posiblemente normal y absolutamente normal; pudiendo pertenecer a varios conjuntos al mismo tiempo. Es decir, la variable $N_{c_i}^j$ tomará los valores del dominio difuso $DDV_{N_{c_i}^j} = \{A_1, A_2, A_3, A_4, A_5\}$, donde cada A_i representa el conjunto difuso en la posición i , mostrado en la Figura 3.4. Este dominio difuso debe verificar las siguientes propiedades [66]:

$$1. \forall A_i \in DDV_{N_{c_i}^j},$$

$$Altura(A_i) = 1. \quad (3.14)$$

$$2. \forall (A_i, A_j) \in DDV_{N_{c_i}^j},$$

$$Nucleo(A_i) \cap Nucleo(A_j) = \emptyset. \quad (3.15)$$

Pertenecen al núcleo de un conjunto difuso A , todos aquellos valores $x \in [0, 1]$ que maximizan la función μ_A , es decir, $\mu_A(x) = 1$.

$$3. \text{ Para todo } x \text{ definido en el dominio de la variable } N_{c_i}^j,$$

$$\sum_{j=1}^{|DDV_{N_{c_i}^j}|} \mu_{A_j}(x) = 1. \quad (3.16)$$

Los valores límites de los conjuntos pueden variar dependiendo de las características del entorno a vigilar, el nivel de seguridad deseado y el criterio del experto encargado de modificar los parámetros de configuración. En la Figura 3.4 se puede apreciar como los valores asignados a los conjuntos que representan las situaciones anómalas no son muy elevados, en este caso el sistema no será demasiado estricto y evitará una activación frecuente y, posiblemente, innecesaria de las alarmas. De todas formas, los valores establecidos para cada uno de los conjuntos de la partición difusa deben ir adaptándose gracias a la realimentación que existe entre la interacción del experto con el sistema y la salida del mismo mediante la activación de alarmas.

Esta forma de solucionar el mismo problema presenta algunas ventajas con respecto al método anterior:

- La función de salida deja de ser escalonada por lo que el cambio de un tipo de situación a otra es mucho más gradual.
- La capa del sistema encargada de tomar las decisiones recibe una información mucho más rica sobre la situación actual, lo cual le ofrece un mayor abanico de posibilidades para tomar las decisiones oportunas. De acuerdo a la solución anterior, la capa encargada de tomar las decisiones únicamente sabía si una situación era normal o no. Por contra, la nueva solución proporciona a esta capa diferentes grados de pertenencia a distintas clases de situaciones.
- Al igual que sucede con los valores umbrales α_{c_i} el experto también debe especificar los valores de los conjuntos difusos. Sin embargo, ya no depende de un único valor numérico por lo que el margen de error para el experto es más amplio.

La decisión de si una situación es normal o no lo es, en base a la salida de un componente de normalidad, depende de cada sistema en particular y, en concreto, de la capa encargada de tomar las decisiones. Habrá sistemas en los que una situación sospechosa sea considerada como anómala, mientras que en otros puede que no sea un motivo suficiente de alarma y, únicamente, las situaciones posiblemente anómalas y absolutamente anómalas son las que producen las activaciones de las alarmas. En la siguiente Sección se estudiará como combinar las salidas de los componentes de normalidad para obtener así un análisis global y determinar si el comportamiento de cada objeto es normal en general en un entorno particular.

3.3. Análisis global de normalidad mediante la fusión de los análisis particulares

Cuando un componente de normalidad analiza el comportamiento de un objeto genera un valor numérico de salida y se estudia la pertenencia de este valor a los conjuntos difusos definidos en la Figura 3.4, tal como se ha descrito en la Sección 3.2.9. El resultado de dicho estudio permite al sistema ser consciente de la normalidad del comportamiento del objeto con respecto a un determinado concepto. Sin embargo, la normalidad de los actos de un objeto no depende únicamente de un concepto, sino de una valoración global de todos ellos. Por tanto, es necesario establecer una serie de mecanismos que permitan la combinación de los valores de salida de los componentes para obtener un valor global que represente la normalidad del comportamiento de un objeto de manera general. En los Capítulos 4 y 5 se describirán dos componentes de normalidad en el que se aplican los métodos propuestos en este Capítulo. El primero de ellos representa y analiza si un objeto sigue trayectorias normales o permitidas en un entorno, y el segundo si los objetos circulan a una velocidad adecuada. Para que un objeto se comporte de forma normal debe seguir trayectorias normales y circular a la velocidad adecuada. En el momento que se incumpla la normalidad definida para alguno de estos componentes puede ser motivo suficiente para activar alguna de las alarmas que forma parte del sistema.

Una posible alternativa para combinar los valores de salida de los componentes de normalidad podría ser el uso de t-normas o t-conormas tal como se emplearon para el cálculo de los grados de normalidad en las Secciones 3.2.7 y 3.2.8. Si se emplearan t-normas, por ejemplo la t-norma del mínimo, el valor global de normalidad será bajo en el momento en el que no se satisfaga la normalidad de alguno de los componentes. En cambio, si se emplean t-conormas, por ejemplo la t-conorma del máximo, el valor global de normalidad será alto en el momento que se satisfaga la normalidad de alguno de los componentes. En el caso particular de los sistemas de seguridad, quizás tiene más sentido emplear el uso de t-normas para detectar cualquier anomalía. De cualquier forma, en este trabajo se propone el uso de operadores de agregación OWA que pueden comportarse tanto como los operadores t-norma como los t-conorma, y se caracterizan por su gran flexibilidad en la toma de decisiones a partir de múltiples criterios.

Los operadores de agregación OWA (media ponderada ordenada) fueron propuestos por R. Yager [207] y facilitan la toma de decisiones en base a múltiples criterios con diferentes pesos. Formalmente, un operador de agregación OWA se puede representar mediante una función $F : R^n \rightarrow R$, que tiene asociado un vector de pesos W de longitud n ; $W = [w_1, w_2, \dots, w_n]$, donde cada $w_i \in [0, 1]$ y $\sum_{i=1}^n w_i = 1$.

$$OWA(a_1, a_2, \dots, a_n) = \sum_{j=1}^n w_j \cdot b_j \quad (3.17)$$

$$OWA(a_1, a_2, \dots, a_n) = [w_1, w_2, \dots, w_n] \begin{bmatrix} b_1 \\ \cdot \\ \cdot \\ b_n \end{bmatrix}$$

tal que $(a_1, a_2, \dots, a_n)$ representa el conjunto de valores iniciales o criterios de los que dispone el operador para tomar una decisión, y $(b_1, b_2, \dots, b_n)$ es un conjunto ordenado donde b_j es el j -ésimo valor mayor del conjunto inicial $(a_1, a_2, \dots, a_n)$. Además, los valores de los pesos w_i pertenecientes al vector W están ligados a posiciones y no a valores concretos del conjunto inicial. En el modelo planteado en este Capítulo, el conjunto de criterios o valores iniciales $(a_1, a_2, \dots, a_n)$ corresponde con los valores de normalidad calculados por cada componente $(N_{c_1}^j, N_{c_2}^j, \dots, N_{c_n}^j)$.

La Figura 3.5 muestra las relaciones existentes entre los valores obtenidos por los operadores de intersección t-norma, los operadores OWA y los operadores de unión t-conorma. Tal como se puede apreciar en el gráfico, los valores obtenidos por los operadores OWA se sitúan en un lugar intermedio entre los valores obtenidos por las otras dos clases de operadores.

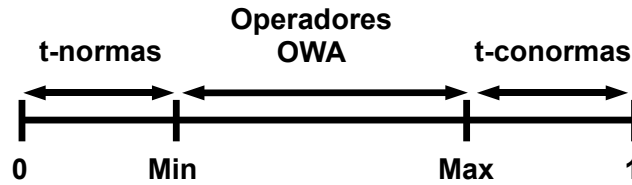


Figura 3.5: Comparación visual entre el rango de valores obtenidos por operadores t-normas, t-conormas y operadores OWA

Una de las principales características de esta familia de operadores de agregación es su flexibilidad para variar su comportamiento en base a los valores asignados al vector de pesos W . En función de la asignación de los pesos, el comportamiento del operador puede tender más a la forma de comportarse de los operadores de unión “or”, o bien, al comportamiento de los operadores de intersección “and”. De hecho es posible configurar el operador para que se comporte como la t-norma del mínimo y la t-conorma del máximo y calcular los valores de normalidad de la misma forma que en las Secciones 3.2.7 y 3.2.8. Un ejemplo del primer caso se muestra en la Tabla 3.3, donde la primera columna representa los valores de salida calculados para cada uno de los tres componentes de normalidad empleados, y que a su vez sirven como criterios para determinar el valor global de normalidad.

Criterios ($b_1, \dots, b_n$)	Valor de la restricción	Peso	Evaluación
$N_{c_1}^j(obj)$	0.95	0	0
$N_{c_2}^j(obj)$	0.8	0	0
$N_{c_3}^j(obj)$	0.2	1	0.2
		TOTAL	0.2

Tabla 3.3: Configuración de un operador OWA para que su comportamiento sea el mismo que el de la t-norma del mínimo

Por otro lado, para que el operador OWA se comporte como una t-conorma, en concreto la t-conorma del máximo es suficiente con que todo el peso se asigne al criterio de mayor valor. Un ejemplo se muestra en la Tabla 3.4.

Criterios ($b_1, \dots, b_n$)	Valor de la restricción	Peso	Evaluación
$N_{c_1}^j(obj)$	0.95	1	0.95
$N_{c_2}^j(obj)$	0.8	0	0
$N_{c_3}^j(obj)$	0.2	0	0
			0.95

Tabla 3.4: Configuración de un operador OWA para que su comportamiento sea el mismo que el de la t-conorma del máximo

Finalmente, otra de las configuraciones del vector W en los operadores OWA que se utilizan con mayor frecuencia es la de la media ponderada. La Tabla 3.5 muestra un ejemplo en el que se vuelve a calcular el mismo valor de normalidad que en la Tabla 3.3, pero en este caso se asigna el mismo peso o la misma importancia a todas las restricciones de normalidad. Tal como puede apreciarse, el valor global de normalidad que se obtiene 0.64, es un valor que dista bastante del 0.2 obtenido con el cálculo del mínimo. En este caso, los tres componentes tienen el mismo peso para determinar el valor global del comportamiento de un objeto.

Criterios ($b_1, \dots, b_n$)	Valor de la restricción	Peso	Evaluación
$N_{c_1}^j(obj)$	0.95	0.33	0.31
$N_{c_2}^j(obj)$	0.8	0.33	0.264
$N_{c_3}^j(obj)$	0.2	0.33	0.066
			0.64

Tabla 3.5: Operador OWA en el que los pesos se asignan de forma ponderada

R. Yager propone en [207] una medida interesante para analizar el comportamiento del operador en base a la distribución de los pesos. Dicha medida permite estudiar el grado de unión (o-lógico) e intersección (y-lógico). Para el primer caso, R. Yager propone la siguiente fórmula:

$$orness(W) = (1/n - 1) \sum_{i=1}^n ((n - 1) \cdot w_i) \quad (3.18)$$

3.3. Análisis global de normalidad

| 101 |

El valor de esta medida en los tres ejemplos anteriores (Tablas 3.3, 3.4 y 3.5) sería el siguiente:

- **Tabla 3.3:** $W = [0, 0, 1]$; $orness(W) = 0$. El grado de unión es cero y, por tanto, el operador se comporta como un operador de intersección puro.
- **Tabla 3.4:** $W = [1, 0, 0]$; $orness(W) = 1$. El grado de unión es uno y, por tanto, el operador se comporta como un operador de unión puro.
- **Tabla 3.5:** $W = [0, 33, 0, 33, 0, 33]$; $orness(W) = 0,5$. El grado de unión es 0.5, por lo que el operador de agregación se sitúa en un punto intermedio entre los operadores de unión e intersección.

La elección de los valores de los pesos del vector W dependerá de nivel de vigilancia que se desea alcanzar. La configuración que se muestra en la Tabla 3.3 representa el máximo nivel de exigencia, la configuración de la Tabla 3.4 el caso opuesto y, finalmente, la Tabla 3.5 un caso intermedio.

Una vez que se ha obtenido un valor global de normalidad como resultado de la combinación de los valores de salida de los componentes que han participado en el análisis del comportamiento de un objeto ($OWA(N_{c_1}^j, N_{c_2}^j, \dots, N_{c_n}^j)$), se estudia la pertenencia de este valor a los conjuntos difusos de la partición difusa que se muestra en la Figura 3.4. El valor final de normalidad $N^j(obj_k)$ asociado a un objeto obj_k en un determinado entorno E_j es la suma de los valores de pertenencia del valor de salida del operador $OWA(N_{c_1}^j, N_{c_2}^j, \dots, N_{c_n}^j)$ a los conjuntos difusos CPN: comportamiento parcialmente normal y CAN: comportamiento absolutamente normal (ver Figura 3.4):

$$N^j(obj_k) = \mu_{CPN}(OWA(N_{c_1}^j, N_{c_2}^j, \dots, N_{c_n}^j)) + \mu_{CAN}(OWA(N_{c_1}^j, N_{c_2}^j, \dots, N_{c_n}^j)) \quad (3.19)$$

donde $\mu_{CPN}(OWA(N_{c_1}^j, N_{c_2}^j, \dots, N_{c_n}^j))$ es la pertenencia del valor de salida del operador OWA al conjunto CPN y $\mu_{CAN}(OWA(N_{c_1}^j, N_{c_2}^j, \dots, N_{c_n}^j))$ la pertenencia del mismo valor al conjunto CAN . El valor global de normalidad GN^j en un entorno E_j se calcula como el mínimo de los valores de normalidad N^j calculado para cada objeto obj_k :

$$GN^j = \bigwedge_{k=1}^{|O|} N^j(obj_k) \quad (3.20)$$

donde $|O|$ representa el número de elementos del conjunto O donde se agrupan los objetos que están siendo analizados por el sistema, y $N^j(obj_k)$ el valor de normalidad calculado para cada objeto obj_k . En el momento que un

objeto no se comporte de manera normal el valor global de normalidad GN^j en el entorno E_j será un valor bajo.

Finalmente en la Figura 3.6 se muestran algunos ejemplos de asociaciones entre objetos, conceptos e instancias de conceptos. En particular, la Figura 3.6.(a) muestra una forma general de representar las asociaciones definidas en las secciones anteriores. Cuando el sistema determina la clase a la que pertenece el objeto obj_1 , comienza a analizar la normalidad de su comportamiento en base a los conceptos especificados en $O \times C$. Cada uno de los conceptos tiene asociado un conjunto de instancias particulares c_{iy} para el entorno E_j en el que se encuentra el objeto obj_1 .

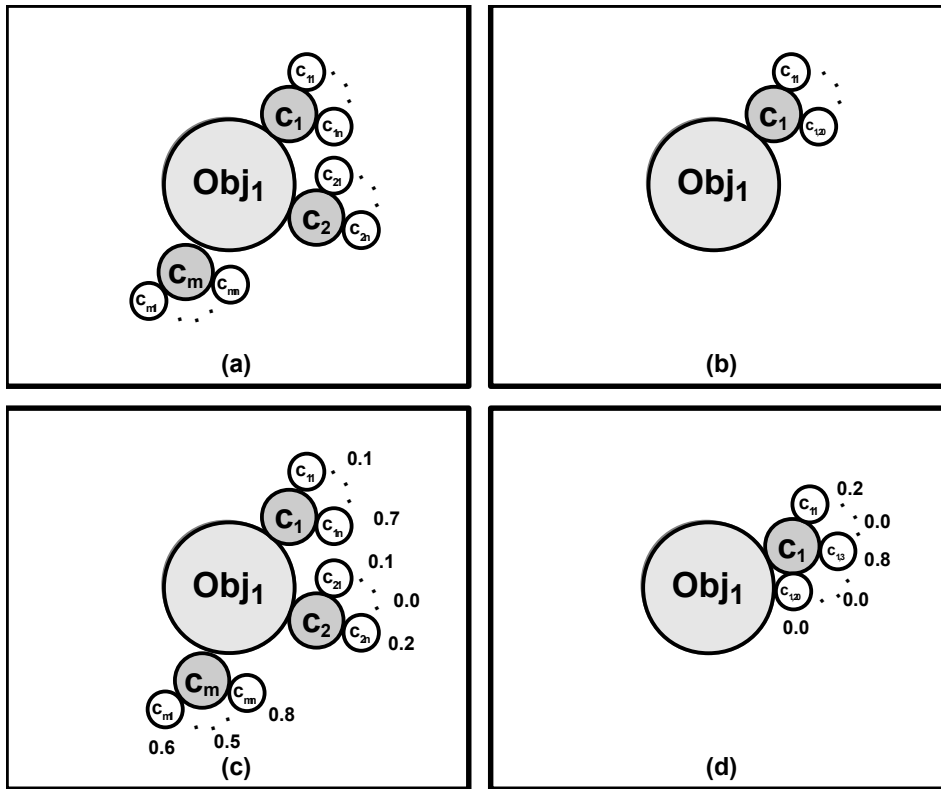


Figura 3.6: Ejemplos de asociaciones entre objetos, conceptos e instancias de conceptos.

La Figura 3.6.(b) representa el análisis del comportamiento del objeto obj_1 en base a un sólo concepto c_1 . Por ejemplo, si c_1 fuera el concepto *trayectoria normal*, cada instancia c_{1y} representaría las trayectorias normales definidas para el entorno E_j . El comportamiento del objeto obj_1 será normal de acuerdo al concepto c_1 siempre y cuando el objeto este siguiendo alguna de las trayectorias normales, es decir, existe algún $N_{c_{1y}}^j(obj)$ cuyo valor sea elevado, tal que la pertenencia a los conjuntos *comportamiento parcialmente normal* y *comportamiento absolutamente normal* es distinta de cero. Finalmente las

figuras (c) y (d) representan un comportamiento anómalo y normal, por parte del objeto obj_1 , si se tiene en cuenta una configuración del operador OWA igual a la que se muestra en la Tabla 3.3. En el caso particular de la Figura (c), el objeto no se comporta de manera normal ya que no satisface la normalidad definida para el segundo concepto, es decir, no existe una instancia c_{2y} tal que $N_{c_{2y}}(obj_1)$ posea un valor elevado. Esta situación no sucede en la Figura (d) donde el objeto obj_1 satisface la normalidad del concepto c_1 , debido a que existe una estancia c_{13} cuyo valor de normalidad es 0.8.

3.4. Adquisición del conocimiento de los componentes de normalidad

Los sistemas de vigilancia propuestos en este trabajo, los cuales están basados en el uso de componentes de normalidad, necesitan conocimiento sobre un problema o escenario concreto sobre el que se va a desplegar para poder analizar e interpretar la información que procede de los sensores. Sin este conocimiento el sistema sería incapaz de interpretar las señales y diferenciar entre lo que es normal y lo que no lo es. Por este motivo, la adquisición del conocimiento juega un papel primordial en este trabajo. El conocimiento puede ser aportado directamente por el experto mediante el uso de herramientas de adquisición de conocimiento, o bien, aprendido automáticamente mediante algoritmos de aprendizaje.

A modo de ejemplo, se mostrará más adelante algunas herramientas de adquisición de conocimiento que han sido diseñadas para instanciar el componente que analiza las trayectorias (ver Capítulo 4 y Anexo A) y un algoritmo de aprendizaje automático que ha sido empleado para instanciar el componente que analiza la normalidad en cuanto a la velocidad con la que se mueven los objetos que son captados por una cámara (ver Capítulo 5 y Anexo B). Este algoritmo será presentado a continuación ya que también ha sido usado para clasificar los objetos que se mueven por la escena. Como ya se ha indicado anteriormente, esta es una tarea fundamental dentro del modelo propuesto ya que el análisis de normalidad depende del tipo de objeto, es decir, la definición de la normalidad para cada tipo de objeto es distinta.

3.4.1. Algoritmo de aprendizaje automático inductivo

En la literatura existen multitud de algoritmos de aprendizaje supervisados y no supervisados que facilitan la adquisición del conocimiento que necesitan los sistemas expertos para resolver problemas del dominio real en el que actúan. Para el diseño de un algoritmo de aprendizaje que sea capaz de adquirir automáticamente la instancias de un concepto c_i en un escenario

E_i , se ha optado por tomar como punto de partida el método propuesto por Castro et al. en [40]⁴. Este método emplea la lógica difusa para aprender un conjunto de reglas de tipo SI-ENTONCES con un índice alto de generalidad e interpretabilidad, a partir de un conjunto de ejemplos que constituyen el conjunto de entrenamiento. Es decir, se trata de un algoritmo inductivo que parte de casos específicos para obtener conocimiento general del entorno observado. Tal elección ha sido motivada por las siguientes razones:

- La lógica difusa proporciona un marco de trabajo matemático bien definido para el tratamiento de la incertidumbre y la vaguedad, que se encuentra inherente en la información visual procedente de las cámaras de seguridad. Además, uno de los objetivos es aprender las instancias de un concepto (Sección 3.2.5), estas instancias se definen en base a una serie de restricciones, y las restricciones han sido definidas como conjuntos difusos. Por este motivo, para ser coherentes y mantener la línea trazada por el modelo descrito en este Capítulo, es necesario utilizar un algoritmo cuyas características sean similares a las que se presentan en [40].
- El conocimiento inferido se representa mediante un conjunto de reglas SI-ENTONCES dotadas de gran interpretabilidad, las cuales son representadas mediante términos lingüísticos. De esta forma, el conocimiento que forma parte del sistema de seguridad es reutilizable, modificable y fácil de compartir. Por otro lado, las reglas de esta clase no sólo tienen la capacidad de clasificar una determinada situación, sino que además justifican por sí mismas su activación. Es decir, no sólo tienen la capacidad de determinar la clase a la que pertenece un elemento, sino que también la capacidad de justificar por qué pertenece a esa clase.
- Los resultados experimentales muestran que el algoritmo consigue mejores resultados que otros algoritmos de su misma clase [40].
- El tiempo de aprendizaje y clasificación son considerablemente bajos.

Básicamente, el algoritmo presentado por Castro et al. [40] trabaja en tres fases. La primera de ellas consiste en la división de los dominios de las variables de entrada. Cada uno de los dominios de las variables de entrada se divide en varias regiones, cada una de las cuales será presentada por medio de una función trapezoidal. La elección de las regiones que conforman tal división se podrá hacer mediante alguna de las siguientes maneras: i) el experto establece el número de regiones, los límites de cada una de ellas y su función de pertenencia; ii) se calculan límites equidistantes. El experto lo único que debe hacer es establecer el número de términos que él emplea, es decir el número de particiones, pero sin fijar los límites. La salida de esta

⁴Otros algoritmos podrían ser utilizados en función de las demandas del componente que se esté diseñando, para los componentes diseñados en este trabajo se eligió como base [40] por ser el que proporcionaba mejores resultados.

fase es un conjunto de variables con sus dominios de definición divididos en conjuntos de regiones que corresponden a los posibles valores que podrán tomar cada una de las variables.

En la segunda fase, cada ejemplo del conjunto de entrenamiento se convierte en una regla inicial o básica, la cual carece de toda generalidad. De hecho, esta regla inicial únicamente especifica cómo clasificar un conjunto de ejemplos muy concreto, es decir, cómo clasificar el ejemplo o ejemplos a partir de los cuales se ha generado la regla. Por tanto, en el peor de los casos, existirán tantas reglas particulares como ejemplos haya en el conjunto de entrenamiento. En cada una de estas reglas, cada valor de cada variable de entrada será presentado por medio de una región de su dominio y la salida del ejemplo será codificada.

Finalmente, la tercera fase tiene como objetivo reducir el conjunto de reglas iniciales para construir reglas más generales. Cada una de las reglas generales abarca varias reglas particulares o iniciales. Para ello se utiliza un proceso de ampliación en el que una regla inicial se extiende cubriendo varias situaciones del espacio global de posibles situaciones, que no ha sido cubierto previamente por otra regla.

Formalmente, el algoritmo original comienza con una fase de entrenamiento a partir de un conjunto de instancias $\Theta = \{e_1, \dots, e_m\}$. Cada instancia $e_i = ((x_{i1}, \dots, x_{in}), y_j)$ es un ejemplo convenientemente descrito mediante un conjunto de variables de entrada $V = \{v_1, \dots, v_n\}$ y una variable de salida y_j , la cual es conocida como la *clase* a la que pertenece el ejemplo e_i . Los elementos $(x_{i1}, \dots, x_{in})$ son los valores concretos que el ejemplo e_i toma para cada variable $v_k \in V$. Además, existe un dominio de definición DDV_i para cada variable en el conjunto V , siendo $DDV = \{DDV_1, \dots, DDV_n\}$ el conjunto de todos los dominios. Cada DDV_i está compuesto de un conjunto de etiquetas lingüísticas $\{L_1, \dots, L_p\}$ que corresponden con los conjuntos difusos representados por medio de funciones trapezoidales tal como se especifica a continuación:

$$\Pi(u; a, b, c, d) = \begin{cases} 0 & u < a \\ \frac{(u-a)}{(b-a)} & a \leq u < b \\ 1 & b \leq u \leq c \\ \frac{(d-u)}{(d-c)} & c < u \leq d \\ 0 & u > d \end{cases} \quad (3.21)$$

donde u representa el valor de entrada de la función, y a, b, c y d los cuatro parámetros a partir de los cuales se define el trapecio, siendo 1 la altura máxima. Cada DDV_i debe verificar las siguientes propiedades:

1. $\forall L_x \in DDV_i, altura(L_x) = 1$
2. $\forall L_x, L_y \in DDV_i, nucleo(L_x) \cap nucleo(L_y) = \emptyset$. Pertenecen al núcleo de

un conjunto difuso, representado mediante una etiqueta lingüística L , aquellos valores $x \in X_i$ que maximizan la función de pertenencia μ_L , es decir, $\mu_L(x) = 1$.

3. $\forall x \in X_i, \sum_{j=1}^{|DDV_i|} \mu_{L_j}(x) = 1$, siendo X_i el dominio donde se define v_i .

Las reglas difusas del tipo SI-ENTONCES pertenecientes al conjunto definitivo de reglas presentan la siguiente forma:

$$\text{SI } v_0 \text{ es } ZD_0 \wedge \dots \wedge v_n \text{ es } ZD_n \text{ ENTONCES } y_j \quad (3.22)$$

donde $v_i \in V$ y ZD_i es un subconjunto de las etiquetas lingüísticas definidas en DDV_i para la variable v_i , es decir, $ZD_i \subseteq DDV_i$. Este tipo de reglas, tal como se verá más adelante, se empleará tanto para clasificar los objetos de una escena como para representar las instancias de un concepto. En el primer caso, las reglas representan los valores que deben tomar las variables para que un objeto pertenezca a una determinada clase, por ejemplo, peatón o vehículo. O dicho de otra forma, las características que deben cumplir para pertenecer a una clase concreta. En el segundo caso, cada una de las reglas representa una instancia de un concepto. Es decir, la regla representa la particularización de las restricciones definidas para el concepto. Si se establece una comparación entre la forma en la que se representa cada una de las reglas aprendidas, y la forma en la que se representa y calcula el valor de normalidad de una instancia de un concepto, es posible encontrar ciertas similitudes:

$$\begin{array}{ccccccc} SI & v_0 \text{ es } ZD_0 & \wedge & \dots & \wedge & v_n \text{ es } ZD_n & ENTONCES y_j \\ c_{iw}^j & \tilde{\mu}_{1i}^1 & \wedge & \dots & \wedge & \tilde{\mu}_{ni}^1 & N_{c_{iw}}^j(obj) \end{array}$$

Ejemplo. A continuación se presenta un ejemplo de cómo este tipo de reglas puede representar una instancia de un concepto, y cómo a partir de ésta se puede calcular el valor de normalidad. Sea v_1 y v_2 dos variables empleadas para la construcción de reglas difusas y, DDV_{v_1} y DDV_{v_2} sus dominios de definición, ambos mostrados en la Figura 3.7.

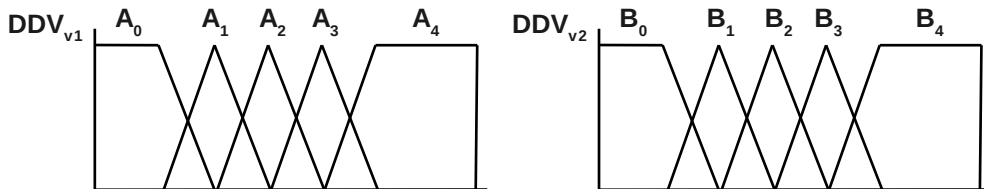


Figura 3.7: Dominios de definición para las variables v_1 y v_2 .

Sea R_1 la siguiente regla: SI v_1 es $\{A_0, A_3, A_4\} \wedge v_2$ es $\{B_2, B_3\}$ ENTONCES y_1 , siendo y_1 el concepto del que se pretende aprender el conjunto de instancias. En este caso, $ZD_1 = \{A_0, A_3, A_4\}$ y $ZD_2 = \{B_2, B_3\}$, ambos construidos mediante la unión de conjuntos difusos y representados gráficamente en la Figura 3.8.

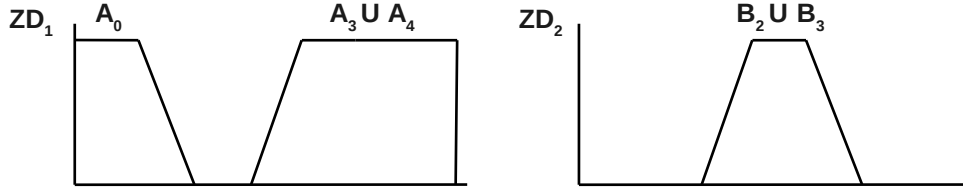


Figura 3.8: Representación gráfica de los subconjuntos ZD_1 y ZD_2 incluidos en la regla difusa R_1 .

Si se tiene en cuenta que una restricción se representa mediante un conjunto difuso con una función de pertenencia asociada, la cual determina el grado con el que se satisface la restricción, v_1 es ZD_1 y v_2 es ZD_2 representan dos restricciones cuyas funciones de pertenencia son $\mu_{A_0 \cup (A_3 \cup A_4)}$ y $\mu_{B_2 \cup B_3}$, respectivamente. El grado de normalidad se calcula como el valor mínimo de los grados de satisfacción de las restricciones $\min(\mu_{A_0 \cup (A_3 \cup A_4)}, \mu_{B_2 \cup B_3})$. Si alguna de las reglas se activa con un valor alto de satisfacción, el objeto satisface la normalidad de una de las instancias del concepto y, por tanto, se comporta de forma normal de acuerdo al concepto. Las siguientes líneas tratan de mostrar tal equivalencia:

1. SI v_1 es $ZD_1 \wedge v_2$ es ZD_2 ENTONCES y_1
2. SI v_1 es $\{A_0, A_3, A_4\} \wedge v_2$ es $\{B_2, B_3\}$ ENTONCES y_1
3. $(\{A_0, A_3, A_4\}, \{B_2, B_3\}, y_1)$
4. $\mu_{A_0 \cup (A_3 \cup A_4)} \wedge \mu_{B_2 \cup B_3}$
5. $\min(\mu_{A_0 \cup (A_3 \cup A_4)}, \mu_{B_2 \cup B_3}) = N_{c_{iw}}^j(obj)$

Por otro lado, aunque la descripción detallada de los pasos del algoritmo original se incluye en [40], se ha creído conveniente incluirlos en esta Sección, en particular, en el Algoritmo 1.

Para una comprensión adecuada del Algoritmo 1 es conveniente recordar los conceptos *subsume* y *amplificación*. Una regla R_i ($(ZD_{i0}, ZD_{i1}, \dots, ZD_{in}) y_p$) subsume en otra regla R_k ($(ZD_{k0}, ZD_{k1}, \dots, ZD_{kn}) y_q$) si $ZD_{i0} \subseteq ZD_{k0}$ y $ZD_{i1} \subseteq ZD_{k1}$ y ... y $ZD_{in} \subseteq ZD_{kn}$ y $y_p = y_q$. Por ejemplo, la regla inicial ($\{VU, CH, ME, SH, HI, LA\} y_1$) subsume en la siguiente regla: SI X_5 es $\{HI, VH\}$ ENTONCES Y es y_1 . Es importante recalcar que cuando una variable de entrada no aparece en una regla final es debido a que ésta puede tomar cualquier valor de su dominio de definición y, por esta razón, se omite.

Por otro lado, el proceso de amplificación consiste en la incorporación de nuevas etiquetas lingüísticas en una variable determinada. Este proceso comienza a partir de una regla inicial $((ZD_0, ZD_1, \dots, ZD_n) y_p)$ y, posteriormente, continúa intentado amplificar cada una de las variables, es decir, el algoritmo intenta amplificar cada ZD_i , $i = 1, \dots, n$ con las etiquetas lingüísticas que todavía no han sido consideradas. Una amplificación es posible siempre y cuando no exista un contraejemplo en otra clase que entre en conflicto con la regla amplificada.

Algorithm 1 Algoritmo de aprendizaje propuesto por Castro et al. [40]

Input: Conjunto de variables de entrada, dominios de definición y conjunto de entrenamiento Θ .

Output: Conjunto de reglas difusas generadas.

1.- Convertir cada ejemplo del conjunto de entrenamiento en una regla inicial (fuzzyficación del ejemplo). Cada elemento $e_i \in \Theta$ es transformado en una regla difusa inicial, donde cada valor de cada variable es representado mediante una etiqueta lingüística. Este paso crea el conjunto de reglas iniciales y un conjunto vacío de reglas generales o definitivas.

2.- Tomar una regla de el conjunto de reglas iniciales. Sea R_i esta regla, R_i : SI v_0 es $ZD_{i0} \wedge \dots \wedge v_n$ es ZD_{in} ENTONCES y_p .

3.- Comprobar si la regla R_i subsume en alguna de las reglas pertenecientes al conjunto de reglas definitivas. Si esto sucede se ignora la regla R_i y se vuelve al paso número 2.

4.- Si R_i no subsume en alguna de las reglas del conjunto definitivo de reglas, intentar amplificarla. Para cada variable v_k :

(4.a) Para cada etiqueta L_x no considerada:

(4.a.i) Intentar amplificar la regla $ZD_{i'k} = ZD_{ik} \cup L_x$. si no es posible ir al paso 4.a. Una regla puede ser amplificada si no existe otra regla R_j : SI v_0 es $ZD_{j0} \wedge \dots \wedge v_n$ es ZD_{jn} ENTONCES y_q , en el conjunto de reglas iniciales, tal que $\forall k \in [0, |V|]$, se cumple que $ZD_{jk} \subseteq ZD_{i'k}$ y $y_p \neq y_q$

(4.a.ii) Amplificar la regla

5.- Si la regla ha sido amplificada, entonces incluirla en el conjunto de reglas definitivas.

6.- Si existen aún reglas sin considerar en el conjunto inicial ir al paso 2, en caso contrario FIN.

A pesar de las ventajas mencionadas anteriormente, el algoritmo general presenta algunos problemas cuando se aplica en el contexto de la vídeo vigilancia. Estos problemas tienen su origen en la producción de reglas sobre-generalizadas. Con sobre-generalización nos referimos a la tendencia que tienen las reglas definitivas a cubrir espacios para los que no existen contraejemplos.

Ejemplo. Supongamos las siguientes reglas pertenecientes al conjunto de reglas iniciales, para la clasificación de objetos a partir de su tamaño y la distancia a la que se encuentran con respecto a la cámara (ver Figura 3.9).

- R_i : ((SMALL, MEDIUM), persona)
- R_j : ((MEDSMALL, MEDIUM), motocicleta)
- R_k : ((MEDBIG, MEDIUM), coche)

estas mismas reglas se pueden representar de la siguiente manera:

- R_i : SI tamaño es {SMALL} $\wedge$ distancia es {MEDIUM} ENTONCES persona
- R_j : SI tamaño es {MEDSMALL} $\wedge$ distancia es {MEDIUM} ENTONCES motocicleta
- R_k : SI tamaño es {MEDBIG} $\wedge$ distancia es {MEDIUM} ENTONCES coche

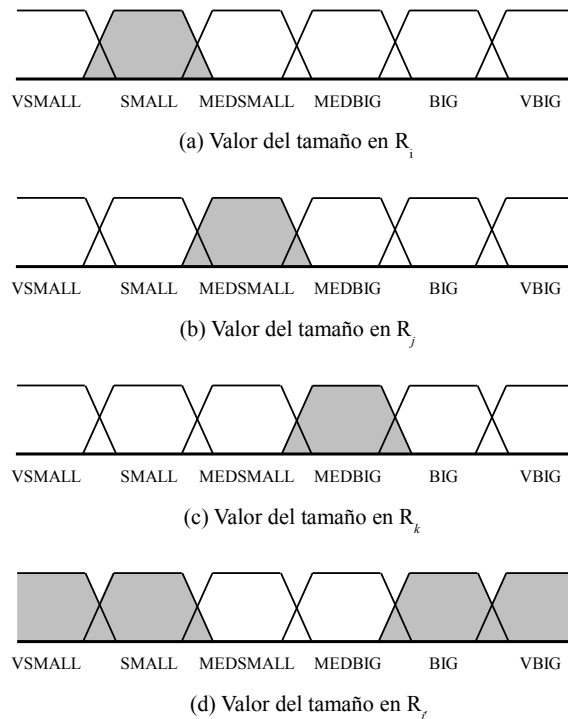


Figura 3.9: Representación gráfica de los valores que toma la variable tamaño para cada una de las reglas R_i , R_j , R_k y $R_{i'}$. La variable distancia con respecto a la cámara no se ha incluido en la imagen debido a que su valor no varía en ninguna de las reglas anteriores.

Si el dominio de definición de la variable tamaño es $DDV_{tamaño} = \{VSMALL, SMALL, MEDSMALL, MEDBIG, BIG, VBIG\}$, una posible ampliación de la regla sencilla R_i a una regla general $R_{i'}$ podría ser la siguiente:

- $R_{i'}$: SI tamaño es $\{VSMALL, SMALL, BIG, VBIG\} \wedge$ distancia es $\{MEDIUM\}$ ENTONCES persona

La regla anterior $R_{i'}$ significa que si un objeto se encuentra a una distancia media con respecto de la cámara y tiene un tamaño muy pequeño (VSMALL), pequeño (SMALL), grande (BIG) o muy grande (VBIG), entonces pertenece a la clase persona. La sobre-generalización se produce por la inclusión de las etiquetas BIG y VBIG en la variable *tamaño*, debido a que no existe ningún ejemplo en las clases *motocicleta* y *coche* que se encuentren a una distancia media y su tamaño sea grande o muy grande. Obviamente, la regla $R_{i'}$ carece de sentido, ya que las personas situadas a una distancia media se caracterizan por tener un tamaño muy pequeño o pequeño y no ser más grandes que los propios vehículos a esa distancia. Para solucionar este problema se ha modificado el algoritmo original [40] restringiendo el proceso de ampliación.

Algorithm 2 Modificaciones sobre el algoritmo propuesto por Castro et al. [40]

Input: Conjunto de variables de entrada, dominios de definición y conjunto de entrenamiento Θ .

Output: Conjunto de reglas difusas generadas.

(4.a) Para cada etiqueta L_x no considerada:

(4.a.i) Intentar ampliar la regla $ZD_{i'k} = ZD_{ik} \cup L_x$. si no es posible ir al paso 4.a. Una regla puede ser ampliada si :

A) No existe otra regla R_j : SI v_0 es $ZD_{j0} \wedge \dots \wedge v_n$ es ZD_{jn} ENTONCES y_q , en el conjunto de reglas iniciales, tal que $\forall k \in [0, |V|]$, se cumple que $ZD_{jk} \subseteq ZD_{i'k}$ y $y_p \neq y_q$

B) Si el área existente entre la última etiqueta añadida a la regla y la etiqueta actual L_x que está siendo considerada en el proceso de ampliación, no excede el valor máximo de separabilidad permitido.

(4.a.ii) Amplificar la regla

Para resolver el problema de sobre-generalización se impone como limitación que se amplie una variable con una nueva etiqueta de su dominio de definición si la última añadida no se encuentra a una distancia excesiva. Para ello es necesario definir una medida de separabilidad entre etiquetas lingüísticas o, lo que es lo mismo, distancia entre conjuntos difusos. La medida de separabilidad entre dos conjuntos difusos L_i y L_j , tal que $L_i < L_j$, se define como el área existente entre ambos conjuntos difusos, $Entre(L_i, L_j)$ (ver Figura 3.10.c).

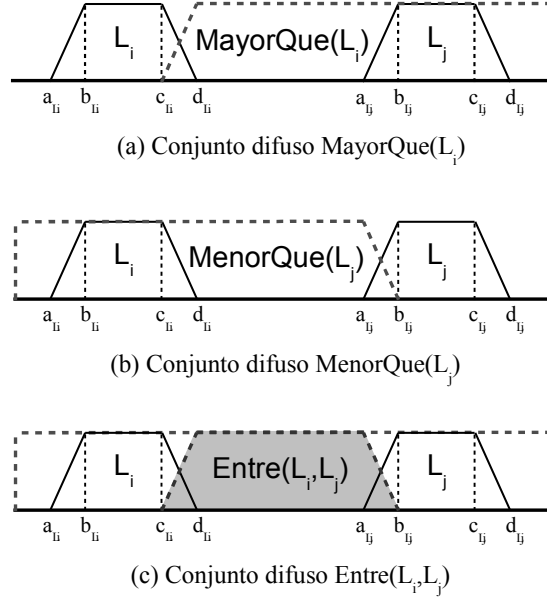


Figura 3.10: Relaciones existentes entre dos conjuntos difusos L_i, L_j para la estimación de la distancia a la que se encuentran.

La relación $Entre(L_i, L_j)$ es un conjunto difuso que se calcula mediante la intersección entre $MayorQue(L_i)$ (Figura 3.10.a) y $MenorQue(L_j)$ (Figura 3.10.b). La expresión 3.23 muestra como se calcula la distancia entre dos conjuntos L_i y L_j :

$$separabilidad(L_i, L_j) = \frac{(b_{L_j} - c_{L_i}) + (a_{L_j} - d_{L_i})}{2} \quad (3.23)$$

siendo $(a_{L_i}, b_{L_i}, c_{L_i}, d_{L_i})$ y $(a_{L_j}, b_{L_j}, c_{L_j}, d_{L_j})$ los trapezoides que definen los conjuntos difusos L_i y L_j , respectivamente, los cuales verifican que $d_{L_i} < a_{L_j}$. Si $d_{L_i} = a_{L_j}$ or $d_{L_i} > a_{L_j}$, entonces $separabilidad(L_i, L_j) = 0$. Actualmente, el valor umbral de separabilidad se calcula de forma empírica, observando los resultados obtenidos por el algoritmo para diferentes valores umbrales. El valor umbral elegido será aquel que produzca mejores resultados en el proceso de clasificación.

Una segunda modificación sobre el algoritmo de aprendizaje original está relacionada con el orden en el que se amplían las variables de una regla. En el algoritmo original se amplifican las variables en el orden en el que se han definido, sin seguir ningún criterio especial. De este modo, las variables que se amplifican en primer lugar pueden impedir la ampliación de variables posteriores con ciertas etiquetas, las cuales podrían discriminar mejor a los elementos de una clase. Por este motivo se ha optado por calcular el valor de varianza para cada una de las variables, siendo las que tienen mayor valor

de varianza las que se amplifican primero. La varianza es una medida de dispersión que se calcula de la siguiente forma:

$$s^2(v) = \frac{1}{n} \sum_{i=1}^n (v_i - \bar{v})^2 \quad (3.24)$$

siendo v_i cada uno de los valores que toma la variable v , $\bar{v}$ el valor medio de los valores de v , y n el número de instancias de la variable v . A la hora de clasificar los objetos de una clase es interesante encontrar alguna característica común que cumplan la mayoría de los elementos de esa clase. Las variables con mayor valor de varianza tienen un grado de dispersión mayor debido a que toman un amplio rango de valores, por lo que no son buenos discriminadores en la clasificación de objetos. Por este motivo, en la nueva versión del algoritmo de aprendizaje se amplifican en primer lugar aquellas variables con mayor valor de dispersión. Al amplificarse en primer lugar, existe una mayor probabilidad de que la variable se amplíe con todas las etiquetas definidas en el dominio de definición de la variable. Cuando esto sucede, la variable no se incluye en la regla. De esta forma, sólo se incluyen en las reglas finales aquellas variables y valores que discriminan mejor a los elementos de una clase.

3.5. Arquitectura multi-agente con soporte al modelo de análisis de la normalidad

Para concluir este Capítulo se describirá brevemente el trabajo llevado a cabo en una línea de investigación paralela en la que se ha diseñado una arquitectura multi-agente con el objetivo de ofrecer soporte y cubrir las necesidades planteadas en el modelo descrito anteriormente. Este tipo de arquitecturas basadas en agentes inteligentes distribuidos, facilitan el despliegue de los sistemas avanzados de vigilancia.

La arquitectura diseñada permite el despliegue de entidades autónomas que gestionan las tareas de vigilancia de entornos en función de los componentes de análisis de normalidad utilizados para llevar a cabo la monitorización. Así mismo, esta arquitectura ha sido concebida para tratar con el problema de la vigilancia de una manera sencilla teniendo en cuenta la división del entorno monitorizado en sub-entornos, como especifica el modelo de normalidad adoptado. Para ello, la arquitectura también proporciona los mecanismos de comunicación requeridos para integrar la información generada mediante el análisis de distintos componentes y gestiona los mecanismos de interacción con los dispositivos de vigilancia.

En este contexto, la arquitectura está inspirada en los Sistemas Multi-Agente [199], estableciendo un *framework* donde estas entidades autónomas

o agentes conviven y cooperan llevando a cabo las tareas para las cuales fueron diseñados. El uso de este tipo de sistemas facilita la delegación de tareas, el reparto de responsabilidades, la colaboración y la gestión del conocimiento distribuido; aspectos directamente relacionados con las necesidades del modelo de normalidad adoptado.

Actualmente existen propuestas basadas en el uso de la tecnología de agentes para solucionar problemas particulares en entornos concretos, como las discutidas en [12], [150], [43] o [145]. Sin embargo, estas aproximaciones no son lo suficientemente escalables y flexibles para soportar el modelo general de conocimiento, el cual establece las bases para un sistema de vigilancia independiente de dominio y basado en el análisis de normalidad. Por esta razón se propone una arquitectura multi-agente basada en dicho modelo, en la que los agentes que la componen son los responsables de tratar con los requisitos establecidos por el mismo. Las principales ventajas de la arquitectura propuesta son las siguientes:

- **Generalidad**, debido a que permite el despliegue de agentes que gestionan el conocimiento derivado del diseño de los componentes de vigilancia.
- **Escalabilidad**, debido a que es posible añadir nuevos agentes al sistema para tratar con una tarea de vigilancia que no se consideró inicialmente, como el preprocesamiento de información recogida por un nuevo sensor o la necesidad de incluir un nuevo componente de análisis en el entorno.
- **Diseño multi-capa**, que permite estructurar la funcionalidad del sistema en distintas capas.
- **Enfoque basado en agentes**, que facilita el diseño de sistemas distribuidos complejos donde las distintas entidades autónomas cooperan para alcanzar un objetivo común. En el caso referente a este trabajo, este objetivo es la vigilancia de entornos complejos.
- **Control distribuido**, que hace posible el desarrollo de soluciones robustas y adaptables en lugar de utilizar enfoques centralizados.

En la Figura 3.11 se muestra un esquema de la arquitectura propuesta para un sistema de vigilancia inteligente, el cual guarda relación con el esquema que se muestra en la Figura 3.1. Dicha arquitectura está estructurada en tres niveles: nivel reactivo, nivel deliberativo y nivel de usuario. Los distintos componentes de cada nivel se comunican mediante un moderno middleware llamado ZeroC ICE [11], que permite la interacción de los mismos y la integración de nuevos elementos. A continuación se resumen las principales características de cada nivel:

- El **nivel reactivo** está compuesto por los dispositivos de vigilancia

utilizados para monitorizar los distintos sub-escenarios del entorno global y por aquellos agentes que mantienen un esquema reactivo.

- El **nivel deliberativo** está compuesto por aquellos agentes que mantienen y utilizan un modelo de conocimiento y que persigue el análisis de normalidad y anormalidad del entorno desde distintos puntos de vista (componentes). Así mismo, en este nivel residen aquellos agentes que llevan a cabo la fusión de la información obtenida de las distintas percepciones (sub-entornos) y la toma de decisiones.
- El **nivel de usuario** permite la interacción del personal en seguridad a través de herramientas de monitorización.

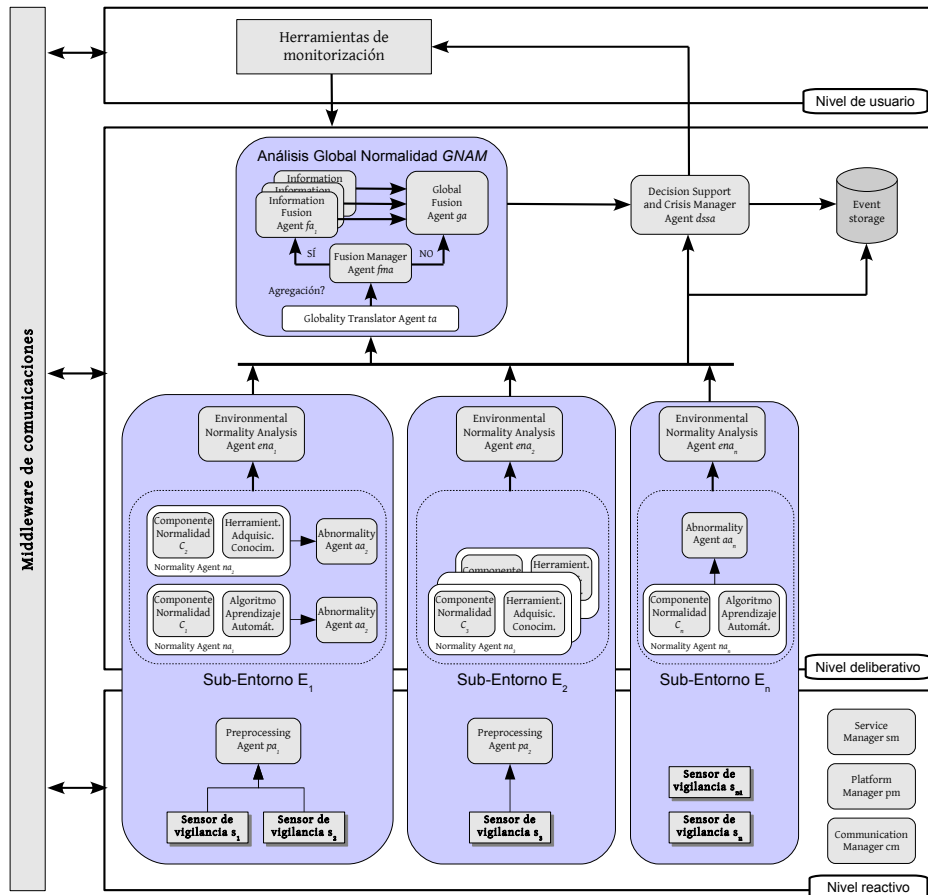


Figura 3.11: Propuesta de arquitectura para sistemas de vigilancia inteligente.

Además de los agentes propuestos para solucionar el problema de la vigilancia en un entorno desde un punto de vista global, resulta interesante definir una serie de agentes de gestión que faciliten el control del sistema multi-agente y proporcionen **servicios básicos**, como por ejemplo un servicio

de directorio o páginas amarillas o un servicio de comunicación entre agentes. Ante esta necesidad, se proponen una serie de agentes de gestión de acuerdo al conjunto de estándares definidos por el comité FIPA [80] para el desarrollo de sistemas multi-agente. Dichos agentes son el *Agent Management System (AMS)*, el cual gestiona los eventos que se producen en el sistema multi-agente y controla el estado de cada uno de los agentes, ofreciendo un servicio de páginas blancas; el *Directory Facilitator (DF)*, responsable de proporcionar un servicio de páginas amarillas; y el *Message Transport Service (MTS)*, el cual gestiona el envío y recepción de los mensajes de los agentes.

A continuación se describe cómo fluye la información por los distintos elementos que componen la arquitectura del sistema (ver Figura 3.12). El objetivo de esta descripción informal es dar una visión general de cómo la información obtenida del entorno por los sensores se va transformando hasta proporcionar conocimiento de alto nivel, como el análisis de comportamientos o la posible toma de decisiones.

Información obtenida del entorno. Los sensores de vigilancia son los encargados de obtener dicha información, la cual está vinculada a la especialización del sensor (audio, vídeo, presencia, etc). En el caso de la videovigilancia, esta información consiste en la secuencia de imágenes estáticas obtenidas por las cámaras de seguridad. La integración de los dispositivos de vigilancia se realizará mediante el *middleware* de comunicaciones que da soporte a la arquitectura del sistema de vigilancia.

Notificación de eventos de vigilancia. La salida generada por los sensores de vigilancia puede requerir un procesamiento para extraer la información relevante que se utilizará para realizar el análisis de la normalidad del entorno. Si éste es el caso, los *preprocessing agents* procesarán esta información antes de enviarla al nivel deliberativo, como ocurre con la información obtenida de los sensores de vídeo. Sin embargo, también es posible hacer uso de sensores más sencillos cuya información será notificada directamente al nivel deliberativo, como es el caso de los sensores de presencia, los cuales generan una salida *booleana*. En ambos casos, la información se encapsula en eventos de vigilancia que se enviarán al nivel deliberativo mediante canales de eventos.

Generación de conocimiento para analizar el entorno. Los eventos de vigilancia obtenidos del nivel reactivo son procesados por los *normality agents*, agentes responsables de analizar la normalidad del entorno monitorizado y capaces de detectar la violación de la normalidad. Todo este conocimiento se comparte gracias a una pizarra en la que los agentes de normalidad van anotando lo que ocurre en el entorno monitorizado. Así mismo, este conocimiento se notifica a los *abnormality agents*, los cuales llevan a cabo la detección de las situaciones anómalas más comunes en los sub-escenarios en los que estén desplegados, y al personal de seguridad a través de las herramientas de monitorización. El conocimiento anotado en la pizarra por parte de los

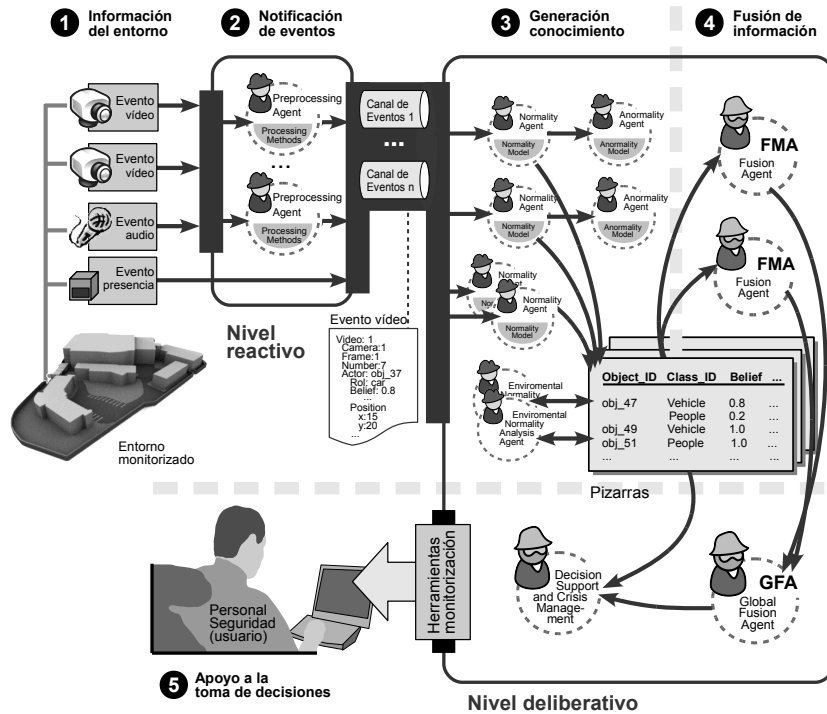


Figura 3.12: Visión general del flujo de información entre los principales componentes de la arquitectura de vigilancia.

normality agents de un escenario es usado por el *environmental normality analysis agent* de dicho escenario para realizar un análisis de normalidad completo a nivel de escenario o sub-entorno, es decir, contemplando la normalidad de todos los aspectos monitorizados en dicho escenario.

Fusión de información. El conocimiento explícito de la pizarra es utilizado por los distintos *fusion information agents*, cada uno de ellos desplegado por cada aspecto a vigilar en el entorno global, para efectuar un análisis global de normalidad a partir de los resultados parciales obtenidos por los *normality agents*. Con el objetivo de establecer relaciones entre la información obtenida a nivel de escenarios y de entorno global, el *global translator agent* será el encargado de este proceso, ofreciendo una visión homogénea de la información del entorno a los *fusion information agents*. Finalmente, el *global fusion agent* será el responsable de agregar el conocimiento de los distintos agentes de fusión, con el objetivo de proporcionar una vigilancia más sofisticada que relacione distintos conceptos de vigilancia entre sí.

Apoyo a la toma de decisiones. Tanto el conocimiento anotado en la pizarra como el generado por los agentes de fusión es utilizado por el *decision support and crisis manager agent* para ofrecer apoyo al personal de seguridad en lo

3.5. Arquitectura multi-agente

| 117 |

que a toma de decisiones y gestión de crisis se refiere. La interacción con los mismos se efectúa mediante herramientas de monitorización.

Una descripción más detallada sobre la arquitectura y plataforma multi-agente propuesta para ofrecer soporte al modelo de normalidad, tipos de agentes inteligentes y mecanismos de comunicación empleados, se puede encontrar en [187, 188, 189, 190].



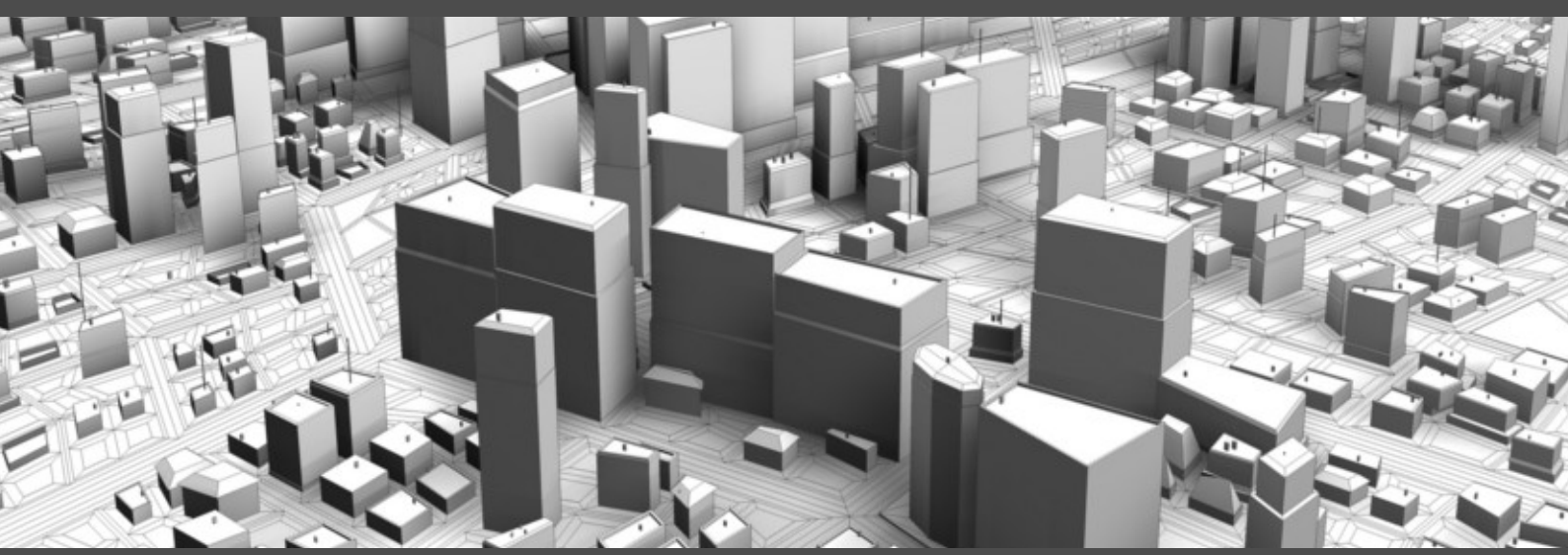
Capítulo

4

Componente de Normalidad
para el Análisis de Trayectorias

4

Capítulo



4

Componente de Normalidad para el Análisis de Trayectorias

En este Capítulo se presenta un componente de normalidad para el análisis de trayectorias, que se ha diseñado siguiendo la línea marcada por el modelo presentado previamente. Para comenzar, se discute la importancia y principal problemática del análisis de trayectorias en los sistemas de vídeo vigilancia a partir de información visual procedente de las cámaras. A continuación se describen las variables empleadas para la definición y el análisis de trayectorias, así como algunos métodos para la obtención de sus valores (preprocesamiento de señales). Entre estos métodos se incluye el algoritmo de aprendizaje presentado en el Capítulo anterior para la generación de un conjunto de reglas difusas que permitan la clasificación de los objetos. Finalmente se definen las restricciones necesarias de manera general y un conjunto de trayectorias particulares para un entorno real concreto.



Figura 4.1: Descripción, con un alto nivel de abstracción, de las entradas y salidas del componente de normalidad encargado de analizar las trayectorias en un entorno vigilando por un conjunto de cámaras.

4.1. Concepto de trayectoria normal

El estudio de las trayectorias que siguen los elementos móviles en un entorno observado es muy frecuente [133], sobre todo en entornos donde se dispone de información visual obtenida a partir de un conjunto de cámaras. La mayoría de entes con capacidad de acción se mueven con el fin de cumplir un objetivo y para alcanzarlo es necesario realizar diversos recorridos [61, 104]. En cada escenario o entorno real existen una serie de recorridos predefinidos que son considerados como normales o adecuados para cada tipo de agente, los cuales suelen coincidir con los recorridos más frecuentes pero no tiene por qué ser siempre así. Cualquier desviación en estos recorridos puede implicar ciertas anomalías y ser motivo de alerta, por ejemplo, personas intentando acceder a edificios a través de ventanas, merodeos, vehículos que invaden zonas peatonales, etc. Por esta razón cobra especial sentido estudiar el camino seguido por los agentes que interactúan en el entorno observado.

Normalmente, para la detección de trayectorias se suele emplear la información visual que procede de las cámaras de vigilancia aunque no es la única forma, por ejemplo, se podría utilizar sensores de presencia instalados en el recinto vigilado. Sin embargo, este tipo de sensores, aunque efectivos en su principal función, no facilitan la obtención de información acerca de las características físicas del objeto detectado, tal que dificultan la clasificación y el seguimiento del mismo a lo largo del tiempo. Si se supiera con absoluta certeza que sólo existe un objeto en el entorno, la secuencia de activaciones de los sensores de presencia ofrecería una idea clara sobre el camino que está siguiendo el objeto. Sin embargo, si existiera una gran cantidad de objetos en movimiento en el entorno, sería complicado conocer la ubicación de cada uno de ellos sin el uso de marcas u otro tipo de identificadores. Por estos motivos, las cámaras de vigilancia son los tipos de sensores más apropiados para el análisis de trayectorias.

La información visual procedente de las cámaras es procesada por algoritmos de segmentación, que diferencian los elementos móviles de los elementos estáticos de la escena, y por algoritmos de *tracking* que siguen a los objetos en movimiento manteniendo una referencia única por cada uno de ellos. Es importante recalcar que no existen sistemas de visión artificial perfectos y que la mayoría de ellos cometen algunos de los siguientes errores:

- Objetos en movimiento que no son detectados. Esta situación puede suceder si el objeto es de pequeñas dimensiones, el movimiento es leve, está parcialmente oculto, o bien, existen condiciones de iluminación o meteorológicas adversas.
- El sistema pierde la referencia de un objeto. Es decir, el sistema cree que el objeto que comenzó a seguir cuando éste apareció en la escena, y ahora se encuentra en una zona intermedia, es un objeto nuevo. Este

error se puede producir debido a oclusiones absolutas o parciales, entre otras causas.

- Objeto de grandes dimensiones en movimiento detectado por el sistema como varios objetos de menor tamaño. Por ejemplo, es frecuente observar como algoritmos de segmentación marcan vehículos de grandes dimensiones con varias elipses en lugar de una sola.
- Varios objetos independientes detectados como un único objeto. Este hecho sucede con frecuencia con grupos de personas, peatones muy próximos entre sí, o bien, peatones muy próximos a vehículos.
- Estimación errónea del lugar o zona en la que se encuentra un objeto.
- Clasificación incorrecta de un objeto.

Cuando alguno de estos errores se produce en los niveles inferiores, la parte del sistema encargada de analizar e interpretar lo que sucede en el entorno se enfrenta a serias dificultades. Si se estableciera un símil con agentes de vigilancia reales, es como si éstos no pudieran ver u oír correctamente y, por tanto, existiría una probabilidad alta de interpretar erróneamente lo que sucede alrededor. Este tipo de errores, dado su origen, podría ser tratado en las capas inferiores para evitar su propagación al resto de capas del sistema, sin embargo, esto no siempre es posible. Por este motivo, el diseño de las capas intermedias debe realizarse teniendo en cuenta este tipo de adversidades. Cuanto mayor sea la tolerancia de los métodos de análisis a los errores procedentes de las capas inferiores, mayor será la robustez del sistema [63]. En el Capítulo 6, una vez evaluado el modelo y los componentes de normalidad a partir de los resultados obtenidos, se discutirá la tolerancia y robustez de los métodos propuestos ante errores procedentes de las capas inferiores.

La mayoría de autores suele definir las trayectorias otorgando toda la importancia a la información espacial, es decir, los puntos o secuencia de zonas que recorre un objeto [133]. En sistemas de visión donde el objetivo principal es la identificación de las trayectorias que realizan los objetos, este enfoque puede ser suficiente. Sin embargo, la representación de trayectorias únicamente en base a información espacial puede ser insuficiente para los sistemas de vigilancia donde, además de determinar el recorrido de cada objeto, es necesario valorar otra serie de aspectos relevantes para la detección de anomalías como, por ejemplo, el tiempo o la clase a la que pertenece cada objeto. Dicho de otro modo, existen trayectorias que son únicamente normales siempre y cuando sean realizadas por una clase concreta de objetos y en un intervalo de tiempo específico. Una trayectoria es una secuencia de puntos o lugares por los que se mueve un objeto a lo largo del tiempo, y se pueden representar como una secuencia de estados y transiciones entre estados, donde cada uno de ellos corresponde con una región o punto del entorno.

En particular, en este trabajo, una trayectoria se define mediante un conjunto de restricciones de rol, espaciales y temporales, que puede ser ampliado de acuerdo a las necesidades de cada entorno a monitorizar. La definición de conceptos y su normalidad en base a un conjunto de restricciones de la forma en la que se planteó en el Capítulo anterior, ofrece una forma flexible y escalable de enriquecer las definiciones de los conceptos.

El primer tipo de restricciones especifica la lista de clases de objetos que pueden realizar una trayectoria. El segundo tipo determina la secuencia de zonas o áreas que el objeto puede recorrer y, finalmente, las restricciones temporales comprueban si la trayectoria se está llevando a cabo en el intervalo de tiempo adecuado y con la duración apropiada. En este trabajo se ha optado por realizar la definición de trayectorias mediante secuencias de áreas ya que, tal como se puede apreciar en la Figura 4.2.a es realmente complicado que puedan existir dos recorridos exactamente iguales ¹. De esta forma, una misma secuencia de áreas, representadas mediante un conjunto de poligonos en una imagen 2D, puede representar una trayectoria que engloba múltiples recorridos similares cuyos puntos están próximos entre sí (ver Figura 4.2.c).

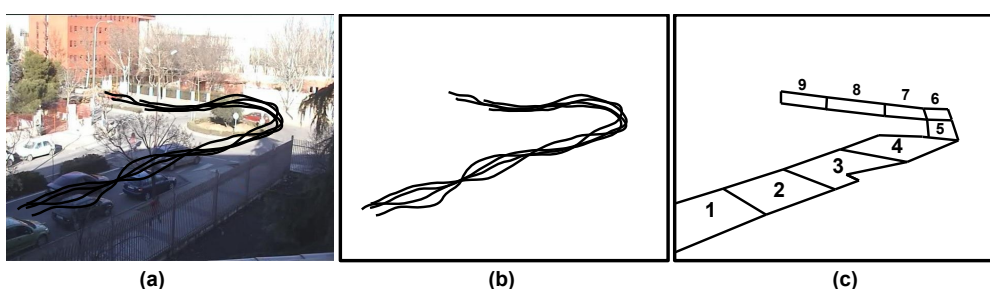


Figura 4.2: (a) Grupo de trayectorias realizadas por vehículos y marcadas mediante líneas continuas sobre la escena. (b) Distinción del grupo de trayectorias sin el uso de la imagen de la escena de fondo. (c) Ejemplo de secuencia de zonas que engloban al grupo de trayectorias mostrado en la Figura (b).

La definición de trayectorias mediante la forma descrita anteriormente implica una serie de necesidades que se detallan a continuación:

- El sistema debe conocer el conjunto de zonas relevantes que existe en el entorno observado.
- El sistema debe conocer en cada momento la zona/as en las que se podría encontrar situado cada objeto.

¹Otros autores optan por utilizar las trayectorias que han realizado los objetos durante un periodo de aprendizaje, como trayectorias modelo. Este tipo de enfoque suele ir acompañado de medidas de distancia que permiten calcular las similitudes entre los puntos pertenecientes a las trayectorias modelo y los puntos de la trayectoria actual del objeto que está siendo observado [133].

- Secuencia de áreas recorridas por el objeto y ordenadas de forma temporal, desde que éste aparece por primera vez en la escena y es captado por las cámaras de vigilancia.
- Clasificación de los objetos a partir de las características físicas, tales como la altura, anchura, desplazamiento en el eje horizontal y vertical, etc.
- Referencia o identificador asignado a cada objeto en el proceso de *tracking*.
- Referencia temporal al momento actual.
- Tiempo empleado para cada una de las posibles trayectorias que podría estar siguiendo el objeto.

Estas necesidades justifican la elección de variables para la representación de trayectorias, presentada en la siguiente Sección.

4.2. Variables para la definición y el análisis de trayectorias

Cuando se pretende diseñar e incluir en el sistema de vigilancia un nuevo componente para analizar un nuevo aspecto de seguridad, tal como las trayectorias que siguen los objetos, es necesario realizar un estudio similar al realizado en la sección anterior. Este estudio proporcionará la información necesaria sobre la principal problemática, necesidades y ventajas que implica el desarrollo y la implantación de dicho componente. Este proceso es de vital importancia para una correcta elección de las variables que deben representar adecuadamente el estado actual del entorno y proporcionar la información que necesite el nuevo componente.

Tal como se indicó en el Capítulo 3, Sección 3.2.3, un concepto se define a partir de una 3-tupla $c_i = \langle V_i; DDV_i; \Phi_i \rangle$, formada por tres elementos, donde V_i es el conjunto de variables necesario para representar el concepto c_i , DDV_i el dominio de definición de cada una de las variables pertenecientes a V_i y, finalmente, la definición del concepto se completa mediante la asociación de un conjunto de restricciones Φ_i . Teniendo en cuenta las necesidades planteadas en la sección anterior, a continuación se describe el conjunto de variables elegido para el componente de normalidad actual y que son necesarias para el análisis de trayectorias.

(a) El sistema debe conocer las zonas del entorno y la zona/as en las que se podría encontrar situado cada objeto. Para ello es necesario que el sistema disponga de una representación interna de las zonas más importantes o con mayor actividad, de tal forma que pueda establecer

correspondencias entre la posición actual de cada objeto y las zonas definidas a priori (proceso descrito en la Sección 4.3.2). En este trabajo se ha optado por definir las áreas como polígonos constituidos por un conjunto de puntos sobre imágenes en dos dimensiones. Estos polígonos son definidos mediante una herramienta de adquisición de conocimiento que será descrita en detalle en la Sección A.1, del Anexo A.

Variable	DDV_A	Descripción
A	$\{a_1, a_2, \dots, a_n\}$	Conjunto de áreas pertenecientes a un entorno observado. Cada a_i es representada mediante un polígono, formado a su vez por un conjunto de puntos.

Tabla 4.1: Conjunto A de áreas que representan las zonas de principal actividad en un entorno observado.

(b) Secuencia de áreas recorridas por el objeto y ordenadas de forma temporal. La Tabla 4.2 completa el conjunto de variables referente a las zonas del entorno y la ubicación de cada objeto. Cada $\mu_a(obj)$ representa la pertenencia de un objeto obj a un área a particular. Esta información no se obtiene en el componente en sí, sino que se genera en módulos de preprocesamiento ubicados en la capa inferior, los cuales toman como entrada la información de salida de los algoritmos de segmentación y *tracking* (ver Figura 3.1). La principal razón de calcular las zonas en las que se encuentra un objeto en un nivel inferior al del componente, es porque otros componentes de normalidad pueden necesitar la misma información. En la Sección 4.3.2 se describirá en detalle la forma en la que se estima la zona/as en las que se encuentra cada objeto. Por otro lado, el sistema mantiene un histórico $\ell(obj)$ para cada objeto obj , en el que queda reflejado la secuencia de zonas por las que ha circulado y que será empleado para establecer similitudes con las trayectorias normales.

Variable (V_i)	DDV_i	Descripción
$\mu_a(obj)$	$[0,1]$	Información vaga procedente de las capas inferiores acerca de la posibilidad de que un objeto obj se encuentre en un área a , donde 1 representa que el objeto está ubicado totalmente en el área y 0 el caso contrario.
$\ell(obj)$	$\{\mu_a(obj)\}^+$	Lista de $\mu_a(obj) \mu_a(obj) > 0$, actualizada hasta el tiempo actual.

Tabla 4.2: Variables utilizadas junto con el conjunto de áreas A para representar la ubicación de los objetos en la escena.

(c) Pertenencia de un objeto a una o varias clases de objetos (Tabla 4.3). Cada $\mu_c(obj)$ representa la pertenencia de un objeto obj a una clase c y es un valor incluido en el intervalo $[0,1]$, donde 1 representa la máxima pertenencia a una clase y 0 el caso contrario. Al igual que sucede con las zonas, la pertenencia de un objeto a una clase es calculada por los preprocesadores

situados en la capa inferior y que utilizan la información espacial procedente de los algoritmos de segmentación y de *tracking*. Por lo que, tanto la zona/as en las que se puede encontrar un objeto, como la clase/s a las que puede pertenecer son valores de entrada del componente encargado de analizar las trayectorias. El preprocesador encargado de determinar la clase a la que pertenece cada objeto hace uso del algoritmo de aprendizaje automático descrito en el Capítulo anterior. El algoritmo produce un conjunto de reglas que se emplearán en el proceso de clasificación. La forma en la que obtiene el conjunto de reglas y se utilizan para determinar la pertenencia de un objeto a una clase se describirá en detalle en la Sección 4.3.3.

Variable (V_i)	DDV_i	Descripción
$\mu_c(obj)$	$[0, 1]$	Información vaga sobre la pertenencia de un objeto a una clase c , $\forall c \in O, \exists \mu_c(obj)$.

Tabla 4.3: Pertenencia de un objeto obj a una clase c .

(d) Referencias temporales. El sistema mantiene en todo momento una referencia temporal al momento actual en el que se está produciendo el análisis y, además, permite la asociación de máximos tiempos de duración e intervalos temporales a las trayectorias (Tabla 4.4).

Variable	DDV	Descripción
t_c	$[1, 31] \cup \{*\} \times [1, 12] \cup \{*\} \times [1900 - 9999] \cup \{*\} \times [0, 24] \cup \{*\} \times [0, 59] \cup \{*\} \times [0, 59] \cup \{*\}$	Referencia absoluta al instante de tiempo actual cuyo formato es el siguiente: $DD/MM/YY - hh : mm : ss$. El símbolo $*$ es utilizado como comodín.
t_{max}	$t_{max} \in [0, 9999]$	t_{max} es el máximo tiempo de duración permitido para una trayectoria y es expresado en segundos. Este parámetro es utilizado con frecuencia para la definición de restricciones temporales.
$Int_j < t_s, t_e >$	$DDV_{t_c} \times DDV_{t_c}$	Definición de intervalos de tiempo mediante instantes iniciales (t_s) e instantes finales (t_e). Al igual que la variable anterior, también se emplea para la definición de restricciones temporales.

Tabla 4.4: Conjunto de variables temporales asociadas a las trayectorias.

Finalmente, la Tabla 4.5 muestra el resto de variables, las cuales son utilizadas principalmente para la definición de restricciones. La tercera columna de la tabla aporta una pequeña descripción sobre cada una de ellas.

Variable	DDV	Descripción
Υ	O	Υ representa el conjunto de clases/roles que pueden seguir una trayectoria determinada. Cada trayectoria definida tiene un Υ asociado, y se utiliza para definir restricciones de rol.
Ψ	$DDV_{\Psi} \subseteq DDV_A$	Ψ representa la secuencia de áreas que se debe recorrer para realizar una determinada trayectoria. Cada trayectoria tiene asociado una secuencia Ψ y es utilizada para la definición de restricciones espaciales.
$\ell_{\Psi}(obj)$	DDV_{Ψ}	Lista con la máxima pertenencia del objeto obj a cada una de las zonas recorridas de Ψ hasta el momento actual. Si un objeto, en un momento dado, no satisface la secuencialidad definida en Ψ (el objeto no se encuentra en ninguna de las zonas de la secuencia Ψ o ha alterado el orden), la lista de áreas recorridas $\ell_{\Psi}(obj)$ se vacía y vuelve a tomar el valor nulo: $\ell_{\Psi}(obj) = \emptyset$

Tabla 4.5: Variables Υ , Ψ y $\ell_{\Psi}(obj)$ para la definición de restricciones de rol y espaciales, respectivamente.

4.3. Módulos de preprocesamiento

Los módulos de preprocesamiento se encargan de preparar los valores de las variables que necesitan los componentes de normalidad para realizar el análisis. En esta sección se describen los procesos necesarios para obtener los valores de algunas de las variables que necesita el componente encargado de analizar las trayectorias y que han sido descritas en la Sección anterior.

4.3.1. Información espacial de un objeto: segmentación y *tracking*

Un paso previo a la estimación de posibles zonas en las que se puede encontrar un objeto y la determinación de las clases a las que pertenece, es la extracción de información espacial referente a su posición y tamaño. El diseño y desarrollo de algoritmos de segmentación y de *tracking* para la obtención de este tipo de información, no es un objetivo prioritario de este trabajo pero sí son necesarios para conseguir los objetivos propuestos. Si el sistema desconoce la existencia de los objetos, difícilmente podrá analizar sus comportamientos. Por este motivo, se ha optado por reutilizar *software*

existente, en concreto la librería *Blob Tracker Facility* [2] perteneciente a la biblioteca de OpenCV². La aplicación **blobtrack.cpp**, implementada en el lenguaje C++, hace uso de estas librerías para segmentar y realizar el seguimiento de objetos a partir del flujo de vídeo obtenido mediante las cámaras de seguridad. Para ello sigue un proceso encadenado que consta de las siguientes fases:

- La primera fase está formada por un discriminador que etiqueta cada píxel, bien como píxel perteneciente a la imagen de fondo (escena), o bien como un píxel perteneciente a un objeto móvil.
- En segundo lugar, un detector de blobs agrupa los píxeles que han sido etiquetados en la fase anterior como píxeles pertenecientes a objetos móviles y los cuales se encuentran próximos entre sí. Cada una de estas agrupaciones corresponde con un blob y, a su vez, un blob representa a un objeto.
- Por último, se asigna un identificador a cada objeto detectado y se lleva a cabo un seguimiento de los mismos durante el tiempo que permanecen en la escena.

En [183] se describe con mayor detalle el proceso anterior. La aplicación original genera un vídeo de salida en el que se marca cada objeto mediante una elipse y se dibuja el identificador asignado a cada uno de ellos. Para cada uno de los objetos se genera información espacial como por ejemplo las coordenadas del punto central de la elipse, así como la anchura y la altura. La aplicación original ha sido modificada para ampliar la información espacial y la funcionalidad de la herramienta. Tras las modificaciones, la herramienta proporciona el desplazamiento horizontal y vertical de cada objeto entre *frames* consecutivos, calculados como la diferencia entre los puntos *x* e *y* del centro de la elipse en el *frame* anterior y el actual. Además, se calcula el desplazamiento de un objeto como la distancia euclídea entre estos dos puntos. Por otro lado, también se ha ampliado el código fuente de la aplicación para generar un fichero de salida en el que la información del proceso de segmentación y *tracking* se hace persistente. Esta información se utiliza posteriormente en los módulos de preprocesamiento y los componentes de normalidad. A continuación se muestra un ejemplo del fichero de salida generado en los procesos de segmentación y *tracking*:

Ejemplo de fichero de salida

```

1  ...
2  528;obj4;275;272;29;19;19;3;19,24
3  528;obj3;411;382;6;10;0;0;0,00
4  529;obj6;307;309;12;18;0;0;0,00
5  529;obj5;293;459;12;14;1;1;1,41

```

²OpenCV es una biblioteca desarrollada inicialmente por la compañía Intel y que fue publicada bajo licencia BSD. La biblioteca está orientada principalmente al desarrollo de aplicaciones de visión artificial por computador.

```

6 529;obj4;286;275;30;19;10;2;10,20
7 529;obj3;411;382;6;10;0;0;0,00
8 530;obj6;307;309;12;17;0;0;0,00
9 530;obj5;294;459;12;14;0;0;0,00
10 530;obj4;287;276;32;19;1;0;1,00
11 530;obj3;411;382;5;9;0;0;0,00
12 531;obj6;306;309;12;17;0;0;0,00
13 531;obj5;294;458;12;14;0;0;0,00
14 531;obj4;284;275;33;20;3;1;3,16
15 531;obj3;411;382;5;9;0;0;0,00
16 532;obj6;306;309;12;17;0;0;0,00
17 532;obj5;295;458;12;13;0;0;0,00
18 532;obj4;283;275;34;20;0;0;0,00
19 532;obj3;413;383;5;9;1;1;1,41
20 ...

```

En el ejemplo anterior, cada una de las líneas representa un objeto detectado en la escena en un momento puntual. Cada uno de los valores alfanuméricos se separa mediante el símbolo “;” y, de izquierda a derecha, representan los siguiente: número del *frame* en el que el objeto es detectado, identificador del objeto, valor de la coordenada x del centro de la elipse, valor de la coordenada y , anchura de la elipse que envuelve al objeto, altura, desplazamiento horizontal, vertical y global. Esta es la información mínima de cada objeto que necesitan los componentes de normalidad para realizar el análisis.

4.3.2. Estimación de la localización de un objeto

La estimación de la localización de un objeto consiste en determinar la zona o zonas en las que se puede encontrar, por ejemplo, peatón en paso de peatones, grupo de personas en jardín, vehículo en la carretera, etc. Dicha estimación es llevada a cabo por un preprocesador ubicado en la capa inferior (ver Figura 3.1) que utiliza la información procedente de los algoritmos que procesan la señal de vídeo y cuyo objetivo principal es la generación de nueva información que sirva como entrada para el nuevo componente. En este caso la información que necesita el preprocesador para estimar la localización de un objeto es la siguiente:

- Conjunto de zonas (A), donde cada una de ellas es definida como un polígono sobre una imagen 2D (imagen procedente de una cámara de seguridad que representa el escenario sobre el que se desea estudiar las trayectorias). La Figura 4.3.b muestra un ejemplo de división de una escena en diferentes zonas.
- Información espacial referente a cada objeto procedente del algoritmo de segmentación utilizado. En particular las coordenadas del centro de la elipse que envuelve al objeto, así como la anchura y la altura.

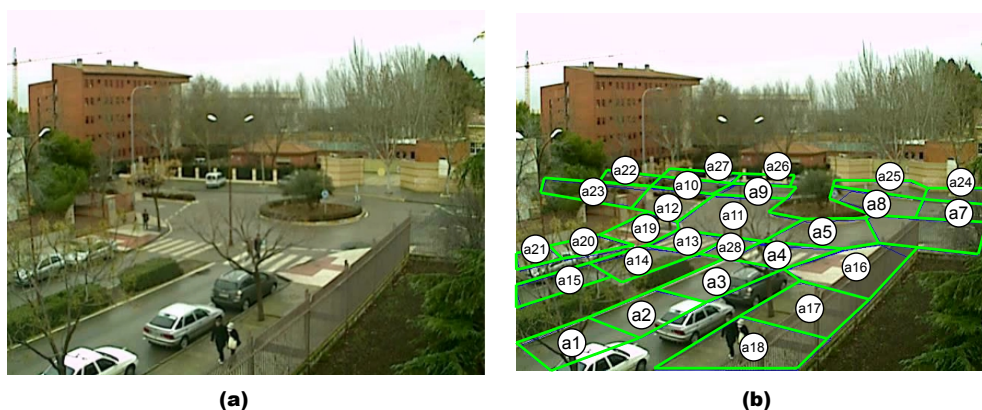


Figura 4.3: (a) Escena captada desde una cámara de vigilancia. (b) Ejemplo de división de la escena mostrada en (a) en diferentes zonas.

En líneas generales, el grado de pertenencia de un objeto a un área particular ($\mu_a(obj)$) es el resultado de dividir el número de puntos de soporte o apoyo del objeto (puntos sobre los que se sustenta) que se encuentran en el interior o en los límites del polígono que representa la zona a , entre el número total de puntos de apoyo del objeto. La zona de soporte o apoyo de un objeto coincide con la mitad inferior de la elipse que lo envuelve. Se consideran puntos de apoyo de un objeto todos aquellos que se encuentren en el interior del rectángulo cuyo vértice superior izquierdo es calculado como $(x - anchura/2, y - altura/2)$ y el vértice inferior derecho como $(x + anchura/2, y + altura/2)$, siendo x e y las coordenadas del punto central del objeto. En la Figura 4.4 se muestran algunos ejemplos de objetos marcados mediante elipses, donde los rectángulos sombreados representan los puntos de apoyo.

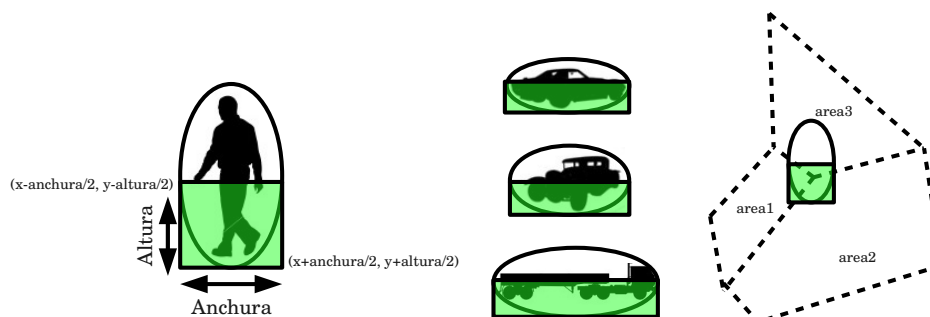


Figura 4.4: Zonas de apoyo de los objetos marcada con un rectángulo sombreado. En la parte derecha de la imagen se puede apreciar como un objeto puede estar situado en varias zonas al mismo tiempo.

En realidad, sólo aquellos puntos que se encuentren en filas y columnas impares de la matriz de puntos, son empleados para calcular la intersección con las zonas. Esta decisión viene motivada por la intención de reducir en la mayor medida de los posible el tiempo empleado en el cálculo de la

localización de un objeto, y obtener así respuestas en un tiempo cercano al real. En la Figura 4.5 se puede apreciar como la elección de los puntos situados en las columnas y filas impares proporciona una distribución representativa de la zona de soporte de un objeto, y es suficiente para determinar correctamente la localización de un objeto.

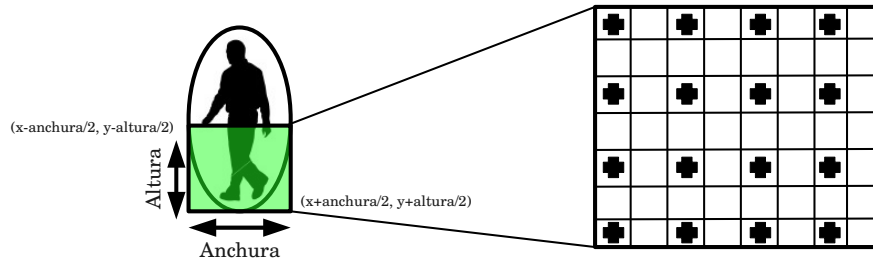


Figura 4.5: Ejemplo de puntos de la matriz de apoyo de un objeto, que son empleados para estimar la zona/as en las que se encuentra un objeto. El algoritmo estudia si cada uno de los píxeles marcados con una cruz en la matriz de la derecha se encuentran en el interior de una determinada zona.

Por otro lado, para determinar si un punto con coordenadas (x, y) se encuentra en el interior de un polígono definido a partir de un conjunto de puntos, se utiliza el algoritmo creado por W. Randolph Franklin [201], basado en un algoritmo anterior [164] y que fue citado con anterioridad en el estado del conocimiento del presente trabajo.

Algoritmo pnpoly

```

1  int pnpoly(int nvert, float *vertx, float *verty, float testx,
2      float testy)
3  {
4      int i, j, c = 0;
5      for (i = 0, j = nvert-1; i < nvert; j = i++) {
6          if ( ((verty[i]>testy) != (verty[j]>testy)) &&
7              (testx < (vertx[j]-vertx[i]) * (testy-verty[i])
8                  / (verty[j]-verty[i]) + vertx[i]) )
9              c = !c;
10     }
11     return c;
12 }
13 
```

Donde *nvert* es el número de vértices del polígono, *vertx* y *verty* los arrays que contienen las coordenadas *x*, *y* de los vértices del polígono y, finalmente, *testx* y *testy* representan las coordenadas del punto *p*. La función devuelve el valor 1 si el punto *p* se encuentra en el interior del polígono, y 0 en caso contrario. Formalmente, el grado de intersección entre un objeto y una zona se calcula tal como se indica en **Algoritmo 3**. Este algoritmo se repite para cada objeto y cada zona, cada vez que el sistema debe estimar la zona/s en las que se encuentra cada objeto.

Algorithm 3 Estimación $\mu_a(obj)$ de la intersección entre un objeto obj y un área a

Input: Conjunto de vértices del área a cuyas coordenadas se encuentran en $vertx$ y $verty$. Información espacial relativa al objeto $(x, y, ancho, alto)$.

Output: $\mu_a(obj) \in [0, 1]$

```

 $i = x - ancho/2$ 
 $puntos\_totales = 0$ 
 $puntos\_interseccion = 0$ 
while  $i \leq x + ancho/2$  do
     $j = y - alto/2$ 
    while  $j \leq y + alto/2$  do
        if  $i$  es impar Y  $j$  es impar then
             $puntos\_totales++$ 
            if  $pnpoly(nvert, vertx, verty, i, j)$  then
                 $puntos\_interseccion++$ 
            end if
        end if
         $j++$ 
    end while
     $i++$ 
end while
 $\mu_a(obj) = puntos\_interseccion/puntos\_totales$ 

```

4.3.3. Clasificación de objetos

Anteriormente, en este documento se ha discutido la necesidad de clasificar los objetos activos en una escena para que los componentes que forman parte del sistema de vigilancia puedan calcular correctamente los grados de normalidad. La clasificación de objetos permite al sistema conocer la clase o clases a las que puede pertenecer un objeto. En concreto, el componente de normalidad encargado de analizar las trayectorias necesita esta información debido a que existen trayectorias que son normales únicamente si son llevadas a cabo por ciertas clases de objetos. Este proceso no se realiza en el propio componente, ya que existen otros componentes que necesitan la misma información y supondría la repetición del mismo proceso en cada uno de ellos. Por este motivo existe un preprocesador ubicado en un nivel inferior al que se encuentran los componentes de normalidad, que se encarga de esta tarea. Éste utiliza la información espacial generada por los módulos encargados de la segmentación y el *tracking* de la escena para clasificar los objetos y su salida se conecta con los componentes de normalidad que necesitan esta información. De esta forma, el proceso de clasificación sólo se ejecuta una sola vez.

Para la clasificación de objetos se ha tomado la determinación de utilizar un conjunto de reglas difusas de tipo SI-ENTONCES adquiridas automáticamente mediante el uso del algoritmo de aprendizaje descrito en

la Sección 3.4, debido a las ventajas ya citadas en esa misma Sección. Estas reglas son de la forma **SI** v_0 es $ZD_0 \wedge \dots \wedge v_n$ es ZD_n **ENTONCES** y_j , y se caracterizan por antecedentes formados por un conjunto de variables v_i (todas ellas conectadas por el operador “and” o “y-lógico”) que toman un subconjunto de valores $ZD_i \subset DDV_{v_i}$, y por un consecuente y_j que representa la clase a la que pertenece el objeto si se activa la regla. Este tipo de reglas se adapta perfectamente al problema de la clasificación de objetos, ya que las variables representan las propiedades que caracterizan a los objetos y cada ZD_i los valores particulares que toman cada una de ellas. Dependiendo de estos valores un objeto pertenecerá a una clase u otra. Así, la elección del conjunto de variables que constituyen el antecedente de las reglas y la especificación de las clases de objetos como valor de salida de las reglas, son dos pasos previos a la etapa de aprendizaje.

El preprocesador encargado de esta tarea dispone del conjunto de variables descrito en la Sección 4.3.1, es decir, las coordenadas x e y del punto central de la elipse que envuelve al objeto, la anchura y altura de la elipse, así como los desplazamientos vertical, horizontal y global. No todas las variables disponibles tienen por qué ser empleadas para la construcción de las reglas, sino únicamente aquellas que discriminen o permitan diferenciar mejor los objetos; o incluso no tienen por qué utilizarse en su forma original y se pueden combinar mediante operaciones matemáticas para la obtención de nuevas variables. Para una correcta elección del conjunto de variables que permita diferenciar a los objetos de una escena de la mejor manera posible, se ha recopilado un conjunto de datos estadísticos sobre los valores que toman cada una de estas variables para cada uno de los tipos de objetos en la escena de la Figura 4.3. Para ello se ha exportado en una hoja de cálculo la salida de los procesos de segmentación y tracking aplicados sobre diferentes grabaciones. Los resultados del estudio se muestran en el conjunto de tablas de la Figura 4.6.

La cinco primeras tablas: A, B, C, D y E, muestran los valores que toman las variables x , y , anchura, altura, desplazamiento horizontal, desplazamiento vertical y desplazamiento global, para las personas, grupos de personas, motocicletas/bicicletas, coches y camiones, respectivamente. En la Tabla F se estudia la relación existente entre las variables anchura/altura y desplazamiento vertical / desplazamiento horizontal. Finalmente, la Tabla G resume los valores mínimo, máximo y medio para cada una de las variables y clases de objetos. Las conclusiones más interesantes provienen de la Tabla F donde se puede observar que los vehículos de grandes dimensiones (coches o camiones) son más anchos que altos más del 93 % de las ocasiones, hecho que sucede al contrario con personas y vehículos de dos ruedas. El tipo de objeto más conflictivo es *grupo de personas*, que dependiendo de su tamaño se puede confundir fácilmente con los vehículos, sin embargo, éstos últimos pueden realizar desplazamientos de mucha mayor longitud. Por otro lado, también se puede observar que el desplazamiento horizontal de los vehículos en la escena sobre la que se han realizado las pruebas, es mayor que el vertical en un 70 % de los casos. Esto es debido a la forma en la que están

4.3. Módulos de preprocesamiento

| 135 |

Estudio de las propiedades de los objetos móviles

TABLA A: Propiedades de las PERSONAS									
	X	Y	Width	Height	Hspeed	Vspeed	Mov		
MIN	91	215	5	9	0	0	0		
MAX	607	479	21	57	11	13	13,67		
Moda	369	224	15	18	0	0	0		
Menores que la Moda	1307	74	1633	1041	0	0	0		
Mayores que la Moda	1136	2328	524	1104	616	583	926		
Num veces Moda	35	49	296	158	1873	1906	1563		
Media	361,87	322,47	12,76	19,91	0,36	0,52	0,77		
Mediana	364	319	12	18	0	0	0		

TABLA B: Propiedades de los GRUPOS de Personas									
	X	Y	Width	Height	Hspeed	Vspeed	Mov		
MIN	38	219	7	9	0	0	0		
MAX	495	477	58	49	21	14	21,1		
Moda	183	325	21	34	0	0	0		
Menores que la Moda	91	752	525	1236	0	0	0		
Mayores que la Moda	1478	812	158	537	314	684	913		
Num veces Moda	28	33	246	203	1060	1283	913		
Media	324,36	336,05	25,52	27,36	0,77	0,39	1,02		
Mediana	348	326	22	27	0	0	0		

TABLA C: Propiedades de las MOTOS/BICICLETAS									
	X	Y	Width	Height	Hspeed	Vspeed	Mov		
MIN	-2	297	12	14	0	0	0		
MAX	214	355	16	43	8	17	17,26		
Moda	25	308	13	15	1	0	1		
Menores que la Moda	9	21	19	2	11	0	5		
Mayores que la Moda	75	57	36	67	50	47	64		
Num veces Moda	2	8	31	17	25	39	17		
Media	115,26	322,5	13,57	23,57	2,01	2,42	3,73		
Mediana	121,5	321	13	18	2	1	2,24		

TABLA D: Propiedades de los Coches									
	X	Y	Width	Height	Hspeed	Vspeed	Mov		
MIN	-2	222	9	7	0	0	0		
MAX	638	414	95	60	46	16	48,27		
Moda	264	267	46	26	0	0	0		
Menores que la Moda	1074	1196	1217	1365	0	0	0		
Mayores que la Moda	1224	1076	1078	872	1759	431	1821		
Num veces Moda	93	119	96	154	632	1960	570		
Media	312,97	273,78	44,87	24,72	2,39	0,42	2,54		
Mediana	265	266	45	24	1	0	2		

TABLA E: Propiedades de los CAMIONES									
	X	Y	Width	Height	Hspeed	Vspeed	Mov		
MIN	-14	213	16	15	0	0	0		
MAX	621	344	106	50	49	23	51,87		
Moda	255	268	46	27	0	0	0		
Menores que la Moda	268	334	110	110	0	0	0		
Mayores que la Moda	211	136	328	328	341	144	375		
Num veces Moda	16	23	57	57	154	351	320		
Media	294,01	255,78	56,42	33,17	2,65	0,77	3,02		
Mediana	230	258	54	32	2	0	2		

TABLA F: Comparación entre altura anchura y Vel. horizontal y vertical									
	X>H	Width<H	Width>H	Width=H	Hs>Vs	Hs=Vs	Hs<Vs		
Personas	10,53	129	68,36	15,27	67,68	16,56	11,58		
GRUPOS	32,44	482	62,74	27,74	60,88	11,58	30,23		
Motos/Bicis	0	0	0	0	0	0	0		
Coches	95,57	0,79	2,68	69,47	26,35	4,18	10,1		
Vehiculos Grandes	93,94	0,4	5,63	62,53	27,07	10,1	10,1		

TABLA G: Valor máximo, mínimo y medio para cada variable y cada tipo de objeto									
	Min	Max	Media	Vel.Horizontal	Vel.Vertical	Desplazamiento			
Personas	5	21	12,76	0,36	0,13	0,52	0	13,6	0,77
GRUPOS	7	58	25,52	0,77	0,14	0,39	0	21,1	1,02
Motos/Bicis	12	16	13,57	2,01	0,17	2,42	0	17,26	3,73
Coches	9	95	44,87	0,42	0,16	0,42	0	46,27	2,54
Vehiculos Grandes	16	106	56,42	2,65	0,23	0,77	0	51,87	3,02

Figura 4.6: Estudio de las propiedades de los objetos con capacidad de movimiento en la escena que se muestra en la Figura 4.3

ubicadas las carreteras con respecto al punto de vista de la cámara. Teniendo en cuenta estas consideraciones el conjunto de variables V que se ha empleado para la clasificación de objetos es el siguiente:

- **Ubicación horizontal** v_1 . La localización de un objeto tanto en el eje horizontal como vertical es vital para clasificar los objetos de una escena. La localización y la distancia con respecto de la cámara son dos factores que afectan en gran medida a las dimensiones de los objetos de cada clase. Un mismo objeto situado en una zona alejada de la cámara tendrá un tamaño considerablemente inferior al tamaño que tendrá cuando se encuentre en una zona cercana. Además, si se tiene en cuenta la perspectiva, un objeto puede variar sus proporciones dependiendo de la zona en la que se encuentre. El sistema debe ser consciente de estos factores para clasificar correctamente los objetos.
- **Ubicación vertical** v_2 . Localización del objeto en el eje vertical.
- **Proporción anchura-altura (W/H)**, v_3 . Valor obtenido a partir de la división de las variables anchura y altura de la elipse. Por ejemplo, el valor 2 en esta división representaría un objeto cuya anchura sería el doble que su altura, siendo muy probable que éste sea un vehículo. En cambio valores próximos a cero representan objetos mucho más altos que anchos.
- **Proporción desplazamiento horizontal y vertical**, v_4 . Al igual que en la variable anterior la proporción se obtiene mediante una división, en este caso entre las variables que contienen los valores del desplazamiento horizontal y vertical del objeto. En algunas escenas esta variable puede ser un discriminador importante de personas y vehículos. En el caso de la escena de la Figura 4.3, la ubicación de la cámara y las carreteras por las que circulan los vehículos dan lugar a que los desplazamientos horizontales de los vehículos en la imagen 2D captada por la cámara, sean normalmente mayores que los verticales.
- **Desplazamiento global (MOV)**, v_5 . Calculado como la distancia euclídea entre los centros de las elipses que envuelven al objeto entre instantes de tiempo consecutivos. Esta variable también es interesante para diferenciar entre vehículos y peatones, ya que existen longitudes de desplazamientos que son imposibles para las personas. La longitud de los desplazamientos varía dependiendo de la zona de la imagen 2D donde se produzcan, es decir, dependiendo de la distancia del objeto con respecto de la cámara. Lógicamente, la longitud de los desplazamientos que se producen en zonas alejadas de la cámara serán mucho menores que los que se producen en zonas cercanas, y no por ello recorren menos espacio en la escena real. Este fenómeno se explicará con mayor detalle en el siguiente Capítulo donde se estudia la velocidad de los objetos a partir de sus desplazamientos en una imagen 2D.

Se podría pensar en la idea de incluir una mayor número de variables como por ejemplo la inclusión directa de la anchura y la altura, o los desplazamientos verticales y horizontales. Sin embargo, cuanto mayor sea el número de variables que forman parte de las reglas, más complicado es construir un conjunto de entrenamiento representativo que abarque el mayor número de casos posibles y mayor es el número de conflictos que se producen entre las reglas. Tal como se podrá apreciar en el Capítulo de resultados, la elección del conjunto de variables realizada para la clasificación de objetos permiten discriminar correctamente a las personas y vehículos, y obtener buenos resultados en el proceso de clasificación en la escena estudiada.

En cuanto a las variables de salida se podría optar por las clases de objetos empleadas en el estudio de la Figura 4.6: personas, grupos de personas, motocicletas/bicicletas, coches y camiones. Sin embargo es importante recordar que el objetivo final es el cálculo de los grados de normalidad y la distinción entre situaciones normales y anómalas. En la escena empleada para probar los métodos propuestos en el Capítulo 3 (ver Figura 4.3), todos los vehículos están sometidos a las mismas normas independientemente de la clase a la que pertenezcan (motocicleta/bicicleta, coche y camión) y lo mismo sucede con los peatones (personas y vehículos). Por lo que no tiene sentido hacer una división tan detallada. De esta forma, las clases de objetos y_j elegidas para los consecuentes de las reglas son:

- y_1 peatón/peatones.
- y_2 vehículo.

Una vez elegido el conjunto de variables V y las clases a las que pueden pertenecer los objetos, es necesario seguir los siguientes pasos para obtener el conjunto de reglas que será empleado en el proceso de clasificación:

1. Construcción de los dominios de definición DDV_{v_i} para cada variable v_i de $V = \{v_1, v_2, v_3, v_4, v_5\}$.
2. Construcción del conjunto de entrenamiento.
3. Sustitución de los valores numéricos de los ejemplos que forman parte del conjunto de entrenamiento, por etiquetas lingüísticas.
4. Agrupación de los ejemplos formados por etiquetas lingüísticas. Aquellos objetos que tengan las mismas etiquetas son agrupados en un único ejemplo.
5. Ordenación de los ejemplos según el número de repeticiones. Aquellos ejemplos que se repitan con más frecuencia serán tratados en primer lugar en el proceso de aprendizaje.

6. Aplicación del algoritmo de aprendizaje descrito en la Sección 3.4.1 para la generación del conjunto de reglas.
7. Uso del conjunto de reglas y la información procedente del proceso de tracking para la clasificación de objetos.

1) Construcción de los dominios de definición. La Tabla 4.6 muestra los dominios de definición contruidos para cada una de las variables del conjunto $V = \{v_1, v_2, v_3, v_4, v_5\}$, donde, como ya se ha comentado anteriormente, v_1 representa la posición horizontal, v_2 la posición vertical, v_3 la relación entre anchura y altura, v_4 la relación entre el desplazamiento horizontal y vertical, y v_5 el movimiento global del objeto. Los límites establecidos para cada uno de los conjuntos que forman parte de las particiones difusas son apropiados para la escena que se muestra en la Figura 4.3. Dichas particiones han sido definidas de tal forma que exista el menor solapamiento entre las clases de objetos y se puedan diferenciar de la mejor manera posible. Para ello se han utilizado los resultados del estudio mostrado en la Figura 4.6, donde se muestran los valores mínimo, medio y máximo para cada una de las variables y para cada clase de objeto.

V_i	DDV_i	V_i	DDV_i
v_1	0 100 160 210 270 370 430 480 540 650 VL = very left L = left C = centre R = right VR = very right	v_2	0 60 100 160 200 300 340 400 440 480 VU = very up U = up C = centre D = down VD = very down
v_3	0 0.1 0.4 0.8 0.95 1.05 1.15 2.0 2.5 VU = very short S = short IP = in proportion L = large VL = very large	v_4	0 0.1 0.4 0.8 0.95 1.05 1.15 2.0 2.5 VU = very short S = short IP = in proportion L = large VL = very large
v_5	0 2 5 10 12 30 40 60 70 90 100 120 130 200 VS = very short S = short MS = medium short M = medium ML = medium large L = large VL = very large		

Tabla 4.6: Dominios de definición de las variables elegidas para la construcción de reglas difusas que permitan clasificar objetos en una escena observada por cámaras de seguridad.

Las particiones definidas para las variables x e y dividen a la escena horizontal y verticalmente. Un ejemplo de tal división se muestra en la Figura

4.7. Con divisiones de este tipo y conociendo el cuadrante en el que se encuentra cada objeto el sistema puede conocer si están situados muy a la izquierda, a la izquierda, el centro, a la derecha, o muy a la derecha; y también, si se encuentra muy arriba, arriba, en el centro, abajo, o muy abajo. El razonamiento llevado a cabo para estimar la ubicación de un objeto es similar al que realizan las personas. Los seres humanos no manejan valores numéricos precisos, sino que emplean valores lingüísticos con un alto grado de vaguedad. Este modo de razonamiento es poco preciso pero muy rápido y, en la mayoría de las ocasiones, muy efectivo. De esta forma, el sistema de vigilancia podrá afirmar, por ejemplo, que un *objeto* situado *muy a la izquierda* y en el *centro*, con un tamaño grande y una velocidad media es un vehículo. Esta forma de razonar sería muy parecida a la de un agente de seguridad humano y, precisamente, es la que se pretende trasladar a un sistema artificial.

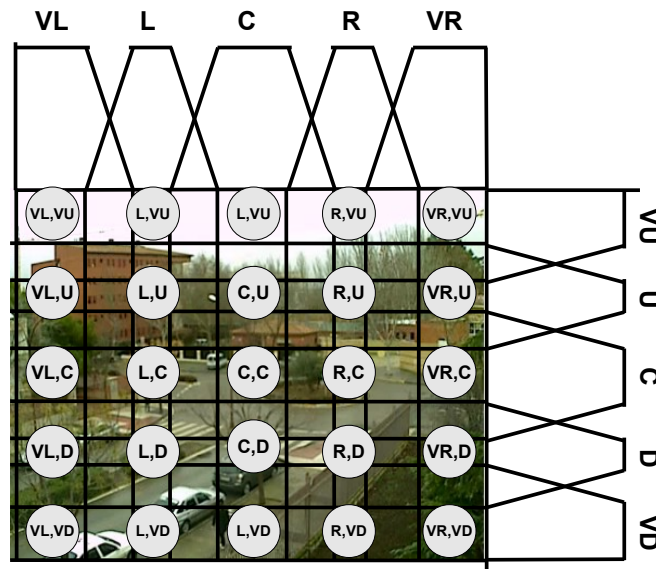


Figura 4.7: Ejemplo de división vertical y horizontal de un escena. El significado de las etiquetas lingüísticas utilizadas para la división horizontal es el siguiente: VL: very left, L: left, C: center, R: right, VR: very right. Por otro lado, las etiquetas lingüísticas utilizadas en la división vertical tienen el siguiente significado: VU: very up, U: up, C: center, D: down, VD: very down. Cada uno de los círculos representa zonas de la imagen en la que la pertenencia a los conjuntos difusos es máxima.

En cuanto a las divisiones realizadas para las variables v_3 y v_4 , la pertenencia a los conjuntos VS y S representa que el objeto es más alto que ancho en el primer caso, y que el desplazamiento vertical es mayor que el horizontal en el segundo. Valores comprendidos entre 0.95 y 1.05, es decir, pertenencia absoluta al conjunto IP indican que el objeto mantiene una proporción en anchura y altura o que el movimiento vertical y horizontal ha sido similar. Finalmente, las pertenencia a los conjuntos L y VL indica todo lo contrario que la pertenencia a VS y S . Una pertenencia a VL en el caso de la variable v_3 indica que el objeto es al menos el doble de ancho que

de alto.

Por otro lado, si se observa el dominio de definición de la variable v_5 , DDV_{v_5} , es posible apreciar que el tamaño de los primeros conjuntos difusos es inferior al de los conjuntos que representan un desplazamiento mediano o mayor que mediano. La partición difusa se ha diseñado de esta forma para diferenciar correctamente los desplazamientos realizados por personas y vehículos. Un vehículo puede realizar cualquiera de los desplazamientos que se muestran en DDV_{v_5} , sin embargo, los peatones no pueden superar el valor límite definido en el conjunto MS , independientemente de la zona en la que se encuentren. Si los conjuntos iniciales fueran de mayor tamaño, no sería posible diferenciar distintos desplazamientos para las personas ya que quedarían todos encuadrados en el primer conjunto difuso de la partición.

2) Construcción del conjunto de entrenamiento. Para la construcción de un conjunto de entrenamiento que facilite la generación de reglas de calidad, es necesario obtener una muestra representativa de ejemplos, donde cada uno de ellos toma un valor particular para cada una de las variables mencionadas anteriormente $V = \{v_1, v_2, v_3, v_4, v_5\}$. En segundo lugar es necesario realizar una división supervisada de los ejemplos del conjunto de entrenamiento de acuerdo a las diferentes clases de salida (y_1 o y_2). Cada ejemplo se representa de la siguiente forma: $e = ((x_1, x_2, x_3, x_4, x_5), y)$, donde cada x_i es el valor que toma la variable $v_i \in V$, e y la clase y_1 peatón, o bien, y_2 vehículo. La Tabla 4.7 muestra algunos ejemplos de objetos detectados en el proceso de segmentación y que han sido catalogados como personas (y_1) o como vehículo (y_2).

Peatón	Vehículo
$((253,304,0.69,1,0),y_1)$	$((482,251,2.4,2,13.15),y_2)$
$((258,300,0.73,0.33,3.16),y_1)$	$((231,296,1.18,1.67,4.47),y_2)$
$((271,311,0.75,1,1.41),y_1)$	$((22,368,1.71,4,8.54),y_2)$
$((282,308,0.82,1,5.66),y_1)$	$((557,298,1.93,1,0),y_2)$
$((410,355,0.77,1,0),y_1)$	$((278,268,1.88,10,13.6),y_2)$

Tabla 4.7: Algunos ejemplos de elementos que forman parte del conjunto de entrenamiento empleado para la generación de reglas difusas, que permiten la clasificación de los objetos detectados por el proceso de segmentación.

Para la construcción del conjunto de entrenamiento el usuario debe observar el vídeo de salida generado en el proceso de segmentación y *tracking*, donde los objetos están marcados con elipses y cada uno de ellos tiene asociado un identificador. Esta información, que se muestra de manera visual, también está contenida en un fichero de salida donde se almacenan los valores numéricos que toma cada uno de los objetos para cada una de las variables en cada *frame*. Para facilitar la división supervisada de objetos en clases se ha elaborado un programa que recibe como entrada el fichero de salida del proceso de segmentación y *tracking*, así como un listado de identificadores de objetos y la clase a la que pertenece. El programa se encarga de buscar automáticamente en el fichero toda la información

relativa a los objetos cuyos identificadores han sido listados por el usuario, y generar un nuevo fichero de salida donde los ejemplos tienen la forma $((x_1, x_2, x_3, x_4, x_5), y_j)$.

3) Sustitución de los valores numéricos de los ejemplos que forman parte del conjunto de entrenamiento por etiquetas lingüísticas. Cada uno de los valores numéricos que forman parte de los ejemplos del conjunto de entrenamiento son sustituidos por etiquetas lingüísticas. Para sustituir cada valor x_i del ejemplo $(x_1, x_2, x_3, x_4, x_5), y_j)$, se estudia la pertenencia del valor a cada uno de los conjuntos difusos definidos en el dominio DDV_{v_i} , mediante el uso de funciones de pertenencia trapezoidales (ver Función 3.21, en Capítulo 3). El sistema elige la etiqueta del conjunto difuso cuya pertenencia sea mayor. La Tabla 4.8 muestra los ejemplos de la Tabla 4.7 con etiquetas lingüísticas.

Peatón	Vehículo
$((C, C, S, IP, VS), y_1)$	$((R, C, VL, L, M), y_2)$
$((C, C, S, S, MS), y_1)$	$((L, C, L, L, MS), y_2)$
$((C, C, S, IP, MS), y_1)$	$((VL, D, L, VL, M), y_2)$
$((C, C, S, IP, MS), y_1)$	$((VR, C, L, IP, VS), y_2)$
$((R, D, S, IP, VS), y_1)$	$((C, C, L, VL, M), y_2)$

Tabla 4.8: Etiquetado de los ejemplos de la Tabla 4.7, donde los valores numéricos son sustituidos por etiquetas lingüísticas.

4) Agrupación de los ejemplos formados por etiquetas lingüísticas. Una vez que se han transformado todos los ejemplos del conjunto de entrenamiento se agrupan todos aquellos que sean iguales en un único ejemplo y se almacena el número de apariciones o instancias para luego poder ordenarlos en la siguiente etapa. El número de veces que aparezca un ejemplo no es de suma importancia para el algoritmo de aprendizaje utilizado en este trabajo, ya que una sola aparición es suficiente para poder generar una regla definitiva a partir de éste. Esta es una ventaja importante frente a los algoritmos que basan su aprendizaje en la repetición o frecuencia con la que suceden los hechos, debido a que no es necesario un número elevado de ejemplos para construir un conjunto de entrenamiento representativo, es suficiente con que aparezcan una sola vez. En la Tabla 4.9 se muestra un número reducido de ejemplos y cómo éstos se agrupan.

5) Ordenación de los ejemplos por número de repeticiones. Aquellos ejemplos que se repiten con más frecuencia son los que se extienden primero a reglas generales o reglas definitivas. Si no se hiciera de esta forma, las reglas generadas a partir de ejemplos que se repiten con menor frecuencia podrían impedir la generación de reglas más generales. La Tabla 4.10 muestra los ejemplos de la Tabla 4.9 ordenados de mayor a menor por número de apariciones.

6) Aplicación del algoritmo de aprendizaje descrito en la Sección 3.4.1. La aplicación de este algoritmo sobre el conjunto de entrenamiento genera

Ejemplos	Ejemplos Agrupados
((C,C,S,IP,VS), y_1)	((C,C,S,IP,VS), y_1) (2)
((C,C,S,S,MS), y_1)	((C,C,S,S,MS), y_1) (1)
((C,C,S,IP,VS), y_1)	((C,C,S,IP,MS), y_1) (3)
((C,C,S,IP,MS), y_1)	((R,D,S,IP,VS), y_1) (4)
((R,D,S,IP,VS), y_1)	
((C,C,S,IP,MS), y_1)	
((C,C,S,IP,MS), y_1)	
((R,D,S,IP,VS), y_1)	
((R,D,S,IP,VS), y_1)	
((R,D,S,IP,VS), y_1)	

Tabla 4.9: Agrupación de los ejemplos que forman parte del conjunto de entrenamiento.

Ejemplos no ordenados	Ejemplos ordenados
((C,C,S,IP,VS), y_1) (2)	((R,D,S,IP,VS), y_1) (4)
((C,C,S,S,MS), y_1) (1)	((C,C,S,IP,MS), y_1) (3)
((C,C,S,IP,MS), y_1) (3)	((C,C,S,IP,VS), y_1) (2)
((R,D,S,IP,VS), y_1) (4)	((C,C,S,S,MS), y_1) (1)

Tabla 4.10: Ordenación de los ejemplos de la Tabla 4.9 según el número de instancias o apariciones.

un conjunto de reglas para clasificar los objetos detectados automáticamente en la escena como peatones o vehículos. La Tabla 4.11 muestra algunos ejemplos de reglas generadas para la escena de la Figura 4.3.

7) Clasificación de objetos. Una vez finalizada la etapa de aprendizaje, el sistema dispone de un conjunto de reglas para clasificar los objetos. El módulo de preprocesamiento encargado de clasificar los objetos recibe un flujo continuo procedente de los módulos de segmentación y *tracking*, con información espacial relativa a cada uno de los objetos detectados en el *frame* actual. Cada objeto se representa con una tupla de valores similar a la empleada en los ejemplos que forman parte del conjunto de entrenamiento, es decir, un valor x_i para cada una de las variables del conjunto V pero, en este caso, sin el valor de salida y_j que es el que se pretende conocer.

Posteriormente, los valores numéricos son sustituidos por las etiquetas lingüísticas correspondientes y se estudia la activación de las reglas difusas. La activación de varias reglas en una misma clase y en clases diferentes al mismo tiempo es posible. El sistema elegirá de cada clase aquella regla con mayor grado de satisfacción, siempre y cuando sea mayor a cero. Esto quiere decir que un objeto en un momento dado puede tener diferentes grados de pertenencia a distintas clases. Estos valores se almacenan y se asocian al objeto, de tal forma que en *frames* posteriores a su aparición, el valor de pertenencia a cada clase se calcula como la media aritmética de los valores anteriores más los actuales obtenidos a partir de las reglas. De esta forma, si un objeto es clasificado como vehículo durante 125 *frames* con una evidencia

Reglas para la clasificación de objetos como personas R_1: SI v_2 es $\{VD, D\} \wedge v_3$ no es $\{L, VL\}$ ENTONCES y es Persona/as R_2: SI v_1 es $\{R, VR\} \wedge v_3$ no es $\{L, VL\}$ ENTONCES y es Persona/as R_3: SI v_1 no es $\{VL\} \wedge v_3$ es $\{VS, S\} \wedge v_5$ es $\{VS, S\}$ ENTONCES y es Persona/as R_4: SI v_1 es $\{C\} \wedge v_2$ es $\{C\} \wedge v_3$ es $\{S\} \wedge v_4$ es $\{E\} \wedge v_5$ es $\{VS\}$ ENTONCES y es Persona/as R_5: SI v_1 es $\{R\} \wedge v_2$ es $\{C\} \wedge v_3$ es $\{L\} \wedge v_4$ es $\{E\} \wedge v_5$ es $\{VS\}$ ENTONCES y es Persona/as Reglas para la clasificación de objetos como vehículos R_6: SI v_1 no es $\{R, VR\} \wedge v_2$ no es $\{D, VD\} \wedge v_3$ es $\{L, VL\}$ ENTONCES y es Vehículo R_7: SI v_2 no es $\{D, VD\} \wedge v_3$ es $\{L, VL\} \wedge v_4$ es $\{L, VL\}$ ENTONCES y es Vehículo R_8: SI v_1 es $\{VR\} \wedge v_3$ es $\{L, VL\}$ ENTONCES y es Vehículo R_9: SI v_4 es $\{L, VL\} \wedge v_5$ es $\{VS, S\}$ ENTONCES y es Vehículo R_{10}: SI v_2 no es $\{D, VD\} \wedge v_3$ es $\{L, VL\} \wedge v_5$ no es $\{VS, S\}$ ENTONCES y es Vehículo

Tabla 4.11: Ejemplos de reglas difusas para la clasificación de objetos.

absoluta 1,0 y, en un momento dado, es clasificado como peatón con un evidencia alta debido a la aparición de algún tipo de ruido o anomalía, el sistema seguirá considerando que es un vehículo. Un ejemplo de este caso se muestra en la Figura 4.8. Finalmente, el Algoritmo 4 enumera en detalle los pasos descritos anteriormente para la clasificación de objetos.

4.4. Definición de restricciones

El último paso para completar la definición del concepto trayectoria normal y finalizar la construcción del componente, consiste en la especificación de un conjunto de restricciones generales Φ_i , incluido en la tupla $\langle V_i, DDV_i, \Phi_i \rangle$. De acuerdo al modelo presentado en el Capítulo 3 (Sección 3.2.7), el cumplimiento o incumplimiento de estas restricciones permitirá al sistema conocer si un objeto sigue correctamente una trayectoria, así como el grado de normalidad asociado al concepto, es decir, cómo de normal es una situación en base al concepto “trayectoria normal”. Un valor alto de este parámetro implicaría que el objeto esta siguiendo adecuadamente alguna de las trayectorias normales propias del entorno vigilado. En particular, para el concepto trayectoria normal se han definido tres tipos de restricciones: restricciones de rol, espaciales y temporales.

Restricción de rol μ_{11} . Cada trayectoria tiene asociada una lista Υ donde

Algorithm 4 Algoritmo de Clasificación de Objetos

Input: Objeto detectado en el *frame* actual cuyos valores para las variables $(v_1, v_2, v_3, v_4, v_5)$ son $(x_1, x_2, x_3, x_4, x_5)$.

Output: Grados de pertenencia a las clases y_1 peatón e y_2 vehículo.

1.- Calcular el grado de satisfacción de cada regla.

2.- **Para cada clase c_i :**

A) Se calcula el valor de pertenencia actual sin tener en cuenta clasificaciones anteriores. Si se activan varias reglas de la misma clase (grado de satisfacción mayor que cero), el valor elegido será el máximo grado de satisfacción de las reglas activadas. En el caso de que no se active ninguna regla, el valor de pertenencia es cero.

B) La pertenencia final a cada una de las clases se calcula como la media aritmética de los valores de pertenencia anteriores y el actual.

FIN PARA

Tiempo	t_1	t_2	t_3	t_4	t_5
Elipse	 obj1	 obj1	 obj1	 obj1	 obj1
Clasificación	 1.0 vehículo	 1.0 vehículo	 0.8 vehículo	 0.5 vehículo	 1.0 vehículo
		$= (1.0+1.0)/2$ $= 1.0$	$= (1+1+0.8)/3$ $= 0.93$	$= (1+1+0.8+0.5)/4$ $= 0.825$	$= (1+1+0.8+0.5+1)/5$ $= 0.86$
 peatón	 0.0	 0.0 $= (0+0)/2$ $= 0.0$	 0.25 $= (0+0+0.25)/3$ $= 0.08$	 0.5 $= (0.5+0.25)/4 = 0.18$	 0.15 $= (0.5+0.25)/5 = 0.15$

Figura 4.8: Ejemplo de clasificación de un objeto en el que se tiene en cuenta la evolución temporal, es decir, el grado de pertenencia actual a cada clase, obtenido mediante la activación de las reglas y los grados de pertenencia obtenidos en instantes anteriores. La primera fila de la Figura representa el tiempo, cada instante situado a la derecha representa un instante posterior a su antecesor. En la fila intermedia se muestra la forma y el tamaño de la elipse que envuelve al objeto en cada uno de los instantes t_i . Finalmente, en la última fila se muestra la clasificación del objeto. Si la imagen que representa la clase tiene un valor numérico asociado a la derecha, quiere decir que el objeto ha sido clasificado en ese instante t_i con ese valor de pertenencia. Sin embargo, el valor definitivo de pertenencia se calcula como la media aritmética de los valores de pertenencia anteriores y el actual.

se especifican las clases de objetos para los que la trayectoria es adecuada. Sólo satisfacen la restricción aquellos objetos cuya clase pertenece a Υ . Por ejemplo, aquellos objetos que sean clasificados como peatones no cumplirán la restricción de rol asociada a las trayectorias específicas para vehículos. La función de pertenencia μ_{11} asociada a esta restricción se define de la siguiente manera:

$$\mu_{11}(obj) = \mu_{c_k}(obj) \iff \forall c \neq c_k; c, c_k \in \Upsilon; \mu_{c_k}(obj) \geq \mu_c(obj)$$

Tal que $\mu_c(obj)$ representa la pertenencia del objeto obj a las clase c , y $\mu_{c_k}(obj)$ el máximo valor de pertenencia a una de las clases de Υ . Otro ejemplo podría ser el siguiente: si $\Upsilon = \{\text{Persona}, \text{Grupo de Personas}\}$ y el objeto obj ha sido clasificado como $\text{Persona} = 0.3$ y $\text{Grupo de Personas} = 0.7$, el grado de satisfacción de la restricción $\mu_{11}(obj)$ será 0.7, es decir, $\max(\mu_{\text{Persona}}(obj), \mu_{\text{GrupoDePersonas}}(obj))$.

Restricción espacial μ_{21} . El tipo de restricción espacial definida en este componente y, asociada a una trayectoria, comprueba si un objeto está siguiendo en orden la secuencia de zonas definida para la trayectoria (Ψ). Cada vez que el componente recibe información espacial procedente de las capas inferiores, calcula la pertenencia a cada una de las zonas en función de la localización del objeto. En segundo lugar se actualiza cada $\ell(obj)$ (Tabla 4.2) y $\ell_\Psi(obj)$, siempre y cuando se supere alguno de los máximos almacenados en dicha lista (Tabla 4.5). Para cada una de las trayectorias asociadas a un objeto, se gestiona una lista $\ell_\Psi(obj)$ donde se almacena las máximas pertenencias a áreas de Ψ , que el objeto ya ha visitado. Es decir, si la máxima pertenencia de un objeto a un área a_k hasta el momento actual es $\mu_{a_k}(obj) = 0.8$, siendo a_k un área perteneciente a Ψ , y la pertenencia actual es $\mu_{a_k}(obj) = 0.3$, el valor de $\mu_{a_k}(obj)$ en $\ell_\Psi(obj)$ no se actualizará. De esta forma, se evitan valores extremadamente bajos en momentos de tránsito entre zonas. Cuando un objeto cambia de zona, la pertenencia a la zona que está abandonando puede ser realmente baja, valor que podría provocar el incumplimiento de la restricción espacial. El valor de satisfacción de la restricción espacial se calcula como el valor medio de todos los $\mu_{a_k}(obj) \in \ell_\Psi(obj)$. Como se comentó en la Tabla 4.5, siempre que el objeto incumpla la secuencialidad definida en Ψ , el contenido de $\ell_\Psi(obj)$ será $\emptyset$ y, por tanto, el grado de satisfacción de la restricción será cero. La función de pertenencia μ_{21} asociada a este tipo de restricción se define de la siguiente manera:

$$\mu_{21}(obj) = \begin{cases} \frac{\sum_k^{\ell(obj)} \mu_{a_k}(obj)}{|\ell(obj)|} & \text{si } |\ell_\Psi(obj)| > 0; \\ 0 & \text{en caso contrario;} \end{cases}$$

Restricciones temporales μ_{31} y μ_{41} . Las restricciones temporales permiten especificar cuándo es normal seguir una determinada trayectoria. El primer tipo de restricción temporal (μ_{31}) comprueba si se ha superado un tiempo límite t_{max} , es decir, un objeto está siguiendo una trayectoria particular de manera normal siempre y cuando lo haga en un tiempo inferior a t_{max} . La función de pertenencia μ_{31} de la restricción se define de la siguiente forma:

$$\mu_{31}(t_{max}, t_c, t_b) = \begin{cases} 1 & \text{if } (t_{max} = \emptyset) \vee (t_{max} \leq (t_c - t_b)) \\ 0 & \text{en caso contrario;} \end{cases}$$

donde t_c representa el instante de tiempo actual y t_b el instante de tiempo en el que el objeto comenzó a seguir la trayectoria.

El segundo tipo de restricción temporal (μ_{41}) comprueba si un objeto está siguiendo una determinada trayectoria en el intervalo de tiempo adecuado. Para ello se han definido cinco relaciones temporales entre instantes de tiempo e intervalos basadas en el álgebra de intervalos de Allen [15] y el trabajo desarrollado por Lin [116]. La Tabla 4.12 muestra estas relaciones.

Relación (t_c, Int_j)	Definición lógica
Before	$t_c < start(Int_j)$
After	$end(Int_j) < t_c$
During	$start(Int_j) < t_c < end(Int_j)$
Start	$start(Int_j) = t_c$
Finish	$end(Int_j) = t_c$

Tabla 4.12: Relaciones temporales entre instantes e intervalos de tiempo

Formalmente, la restricción μ_{41} se define de la siguiente forma:

$$\mu_{41}(Int_j, t_c) = \begin{cases} 1 & \text{if } (Int_j = \emptyset) \vee Start(t_c, Int_j) \vee During(t_c, Int_j) \\ & \vee Finish(t_c, Int_j)) \\ 0 & \text{en caso contrario;} \end{cases}$$

Un objeto satisface una restricción temporal con un intervalo Int_j asociado, si el instante de tiempo actual t_c pertenece a dicho intervalo, y este hecho sucede cuando se cumple alguna de las siguientes condiciones: $Start(t_c, Int_j)$, $During(t_c, Int_j)$ o $Finish(t_c, Int_j)$.

El siguiente paso a la definición formal de un concepto de normalidad mediante el uso del modelo presentado en el Capítulo anterior, es la implementación de un componente software, independiente³, que permita a una máquina realizar el análisis de la escena tal como se ha indicado en la definición. La especificación de variables debe servir para definir fácilmente la interfaz de entrada del componente software y, como salida, los grados de normalidad indicados en el modelo y la información propia de cada componente. En este caso, las posibles trayectorias que podría estar siguiendo un objeto. El componente debe incluir los procedimientos necesarios para representar internamente las restricciones definidas y comprobar el grado en el que se satisfacen, con el objeto de obtener finalmente los grados de normalidad.

³Cambios en el componente no influyen en el funcionamiento del resto de componentes.

4.5. Particularizaciones del concepto trayectoria normal en una escena real

En el modelo descrito en el Capítulo 3 se discutió la necesidad de crear particularizaciones de los conceptos definidos de forma general para problemas de vigilancia concretos (Sección 3.2.5). En el caso particular del concepto trayectoria normal, una instanciación equivale a la creación de una trayectoria concreta en un determinado entorno. En esta sección, a modo de ejemplo, se definirán las trayectorias normales de la escena mostrada en la Figura 4.3.a, captada desde una cámara situada en el laboratorio del grupo de investigación ORETO de la Universidad de Castilla-La Mancha.

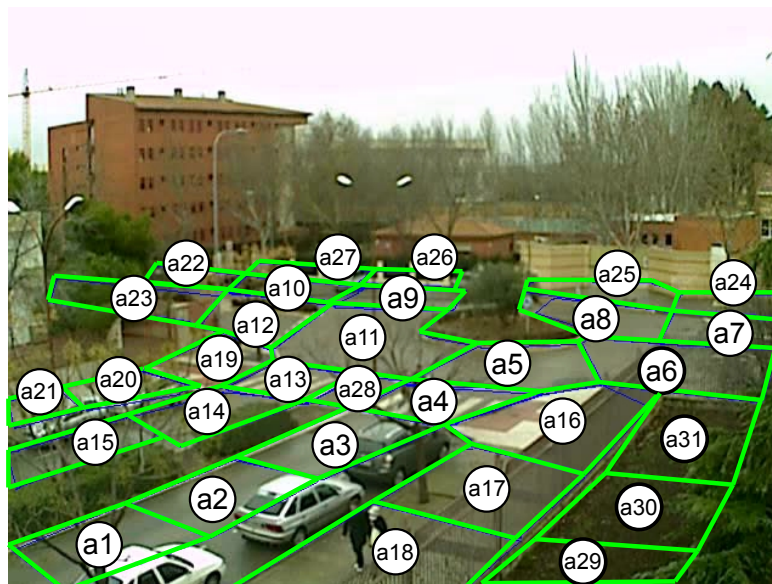


Figura 4.9: División en zonas de la escena elegida para la particularización de trayectorias normales

La escena corresponde con un entorno urbano clásico formado por aceras, jardines, carreteras, edificios, componentes de tráfico tales como una rotonda, señales, pasos de peatones, etc. Existen zonas en las que sólo pueden circular peatones y, de la misma forma, otras zonas para la circulación de vehículos. En concreto, los peatones deben circular por las aceras y únicamente cruzar las carreteras a través de los pasos de peatones no regulados mediante semáforo, además no deben invadir las zonas ajardinadas. Los vehículos únicamente puede circular por las carreteras y siempre en el sentido o dirección que se especifique. Lo que quiere decir que si un vehículo realiza una trayectoria pero en sentido contrario, sería una situación anómala debido al incumplimiento de la restricción espacial. La Figura 4.9 muestra la división en zonas que se ha realizado para la escena

elegida y la Tabla 4.13 la clase a la que pertenece cada una de ellas.

Lista de zonas	Clase
$\{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{14}, a_{15}, a_{22}, a_{23}\}$	Carretera
$\{a_{16}, a_{17}, a_{18}, a_{19}, a_{20}, a_{21}, a_{24}, a_{25}, a_{26}, a_{27}, a_{28}\}$	Acera
$\{a_4, a_{13}\}$	Paso de peatones
$\{a_{29}, a_{30}, a_{31}\}$	Universidad

Tabla 4.13: Clases o categorías a las que pertenece cada zona definida en la escena

Por tanto, el conjunto A de zonas (ver Tabla 4.1) de la escena estará formado por 31 elementos $A = \{a_1, a_2, \dots, a_{31}\}$. Las Tablas 4.14 y 4.15 muestran las instancias del concepto trayectoria normal para la escena de la Figura 4.9. Cada fila representa una trayectoria normal y cada columna, desde la segunda hasta la quinta, una particularización de una restricción. Donde μ_{11} es la restricción de rol, μ_{21} la restricción espacial, μ_{31} restricción temporal con máximo tiempo de duración y, finalmente, μ_{41} restricción temporal con intervalo de tiempo asociado. Para esta escena en concreto no existen trayectorias normales para vehículos y peatones al mismo tiempo, por este motivo la lista Υ de cada particularización está formada por un único elemento.

$c_{1,y}^1$	$\tilde{\mu}_{11}$	$\tilde{\mu}_{21}$	$\tilde{\mu}_{31}$	$\tilde{\mu}_{41}$
$c_{1,1}^1$	$\Upsilon = \{\text{vehículo}\}$	$\Psi = \{a_1, a_2, a_3, a_4, a_5, a_6\}$	150sg	$\emptyset$
$c_{1,2}^1$	$\Upsilon = \{\text{vehículo}\}$	$\Psi = \{a_1, a_2, a_3, a_4, a_5, a_6, a_8, a_9, a_{10}, a_{22}\}$	150sg	$\emptyset$
$c_{1,3}^1$	$\Upsilon = \{\text{vehículo}\}$	$\Psi = \{a_1, a_2, a_3, a_4, a_5, a_6, a_8, a_9, a_{11}, a_{13}, a_{14}, a_{15}\}$	150sg	$\emptyset$
$c_{1,4}^1$	$\Upsilon = \{\text{vehículo}\}$	$\Psi = \{a_7, a_8, a_9, a_{10}, a_{22}\}$	150sg	$\emptyset$
$c_{1,5}^1$	$\Upsilon = \{\text{vehículo}\}$	$\Psi = \{a_7, a_8, a_9, a_{11}, a_{13}, a_{14}, a_{15}\}$	150sg	$\emptyset$
$c_{1,6}^1$	$\Upsilon = \{\text{vehículo}\}$	$\Psi = \{a_7, a_8, a_9, a_{11}, a_5, a_6\}$	150sg	$\emptyset$
$c_{1,7}^1$	$\Upsilon = \{\text{vehículo}\}$	$\Psi = \{a_{23}, a_{12}, a_{11}, a_5, a_6\}$	150sg	$\emptyset$
$c_{1,8}^1$	$\Upsilon = \{\text{vehículo}\}$	$\Psi = \{a_{23}, a_{12}, a_{11}, a_{13}, a_{14}, a_{15}\}$	150sg	$\emptyset$
$c_{1,9}^1$	$\Upsilon = \{\text{vehículo}\}$	$\Psi = \{a_{23}, a_{12}, a_9, a_{11}, a_5, a_6, a_8, a_9, a_{10}, a_{22}\}$	150sg	$\emptyset$

Tabla 4.14: Conjunto de instancias del concepto trayectoria normal para los vehículos.

A modo de ejemplo, en la Figura 4.10 se muestra una secuencia de frames donde se analizan las posibles trayectorias que sigue un vehículo que aparece en la escena por primera vez en la zona a_1 . Los identificadores que se muestran en las imágenes pertenecen a la únicas trayectorias cuyas restricciones asociadas son satisfechas por el objeto móvil. Tal como se puede apreciar en la imagen, el sistema considera, en un principio, que el vehículo puede estar siguiendo las tres primeras trayectorias definidas en la Tabla 4.14. En el instante $t = 4$ el sistema descarta la trayectoria c_{11} debido a que incumple la restricción espacial asociada, es decir, la secuencia de zonas que recorre el vehículo no se corresponde con la secuencia de zonas definida para c_{11} en $\tilde{\mu}_{21}$. Por el mismo motivo se descarta c_{13} en $t = 6$. Otra

4.5. Particularizaciones del concepto trayectoria

| 149 |

$c_{1,y}^1$	$\tilde{\mu}_{11}$	$\tilde{\mu}_{21}$	$\tilde{\mu}_{31}$	$\tilde{\mu}_{41}$
$c_{1,10}^1$	$\Upsilon = \{\text{peatón}\}$	$\Psi = \{a_{29}, a_{30}, a_{31}\}$	$\emptyset$	$[HU]$
$c_{1,11}^1$	$\Upsilon = \{\text{peatón}\}$	$\Psi = \{a_{31}, a_{30}, a_{29}\}$	$\emptyset$	$[HU]$
$c_{1,12}^1$	$\Upsilon = \{\text{peatón}\}$	$\Psi = \{a_{18}, a_{17}, a_{16}\}$	$\emptyset$	$\emptyset$
$c_{1,13}^1$	$\Upsilon = \{\text{peatón}\}$	$\Psi = \{a_{16}, a_{17}, a_{18}\}$	$\emptyset$	$\emptyset$
$c_{1,14}^1$	$\Upsilon = \{\text{peatón}\}$	$\Psi = \{a_{16}, a_4, a_{28}, a_{13}, a_{19}\}$	$\emptyset$	$\emptyset$
$c_{1,15}^1$	$\Upsilon = \{\text{peatón}\}$	$\Psi = \{a_{19}, a_{113}, a_{28}, a_4, a_{16}\}$	$\emptyset$	$\emptyset$
$c_{1,16}^1$	$\Upsilon = \{\text{peatón}\}$	$\Psi = \{a_{19}, a_{20}, a_{21}\}$	$\emptyset$	$\emptyset$
$c_{1,17}^1$	$\Upsilon = \{\text{peatón}\}$	$\Psi = \{a_{21}, a_{20}, a_{19}\}$	$\emptyset$	$\emptyset$
$c_{1,18}^1$	$\Upsilon = \{\text{peatón}\}$	$\Psi = \{a_{24}, a_{25}\}$	$\emptyset$	$\emptyset$
$c_{1,19}^1$	$\Upsilon = \{\text{peatón}\}$	$\Psi = \{a_{25}, a_{24}\}$	$\emptyset$	$\emptyset$
$c_{1,20}^1$	$\Upsilon = \{\text{peatón}\}$	$\Psi = \{a_{26}, a_{27}\}$	$\emptyset$	$\emptyset$
$c_{1,21}^1$	$\Upsilon = \{\text{peatón}\}$	$\Psi = \{a_{27}, a_{26}\}$	$\emptyset$	$\emptyset$

Tabla 4.15: Conjunto de instancias del concepto trayectoria normal para peatones. HU es el identificador del intervalo de tiempo en el que la universidad se encuentra abierta al público.

forma de representar la misma secuencia, en este caso mediante valores numéricos, es tal como se indica en la Tabla 4.16. Cada una de las filas corresponde con un instante de tiempo y cada columna con los grados de satisfacción de las restricciones para las trayectorias $c_{1,1}^1$, $c_{1,2}^1$ y $c_{1,3}^1$. La última columna muestra el grado de normalidad asociado al componente de análisis de trayectorias, que se calcula tal como especifica en el modelo de normalidad. En todo momento, el comportamiento del objeto es normal debido a que satisface las restricciones de al menos una trayectoria normal de las definidas previamente.

Finalmente, la Figura 4.11 muestra la misma secuencia de la Figura 4.10 pero con una ligera modificación: el vehículo en el instante de tiempo $t = 6$ se encuentra en la zona a_{12} en lugar de a_{10} , esto supondría un comportamiento anómalo ya que el vehículo estaría circulando por dirección contraria. La primera fila del mosaico muestra la secuencia de vídeo resumida en tres frames. La segunda fila muestra como un agente humano deduciría en el instante de tiempo $t = 6$ que el comportamiento del vehículo que está siguiendo no es normal, ya que la trayectoria que está realizando no está permitida en el entorno vigilado. La tercera fila muestra las trayectorias que serían normales para el vehículo, dada la zona en la que aparece en la escena. Estas trayectorias dependen de las instanciaciones que se hayan hecho del concepto trayectoria normal. Por último, la cuarta fila muestra como el modelo de normalidad realiza un razonamiento similar al que pueda hacer un agente humano. En el instante $t = 6$, ninguna de las instancias del modelo posee un grado de normalidad elevado debido al incumplimiento de las restricciones, por lo que el sistema activaría la alarma correspondiente.

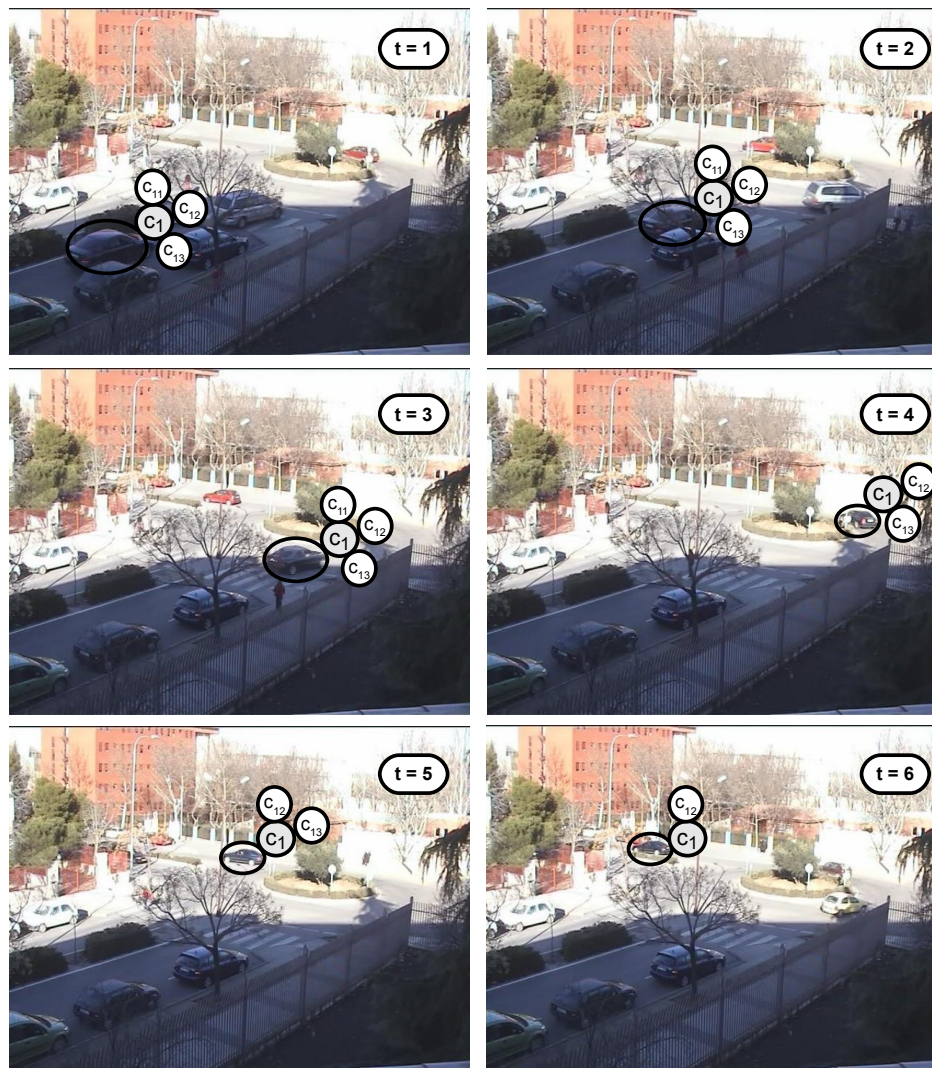


Figura 4.10: Secuencia de frames en los que el sistema de vigilancia analiza las posibles trayectorias que está siguiendo un vehículo.

4.6. Consideraciones finales

El análisis de trayectorias es uno de los más comunes en los sistemas de visión debido, en gran parte, a que la información espacial necesaria puede ser obtenida fácilmente con el uso de cámaras y algoritmos de *tracking*. En este Capítulo se ha podido apreciar como el modelo formal definido en el Capítulo 3 puede ser aplicado correctamente a la definición y el análisis de trayectorias. La definición de trayectorias mediante el modelo formal implica tres ventajas importantes, que se detallan a continuación.

4.6. Consideraciones finales

| 151 |

t	Clasificación	Ubicación	Grados de Satisfacción.	$N_{c_1}^1(obj)$
1	$\mu_{c_1}(obj1) = 1$	$\mu_{a_1}(obj1) = 0,2$ $\mu_{a_2}(obj1) = 0,8$	$c_{1,1}^1 \rightarrow \{\tilde{\mu}_{11} = 1, \tilde{\mu}_{21} = 0,8, \tilde{\mu}_{31} = 1, \tilde{\mu}_{41} = 1\}$	0.8
			$c_{1,2}^1 \rightarrow \{\tilde{\mu}_{11} = 1, \tilde{\mu}_{21} = 0,8, \tilde{\mu}_{31} = 1, \tilde{\mu}_{41} = 1\}$	0.8
			$c_{1,3}^1 \rightarrow \{\tilde{\mu}_{11} = 1, \tilde{\mu}_{21} = 0,8, \tilde{\mu}_{31} = 1, \tilde{\mu}_{41} = 1\}$	0.8
2	$\mu_{c_1}(obj1) = 1$	$\mu_{a_3}(obj1) = 0,9$ $\mu_{a_4}(obj1) = 0,1$	$c_{1,1}^1 \rightarrow \{\tilde{\mu}_{11} = 1, \tilde{\mu}_{21} = 0,9, \tilde{\mu}_{31} = 1, \tilde{\mu}_{41} = 1\}$	0.9
			$c_{1,2}^1 \rightarrow \{\tilde{\mu}_{11} = 1, \tilde{\mu}_{21} = 0,9, \tilde{\mu}_{31} = 1, \tilde{\mu}_{41} = 1\}$	0.9
			$c_{1,3}^1 \rightarrow \{\tilde{\mu}_{11} = 1, \tilde{\mu}_{21} = 0,9, \tilde{\mu}_{31} = 1, \tilde{\mu}_{41} = 1\}$	0.9
3	$\mu_{c_1}(obj1) = 0,9$ $\mu_{c_2}(obj1) = 0,1$	$\mu_{a_4}(obj1) = 0,2$ $\mu_{a_5}(obj1) = 0,8$	$c_{1,1}^1 \rightarrow \{\tilde{\mu}_{11} = 0,9, \tilde{\mu}_{21} = 0,8, \tilde{\mu}_{31} = 1, \tilde{\mu}_{41} = 1\}$	0.8
			$c_{1,2}^1 \rightarrow \{\tilde{\mu}_{11} = 0,9, \tilde{\mu}_{21} = 0,8, \tilde{\mu}_{31} = 1, \tilde{\mu}_{41} = 1\}$	0.8
			$c_{1,3}^1 \rightarrow \{\tilde{\mu}_{11} = 0,9, \tilde{\mu}_{21} = 0,8, \tilde{\mu}_{31} = 1, \tilde{\mu}_{41} = 1\}$	0.8
4	$\mu_{c_1}(obj1) = 0,8$ $\mu_{c_2}(obj1) = 0,2$	$\mu_{a_8}(obj1) = 0,7$ $\mu_{a_{25}}(obj1) = 0,2$ $\mu_{a_6}(obj1) = 0,1$	$c_{1,1}^1 \rightarrow \{\tilde{\mu}_{11} = 0,8, \tilde{\mu}_{21} = 0, \tilde{\mu}_{31} = 1, \tilde{\mu}_{41} = 1\}$	0.0
			$c_{1,2}^1 \rightarrow \{\tilde{\mu}_{11} = 0,8, \tilde{\mu}_{21} = 0,7, \tilde{\mu}_{31} = 1, \tilde{\mu}_{41} = 1\}$	0.7
			$c_{1,3}^1 \rightarrow \{\tilde{\mu}_{11} = 0,8, \tilde{\mu}_{21} = 0,7, \tilde{\mu}_{31} = 1, \tilde{\mu}_{41} = 1\}$	0.7
5	$\mu_{c_1}(obj1) = 0,7$ $\mu_{c_2}(obj1) = 0,3$	$\mu_{a_9}(obj1) = 0,7$ $\mu_{a_{26}}(obj1) = 0,3$	$c_{1,1}^1 \rightarrow \{\tilde{\mu}_{11} = 0,7, \tilde{\mu}_{21} = 0,0, \tilde{\mu}_{31} = 1, \tilde{\mu}_{41} = 1\}$	0.0
			$c_{1,2}^1 \rightarrow \{\tilde{\mu}_{11} = 0,7, \tilde{\mu}_{21} = 0,7, \tilde{\mu}_{31} = 1, \tilde{\mu}_{41} = 1\}$	0.7
			$c_{1,3}^1 \rightarrow \{\tilde{\mu}_{11} = 0,7, \tilde{\mu}_{21} = 0,7, \tilde{\mu}_{31} = 1, \tilde{\mu}_{41} = 1\}$	0.7
6	$\mu_{c_1}(obj1) = 0,7$ $\mu_{c_2}(obj1) = 0,3$	$\mu_{a_{27}}(obj1) = 0,3$ $\mu_{a_{10}}(obj1) = 0,7$	$c_{1,1}^1 \rightarrow \{\tilde{\mu}_{11} = 0,7, \tilde{\mu}_{21} = 0, \tilde{\mu}_{31} = 1, \tilde{\mu}_{41} = 1\}$	0.0
			$c_{1,2}^1 \rightarrow \{\tilde{\mu}_{11} = 0,7, \tilde{\mu}_{21} = 0,7, \tilde{\mu}_{31} = 1, \tilde{\mu}_{41} = 1\}$	0.7
			$c_{1,3}^1 \rightarrow \{\tilde{\mu}_{11} = 0,7, \tilde{\mu}_{21} = 0, \tilde{\mu}_{31} = 0,1, \tilde{\mu}_{41} = 1\}$	0.0

Tabla 4.16: Grados de satisfacción de las restricciones y grados de normalidad correspondiente a la secuencia mostrada en la Figura 4.10. En la primera columna, c_1 representa la clase *vehículo*, mientras que c_2 la clase *peatón*. Los valores numéricos marcados en negrita en la última columna representan el máximo grado de normalidad, es decir $N_{c_1}^1(obj)$.

La primera de ellas es referente a la interpretabilidad. Si se observa la Tabla 4.17 es posible interpretar fácilmente que la trayectoria $c_{1,10}^1$, únicamente puede ser realizada por peatones, consta de una secuencia de tres áreas: a_{29}, a_{30}, a_{31} , y sólo es normal si se produce en el intervalo de tiempo con identificador $[HU]$. El operador encargado de mantener la base de conocimiento del sistema experto podría modificar fácilmente los valores de la trayectoria o incluso crear nuevas de forma manual sin mucho esfuerzo. Por el contrario, aquellos sistemas en los que las trayectorias se definen como una secuencia de puntos (x, y) , la interpretabilidad es mucho menor y son necesarios algoritmos de aprendizaje automáticos que gestionen la información de manera interna. Una trayectoria representada como una secuencia de puntos podría ser la siguiente: $\{(200,45), (220,47), (220, 50), \dots, (230, 80)\}$. Evidentemente, la comprobación y creación de trayectorias por agentes humanos con este segundo formato resulta mucho más complicado

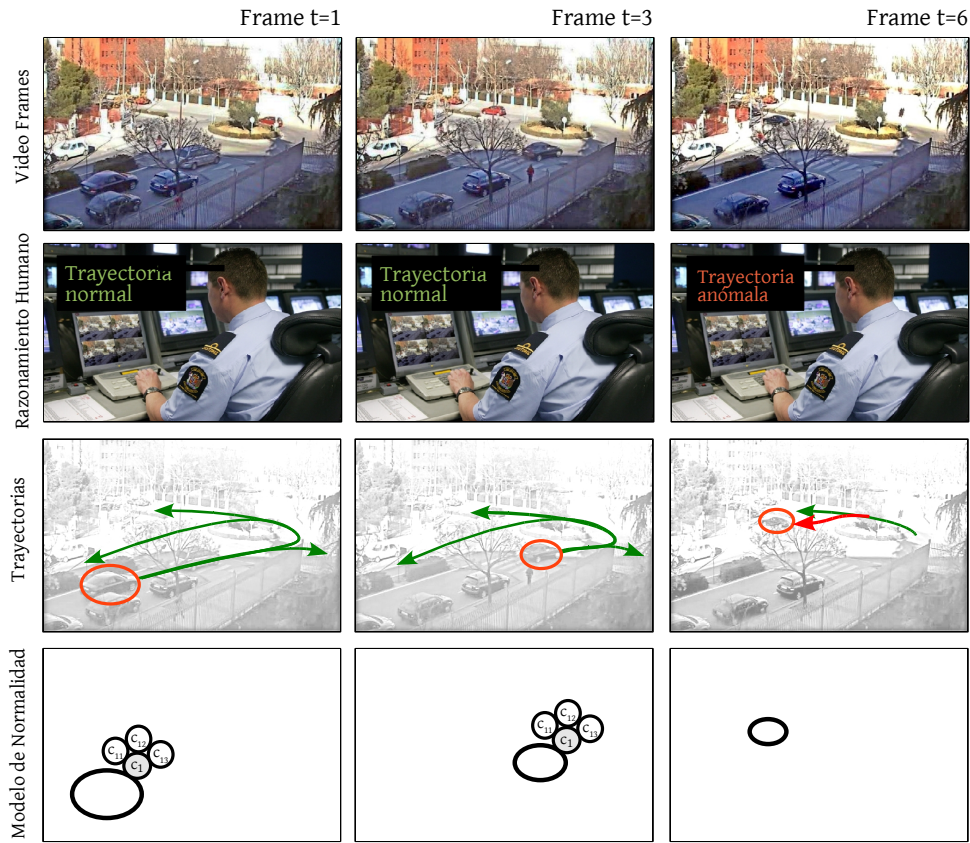


Figura 4.11: Comportamiento anómalo de un vehículo debido a que no sigue ninguna de las trayectorias normales definidas en el sistema.

que con el primero.

$c_{1,y}^1$	$\tilde{\mu}_{11}$	$\tilde{\mu}_{21}$	$\tilde{\mu}_{31}$	$\tilde{\mu}_{41}$
$c_{1,10}^1$	$\Upsilon = \{\text{peatón}\}$	$\Psi = \{a_{29}, a_{30}, a_{31}\}$	$\emptyset$	$[HU]$

Tabla 4.17: Ejemplo de definición de trayectoria normal mediante el modelo formal

La segunda ventaja es la posibilidad de reutilizar el componente en diferentes entornos y con distintos tipos de objetos. El componente es definido una sola vez y posteriormente puede ser instanciado en aquellos entornos que se desean monitorizar.

Finalmente, la definición de trayectorias mediante restricciones de acuerdo a la forma que indica el modelo, permite definiciones más ricas frente a aquellas que se centran únicamente en información espacial. Este tipo de definiciones es más apropiado para sistemas de seguridad donde

4.6. Consideraciones finales

| 153 |

es necesario tener en cuenta otros tipos de aspectos como la clase a la que pertenece un objeto o el momento en el que se producen las acciones. Además, el modelo no sólo es escalable en cuanto a la inclusión de nuevos métodos de análisis mediante la definición de nuevos componentes, sino que también las definiciones de los propios conceptos pueden ser ampliadas con la inclusión de nuevas restricciones.



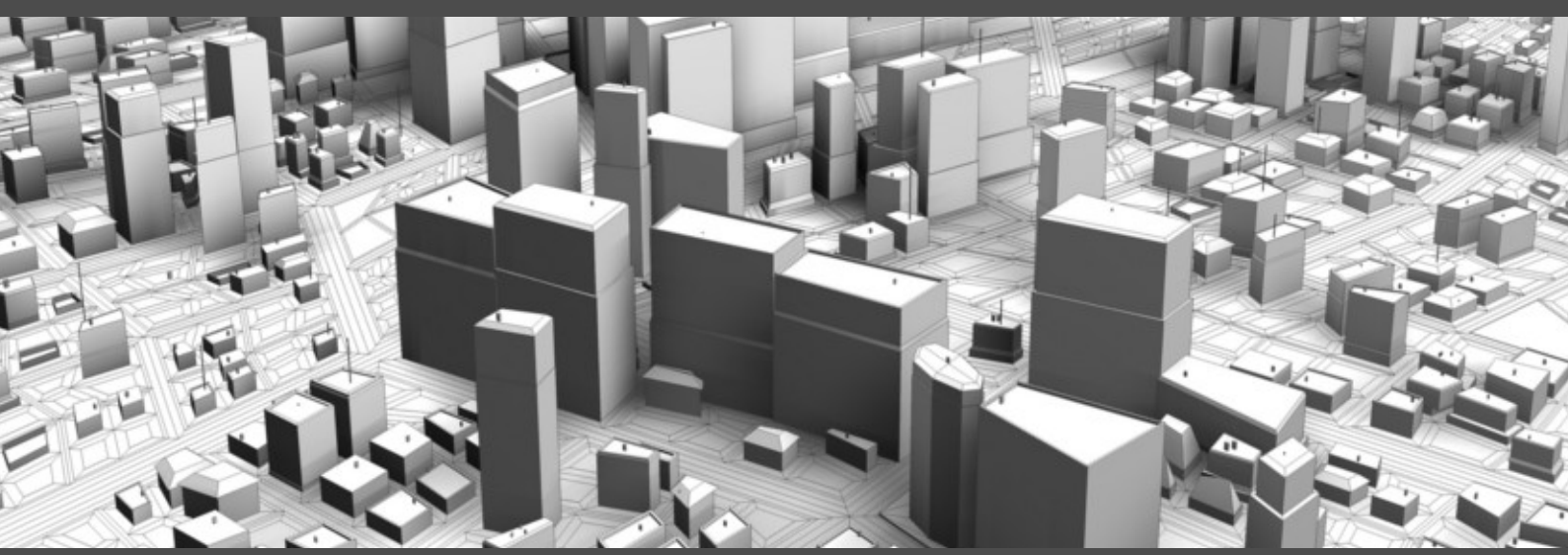
Capítulo

5

Componente de Normalidad
para el Análisis de Velocidades

5

Capítulo



5

Componente de Normalidad para el Análisis de Velocidades

En este Capítulo se presenta un componente de normalidad que analiza la velocidad de los objetos detectados en la escena, a partir de una imagen 2D y sin la necesidad de conocer los parámetros de la cámara como, por ejemplo, el ángulo de inclinación o la altura a la que está situada. En la primera Sección se analiza la principal problemática del análisis de velocidades a partir de imágenes en dos dimensiones procedentes de las cámaras de seguridad. Dicho análisis da pie a la elección de variables descrita en la siguiente Sección y, en consecuencia, a la construcción de los módulos de preprocesamiento encargados de obtener los valores para estas variables. Posteriormente, se definen las restricciones de normalidad necesarias para el análisis de las velocidades de los objetos. Finalmente, se muestran un conjunto de particularizaciones del concepto “velocidad normal” para una escena concreta, las cuales han sido aprendidas mediante el algoritmo de aprendizaje presentado en el Capítulo 3.

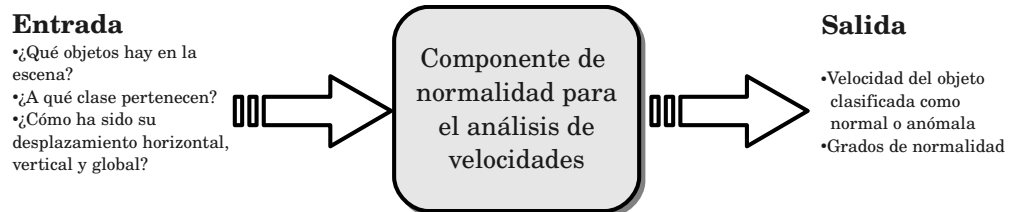


Figura 5.1: Descripción, con un alto nivel de abstracción, de las entradas y salidas del componente de normalidad encargado de analizar las velocidades en un entorno vigilando por un conjunto de cámaras.

5.1. Concepto de velocidad normal

La velocidad con la que un objeto se mueve en una escena es uno de los aspectos a los que usualmente hay que prestar atención en entornos monitorizados, ya que velocidades altas pueden ser indicios de que algo no transcurre normalmente. Un vehículo abandonando rápidamente una zona puede indicar que se está dando a la fuga o también puede ser motivo de un futuro problema (por ejemplo causa de un accidente), o un individuo corriendo por una zona en la que normalmente se suele ir andando puede indicar también algún tipo de problema. Es por esto por lo que el diseño de componentes de análisis de normalidad de velocidades es muy interesante.

La velocidad a la que circula un objeto en un entorno particular se considera normal, siempre y cuando su valor se encuentre entre los límites de velocidad establecidos para dicho entorno. Estos límites dependerán de las características del entorno y pueden ser superiores, cuando el objeto no puede exceder una velocidad límite, e inferiores, cuando el objeto debe circular siempre a una velocidad superior a la mínima establecida. Ambos límites no tienen por qué estar presentes siempre; por ejemplo, en la mayoría de entornos urbanos no existe una velocidad mínima, ya que los vehículos y las personas pueden estar detenidos, sin movimiento, siendo una situación normal. Sin embargo, si el sistema de seguridad se empleara para analizar eventos en una autopista, éste debería tener en cuenta que los vehículos deben circular siempre a una velocidad mínima. Por el contrario, el límite superior de velocidad suele estar presente en la mayoría de escenarios.

El principal objetivo de este nuevo componente es determinar si la velocidad de cada objeto en movimiento es normal o no lo es, haciendo uso únicamente de una imagen 2D procedente de una cámara de seguridad. En ningún momento el objetivo es calcular el valor real de velocidad con el que se mueven los objetos. Se trata de trasladar al sistema experto la capacidad que tienen las personas para analizar las velocidades sin la necesidad de conocer valores numéricos precisos. Una persona podría deducir perfectamente si un objeto se mueve a gran velocidad en la visualización de un vídeo, únicamente mediante el estudio del desplazamiento y tomando como referencia los elementos cercanos al objeto en cuestión. Este enfoque, al igual que cualquier otro, tiene sus ventajas y desventajas. Una de las grandes ventajas es el bajo coste en términos de recursos físicos y económicos, ya que tan sólo es necesario una cámara para poder analizar las velocidades. Como desventaja se podría mencionar una precisión inferior con respecto a los sistemas que calculan la posición 3D de cada objeto mediante mecanismos de transformación y proyección [44, 113, 41, 23, 143, 123], obteniendo una representación real y más exacta de la localización y el movimiento de cada objeto. Sin embargo, frente a este tipo de sistemas, también hay que decir que la solución adoptada no necesita conocer los parámetros de la cámara a priori y ofrece una gran velocidad de cómputo, aspecto, que como ya se ha mencionado anteriormente, es importante en los sistemas de seguridad.

Además, si se tiene en cuenta que el objetivo principal es la clasificación de las velocidades como normales o anómalas, y no la estimación de un valor numérico que represente la velocidad del objeto, la precisión en este caso pierde importancia.

La clasificación de las velocidades como normales o anómalas a partir del uso único de una imagen bidimensional, es posible mediante el análisis de los desplazamientos entre instantes de tiempo consecutivos, sin la necesidad de realizar una reconstrucción tridimensional de la posición y el movimiento del objeto. Cuanto mayor sea el desplazamiento mayor será la velocidad. También es importante tener en cuenta el incremento de tiempo producido ya que, por ejemplo, no es lo mismo un desplazamiento de 20 píxeles en 0.5 segundos que en 1 segundo. El desplazamiento global de un objeto en una imagen 2D se puede calcular como la distancia euclídea existente entre los centros de las elipses que lo envuelven, dividido entre el incremento de tiempo:

$$d = \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}{t_2 - t_1} \quad (5.1)$$

siendo (x_1, y_1) las coordenadas del centro de la elipse en el instante t_1 , y (x_2, y_2) las coordenadas del centro en el instante t_2 .

Esta distancia no es la distancia real que recorre el objeto en el entorno, sino el desplazamiento de la elipse del objeto en una imagen plana. La principal dificultad a la que se enfrenta este método de estimación es la perspectiva de la cámara. Un mismo desplazamiento en una zona cercana a la cámara puede implicar una velocidad normal, mientras que en zonas alejadas (donde la profundidad de la escena es mayor) se puede tratar de una velocidad fuera de los límites establecidos. Es por esto, por lo que es conveniente realizar el análisis de las velocidades de los objetos en función de sus localizaciones en la imagen 2D, teniendo en cuenta la perspectiva de la cámara. La parte izquierda de la Figura 5.2 muestra un ejemplo de la división de una escena de acuerdo a las distancias con respecto a la cámara: A) cerca, B) distancia media y C) lejos. En la imagen de la derecha se ha trasladado la misma elipse que envuelve a un vehículo en una zona cercana a la cámara, a otro vehículo de las mismas características situado en una zona lejana. En esta imagen se puede observar como la elipse tiene un tamaño suficiente como para envolver a varios vehículos del mismo tipo en zonas alejadas de la cámara. Si se habla de desplazamientos en lugar de dimensiones de objetos, es posible afirmar que un desplazamiento en píxeles de un objeto sobre la zona A, implica un desplazamiento real mucho mayor en la zona C. Es decir, un desplazamiento de 10 píxeles en la zona A de la imagen puede equivaler a 3 metros reales, mientras que ese mismo desplazamiento en la zona C puede equivaler a 8 metros. El objetivo en este componente es aprender el conjunto de desplazamientos permitidos para cada región y cada tipo de objeto, a partir de un conjunto de observaciones previas.

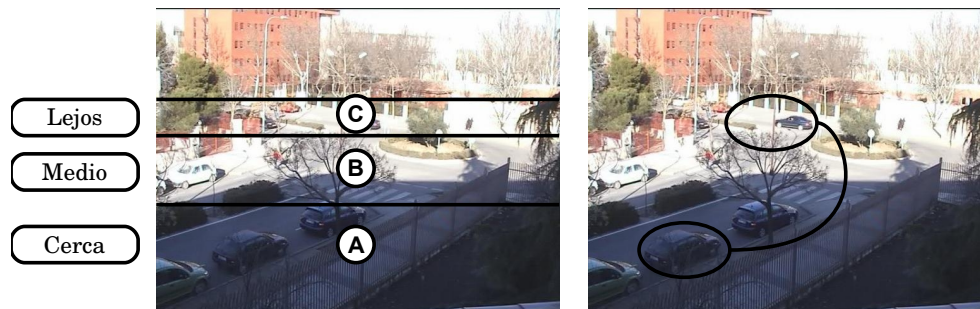


Figura 5.2: División de una escena teniendo en cuenta las distancias con respecto a la cámara. La parte derecha muestra como pueden variar notablemente las dimensiones de los objetos dependiendo de si se encuentra en una zona cercana o lejana a la cámara.

En base a los requisitos planteados anteriormente, para que un sistema artificial pueda estimar las velocidades a partir del desplazamiento de los objetos en una imagen 2D, necesita como mínimo la siguiente información:

- Conocer los objetos que existen en la escena.
- Posición de cada uno de los objetos detectados.
- Clases a las que pertenecen los objetos. Para determinar si la velocidad de un objeto es normal o no, es muy importante conocer la clase a la que pertenece. Por ejemplo, velocidades medias para los vehículos pueden implicar velocidades excesivas para las personas. Para tratar este problema el componente deberá utilizar restricciones de rol, similares a las empleadas para el estudio de trayectorias.
- Desplazamiento horizontal, vertical y global del objeto.

Para que el componente determine si la velocidad de un objeto es normal debe conocer para cada clase y cada zona cuáles son los desplazamientos permitidos, medidos en píxeles. En el componente del Capítulo anterior, el conocimiento requerido para el análisis de normalidad se hacía explícito mediante el uso de herramientas de adquisición de conocimiento. En el caso de las trayectorias, el uso de este tipo de herramientas tiene sentido ya que el experto encargado de manejarlas puede saber con total precisión la secuencia de zonas por las que debe circular un objeto y el tiempo en el que debe hacerlo. Sin embargo, conocer los desplazamientos máximos permitidos en cada una de las zonas de la imagen de una manera precisa es mucho más complicado, por lo que la opinión del experto, en este caso, puede llegar a ser incluso subjetiva. Tal como se comentó anteriormente, una persona que observa un vídeo, sin manejar valores numéricos, podría deducir fácilmente si un objeto se mueve a gran velocidad, tomando como referencia el resto de elementos del entorno, tanto los estáticos como los dinámicos. Sin embargo, no podría ser capaz de conocer con exactitud el

número de píxeles que se desplaza en cada momento sin el uso de alguna herramienta de medida. Aún disponiendo de una herramienta de medida sería una tarea demasiado tediosa. Por este motivo, en este componente se ha optado por el uso de un algoritmo de aprendizaje que facilite la instanciación del concepto velocidad normal. El algoritmo aprende los desplazamientos que son normales en cada una de las zonas, independientemente de si están a una distancia cercana, media o alejada de la cámara; todo ello a partir de un conjunto de entrenamiento formado por ejemplos en los que los objetos circulan a una velocidad normal.

El papel que desempeña el algoritmo de aprendizaje en este componente es el mismo que el de la herramienta de adquisición de conocimiento con la interacción del experto en el componente anterior. Para que el algoritmo de aprendizaje pueda desempeñar las mismas funciones y reemplazar con ciertas garantías la intervención del experto humano y el uso de herramientas, es necesario que satisfaga las siguientes necesidades:

- El conocimiento aprendido por el algoritmo de aprendizaje debe ser interpretable. Cuanto más cercano sea al lenguaje natural, más sencillo será compartir, modificar y reutilizar este conocimiento. Tal como se representaron las trayectorias en el Capítulo anterior y la forma en la que se realizaron las particularizaciones (ver Sección 4.5), es sencillo apreciar la clase de objetos que pueden realizar la trayectorias, los lugares por los que deben circular y el tiempo en el que deben hacerlo. Para este nuevo componente se busca la misma interpretabilidad en la definición de velocidades.
- El algoritmo debe tener la capacidad de aprender todas las instancias del concepto velocidad normal a partir de un conjunto de observaciones o casos previos. Cada instancia corresponde con un caso de velocidad normal de un tipo de objeto, en una zona/zonas en base al desplazamiento realizado por el objeto. Es conveniente recordar en este punto que las instancias de los conceptos se definen en base a la particularización de un conjunto de restricciones (Sección 3.2.5), por lo que el algoritmo debe tener la capacidad de aprender estas particularizaciones.

Una vez analizado el problema de la clasificación de velocidades de los objetos como normales o anómalas, detectados los principales problemas y dificultades, así como la necesidad de usar algoritmos de aprendizaje automático para instanciar la definición general, en las sucesivas secciones se discutirá la elección de variables y la definición de restricciones para representar el concepto de velocidad normal en un entorno monitorizado, de acuerdo al modelo presentado en el Capítulo 3.

5.2. Variables para la definición y el análisis de velocidades

La elección de variables para la definición de velocidades normales surge como consecuencia del análisis realizado en la Sección anterior. El conjunto de variables V_2 para definir el concepto de velocidad normal c_2 presentado en este Capítulo ($c_i = \langle V_i, DDV_i, \Phi_i \rangle$), está compuesto por cuatro grupos de variables: i) un primer grupo para determinar la localización horizontal y vertical del objeto, ii) un segundo grupo formado por variables que determinan la pertenencia de un objeto a una clase, iii) el tercero contiene información sobre el desplazamiento del objeto y, finalmente, iv) el último grupo es empleado para la definición de las restricciones de normalidad necesarias para representar el concepto de velocidad normal. A continuación se mostrarán las variables que conforman cada grupo:

a) Localización horizontal y vertical de un objeto. La localización de un objeto en una imagen 2D se calcula a partir de las coordenadas x e y del centro de la elipse que lo envuelve; donde x representa la posición en el eje horizontal, e y en el eje vertical. El método empleado para estimar la región en la que se encuentra un objeto es exactamente el mismo que se utilizó en la Sección 4.3.3 (ver Figura 4.7 del Capítulo anterior) para la clasificación de objetos. La escena se divide horizontal y verticalmente mediante particiones difusas. Cada sector de la división tiene asociado etiquetas lingüísticas de tal forma que un objeto podrá estar situado muy a la izquierda (VL), a la izquierda (L), el centro (C), a la derecha (R) o muy a la derecha (VR) en el eje horizontal, y muy arriba (VU), arriba (U), en el centro (C), abajo (D), o muy abajo (VD) en el vertical.

La Tabla 5.1 muestra las dos variables empleadas para determinar la localización de un objeto y sus dominios de definición: posición horizontal (HP) y posición vertical (VP). En concreto, los valores numéricos mostrados para cada uno de los conjuntos difusos de la Tabla 5.1, son los que se han elegido para la escena sobre la que se han desarrollado las pruebas. Sin embargo estos valores pueden variar dependiendo de la escena y la resolución de la cámara utilizada.

b) Clase a la que pertenece un objeto. En este trabajo se ha discutido en varias ocasiones la importancia de conocer la clase a la que pertenece cada objeto para catalogar un comportamiento como normal o anómalo. Al igual que en las variables empleadas para el análisis de trayectorias, cada $\mu_c(obj)$ representa la pertenencia de un objeto obj a una clase c y es un valor incluido en el intervalo $[0, 1]$, donde 1 representa la máxima pertenencia a una clase y 0 el caso contrario. Véase la Sección 4.3.3 del Capítulo 4, para una descripción detallada de la forma en la que se calcula cada $\mu_c(obj)$. La Tabla 5.2 muestra el dominio de definición de la variable $\mu_c(obj)$ y una breve descripción de la misma. En la misma Tabla también se describe la variable Υ que se utiliza para la definición de restricciones de rol asociadas

5.2. Variables para la representación de velocidades

| 163 |

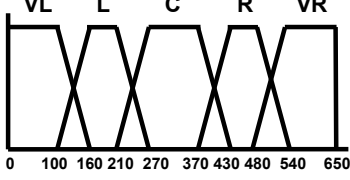
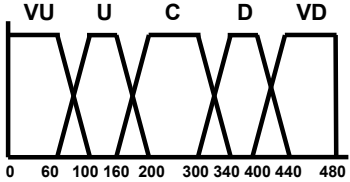
Variable V_i	DDV_i	Descripción
HP	 VL = very left L = left C = centre R = right VR = very right	Posición horizontal.
VP	 VU = very up U = up C = centre D = down VD = very down	Posición vertical.

Tabla 5.1: Conjunto de variables utilizado para determinar la localización de un objeto. Los valores que definen los conjuntos difusos pueden variar dependiendo de la escena y la resolución de la cámara empleada. En este caso, los valores corresponden con la escena descrita en 4.5.

a las velocidades. Finalmente v_c representa el identificador de la clase c cuyo $\mu_c(obj)$ sea máximo. Tal como se verá más adelante, esta variable formará parte de los ejemplos incluidos en el conjunto de entrenamiento que se utilizará para la generación de reglas que permitan clasificar las velocidades de los objetos. De acuerdo a la naturaleza del algoritmo de aprendizaje empleado, el conjunto de entrenamiento no debe contener ningún tipo de incertidumbre, por lo que se debe indicar de alguna forma si un objeto es un vehículo o un peatón; hecho que puede ser fácilmente diferenciable por la persona que construye el conjunto de entrenamiento. Posteriormente, el formato que deben cumplir los elementos que se pretenden clasificar debe ser el mismo que poseían los ejemplos empleados en el conjunto de entrenamiento. La cuestión es que el sistema de vigilancia diseñado en este trabajo calcula la pertenencia $\mu_c(obj)$ de cada objeto a cada clase c , es decir, un objeto puede pertenecer a varias clases con distintos grados. Sin embargo, para hacer uso del conjunto de reglas que determinen si la velocidad del objeto es normal o no, será necesario especificar si el objeto es un peatón ($v_c = 0$) o un vehículo ($v_c = 1$). Por este motivo, el componente de normalidad encargado de analizar las velocidades elegirá como clase v_c aquella cuyo valor de pertenencia sea mayor.

c) Desplazamiento de un objeto. El desplazamiento de un objeto se determina a partir del punto central de la elipse que lo envuelve. Cada vez que éste realiza un movimiento y se desplaza en la escena, el punto central varía de posición modificando su valor en el eje vertical y horizontal. A partir del incremento y decremento de estos dos parámetros es posible estudiar dos tipos de desplazamientos: verticales y horizontales. Por otro lado, la

Variable (V_i)	DDV_i	Descripción
$\mu_c(obj)$	$[0, 1]$	Información vaga sobre la pertenencia de un objeto a una clase c , $\forall c \in O, \exists \mu_c(obj)$.
Υ	O	Υ representa el conjunto de clases/roles que pueden alcanzar una velocidad determinada. Cada velocidad definida tiene un Υ asociado, y se utiliza para definir las restricciones de rol.
v_c	$[0, (O - 1)]$	Determina el identificador de la clase a la que pertenece un objeto. Si el conjunto de clases de objetos O a los que el sistema debe prestar especial atención está formado por cuatro clases, el dominio de definición de la variable v_c será $DDV_{v_c} = \{0, 1, 2, 3\}$.

Tabla 5.2: Pertenencia de un objeto obj a una clase c .

combinación de ambos proporciona el desplazamiento global del objeto. La Tabla 5.3 agrupa el conjunto de variables que se utilizan para estudiar estos tipos de desplazamientos.

c) Variables utilizadas para la definición de restricciones espaciales. En la Tabla 5.4 se muestra el resto de variables utilizadas en el componente de normalidad para el análisis de velocidades, las cuales son empleadas para la definición de restricciones espaciales asociadas a instancias del concepto velocidad normal. Cada una de estas variables representa un subconjunto de los dominios de definición de las variables anteriores y determinan las posibles posiciones horizontales y verticales de un objeto para una instancia del concepto velocidad normal, así como los posibles desplazamientos verticales, horizontales y globales permitidos.

5.3. Módulos de preprocesamiento

En esta Sección se describen los módulos de preprocesamiento encargados de generar la información de entrada que necesita el componente para analizar las velocidades con las que circulan los objetos. En concreto, el componente de velocidades necesita cuatro módulos de preprocesamiento: i) módulo para la segmentación y el *tracking* de la escena, ii) clasificación de los objetos, iii) localización horizontal y vertical del objeto y, iv) cálculo del desplazamiento.

Los módulos de preprocesamiento para el *tracking* de la escena y para la

Variable V_i	DDV_i	Descripción
HD	VS S MS M ML L VL 0 2 5 10 12 30 40 60 70 90 100 120 130 200 VS = very short S = short MS = medium short M = medium ML = medium large L = large VL = very large	Desplazamiento horizontal del objeto. El valor numérico de la variable HD proviene del módulo de preprocesamiento encargado de calcular los desplazamientos de los objeto y que es descrito en la Sección 5.3.1.
VD	VS S MS M ML L VL 0 2 5 10 12 30 40 60 70 90 100 120 130 200 VS = very short S = short MS = medium short M = medium ML = medium large L = large VL = very large	Desplazamiento vertical del objeto. El valor numérico de la variable VD proviene del módulo de preprocesamiento descrito en la Sección 5.3.1.
MOV	VS S MS M ML L VL 0 2 5 10 12 30 40 60 70 90 100 120 130 200 VS = very short S = short MS = medium short M = medium ML = medium large L = large VL = very large	Desplazamiento global del objeto. Al igual que las dos variables anteriores el valor de esta variable también proviene del módulo de preprocesamiento que calcula los desplazamientos de los objetos.

Tabla 5.3: Conjunto de variables utilizado para determinar el desplazamiento de un objeto. Al igual que en la Tabla 5.1 los valores numéricos que establecen los límites de los conjuntos difusos corresponden con la escena utilizada en las pruebas y que ha sido descrita previamente en 4.5. En esta escena, el estudio de los desplazamientos de los objetos y la clasificación de la velocidad como normal o anómala se ha realizado cada 25 *frames*. En otro tipo de escenas en los que los vehículos pueden circular a velocidades mucho mayores, el análisis se debe realizar con mayor frecuencia. Cuanto menor sea el intervalo de tiempo entre análisis, mayor será la carga computacional para el sistema.

clasificación de objetos son los mismos que se han empleado en el Capítulo anterior, ver 4.3.1 y 4.3.3 para una descripción detallada. La forma en la que el tercer preprocesador determina la posición horizontal y vertical de un objeto mediante una división difusa de la imagen 2D captada por la cámara, también ha sido descrita en la Sección 4.3.3. En el esquema de la Figura 5.3 se puede apreciar las conexiones entre los preprocesadores que emplea el componente descrito en este Capítulo, así como las entradas y salidas de cada uno de los módulos.

5.3.1. Desplazamiento de un objeto

Este preprocesador mantiene para cada objeto un histórico con los puntos donde ha estado situado en cada instante tiempo. Si (x_1, y_1) son las coordenadas del centro de la elipse que envuelve a un objeto en un instante

Variable (V_i)	DDV_i	Descripción
$\ell_{HP}(obj)$	DDV_{HP}	Conjunto de etiquetas lingüísticas de DDV_{HP} que determinan las posiciones horizontales en las se puede encontrar un objeto para seguir una determinada velocidad.
$\ell_{VP}(obj)$	DDV_{VP}	Conjunto de etiquetas lingüísticas de DDV_{VP} que determinan las posiciones verticales en las se puede encontrar un objeto para seguir una determinada velocidad.
$\ell_{HD}(obj)$	DDV_{HD}	Movimientos horizontales permitidos para una estancia del concepto velocidad.
$\ell_{VD}(obj)$	DDV_{VD}	Movimientos verticales permitidos para una estancia del concepto velocidad.
$\ell_{MOV}(obj)$	DDV_{MOV}	Desplazamientos globales permitidos para una estancia del concepto velocidad.

Tabla 5.4: Listado del resto de variables que se emplean para definir las restricciones espaciales del concepto velocidad normal.

t_1 , y (x_2, y_2) las coordenadas en un instante t_2 , los desplazamientos del objeto se calculan de la siguiente forma:

- **Desplazamiento horizontal:** $HD = \frac{\Delta x}{\Delta t} = \frac{|x_2 - x_1|}{t_2 - t_1}$
- **Desplazamiento vertical:** $VD = \frac{\Delta y}{\Delta t} = \frac{|y_2 - y_1|}{t_2 - t_1}$
- **Desplazamiento general:** $MOV = \frac{\sqrt{(\Delta x)^2 + (\Delta y)^2}}{\Delta t} = \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}{t_2 - t_1}$

5.4. Definición de restricciones

Las restricciones para definir el concepto de “velocidad normal” se construyen a partir de las variables presentadas anteriormente y permiten comprobar si un objeto en una escena se mueve a una velocidad normal o no. Las restricciones del componente son las siguientes:

Restricción de rol μ_{12} . Cada instancia del concepto velocidad normal tiene asociado una lista Υ donde se especifican los identificadores de las clases de objetos para los que la velocidad es adecuada. Sólo satisfacen la restricción aquellos objetos cuya clase pertenece a Υ . La función de pertenencia μ_{12} asociada a esta restricción se define de la siguiente manera:

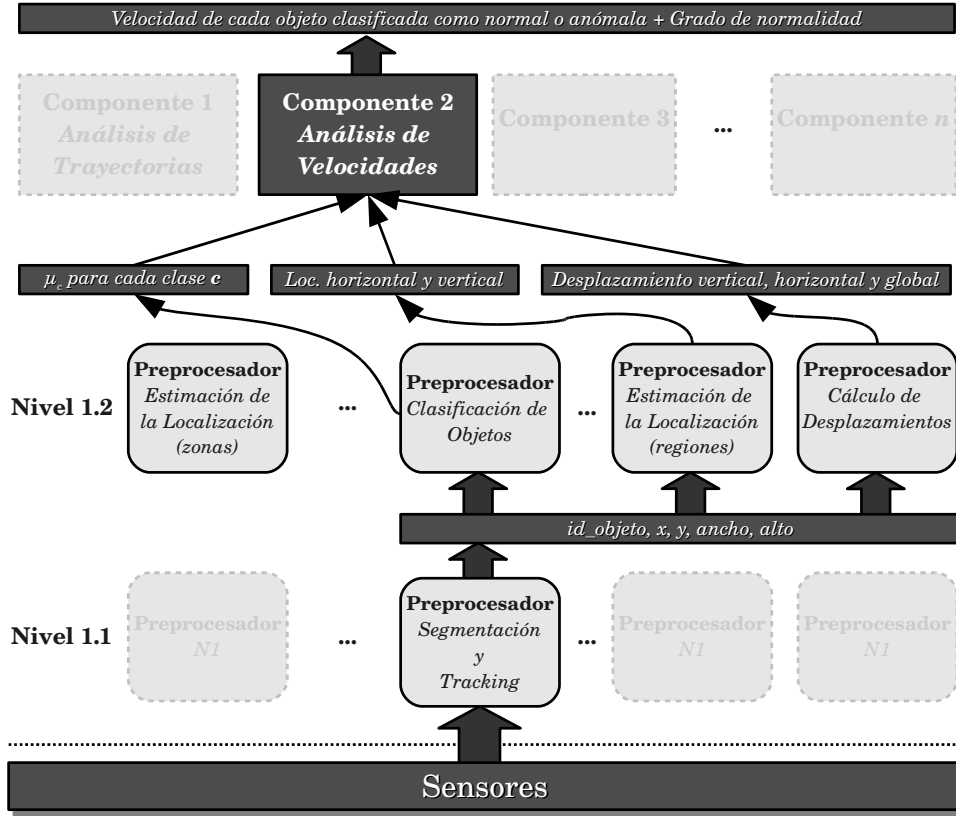


Figura 5.3: Conexión entre los preprocesadores y el componente de normalidad encargado de analizar las velocidades de los objetos.

$$\mu_{12}(obj) = \begin{cases} 1 & \text{if } v_c \text{ es un identificador de una clase } c \in \Upsilon; \\ 0 & \text{en caso contrario;} \end{cases}$$

tal que v_c representa el identificador de la clase cuyo $\mu_c(obj)$ sea máximo.

El resto de restricciones son restricciones espaciales y sirven para comprobar la localización y el desplazamiento del objeto.

Restricción de localización horizontal del objeto μ_{22} . Este tipo de restricción se emplea para comprobar si un objeto se encuentra en una determinada región horizontal de la imagen procedente de la cámara. Para ello, se tiene en cuenta el valor de la variable HP y se comprueba si el objeto se encuentra en alguna de las regiones indicadas en ℓ_{HP} . La variable ℓ_{HP} es en realidad un subconjunto del conjunto $\{VL: \text{very left}, L: \text{left}, C: \text{centre}, R: \text{right}, VR: \text{very right}\}$, es decir, $\ell_{HP} \subseteq DDV_{HP}$ (ver Tabla 5.1). El componente de normalidad c_2 , a partir del valor de la variable HP para un objeto obj , estudia la pertenencia a cada uno de los conjuntos en ℓ_{HP} . La suma de todos

estos valores de pertenencia dan como resultado el grado de satisfacción de la restricción μ_{22} :

$$\mu_{22}(obj) = \sum_{hz \in \ell_{HP}} \mu_{hz}(obj)$$

donde $\mu_{hz}(obj)$ es el valor de pertenencia del objeto obj a la región horizontal hz , siendo hz una región incluida en ℓ_{HP} . Un ejemplo visual del cálculo del grado de satisfacción de este tipo de restricciones se muestra en la Figura 5.4.

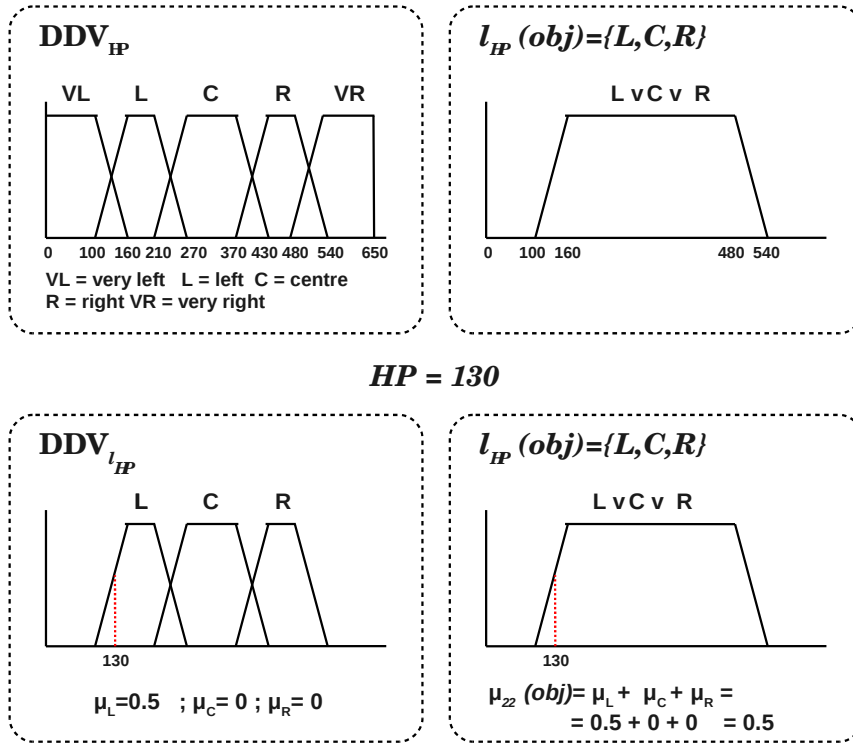


Figura 5.4: Ejemplo del cálculo del grado de satisfacción la restricción de localización horizontal de un objeto (μ_{22}), para $HP = 130$ y $\ell_{HP} = \{L, C, R\}$.

Restricción de la localización vertical del objeto μ_{32} . Este tipo de restricción pretende comprobar si el objeto se encuentra en una región vertical determinada y su definición es similar a la anterior. En este caso se analiza el valor de la variable VP para comprobar si el objeto se encuentra en alguna de las zonas indicadas en ℓ_{VP} . La variable ℓ_{VP} es un subconjunto del conjunto $\{VU: \text{very up}, U: \text{up}, C: \text{centre}, D: \text{down}, VD: \text{very down}\}$ (ver Tabla 5.1), es decir, $\ell_{VP} \subseteq DDV_{VP}$. La restricción μ_{32} se define de la siguiente manera:

$$\mu_{32}(obj) = \sum_{vz \in \ell_{VP}} \mu_{vz}(obj)$$

donde $\mu_{vz}(obj)$ es el valor de pertenencia del objeto obj a la zona vertical vz , siendo vz una zona incluida en ℓ_{VP} .

Restricción de desplazamiento horizontal de un objeto μ_{42} . Con esta restricción se pretende controlar que un objeto se desplace horizontalmente de manera normal por la escena. Para ello, la variable ℓ_{HD} engloba el conjunto de desplazamientos horizontales que un objeto puede realizar para circular a una determinada velocidad. Los elementos de este conjunto pertenecen al dominio $DDV_{HD} = \{VS: \text{very short}, S: \text{short}, MS: \text{medium short}, M: \text{medium}, ML: \text{medium large}, L: \text{large}, VL: \text{very large}\}$, es decir, $\ell_{HD} \in DDV_{HD}$. La variable HD contiene el valor del último desplazamiento del objeto en la componente horizontal. El componente c_2 estudia la pertenencia de este valor numérico a los conjuntos definidos en ℓ_{HD} , siendo el grado de satisfacción de la restricción la suma de todos los grados de pertenencia:

$$\mu_{42}(obj) = \sum_{hd \in \ell_{HD}} \mu_{hd}(obj)$$

donde $\mu_{hd}(obj)$ representa el valor de pertenencia del objeto obj a los conjuntos definidos en ℓ_{HD} .

Restricción de desplazamiento vertical de un objeto μ_{52} . A diferencia de la restricción anterior, en este caso se pretende comprobar que el objeto se desplace verticalmente de una forma normal. La variable ℓ_{VD} engloba el conjunto de desplazamientos verticales que un objeto puede realizar para circular a una determinada velocidad. Los elementos de este conjunto pertenecen al dominio DDV_{VD} , es decir, $\ell_{VD} \subseteq DDV_{VD}$. La variable VD contiene el valor del último desplazamiento vertical realizado por el objeto. La restricción μ_{52} se define de la siguiente manera:

$$\mu_{52}(obj) = \sum_{vd \in \ell_{VD}} \mu_{vd}(obj)$$

donde $\mu_{vd}(obj)$ representa el valor de pertenencia del objeto obj a los conjuntos definidos en ℓ_{VD} .

Restricción de desplazamiento global de un objeto μ_{62} . Con este tipo de restricción se pretende controlar que los desplazamientos globales de un objeto en la escena sean adecuados, teniendo en cuenta ambos tipos de desplazamientos: vertical y horizontal. Esta restricción se construye en base a la variable MOV que contiene el valor del desplazamiento global de un objeto entre instantes de tiempo consecutivos. Para comprobar si se

satisface la restricción se estudia la pertenencia del valor de MOV a los conjuntos incluidos en $\ell_{MOV}(obj)$, los cuales pertenecen a su vez a DDV_{MOV} , y determinan los desplazamientos globales permitidos. Al igual que sucede en las restricciones anteriores, el grado de satisfacción corresponderá con la suma de todos los grados de pertenencia del valor de la variable MOV a los conjuntos definidos ℓ_{MOV} :

$$\mu_{62}(obj) = \sum_{mov \in \ell_{MOV}} \mu_{mov}(obj)$$

donde $\mu_{mov}(obj)$ representa el valor de pertenencia del objeto obj a los conjuntos definidos en ℓ_{MOV} .

Una vez definidas las restricciones de forma general es posible instanciar el componente para entornos particulares. En la siguiente Sección de describirá la forma en la que se particulariza el concepto “velocidad normal” y, a modo de ejemplo, se mostrarán algunas de las particularizaciones para la misma escena en la que se desplegó el componente de trayectorias (Sección 4.5).

5.5. Particularizaciones del concepto velocidad normal

El componente de normalidad encargado de analizar las velocidades, al igual que cualquier otro componente, necesita conocimiento específico del entorno que pretende analizar, para su uso en un problema concreto. En este caso particular, el componente necesitará conocer cuáles son los desplazamientos permitidos en base a la clase a la que pertenece un objeto, la zona o región de la imagen en la que se encuentra y su desplazamiento entre instantes de tiempo. Una vez especificadas las variables y las restricciones necesarias para definir el concepto o aspecto a vigilar, el siguiente paso es asignar valores particulares a estas variables y restricciones para la construcción de instancias. En esta Sección se expone cómo se hace uso del algoritmo de aprendizaje¹ descrito en la Sección 3.4.1 para aprender el conjunto de instancias o particularizaciones del concepto velocidad normal en un entorno concreto.

La Tabla 5.5 muestra un ejemplo de instanciación del concepto velocidad normal en base a las variables y restricciones definidas en las Secciones 5.2 y

¹El algoritmo de aprendizaje automático y su aplicación a un problema concreto, como es la clasificación de objetos, ya ha sido descrito detalladamente en las Secciones 3.4.1 y 4.3.3, por este motivo la descripción sobre la forma en la que se aplica el algoritmo al aprendizaje de instancias del concepto “velocidad normal”, tendrá un nivel de detalle inferior al de las secciones anteriores.

5.5. Particularizaciones del concepto velocidad normal

| 171 |

5.4, respectivamente. Según esta instancia del concepto “velocidad normal”, un objeto que sea un vehículo, esté situado en el centro (C) o la derecha (R), y muy abajo (VD) o abajo (D), realice un desplazamiento muy corto (VS), corto (S) o medio-corto (MS) en el eje horizontal, muy corto (VS) en el vertical y, finalmente, su desplazamiento global sea muy corto (VS), corto (S) o medio-corto (MS), entonces su velocidad es normal para el entorno que el sistema de vigilancia está observando.

$\tilde{\mu}_{12}$	$\tilde{\mu}_{22}$	$\tilde{\mu}_{32}$	$\tilde{\mu}_{42}$	$\tilde{\mu}_{52}$	$\tilde{\mu}_{62}$
$\Upsilon = \{\text{vehículo}\}$	$\ell_{HP} = \{C, R\}$	$\ell_{VP} = \{VD, D\}$	$\ell_{HD} = \{VS, S, MS\}$	$\ell_{VD} = \{VS\}$	$\ell_{MOV} = \{VS, S, MS\}$

Tabla 5.5: Ejemplo de instancia de velocidad normal en base a las restricciones definidas en la sección.

En la Tabla anterior cada una de las columnas representa una instancia de una restricción donde las variables toman valores específicos de sus dominios de definición. Para que un objeto mantenga esa velocidad es necesario que satisfaga absolutamente todas las instancias de restricciones definidas. Es decir, es posible afirmar que las instancias de las restricciones están relacionadas o conectadas mediante operadores y-lógicos. Por otro lado, si se tiene en cuenta que el algoritmo de aprendizaje descrito en la Sección 3.4.1 genera reglas del tipo **SI** v_0 es $ZD_0 \wedge \dots \wedge v_n$ es ZD_n **ENTONCES** y_j , es posible apreciar una correspondencia directa entre cada una de la instancias del concepto velocidad normal y las reglas. En concreto, la regla equivalente a la instancia de la Tabla 5.5 sería la siguiente:

SI $\Upsilon = \text{es } \{\text{vehículo}\} \wedge \ell_{HP} \text{ es } \{C, R\} \wedge \ell_{VP} \text{ es } \{VD, D\} \wedge \ell_{HD} \text{ es } \{VS, S, MS\} \wedge \ell_{VD} \text{ es } \{VS\} \wedge \ell_{MOV} \text{ es } \{VS, S, MS\}$ **ENTONCES** y_j es *velocidad normal*.

Por tanto, el algoritmo de aprendizaje automático que se emplea para la clasificación de objetos, también es apropiado para aprender el conjunto de instancias del concepto velocidad. Uno de los motivos por los cuales el algoritmo de aprendizaje se adapta correctamente al modelo de análisis de normalidad, es el hecho de haber definido las restricciones como conjuntos difusos. Cada una de las partes que componen el antecedente de una regla es en realidad un conjunto difuso y, a su vez, una restricción que deben satisfacer los objetos. Por ejemplo, en la regla anterior, “ ℓ_{HP} es $\{C, R\}$ ” representa el conjunto difuso al que pertenecen todos aquellos elementos cuya posición horizontal HP está localizada en el centro (C) o a la derecha (R) de la imagen procedente de la cámara de seguridad. La función de pertenencia asociada al conjunto difuso $C \vee R$, $\mu_{C \vee R}$, determinará el grado de pertenencia de un elemento a dicho conjunto, o lo que es lo mismo, el grado de satisfacción de la restricción espacial relativa a la posición horizontal. Finalmente, la suma de los grados de pertenencia a los conjuntos que forman parte del antecedente de una regla determinará el valor de satisfacción global de la regla. Esta forma de proceder es exactamente la misma que se indicó en el modelo de normalidad del Capítulo 3 para calcular el grado de normalidad asociado a una instancia de un concepto (ver Sección 3.2.7). Por tanto,

siempre que el movimiento de un objeto provoque la activación de alguna de las reglas querrá decir que el objeto satisface la normalidad de alguna de las instancias del concepto y, por consiguiente, circulará a una velocidad normal.

En cuanto a los pasos a seguir para la generación de las reglas o particularizaciones del concepto velocidad normal, son exactamente los mismos que se describieron de forma general en la Sección 3.4.1 y en 4.3.3, donde se aplicó el algoritmo a la clasificación de objetos detectados por los procesos de segmentación y *tracking*. En primer lugar es necesario construir los dominios de definición de las variables que formarán parte del antecedente de las reglas. Como ejemplo se muestran en las Tablas 5.1, 5.2 y 5.4 los dominios de definición para las variables que representan la posición horizontal de un objeto, posición vertical, clase a la que pertenece, desplazamiento horizontal, vertical y global, para la escena descrita en la Sección 4.5.

Una vez definidos los dominios de definición de las variables, el segundo paso consiste en la construcción del conjunto de entrenamiento donde los ejemplos están divididos en dos posibles clases: y_1 (velocidad normal) e y_2 (velocidad anómala). La máquina de aprendizaje generará reglas para ambas clases, pero como se comentó anteriormente, este trabajo se centra principalmente en el análisis de normalidad, de esta forma, el componente encargado de analizar las velocidades de los objetos tan sólo utilizará las reglas generadas para la clase y_1 (velocidad normal); la ausencia de activaciones de reglas implicará que un objeto no está siguiendo una velocidad normal. Cada uno de los ejemplos es una tupla formada por los siguientes elementos $((x, y, v_c, hd, vd, mov), y_j)$, donde x representa la posición horizontal, y la posición vertical, v_c el identificador de la clase a la que pertenece el objeto, es decir:

$$\max_{c \in O} \{\mu_{v_c}(obj)\} \quad (5.2)$$

y que será representada en los ejemplos con un identificador de clase i que será distinto para cada clase ($i = 0 \dots |O| - 1$), hd desplazamiento horizontal, vd desplazamiento vertical y, finalmente, mov desplazamiento global. En la Tabla 5.6 se muestran algunos de los ejemplos que forman parte del conjunto de entrenamiento.

La definición de los dominios de las variables y la construcción del conjunto de entrenamiento son dos pasos previos necesarios para comenzar el proceso de aprendizaje, en donde cada uno de los ejemplos es etiquetado y los valores numéricos son sustituidos por valores lingüísticos. La Tabla 5.7 muestra los ejemplos de la Tabla 5.6, los cuales representan un conjunto de reglas iniciales.

Una vez etiquetados los ejemplos del conjunto de entrenamiento, éstos

Velocidad normal	Velocidad Anómala
((388,324,0,19.0,4.0,19.41), y_1)	((19,371,1,131.0,47.0,139.17), y_2)
((386,332,0,10.0,0.0,10.0), y_1)	((564,292,1,168.0,14.0,168.58), y_2)
((186,307,1,42.0,3.0,42.10), y_1)	((568,261,1,126.0,3.0,126.03), y_2)
((114,340,1,90.0,31.0,95.18), y_1)	((497,283,0,90.0,5.0,90.13), y_2)
((200,250,1,52.0,12.0,53.36), y_1)	((492,253,0,75.0,11.0,75.80), y_2)

Tabla 5.6: Algunos ejemplos de elementos que forman parte del conjunto de entrenamiento empleado para la generación de reglas difusas que permiten clasificar la velocidad de los objetos detectados. En este caso sólo se emplean dos clases, peatón y vehículo, cuyos identificadores son 0 y 1 respectivamente.

Velocidad normal	Velocidad Anómala
((C,D,Pedestrian,MS,S,MS), y_1)	((VL,D,Vehicle,VL,M,VL), y_2)
((C,D,Pedestrian,S,VS,S), y_1)	((VR,C,Vehicle,VL,MS,VL), y_2)
((L,C,Vehicle,M,VS,M), y_1)	((VR,C,Vehicle,VL,VS,VL), y_2)
((VL,D,Vehicle,ML,MS,L), y_1)	((R,C,Pedestrian,ML,S,ML), y_2)
((L,C,Vehicle,M,MS,M), y_1)	((R,C,Pedestrian,ML,S,ML), y_2)

Tabla 5.7: Conjunto de ejemplos de la Tabla 5.6 etiquetados. Cada uno de ellos representa reglas iniciales a partir de las cuales se construyen reglas generales.

son agrupados y ordenados según el número de repeticiones, es decir, todos aquellos ejemplos que sean etiquetados de la misma forma representan la misma regla inicial. Finalmente, comienza el proceso de ampliación en el que las reglas iniciales se convierten en reglas generales y pasan a formar parte del conjunto definitivo de reglas. La Tabla 5.8 muestra algunas de las reglas obtenidas con el algoritmo de aprendizaje automático para la clasificación de velocidades en la escena concreta presentada en la Sección 4.5. El primer bloque de reglas representa instancias del concepto “velocidad normal”; otra forma de representarlas es tal como se muestra en la Tabla 5.9.

Finalmente, la Figura 5.5 resumen en un esquema los pasos necesarios para la obtención del conjunto de reglas finales que han sido descritos anteriormente. En el Anexo A, se puede encontrar una descripción de las herramientas de adquisición de conocimiento que han sido desarrolladas para agilizar la instanciación del componente descrito en este Capítulo.

5.6. Consideraciones finales

La clasificación de la velocidad de un objeto como normal o anómala, al igual que las trayectorias, también es posible a partir de la información espacial obtenida como consecuencia del análisis de imágenes procedentes de las cámaras. En este Capítulo se ha demostrado como el modelo formal planteado en el Capítulo 3 se puede adaptar correctamente a este problema. Dicha adaptación ofrece una serie de ventajas frente a otros métodos de

Velocidades normales
R_1 : SI y es {VD, D} $\wedge$ vd es {VS, S} ENTONCES velocidad normal
R_2 : SI c es {Vehicle} $\wedge$ mov no es {L, VL} ENTONCES velocidad normal
R_3 : SI x no es {VR} $\wedge$ vd es {VS} ENTONCES velocidad normal
R_4 : SI x es {R, VR} $\wedge$ c es {Vehicle} $\wedge$ vd es {S}
ENTONCES velocidad normal
R_5 : SI c es {Vehicle} $\wedge$ hd no es {L, VL} ENTONCES es velocidad normal
Velocidades anómalas
R_6 : SI y es {D, VD} $\wedge$ hd es {L, VL} ENTONCES velocidad anómala
R_7 : SI x no es {D, VD} $\wedge$ c es {Pedestrian} $\wedge$ mov no es {VS, S, SM}
ENTONCES velocidad anómala
R_8 : SI x es {VL, L} $\wedge$ y no es {D, VD} $\wedge$ mov es {L, VL}
ENTONCES velocidad anómala
R_9 : SI x es {C} $\wedge$ y es {C} $\wedge$ c es {Vehicle} $\wedge$ hd es {L} $\wedge$ vd es {S} $\wedge$
mov es {L} ENTONCES velocidad anómala
R_{10} : SI x es {VL} $\wedge$ y es {D} $\wedge$ c es {Human} $\wedge$ hd es {LM} $\wedge$ vd es {SM} $\wedge$
mov es {LM} ENTONCES velocidad anómala

Tabla 5.8: Ejemplos de reglas difusas para la clasificación de velocidades.

$c_{2,y}^1$	$\tilde{\mu}_{12}$	$\tilde{\mu}_{22}$	$\tilde{\mu}_{32}$	$\tilde{\mu}_{42}$	$\tilde{\mu}_{52}$	$\tilde{\mu}_{62}$
$c_{2,1}^1$	$\Upsilon = \{\emptyset\}$	$\ell_{HP} = \{\emptyset\}$	$\ell_{VP} = \{VD, D\}$	$\ell_{HD} = \{\emptyset\}$	$\ell_{VD} = \{VS, S\}$	$\ell_{MOV} = \{\emptyset\}$
$c_{2,2}^1$	$\Upsilon = \{Vehicle\}$	$\ell_{HP} = \{\emptyset\}$	$\ell_{VP} = \{\emptyset\}$	$\ell_{HD} = \{\emptyset\}$	$\ell_{VD} = \{\emptyset\}$	$\ell_{MOV} = \{VS, S, MS, M, ML\}$
$c_{2,3}^1$	$\Upsilon = \{\emptyset\}$	$\ell_{HP} = \{VL, L, C, R\}$	$\ell_{VP} = \{\emptyset\}$	$\ell_{HD} = \{\emptyset\}$	$\ell_{VD} = \{VS\}$	$\ell_{MOV} = \{\emptyset\}$
$c_{2,4}^1$	$\Upsilon = \{Vehicle\}$	$\ell_{HP} = \{R, VR\}$	$\ell_{VP} = \{\emptyset\}$	$\ell_{HD} = \{\emptyset\}$	$\ell_{VD} = \{VS\}$	$\ell_{MOV} = \{\emptyset\}$
$c_{2,5}^1$	$\Upsilon = \{Vehicle\}$	$\ell_{HP} = \{\emptyset\}$	$\ell_{VP} = \{\emptyset\}$	$\ell_{HD} = \{VS, S, MS, M, ML\}$	$\ell_{VD} = \{\emptyset\}$	$\ell_{MOV} = \{\emptyset\}$

Tabla 5.9: Ejemplo de instancias del concepto velocidad normal aprendidas mediante el algoritmo de aprendizaje automático, para la escena descrita en la Sección 4.5.

estimación de velocidades:

- Interpretabilidad de las instancias del concepto “velocidad normal” y, en consecuencia, facilidad para mantener la base de conocimiento.
- Reutilización de componente para distintos entornos y tipos de objetos. Este proceso se automatiza en gran medida con la máquina de aprendizaje propuesta para la particularización de velocidades.
- Número reducido de ejemplos en el conjunto de entrenamiento. Para generar una regla que represente una determinada situación, tan sólo es necesario un ejemplo de dicha situación.
- Escalabilidad en la definición del concepto velocidad ante la posibilidad de incluir nuevos tipos de restricciones o reducir los existentes.
- Velocidad en la obtención de los resultados.

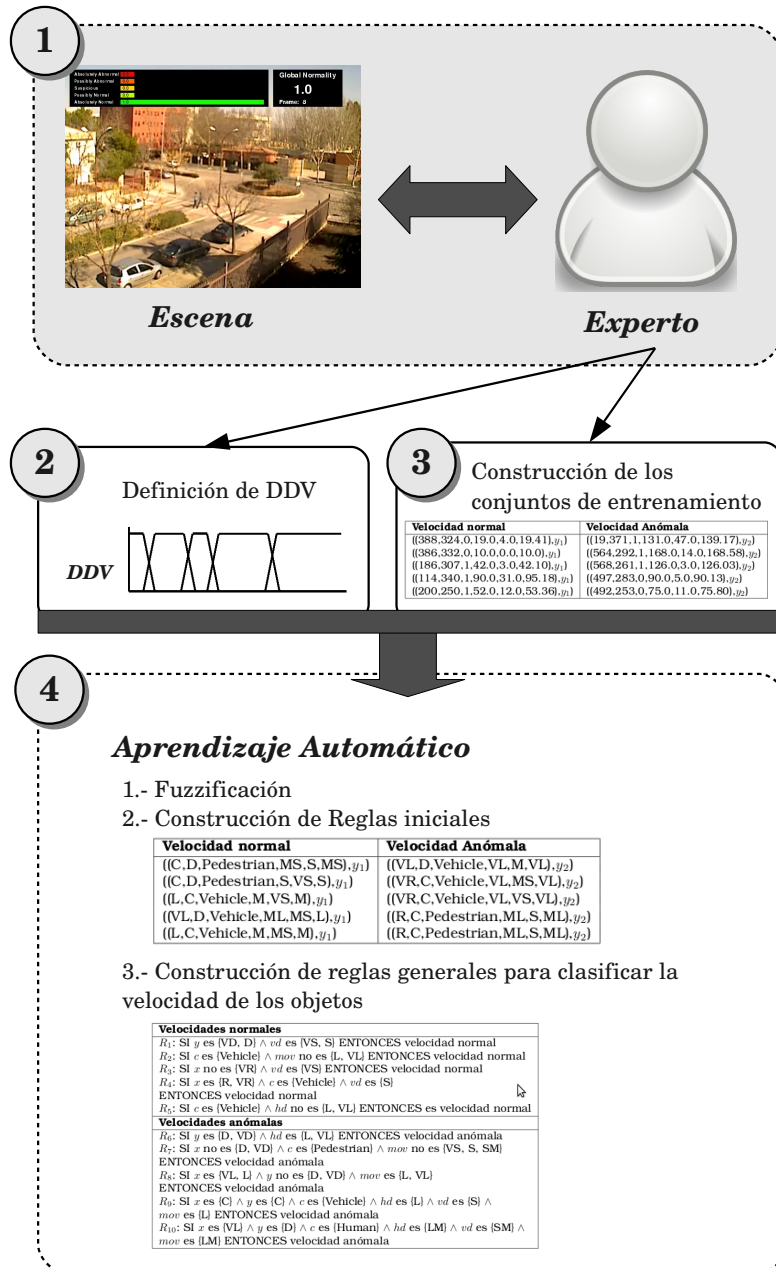


Figura 5.5: Secuencia de pasos para obtener el conjunto de reglas difusas que clasifican la velocidad de los objetos como normales o anómalas.

- Bajo coste económico (tan sólo es necesario una cámara).
- No es necesario conocer a priori los parámetros de la cámara.

En el Capítulo siguiente se discutirán los resultados obtenidos para este componente, así como las principales ventajas e inconvenientes encontrados en las pruebas realizadas.

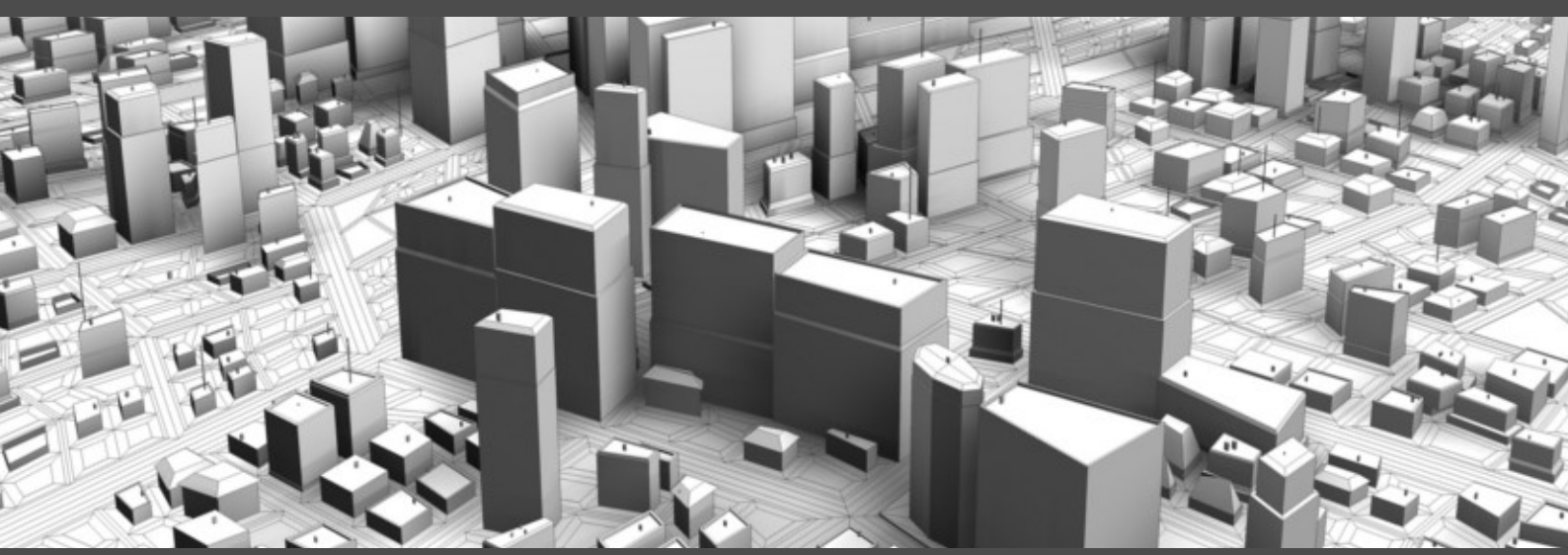


Capítulo

Resultados

6

Capítulo 6



6

Resultados

La propuesta de un nuevo modelo, técnica o método para resolver un determinado problema implica su formalización, aplicación y, finalmente, evaluación para demostrar su validez. En este trabajo se definió un modelo formal para la construcción de componentes que analizan la normalidad de un entorno de acuerdo a un aspecto a vigilar (Capítulo 3), y se puso en práctica con la creación de dos componentes para el análisis de trayectorias y velocidades, descritos en los Capítulos 4 y 5, respectivamente. En este Capítulo se muestran y discuten los resultados obtenidos en el análisis de un entorno urbano real, los cuales demuestran la validez tanto del modelo como de los componentes diseñados en base a dicho modelo.

6.1. Descripción de los experimentos

A la hora de evaluar tanto el modelo formal como los componentes para el análisis de trayectorias y velocidades, se valoraron dos posibilidades: emplear varios entornos en los experimentos o centrarse en uno en concreto. La primera de ellas se descartó ante la dificultad e incompatibilidad legal de ubicar cámaras de vigilancia en lugares públicos, por lo que finalmente se optó por la segunda opción. Una de las ventajas de centrarse en un entorno en concreto, es la posibilidad de encontrar situaciones críticas para someter al sistema a duras pruebas de evaluación; situaciones que serían realmente complicadas de encontrar si no se dedicara un tiempo prudencial al estudio de un entorno.

El entorno elegido para desarrollar las pruebas ya se describió en el Capítulo 4 (Figura 4.9), en concreto en la Sección 4.5 donde se particularizó el concepto de trayectoria para dicho entorno. Se trata de un entorno urbano clásico donde los vehículos y los peatones están sometidos a unas normas de circulación, entre las que se incluyen trayectorias y velocidades a las

que deben circular. Por este motivo, se considera que el entorno elegido es apropiado para evaluar las nuevas propuestas realizadas en este trabajo.

Para la obtención de material visual se ubicó una cámara modelo AXIS IP 2007w en uno de los laboratorios de la Escuela Superior de Informática de la Universidad de Castilla la Mancha. La calidad de los frames obtenidos en los procesos de grabación es suficiente para que el sistema pueda detectar objetos en movimiento y clasificarlos como personas y vehículos, pero no para identificarlos. Es decir, en ningún momento el sistema es conocedor de la identidad de las personas y vehículos detectados. Por otro lado, también se ha empleado un computador para el procesamiento de imágenes y el análisis de comportamientos, cuyas características más relevantes son: CORE 2 DUO 3 GHZ, 3GHZ de memoria RAM y 1 TeraByte de espacio en disco duro.

Es importante recalcar que las pruebas no se han realizado sobre grabaciones en vivo ya que los algoritmos de segmentación y *tracking* empleados en esta Tesis no trabajan en tiempo real. Para la obtención de las secuencias de vídeo se han realizado diferentes grabaciones del mismo entorno en distintos intervalos horarios y con diferentes condiciones de iluminación. A partir del material obtenido se ha realizado una selección de diez escenas de vídeo en la que suceden situaciones interesantes a analizar con las que poner a prueba al sistema de seguridad. El hecho de no haber realizado las pruebas sobre flujo de vídeo en tiempo real no quiere decir que los métodos propuestos en este trabajo (independientes de cualquier algoritmo de segmentación y *tracking*) no puedan trabajar en tiempo real, tal como se demostrará en la Sección 6.2-Tabla 6.10 donde se muestran los tiempos invertidos en cada uno de los procesos que forman parte del análisis de normalidad.

El principal objetivo de los experimentos realizados es la evaluación exclusiva de los métodos propuestos para la clasificación de objetos, así como del análisis de la normalidad en base a las trayectorias y las velocidades. Por este motivo, aquellos objetos en movimiento que no han sido detectados por los algoritmos de segmentación y *tracking*, aún siendo un error global del sistema de vigilancia, no han quedado reflejados en los resultados expuestos en este Capítulo. En otras palabras, los componentes de normalidad no pueden analizar el comportamiento de aquellos objetos de los que se desconoce su existencia y, por tanto, se tratan de errores propios de los algoritmos empleados en las capas inferiores para la obtención de información espacio-temporal.

Antes de explicar la forma en la que se han evaluado los métodos propuestos en este trabajo, la Tabla 6.1 resume las situaciones de mayor interés en cada una de las escenas seleccionadas, así como el intervalo de frames (marcado entre paréntesis) en los que transcurren. Estas situaciones justifican la elección de las escenas en el proceso experimental y se pueden agrupar en ocho tipos, los cuales se describen a continuación:

- Vehículos que circulan a una velocidad excesiva.
- Peatón o peatones que circulan por la carretera en lugar de utilizar zonas habilitadas para ellos, como por ejemplo las aceras o los pasos de peatones.
- Motocicletas y personas montadas sobre bicicletas. Esta es una situación crítica para el sistema de clasificación propuesto en este trabajo ya que éste se basa en las proporciones de los objetos, la velocidad a la que se mueven y las zonas en las que se encuentren. Las proporciones de una persona montada en un vehículo de dos ruedas se encuentran en un punto medio entre las de un vehículo de cuatro ruedas y los peatones, de ahí la dificultad del sistema a la hora de clasificar este tipo de objetos. La incertidumbre es mucho menor cuando el vehículo de dos ruedas se mueve a una velocidad imposible de alcanzar para una persona que camine a pie, en este caso el sistema clasificaría al objeto como un vehículo.
- Grupos de personas caminando a la misma velocidad y muy próximos entre sí. Si el sistema detecta al grupo como un objeto único podría creer en ciertos momentos que se trata de un vehículo.
- Vehículos de grandes dimensiones. La aparición de vehículos de grandes dimensiones en la escena siempre es de gran interés para observar la respuesta del sistema, ya que en ciertas ocasiones pueden ser detectados como varios objetos de menor tamaño.
- Cambios de sentido y giros completos realizados en circunvalaciones. En concreto, en la rotonda que se encuentra en el entorno elegido para realizar las pruebas, existen un conjunto de árboles que provocan oclusiones parciales y absolutas cuando el sistema sigue a los vehículos.
- Cambio de elipses entre objetos próximos entre sí. En algunas ocasiones, cuando un objeto en movimiento se encuentra próximo a un segundo objeto, puede suceder un intercambio de elipses entre los objetos, o bien, si uno de ellos no había sido detectado previamente podría tomar la elipse del objeto que sí había sido detectado. Esta situación es crítica cuando los objetos pertenecen a diferentes clases, por ejemplo, un peatón que toma la elipse de un vehículo. Dadas las condiciones del algoritmo de clasificación descrito en este trabajo, donde se tiene en cuenta la forma en la que se ha clasificado un objeto en los frames anteriores, es muy posible que durante cierto tiempo siga creyendo que se trata de un vehículo.
- Objetos que no son personas o vehículos que se mueven a causa del viento.

Escena	Situaciones relevantes
Escena 1	■ Vehículo que circula con exceso de velocidad (837-881) ■ Grupo de dos peatones que no siguen una trayectoria normal al cruzar por la carretera en lugar de utilizar los pasos de peatones (1603-1843) ■ Cuatro personas andando en linea que el sistema detecta como vehículo (2100-2600) ■ Vehículo de grandes dimensiones detectado como varios objetos de menor tamaño (3060-3290)
Escena 2	■ Motocicleta circulando con exceso de velocidad (110-201) ■ Bicicleta detectada como peatón (580-650) ■ Vehículo que circula con exceso de velocidad (705-732)(2498-2529) ■ Cambio de elipse entre dos objetos (2529-2585)
Escena 3	■ Tres vehículos que circulan con exceso de velocidad (545-661) ■ Vehículo que realiza una vuelta completa a la rotonda (672-1030) ■ Vehículo que circula con exceso de velocidad (985-1034)(1750-1782)(1950-2046)(2266-2284)(2291-2309) ■ La elipse de un coche pasa a formar parte de una persona (1261-1295) ■ Peatón circulando por la carretera (1439-1531) ■ Motocicleta con exceso de velocidad (2869-2957)

6.1. Descripción de los experimentos

| 183 |

Escena 4	■ Vehículo que cambia de sentido en la rotonda (316-491) ■ Peatones andando por la carretera (790-847) ■ Vehículo con exceso de velocidad (1620-1674)(1888-1950)(2480-2566)(2619-2691) ■ Vehículo de grandes dimensiones (2300-2560) ■ Peatón que se apropia de la elipse con la que se ha marcado a un vehículo previamente (2740-2766) ■ Cambio de sentido de un vehículo en la rotonda (2783-3023)
Escena 5	■ Objetos de tamaño reducido (bolsas de plástico y papeles) revoloteando en una zona cercana a la rotonda a causa del aire (1-410). ■ Vehículo que cambia de sentido en la rotonda (481-749)
Escena 6	■ Vehículo que circula con exceso de velocidad(176-225)(1352-1396) ■ Peatón en bicicleta haciendo acrobacias por la carretera (760-846) ■ Vehículo que cambia de sentido en la rotonda (916-1100)

Escena 7	■ Vehículo que cambia de sentido en la rotonda (1-168) ■ Vehículo de pequeñas dimensiones toma elipse de vehículo de mayor tamaño al cruzarse (672-710) ■ Cambio de elipse entre peatón y vehículo (729-737) ■ Vehículo de grandes dimensiones (1788-2040) ■ Vehículo que circula con exceso de velocidad (2801-2835) ■ Un peatón que aparece en la escena toma la elipse de otro detectado previamente (4115-4140) ■ Motocicleta que circula a una velocidad normal(4542-4657)
Escena 8	■ Persona que sale de un vehículo en el lado de la carretera (750-770). ■ Peatón circulando por la carretera durante un tiempo prolongado(1064-1250) ■ Vehículo de grandes dimensiones circulando a gran velocidad (1580-1906)
Escena 9	■ Vehículo que circula con exceso de velocidad (138-202) ■ Grupos numerosos de personas (923-1340)(1550-2070) ■ Peatón andando por la carretera (951-1189) ■ Un vehículo toma la elipse de un peatón (1720-1749) ■ Vehículos de grandes dimensiones (1700-1960) ■ Vehículo de dimensiones intermedias circulando con exceso de velocidad (1932- 2046)

Escena 10	■ Dos vehículos que circulan con exceso de velocidad (316-413) ■ Vehículo que adquiere la elipse asignada previamente a un grupo de personas (540-572) ■ Grupos de Personas andando por la acera (700-772)

Tabla 6.1: Situaciones relevantes en cada una de las escenas elegidas para probar el sistema.

6.1.1. Evaluación de los métodos propuestos

Los experimentos realizados en este trabajo están enfocados a la prueba y evaluación de los diferentes módulos que componen el sistema. Cada uno de los objetos detectados en un *frame* se considera como una situación a analizar y por cada uno de ellos se evalúa si el sistema ha determinado correctamente la clase a la que pertenece, la zona/as donde se encuentra, trayectoria/as que está siguiendo, clasificación de la velocidad como normal o anómala y valoración global de su comportamiento. La evaluación de cada uno de los módulos se realiza de manera manual *frame* a *frame*, tarea enormemente tediosa pero necesaria a la vez ante la inexistencia de un método automático que determine si el sistema artificial ha realizado una interpretación correcta.

En primer lugar, para evaluar el algoritmo de aprendizaje se ha tenido en cuenta el tiempo invertido en el proceso de aprendizaje y el número de reglas generadas. La calidad de las reglas se ha valorado en función del porcentaje de acierto en los procesos de clasificación llevados a cabo por el preprocesador que determina la clase de los objetos, y el segundo componente de normalidad donde se clasifican las velocidades de los objetos como normales o anómalas. Ambos módulos emplean reglas difusas generadas por el algoritmo de aprendizaje.

En el preprocesador encargado de clasificar los objetos detectados en la escena, además de contabilizar los aciertos, se han contabilizado los fallos producidos en el proceso de clasificación y se han organizado de acuerdo a cuatro posibles causas: i) activación incorrecta de las reglas (se activan reglas de clases a las que no pertenece el objeto), ii) no se produce la activación de ninguna regla, iii) clasificación incorrecta debido al peso que tienen las clasificaciones anteriores (Histórico): si un objeto ha sido clasificado incorrectamente como vehículo durante varios frames y, a partir de un momento dado sólo se produce la activación de reglas pertenecientes a la clase peatón, el sistema podría seguir clasificándolo como vehículo durante

un tiempo al tener en cuenta las clasificaciones anteriores. iv) Finalmente, también se pueden producir errores por cambios de elipses entre objetos de diferentes clases.

De la misma forma, para evaluar el preprocesador encargado de estimar la zona/as en las que se encuentran los objetos se han tenido en cuenta tanto los aciertos como los fallos y sus causas. En este caso particular, las causas de error que pueden llevar al sistema a creer que el objeto se encuentra en una zona donde realmente no está, son las siguientes: i) dimensiones y posición de la elipse incorrectas, por ejemplo, una elipse que marca únicamente la parte superior de un peatón. Es importante recordar que la localización de un objeto se estimaba en base a sus puntos de apoyo y, por tanto, si la parte inferior de la elipse no envuelve los puntos de apoyo reales, el sistema podría estimar incorrectamente su localización. ii) La segunda causa de error se debe a que un objeto puede ser detectado como varios objetos de menor tamaño, parte de estas elipses pueden envolver puntos que no pertenecen a los de apoyo.

Siguiendo la misma tónica, la evaluación de los componentes de normalidad y el análisis global también se ha realizado en base al número de aciertos en las interpretaciones y el número de fallos distribuidos de acuerdo a diferentes causas. En el caso del componente de normalidad encargado de analizar las trayectorias las posibles causas de error son las siguientes: i) clasificación incorrecta del objeto y, en consecuencia, incumplimiento de las restricciones de rol, ii) posición incorrecta de la elipse e incumplimiento de las restricciones espaciales y, finalmente, iii) intercambio de elipses entre objetos de distintas clases, en este caso ambos tipos de restricciones se pueden incumplir. Por otro lado, el segundo componente de normalidad encargado de analizar las velocidades tiene algunas causas de error en común con el primero, sus causas particulares son las siguientes: i) clasificación incorrecta de objetos, ii) cambios bruscos en la posición de la elipse que envuelve al objeto, dando lugar a que el sistema crea que el desplazamiento ha sido mayor del real, iii) activación incorrecta de las reglas utilizadas para clasificar las velocidades, iv) cambios de elipses entre objetos de distintas clases, por ejemplo, un vehículo que adquiriera la elipse de un peatón llevaría al sistema a creer fácilmente que su velocidad es excesiva.

Los aciertos y fallos de los componentes de normalidad y el análisis global se pueden clasificar como verdaderos positivos (VP), verdaderos negativos (VN), falsos positivos (FP) y falsos negativos (FN), cada uno de ellos representa la siguiente situación:

- **Verdadero Positivo (VP):** situación normal clasificada correctamente como normal.
- **Verdadero Negativo (VN):** situación anómala clasificada correctamente como anómala.
- **Falso Positivo (FP):** situación anómala clasificada incorrectamente

como normal.

- **Falso Negativo (FN):** situación normal clasificada incorrectamente como anómala.

Además, por cada una de las escenas se ha determinado la relación entre estos parámetros con el cálculo de los siguientes coeficientes [72]:

- Sensibilidad o razón de verdaderos positivos (VPR): $VPR = VP/(VP + FN)$, o lo que es lo mismo, razón de éxitos.
- Razón de Falsos Positivos o especificidad (FPR): $FPR = FP/FP + VN$
- Precisión o exactitud (ACC): $ACC = (VP + VN)/(P + N)$, donde P es el número de casos positivos y N el caso de números negativos.
- Especificidad (SPC) o Razón de Verdaderos Negativos: $SPC = VN/N = VN/(FP + VN) = 1 - FPR$
- Valor Predictivo Positivo (PPV): $PPV = VP/(VP + FP)$
- Valor Predictivo Negativo (NPV): $NPV = VN/(VN + FN)$
- Razón de Falsos Descubrimientos(FDR): $FDR = FP/(FP + VP)$

Finalmente, se ha evaluado la eficiencia en términos de tiempo para cada uno de los módulos mencionados anteriormente. Para cada uno de ellos se ha contabilizado el tiempo total invertido en cada una de las escenas y el tiempo medio invertido por cada uno de los objetos detectados, ambos medidos en milisegundos.

6.2. Discusión de los resultados

En esta Sección se mostrarán y discutirán los resultados en base a los criterios presentados en la Sección anterior.

6.2.1. Clasificación de objetos

La Tabla 6.2 muestra el porcentaje de acierto en el proceso de clasificación de objetos para cada una de las diez escenas empleadas en las pruebas. El valor medio es un 95 %, siendo 99 % el valor más alto obtenido y 91 % el más bajo; estos valores suponen un porcentaje alto de acierto. Las tres primeras causas de error son propias del algoritmo de clasificación diseñado en este trabajo, mientras que la cuarta es propia de los algoritmos de visión

empleados en las capas inferiores y difícilmente pueden ser evitados por el algoritmo de clasificación. Precisamente, los algoritmos de visión encargados de detectar objetos y marcarlos mediante elipses no son perfectos y, en multitud de ocasiones, estas elipses no envuelven completamente a los objetos. Este hecho da lugar a que aparezcan situaciones en las que el sistema marca únicamente la parte frontal de un vehículo, o la parte superior de una persona (de cintura hacia arriba), o incluso en días soleados como por ejemplo ocurre en las escenas 7, 8, 9 y 10, las elipses también podrían incluir las sombras de los objetos, adoptando unas dimensiones mayores de las que realmente poseen los objetos (ver Figuras 6.1 y 6.2).



Figura 6.1: Captura de cada una de las escenas empleadas en las pruebas donde es posible observar las condiciones de iluminación.

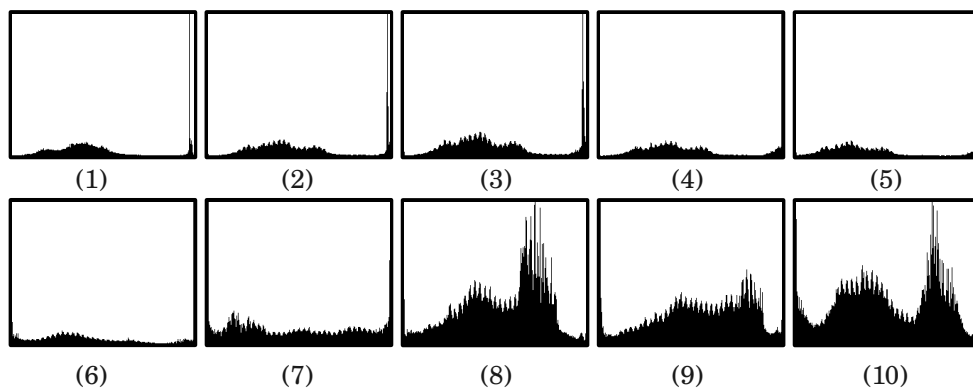


Figura 6.2: Conjunto de histogramas que representan los valores de brillo y contraste en cada una de las escenas empleadas en las pruebas. Cuanto más altos sean los valores alcanzados en el eje Y, mayor es la iluminación de la escena. Las seis primeras escenas corresponden a un día nublado donde los objetos carecen de sombras acentuadas, mientras que en las cuatro últimas se puede apreciar que el grado de iluminación es mucho mayor.

Una de las ventajas de que el algoritmo de clasificación de objetos se base en las proporciones anchura/altura y no en el tamaño real del objeto, es que

no es sensible al problema mencionado anteriormente. Es decir, aunque una elipse no englobe a un objeto en su totalidad, o bien incluya su sombra en escenas iluminadas, la mayor parte de las veces sigue manteniendo unas proporciones propias de la clase a la que pertenece. El peor problema que puede suceder con la aparición de sombras en escenas iluminadas es que los algoritmos de visión las consideren como objetos independientes. Sin embargo, en las pruebas realizadas en este trabajo, este fenómeno ha aparecido en un número reducido de ocasiones debido a que los algoritmos empleados resuelven con gran destreza este problema y no suelen incluir en la elipse píxeles que pertenecen a sombras.

Clasificación de objetos					
Escena	Aciertos	Fallos			
		Regla Errónea	No se disparan Reglas	Histórico	Cambio de Elipse
1	7176 (95 %)	287	1	35	0
2	6760 (96 %)	188	1	56	0
3	6657 (95 %)	227	28	49	30
4	11065 (99 %)	84	4	0	19
5	1397 (99 %)	0	1	0	0
6	2242 (93 %)	153	1	0	0
7	7498 (95 %)	345	0	15	8
8	2866 (97 %)	19	4	0	64
9	6115 (91 %)	359	0	123	92
10	1887 (98 %)	3	1	0	27
MEDIA	95,8 %				

Tabla 6.2: Resultados obtenidos en el proceso de clasificación de objetos como peatones y vehículos.

6.2.2. Estimación de las zonas en las que se encuentran los objetos

La Tabla 6.3 muestra los resultados obtenidos en el proceso de la estimación de zona/as en las que se encuentran los objetos detectados en una escena, a partir de la intersección entre sus puntos de apoyo y los polígonos que representan las zonas relevantes. Este proceso depende en gran medida de la calidad del algoritmo de segmentación empleado y la construcción de las elipses que envuelven a los objetos, ya que si éstas no englobaran frecuentemente los puntos de apoyo de los objetos, el porcentaje de error en el cálculo de la localización sería alto. Este tipo de errores suele dar lugar a insatisfacciones en las restricciones espaciales provocando falsos positivos y falsos negativos.

Estimación de la localización de objetos (calculo de zonas)			
Escena	Aciertos	Fallos	
		Elipse Incorrecta	Objeto detectado con varias elipses
1	7069 (94 %)	238	192
2	7001 (99 %)	4	0
3	6920 (98 %)	71	0
4	11130 (99 %)	23	16
5	1398 (100 %)	0	0
6	2396 (100 %)	0	0
7	7732 (98 %)	137	0
8	2910 (98 %)	26	17
9	6654 (99 %)	35	0
10	1906 (99 %)	12	0
MEDIA	98,4 %		

Tabla 6.3: Resultados obtenidos en la estimación de la zona/as en las que se encuentran cada uno de los objetos detectados en el entorno monitorizado

6.2.3. Análisis de trayectorias

En la Tabla 6.4 se muestran los resultados obtenidos en el proceso de análisis de normalidad en base al concepto trayectoria, siendo 96,4 % el porcentaje de acierto medio. Tal como se comentó anteriormente, se considera cualquier objeto detectado en un *frame* como una situación a analizar; de esta forma, si un objeto fuera detectado por el sistema en 7000 *frames*, se contabilizarían 7000 situaciones, es decir, cada vez que se activan los módulos de análisis. En la Tabla 6.4 se puede apreciar que algunos de los errores de interpretación del primer componente se producen debido a errores causados en etapas anteriores como por ejemplo, la clasificación de objetos o la estimación de zonas incorrectas. Estos errores, en algunas ocasiones, provocan el incumplimiento de las restricciones definidas para las instancias del concepto trayectoria, de tal forma que el objeto no tendrá asociada una trayectoria normal dando lugar a un falso positivo o falso negativo. Un error en los preprocesadores que preparan la información de los componentes de normalidad, no siempre produce un error de interpretación en los propios componentes. Por ejemplo, un vehículo que se encuentra en un paso de peatones y es clasificado como peatón, o un peatón que se encuentre en una zona habilitada para peatones y el sistema considera que se encuentra en otra zona habilitada también para peatones. Esta última situación es normal y el sistema la clasifica como tal, aunque la estimación de la zona en la que se encuentra el objeto haya sido incorrecta. Por este motivo, pueden existir en los tests un mayor número de errores en clasificación de objetos o estimaciones de las zonas, que en el análisis de trayectorias del primer componente de normalidad.

La salida de los componentes de normalidad en cada uno de los *frames* es

un valor numérico a partir del cual se estudia la pertenencia a los intervalos de normalidad: absolutamente normal, posiblemente normal, sospechoso, posiblemente anómalo, absolutamente anómalo. Cualquier situación normal con pertenencia única a los intervalos absolutamente y posiblemente normal ha sido considerada como acierto del componente 1 y, de la misma forma, cualquier situación anómala con pertenencia única a los intervalos absolutamente y posiblemente anómalo también ha sido considerada como acierto; los casos contrarios implican la aparición de falsos positivos y falsos negativos los cuales son contabilizados como errores. En el caso de existir pertenencia al intervalo de normalidad “sospechoso”, se ha considerado como error aquellas situaciones normales en las que la pertenencia a dicho intervalo es mayor que al intervalo posiblemente normal y, de la misma forma, situaciones anómalas con mayor pertenencia al intervalo sospechoso que al intervalo posiblemente anómalo, también han sido consideradas como errores. La decisión de considerar las situaciones sospechosas como críticas o no, debería recaer sobre las capas encargadas de tomar las decisiones en función del grado de seguridad que se pretenda alcanzar. Cuando más estricto sea el sistema, mayor será la probabilidad de un aumento de falsas alarmas, y menor la probabilidad de que una situación anómala no sea detectada.

Análisis de Trayectorias (Componente 1)							
Escena	Número de Frames	Número de Situaciones		Aciertos	Fallos		
		Normales	Anómalas		C1	C2	C3
1	3943	7224	275	7093 (94 %)	184	222	0
2	3000	7142	137	6847 (97 %)	154	4	0
3	3000	6751	240	6731 (96 %)	209	21	30
4	3250	11051	121	11026 (99 %)	88	39	19
5	995	1398	0	1397 (99 %)	1	0	0
6	1471	2343	53	2242 (93 %)	154	0	0
7	5233	7831	35	7590 (96 %)	134	134	8
8	1906	2738	215	2920 (98 %)	7	26	0
9	2165	6390	299	6420 (95 %)	234	35	0
10	772	1843	75	1875 (97 %)	4	12	27
TOTAL				96,4 %			

Tabla 6.4: Resultados obtenidos por el componente de normalidad encargado de analizar las trayectorias de los objetos detectados. Las etiquetas C1, C2 y C3 se refieren a las causas de los fallos y su significado es el siguiente: C1: Clasificación incorrecta de objetos, C2: Posición incorrecta de la elipse, C3: Cambio de elipses entre objetos.

Por otro lado la Tabla 6.5 muestran el número de verdaderos positivos (VP), falsos positivos (FP), verdaderos negativos (VN) y falsos negativos (FN) para el primer componente. En esta Tabla también se pueden apreciar las relaciones que existen entre ellos mediante los coeficientes descritos en la Sección 6.1.1. Estos parámetros miden correctamente la calidad de los métodos propuestos en este trabajo ya que con ellos se puede apreciar

si las situaciones normales son clasificadas como tales, si las situaciones anómalas son detectadas cuando suceden y la cantidad de falsas alarmas que se producen. En particular, los resultados obtenidos y mostrados en la Tabla 6.5 son satisfactorios ya que el número de falsos positivos y falsos negativos es reducido teniendo en cuenta el número de situaciones totales analizadas. La situaciones más críticas son las situaciones anómalas detectadas como normales, o lo que es lo mismo, número de falsos positivos (FP). En la Tabla también se puede apreciar como el valor de este parámetro es mínimo en las diez escenas.

Cálculo de coeficientes para el Componente 1											
Escena	VP	VN	FP	FN	VPR	FPR	ACC	SPC	PPV	NPV	FDR
1	6928	165	71	335	0.95	0.30	0.94	0.699	0.98	0.330	0.01
2	6847	0	0	158	0.98	-	0.977	-	1	0	0
3	6684	47	5	255	0.963	0.096	0.963	0.904	0.999	0.156	0
4	11026	0	4	142	0.987	1	0.987	0	1	0	0
5	1397	0	0	1	0.99	-	0.99	-	1	0	0
6	2228	14	0	154	0.935	0	0.936	1	1	0.083	0
7	7590	0	0	276	0.965	-	0.965	-	1	0	0
8	2766	154	26	7	0.997	0.144	0.989	0.856	0.991	0.957	0.009
9	6181	239	0	269	0.958	0	0.96	1	1	0.470	0
10	1875	0	0	43	0.978	-	0.978	-	1	0	0

Tabla 6.5: Relaciones existentes entre Verdaderos/Falsos positivos y Verdaderos/Falsos negativos en el componente de normalidad encargado de analizar las trayectorias.

6.2.4. Análisis de velocidades

La Tabla 6.6 muestra los resultados obtenidos en el proceso de clasificación de velocidades como normales o anómalas, realizado por el segundo componente de normalidad desarrollado en este trabajo. En este caso el porcentaje medio de acierto obtenido es de un 98,7 %, mayor que la cifra alcanzada por el primer componente. Este componente hace uso del algoritmo de aprendizaje para obtener un conjunto de reglas que determinan si la velocidad de un objeto es normal en función de la región de la imagen en la que se encuentre y el desplazamiento que realice cada 25 *frames*. En las pruebas realizadas, tanto los vehículos como los peatones suelen circular durante la mayor parte del tiempo a una velocidad normal, cuando no ha sido así, estas situaciones han sido normalmente detectadas de manera satisfactoria.

Dado que este componente se basa en el desplazamiento de la elipse que envuelve el objeto cada cierto tiempo, es importante tener en cuenta que las oclusiones parciales pueden provocar cambios bruscos en la posición de la elipse y, por tanto, en la estimación de la velocidad. Es decir, mientras se está produciendo una oclusión parcial la elipse se mantiene en la parte visible del objeto y puede cambiar de posición bruscamente cuando la parte

Análisis de Velocidades (Componente 2)								
Escena	Número de Frames	Número de Situaciones		Aciertos	Fallos			
		Normal	Anómala		C1	C2	C3	C4
1	3943	7224	275	7478 (99 %)	0	0	21	0
2	3000	7142	137	6923 (98 %)	40	0	25	17
3	3000	6751	240	6842 (97 %)	0	25	124	0
4	3250	11051	121	11095 (99 %)	0	0	77	0
5	995	1398	0	1398 (100 %)	0	0	0	0
6	1471	2343	53	2344 (97 %)	13	39	0	0
7	5233	7831	35	7813 (99 %)	0	18	35	0
8	1906	2738	215	2943 (99 %)	0	0	10	0
9	2165	6390	299	6639 (99 %)	0	15	10	25
10	772	1843	75	1918 (100 %)	0	0	0	0
TOTAL				98,7 %				

Tabla 6.6: Resultados obtenidos por el segundo componente de normalidad encargado de analizar las velocidades que alcanzan los objetos detectados. Las etiquetas C1, C2, C3 y C4 hacen referencia a las causas de error en el componente y sus significados son los siguientes: C1: clasificación incorrecta de objetos, C2: Cambios bruscos en la posición y el tamaño de la elipse, C3: Activación incorrecta de las reglas que clasifican las velocidades como normal o anómalas, C4: Cambio de elipse entre objetos de distintas clases

que no era visible pasa a serlo. En el caso particular del entorno empleado en las pruebas, este fenómeno sucede en tres lugares (ver Figura 4.9): parte inferior izquierda donde se encuentran las ramas un árbol, parte superior-central donde se encuentra la rotonda y, finalmente, parte central derecha donde se encuentra la valla que establece los límites de la universidad. El algoritmo de aprendizaje empleado en este trabajo está preparado para solventar este problema, ya que aprende por sí mismo que en este tipo de zonas los desplazamientos pueden ser mayores, considerando la velocidad del objeto como normal.

La Tabla 6.7 muestra la relación existente entre VP, VN, FP y FN. Como es lógico los buenos resultados obtenidos en este componente son una consecuencia directa del número reducido de falsos positivos/negativos, y un número elevado de verdaderos positivos/negativos.

6.2.5. Análisis global

La Tablas 6.8 y 6.9 muestran los resultados obtenidos en el análisis global, en el que se hace uso de operadores de agregación OWA para combinar la salida de los dos componentes anteriores y obtener así un valor global de normalidad. El causante del error en el análisis global puede ser el primer componente, el segundo o ambos al mismo tiempo; esta última situación se contabiliza como un único error producido en el análisis global.

Cálculo de coeficientes para el Componente 2											
Escena	VP	VN	FP	FN	VPR	FPR	ACC	SPC	PPV	NPV	FDR
1	7460	18	21	0	1	0.53	0.99	0.462	0.99	1	0,002
2	6786	137	0	82	0,99	0	0,988	1	1	0,626	0
3	6641	201	39	110	0,984	0,163	0,979	0,838	0,994	0,646	0,005
4	10993	102	15	62	0,994	0,128	0,993	0,872	0,999	0,622	0,001
5	1398	0	0	0	1	-	1	-	1	-	0
6	2344	0	39	13	0,994	1	0,978	0	0,984	0	0,02
7	7788	25	10	43	0,995	0,286	0,995	0,714	0,999	0,368	0,001
8	2918	25	10	0	1	0,286	0,997	0,714	0,997	1	0
9	6589	50	10	40	0,994	0,167	0,99	0,83	0,998	0,556	0,001
10	1843	75	0	0	1	0	1	1	1	1	0

Tabla 6.7: Relaciones existentes entre Verdaderos/Falsos positivos y Verdaderos/Falsos negativos en el componente de normalidad encargado de analizar las velocidades de los objetos detectados.

En algunas ocasiones los errores producidos en algunos de los componentes no producen error en el análisis global debido al uso del vector de pesos en el operador de agregación. Por ejemplo, una situación que sea considerada como sospechosa por el primer componente y absolutamente normal por el segundo podría ser considerada como posiblemente normal en el análisis global.

Análisis Global: trayectorias y velocidades							
Escena	Número de Frames	Número de Situaciones		Aciertos	Fallos		
		Normal	Anómalas		C1	C2	Ambos
1	3943	7224	275	7072 (94 %)	406	21	0
2	3000	7142	137	6805 (97 %)	118	42	40
3	3000	6751	240	6504 (93 %)	260	149	0
4	3250	11051	121	10949 (98 %)	146	77	0
5	995	1398	0	1397 (99 %)	1	0	0
6	1471	2343	53	2190 (91 %)	154	39	13
7	5233	7831	35	7537 (95 %)	276	53	0
8	1906	2738	215	2910 (98 %)	33	10	0
9	2165	6390	299	6370 (95 %)	269	50	0
10	772	1843	75	1875 (97 %)	43	0	0
TOTAL				95,7 %			

Tabla 6.8: Resultados obtenidos en el análisis global como combinación de los análisis realizados por el primer y segundo componente de normalidad. Las etiquetas C1 y C2 significan errores procedentes del primer componente y del segundo, respectivamente.

Para mostrar la relación existente entre la razón de verdaderos positivos (VPR) y la razón de falsos positivos (FPR) se han empleado curvas ROC [72] (Figura 6.3). El parámetro VPR representa la sensibilidad del clasificador, es decir, la relación que existe entre los verdaderos positivos y el total

Cálculo de coeficientes para el Análisis Global											
Esc.	VP	VN	FP	FN	VPR	FPR	ACC	SPC	PPV	NPV	FDR
1	6889	183	92	335	0.95	0.33	0.94	0.67	0.98	0.353	0.01
2	6668	137	0	200	0.97	0	0.971	1	1	0.407	0
3	6334	248	44	365	0.946	0.151	0.941	0.849	0.993	0.405	0.01
4	10847	102	19	204	0.982	0.157	0.98	0.843	0.998	0.333	0.001
5	1397	0	0	1	0.99	-	0.99	-	1	0	0
6	2189	14	39	154	0.934	0.736	0.919	0.264	0.982	0.083	0.02
7	7512	25	10	319	0.959	0.286	0.958	0.714	0.999	0.073	0.001
8	2731	179	36	7	0.997	0.167	0.985	0.833	0.987	0.962	0.01
9	6081	289	10	309	0.952	0.033	0.95	0.967	0.998	0.483	0.001
10	1830	75	0	43	0.977	0	0.978	1	1	0.636	0

Tabla 6.9: Relaciones existentes entre Verdaderos/Falsos positivos y Verdaderos/Falsos negativos en el análisis global de normalidad.

de casos positivos existentes en la prueba. Por el contrario, el parámetro FPR representa la relación existente entre los falsos positivos (situaciones anómalas clasificadas como normal) y el número de situaciones negativas totales de la prueba (verdaderos negativos y falsos positivos). Además, si se tiene en cuenta que VPR también representa la sensibilidad del clasificador y FPR la especificidad, el uso de curvas ROC es una forma apropiada de representar la especificidad frente a la sensibilidad. Tal como se puede apreciar en la Figura 6.3, este tipo de gráficas incluyen una línea diagonal que dividen el espacio ROC en dos partes, todos los puntos que permanezcan en la parte superior son considerados como resultados positivos. El clasificador perfecto es aquel que sitúa todos sus puntos en la esquina superior izquierda, es decir, un 100% de sensibilidad o ningún falso negativo y un 100% de especificidad o, lo que es lo mismo, ningún falso positivo. En las gráficas de la Figura 6.3 se puede apreciar que la interpretación de trayectorias y velocidades se acerca a estas cifras en algunas de las pruebas realizadas. Finalmente, la última de las gráficas mostradas en la Figura 6.3 muestra la relación existente entre VP, VN, FP y FN en el análisis global para cada una de las diez escenas empleadas en los experimentos.

La Figura 6.4 muestra la relación existente entre los dos parámetros que hacen referencia a las situaciones anómalas VN y FP. El número de verdaderos negativos (VN) representa el número de situaciones anómalas que el sistema ha detectado e interpretado correctamente, mientras que el número de falsos positivos (FP) hace referencia a situaciones anómalas que han pasado desapercibidas para el sistema de seguridad. Cuanto mayor sea el número de falsos positivos y menor sea el número de verdaderos negativos, menor es la calidad y eficiencia de los métodos empleados para la interpretación de eventos y situaciones. En las gráficas de la Figura 6.4 se puede apreciar que los métodos propuestos en este trabajo no sólo detectan de forma óptima las situaciones normales sino que también son aptos para

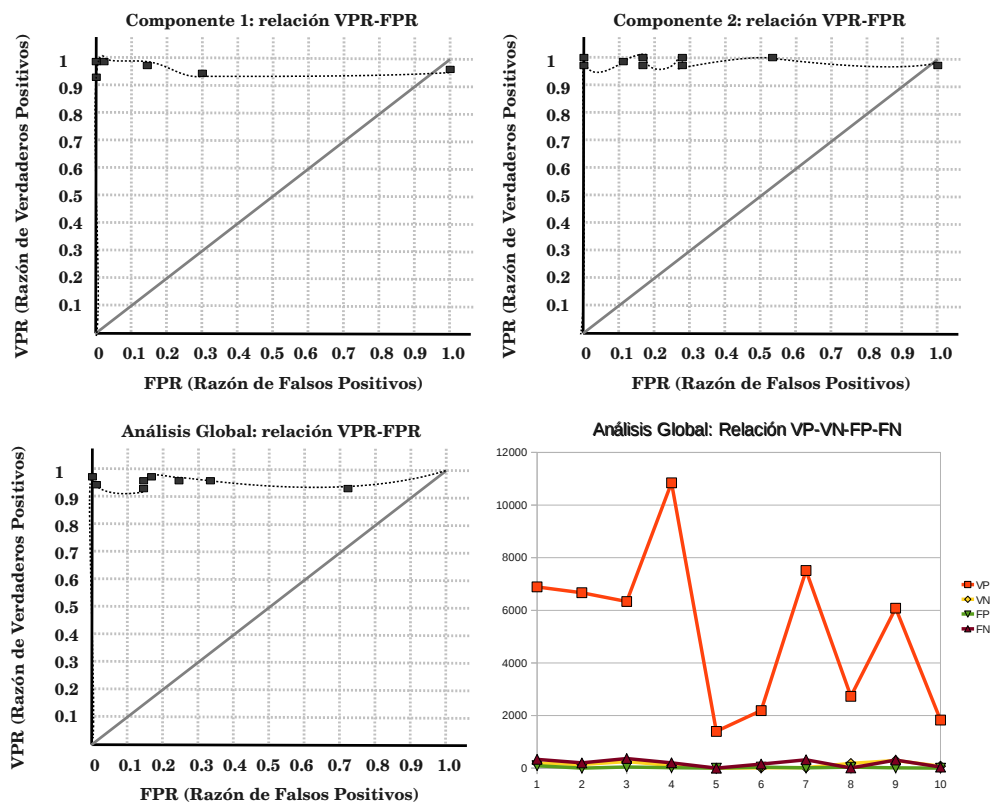


Figura 6.3: Las tres primeras gráficas representan espacios ROC donde se muestran las relaciones entre la razón de verdaderos positivos (VPR) y la razón de falsos positivos (FPR) para el primer y segundo componente, y el análisis global de normalidad. La última gráfica muestra la relación entre VP, VN, FP y FN en el análisis global para cada una de las escenas empleadas en los experimentos.

detectar las situaciones anómalas que suceden con menor frecuencia.

6.2.6. Tiempos

Las Tablas 6.10 y 6.11 muestran los tiempos totales y medios invertidos en cada uno de los procesos que forman parte del sistema global de vigilancia. Entre todos los procesos, la estimación de las zonas en las que podría encontrarse un objeto es el que más tiempo consume. Aún así, el tiempo máximo invertido para calcular la zona/as en las que se encuentra un objeto ha sido 17.33 ms, tiempo que permite al sistema responder ante cualquier suceso rápidamente. En trabajos futuros se propondrán nuevas versiones del algoritmo para reducir el tiempo invertido.

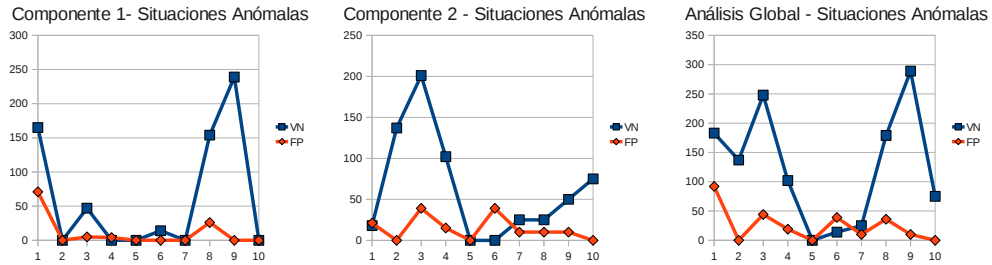


Figura 6.4: Relaciones entre las situaciones anómalas que suceden en los componentes de normalidad y el análisis global. Los verdaderos negativos (VN) representan situaciones anómalas detectadas, mientras que los falsos positivos (FP) representan situaciones anómalas interpretadas como normales.

TIEMPOS (ms)											
Esc.	Elem.	Clasif. de objetos		Localización del objeto		An. de Trayectorias		An. de Veloc.		Análisis Global	
		T	M	T	M	T	M	T	M	T	M
1	7499	255,86	0,03	58020,09	7,73	113,54	0,02	87,57	0,01	78,75	0,01
2	7005	198,65	0,02	52288,12	7,46	141,93	0,02	78,2	0,01	61,29	0,008
3	6991	214,94	0,03	80829,99	11,56	235,01	0,03	96,84	0,01	79,1	0,01
4	11172	376,16	0,03	98661,42	8,83	272,14	0,02	99,06	0,01	80,21	0,007
5	1398	54,77	0,03	12375,41	8,85	74,33	0,05	14,96	0,01	50,77	0,03
6	2396	68,79	0,02	23938,75	10,02	125,02	0,05	25,31	0,01	39,79	0,01
7	7866	288,05	0,03	69862,03	8,88	183,93	0,02	82,99	0,01	85,68	0,01
8	2953	79,76	0,02	51176,20	17,33	133,56	0,04	26,44	0,008	37,92	0,01
9	6689	221,45	0,03	75399,88	11,27	176,43	0,02	48,33	0,007	73,26	0,01
10	1918	70,41	0,03	15376,81	8,01	63,7	0,03	29,93	0,01	37,88	0,01

Tabla 6.10: Tiempos invertidos en cada uno de los procesos que constituyen el sistema (medidos en milisegundos). El significado de las etiquetas empleadas en la cabecera de la tabla es el siguiente: Esc: escena, Elem: elementos a analizar, T: tiempo total invertido, M: tiempo medio.

Tiempos invertidos en el proceso de aprendizaje			
Proceso	Ejemplos del Conjunto de Entrenamiento	Tiempo invertido en el aprendizaje (ms)	Número de reglas generadas
Aprendizaje de reglas para clasificar objetos	4880	1921	56
Aprendizaje de reglas para clasificar velocidades	2680	1610	15

Tabla 6.11: Tiempos invertidos en el proceso de aprendizaje de reglas difusas para la clasificación de objetos y velocidades.

6.3. Consideraciones finales

El objetivo principal de este trabajo era el diseño de un modelo formal que permitiera definir la normalidad en un entorno cualquiera y construir

sistemas de monitorización flexibles y escalables. En los Capítulos 4 y 5 se expuso como este modelo se podía aplicar al diseño de componentes que permitieran el análisis de las trayectorias y velocidades de los objetos. Los resultados obtenidos demuestran que los componentes diseñados mediante el modelo formal del Capítulo 3 pueden obtener resultados realmente satisfactorios, ofrecer un rendimiento alto y respuestas en un tiempo cercano al real y, finalmente, representar su conocimiento con una interpretabilidad alta. Además, ambos componentes han sido combinados con éxito (gracias al uso de operadores de agregación OWA) para obtener una valoración global del comportamiento de los objetos, es decir, el comportamiento de un objeto es normal siempre que realice una trayectoria normal y circule a una velocidad adecuada. Aunque los porcentajes de acierto representan valores numéricos altos (oscilan entre el 91 %-99 %), hay que tener en cuenta que la duración de las escenas elegidas varía entre 30" y 3' minutos. Lo que implica que en un periodo prolongado de observación el número de falsas alarmas todavía sería elevado y, por tanto, siguen siendo necesarias modificaciones futuras para mejorar los resultados.

Por otro lado, los buenos resultados obtenidos también se deben en gran parte a la robustez que el sistema ha demostrado ante errores procedentes de las capas inferiores. Para finalizar esta Sección se expondrán algunos de estos errores y como son tratados para evitar falsos positivos/negativos en la interpretación de eventos y conductas:

- El preprocesador encargado de clasificar los objetos mantiene un histórico sobre cómo ha sido clasificado un objeto desde que es detectado por el sistema hasta que desaparece, por lo que la clasificación de un objeto no sólo depende del último instante de tiempo. De esta forma si un objeto es clasificado correctamente durante un tiempo prolongado, las clasificaciones incorrectas en momentos puntuales no producen errores de clasificación.
- La clasificación de objetos se basa en las proporciones de los mismos, sus movimientos y las regiones de la imagen en las que se encuentre, sin importar el tamaño. Por tanto, aunque la elipse definida para un objeto en el proceso de *tracking* no lo envuelva perfectamente, el algoritmo de clasificación puede clasificarlo correctamente.
- El sistema también mantiene un histórico de las zonas en las que ha estado cada objeto y la máxima pertenencia a cada una de ellas. De esta forma, aunque se produzcan errores de segmentación en momentos puntuales y la elipse no esté situada plenamente sobre la zona en la que está localizado el objeto, no afecta al grado de satisfacción de las restricciones espaciales. Es decir, el sistema creerá que el objeto está siguiendo correctamente la secuencia de zonas que forman parte de una trayectoria, aunque en momentos puntuales se produzcan ciertas desviaciones.
- Los fallos bruscos en la posición de las elipses ante oclusiones parciales

producidas por elementos estáticos del entorno, suelen producir fallos en la estimación de la velocidad, ya que el algoritmo de aprendizaje propuesto en este trabajo es capaz de aprender las zonas donde se produce este fenómeno.

- Las zonas que forman parte de un entorno se pueden clasificar como zonas de entrada y salida por las que los objetos aparecen y desaparecen del campo de visión de la cámara, y zonas intermedias. Las oclusiones pueden provocar que el sistema pierda la referencia de un objeto y asigne una nueva elipse e identificador cuando éste sea visible de nuevo. La definición de trayectorias mediante el modelo formal expuesto en el Capítulo 3, permite la creación de trayectorias cuya secuencia este formada únicamente por áreas intermedias. De esta forma, cuando un objeto sea detectado de nuevo en una zona intermedia debido a una oclusión, el sistema podrá seguir asignándole trayectorias normales.
- El valor global de normalidad para un determinado objeto depende de la combinación de los análisis realizados por cada uno de los componentes. De esta forma, aunque la valoración de uno de los componentes no sea del todo acertada, el grado de normalidad final puede ser el correcto.

6.4. Publicaciones

Durante el desarrollo de la presente Tesis se han publicado los siguientes artículos en congresos y revistas nacionales e internacionales en el ámbito del aprendizaje automático, sistemas multi-agente y análisis de vídeo para la interpretación de eventos y conductas:

Revistas internacionales

- Javier Albusac, J.J Castro-Schez, L.M. Lopez-Lopez, David Vallejo, L. Jimenez-Linares. *A Supervised Learning Approach to Automate the Acquisition of Knowledge in Surveillance Systems*. J. Signal Processing. Vol. 89, No: 12, 2400-2414, 2009.
- Javier Albusac, David Vallejo, L. Jimenez-Linares, J.J Castro-Schez, L. Rodriguez-Benitez. *Intelligent Surveillance based on Normality Analysis to Detect Abnormal Behaviors*, in International Journal of Pattern Recognition and Artificial Intelligence, Special issue on Visual Analysis and Understanding for Surveillance Applications, 2009, IN PRESS.
- D. Vallejo, J. Albusac, L. Jimenez, C. Gonzalez, and J. Moreno, *A Cognitive Surveillance System for Detecting Incorrect Traffic Behaviors*, Expert System with Applications, Elsevier, 36(7):10503-10511, 2009.

- D. Vallejo, J. Albusac, J.A. Mateos, C. Glez-Morcillo, and L. Jiménez, *A Modern Approach to MultiAgent Development*, Journal of Systems and Software, Elsevier, (Accepted).
- L. Rodriguez-Benitez, Juan Moreno-Garcia, J.J Castro-Schez, Javier Albusac, L. Jimenez-Linares. *Automatic Object Behavior Recognition from Compressed Video Domain*. Image Vision and Computing, 27(6), pp. 648-657, 2009.

Congresos internacionales

- Javier Albusac, J.J Castro-Schez, L.M. Lopez-Lopez, D. Vallejo, L. Jimenez-Linares. *Learning and Classification of Events in Monitored Environments*, International Fuzzy Systems Association - European Society for Fuzzy Logic and Technology (IFSAC-EUSFLAT), Julio 2009, Lisboa (Portugal).
- Javier Albusac, J.J Castro-Schez, L. Jimenez-Linares, D. Vallejo, L.M. Lopez-Lopez. *Intelligent Surveillance for Trajectory Analysis*, in Proceedings of the International Conference on Enterprise Information Systems (ICEIS), Mayo 2009, Milan (Italia).
- C.J. Solana-Cipres, J. Albusac, J.J. Castro-Schez, L. Rodriguez-Benitez. *Automatic Object Labelling for Monitored Environments Using Clustering Techniques*. Proceedings of the 3rd International Conference on Imaging for Crime Detection and Prevention (ICDP'09), Diciembre 2009, Kingston (Inglaterra).
- L. Lopez-Lopez, Raul Miguel, J. Albusac, J.J. Castro-Schez. *Improving Searching Usability in Direct Shopping Portals*, The 28th North American Fuzzy Information Processing Society Annual Conference (NAFIPS), Junio 2009, Ohio (USA).
- Lorenzo M. Lopez-Lopez, Javier Albusac, J.J Castro-Schez, Luis Jimenez-Linares. *Semantics-Provided Environment Views for Normality Analysis-based Intelligent Surveillance*, in Proceedings of the ICAART, Lisboa, Jan 2009.
- L.M. Lopez-Lopez, J.J. Castro-Sanchez, D. Vallejo, J. Albusac. *A Recommender System based on a Machine Learning Algorithm for B2C Portals*, IEEE/WIC/ACM International Conference on Web Intelligence. Milan, Italia, Septiembre 15-18, 2009.
- J. Albusac, L.M. Lopez-Lopez, J.M. Murillo, and J.J. Castro-Schez. *Supporting Customer Searches in e-Marketplaces by means of Fuzzy Logic-based Machine Learning*. wi-iat, vol. 1, pp.892-896, 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2008.

- D. Vallejo, J. Albusac, C. Glez-Morcillo, L. Jimenez-Linares. *A Service-Oriented MultiAgent Architecture for Cognitive Surveillance*. Cooperative Information Agents CIA 2008, Praga (Czech Republic). Lecture Notes in Artificial Intelligence.
- D. Vallejo J.J. Castro-Schez, J. Albusac, C. Glez-Morcillo. *Integrating a Standard Communication Protocol into an E-Commerce Environment Based on Intelligent Agents*. ICEIS 2008, Barcelona (Spain). Junio 2008.
- J. Albusac, J.J Castro-Schez, D. Vallejo, L. Jimenez-Linares. *Learning Maximal Structure Rule with Pruning Based on Distance Between Fuzzy Sets*. 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'08), 441-447, Malaga, Spain, 2008.
- L. Rodriguez-Benitez, J. Moreno-Garcia, J. Albusac. *An approximate reasoning technique for segmentation on compressed mpeg video*. The International Joint Conference on Computer Vision and Computer Graphics, Theory and Applications. VISAPP'07. Barcelona (Spain). pp 184-191. Marzo 2007.

Congresos nacionales

- C.J. Solana-Ciprés, J. Albusac, J.J. Castro-Schez, J. Moreno, L. Rodriguez-Benitez. *Construcción de conjuntos de reglas difusas para la clasificación de objetos*. XV Congreso Español sobre Tecnologías y Lógica Fuzzy, ESTYLF 2010, Huelva, España.
- J. Albusac, J.J Castro-Schez, D. Vallejo, L. Jimenez-Linares. *Aprendizaje de reglas difusas para la clasificación de comportamientos en un sistema de vigilancia cognitiva*. IX Congreso Español sobre Tecnología y Lógica Fuzzy (ESTYLF'08), 557-562, Mieres, España, 2008.
- L. Rodriguez-Benitez, J. Moreno-Garcia, J. Albusac, J.J Castro-Schez. *Seguimiento de objetos representados lingüísticamente utilizando técnicas de razonamiento aproximado*. CEDI 2007, Zaragoza (Spain). Septiembre de 2007

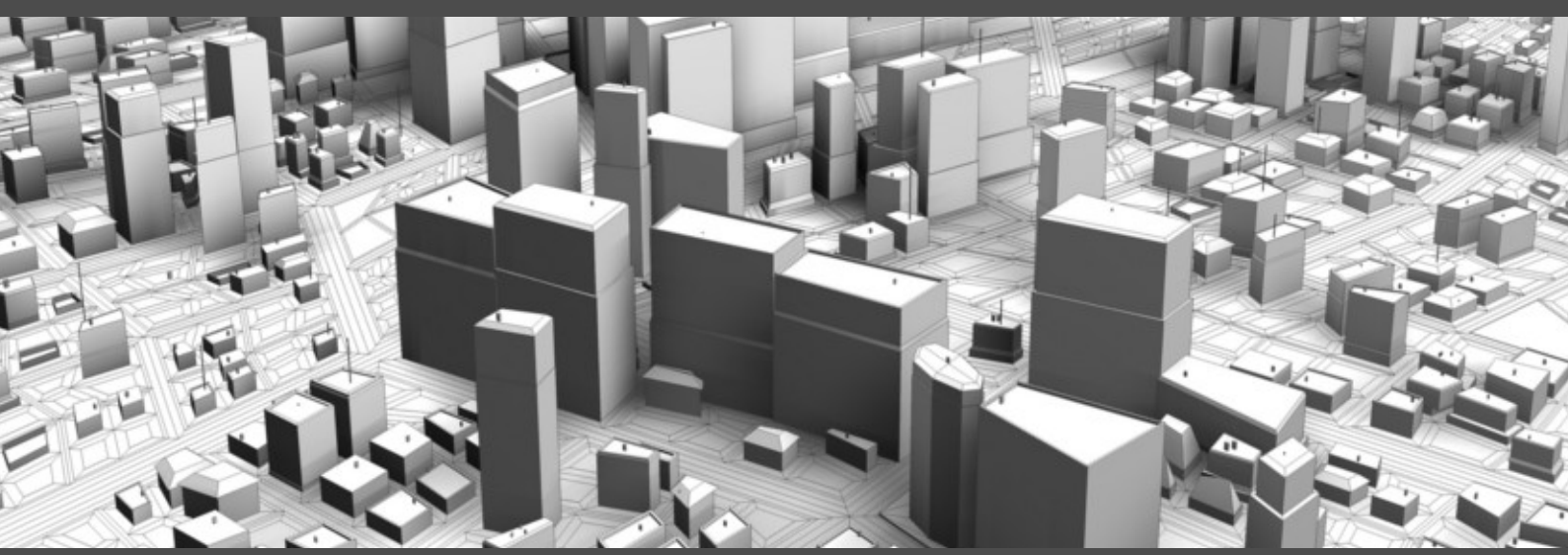


Chapter

Conclusions and Future Works

7

Chapter



7

Conclusions and Future Works

The focus of this thesis work has been to define a formal model for designing scalable and flexible surveillance systems. This model is mainly based on the definition of normal events and behaviours by means of independent and reusable components. In this chapter, the conclusions, applications and future work regarding the different aspects of the research presented in this thesis are discussed.

7.1. Conclusions

Since their first developments, traditional video surveillance systems have been designed to monitor environments. However, these systems have several limitations to satisfy the security demands required by the society. This need together with the arrival of new technologies and the price reduction of dedicated hardware for security represent some of the relevant reasons why intelligent surveillance is currently one of the hot research topics.

One of the main challenges in this field is to provide security expert systems with the autonomy and ability required to automatically understand events and analyse behaviours in order to improve the productivity and effectiveness in surveillance tasks. In complex environments where multiple situations take place at the same time, human agents have to deal with all them, being affected by negative factors such as fatigue or tiredness after a prolonged period of observation. Nevertheless, artificial expert systems do not have these limitations due to their processing capabilities. Furthermore, this kind of systems can be more effective than people when recognising certain kinds of events, such as the detection of suspicious or unattended objects.

In the last decade, different second and third generation surveillance systems have been developed both in the commercial and academic field. In

this last field, multiple novel approaches have been proposed, advancing the state of the art and developing research prototypes that establish interesting starting points for commercial developments. Most of proposed systems have been designed to solve particular problems in specific scenarios. They can be understood as a set of pieces that act separately to detect specific events in particular scenarios, without establishing relationships between them to get a global analysis.

In spite of that, these systems provide important advantages over the traditional ones, but two goals need to be reached in order to further progress. i) First, the design of scalable surveillance systems that allow to include new modules to increase the analysis capacity and to aggregate their output to get a global view of the environment state. ii) Secondly, there exists a need for a higher flexibility to vary the artificial system behaviour depending on the requirements of the analysed environment and the kinds of monitored objects. The way of monitoring the behaviour of a particular class of objects may change in relation to the rest of classes. Depending on the kind of objects that appear in the scenario and the security level established, the system must provide the flexibility to configure the existing analysis modules so that the system designed in a global way can be adapted to particular environments.

These problems are some of our main motivations to propose a novel model for the design of scalable and flexible surveillance systems. In Chapter 3, the surveillance problem was analysed in depth, describing how to structure this problem into several subproblems to deal with the complexity. To do that, the idea of using a divide and conquer approach based on independent and reusable modules in order to solve each subproblem was studied. Thus, the formal model is based on the normality definition of events and behaviours according to a set of independent components, which can be instantiated in particular monitored environments.

These modules, named as *normality components*, specify how each kind of object should ideally behave depending on the aspect to watch, such as the trajectory that follows or the moving speed. The integration of new components into the surveillance system does not imply modifications on the remaining ones, increasing the analysis capacity of the artificial system. The output of these components are combined by means of OWA aggregation operators in order to obtain a global normality value. This kind of operators can be easily configured to assign more or less relevance to such normality components depending on their output values, increasing the flexibility of the system.

Two normality components, trajectories and speed analysis, have been defined by means of this model. Both make use of the spatial information gathered by the security cameras, previously processed by modules of low-level layers. In particular, four preprocessors have been developed in this thesis work:

- Tracking of moving objects. This preprocessor detects and follows each moving object on the scene.
- Object classification. This preprocessor classifies each detected object and determines the degree of membership to each class by means of a set of fuzzy rules previously learned by a machine learning algorithm.
- Zone estimation. This preprocessor provides the first normality component (trajectory analysis) with the information about the location of each object, that is, the membership value to each area where the object is located.
- Horizontal and vertical position of each detected object in a 2D image. This information is used by the second normality component to estimate speeds. An important advantage of this component is that it does not need to know the camera configuration parameters to work.

Although the analysis of trajectories and speed have been widely studied by other researchers, most of the existing approaches are based on the analysis of spatial and (occasionally) temporal information. This information may be enough to recognise the path followed by objects or to estimate their speed, but it is not enough for surveillance systems that need to deal with a higher number of factors. For example, the normal trajectories of vehicles may be not normal for pedestrians. In this case, the object class is very important to define normal trajectories. Another example could be trajectories which are only normal if they are followed at a particular time interval or with a shorter duration than a maximum time allowed. In this second case, more complex temporal constraints are needed to define normal trajectories.

The proposed model defines the normality of a concept depending on a set of fuzzy constraints. This fact allows to develop more complete and suitable definitions for surveillance systems. In this way, the system also provides flexibility in the internal analysis performed by the normality components, since multiple constraints can be easily added or removed. The use of fuzzy logic not only deals with the uncertainty and vagueness of low-level information, but also allows us to represent the expert knowledge used by the surveillance system with a high interpretability by means of linguistic labels, providing it with the expressiveness required to justify the decision making.

The formal model establishes a guide which specifies how to define new normality components in a general way. This definition includes variables, constraints and analysis mechanisms. However, each component must be particularised before being deployed in a specific environment. For instance, the particularisation of a normality component for trajectory analysis includes the normal trajectories of the environment where the component will be deployed. In order to ease this process, the normality components have associated knowledge acquisition tools and learning algorithms. In

this thesis work, both alternatives have been proposed to instantiate such components.

A knowledge acquisition tool to define relevant areas and normal trajectories has been developed for the normality component that analyses the trajectories. Besides, an inductive learning algorithm has been designed to automatically learn the normal speed in a monitored environment. This algorithm generates a set of fuzzy rules that classify object speeds as normal or abnormal. Each rule represents an instance of the *normal speed* concept. An important advantage of this algorithm is that it does not need a huge set of examples to cover all possible normal speeds in an environment. In fact, only one example per situation is needed. In addition, learned rules are high interpretable and justify by themselves why a particular speed is normal or not.

On the other hand, a prototype of monitoring system, named as OCULUS, has been developed to evaluate the model and the designed components. The system includes a knowledge acquisition tool to define relevant areas and normal trajectories by using a 2D image that represent the environment to be monitored; a learning algorithm to generate set of fuzzy rules that classify objects and speeds; and a monitoring tool to analyse the output generated by preprocessors and normality components.

Finally, the experimental results prove the feasibility of the proposed model and the designed normality components, both in relation to the events and behaviours understanding and the time response to provide such results, two key aspects in surveillance systems. Moreover, normality components and algorithms proposed in this work have demonstrated robustness against the errors made by low-level layers. However, it is important to remark that scenes used for the evaluation have a short duration time (30"- 3'30"). Thus, although the global percentage classification accuracy is high (91%-99%, and 95,7% as average value), a 5% of errors every 3 minutes still implies an important number of false alarms which should be reduced with new proposals in future works.

7.2. Future works

One of the expected future works within a short time is the design of new normality components in order to increase the analysis capacity of OCULUS. Some examples of these new components are the following:

- Normality component for pedestrian and vehicle behaviour analysis in pedestrian crossings. This component has to carry out objects classification and to study the proximity relationships between them. An anomalous situation could happen when a vehicle is over a pedestrian crossing at the same time that a pedestrian. In previous

works, we proposed a traffic system to detect anomalies in pedestrian crossings [189]. The main limitation of the proposed system is the lack of scalability to include new analysis modules and combine all of them to obtain a global analysis. It is our intention to reuse part of this work when defining the new component.

- Normality component to detect damages to urban infrastructures. For instance, painted walls, broken litters, damaged benches, etc. The component would have a normal model that represents the monitored environment in normal conditions, comparing it with the current state of the environment to detect possible differences.
- Normality component to detect unattended or abandoned objects. People do not normally abandon objects, so this kind of situations may be suspicious and must be detected.
- Normality component to analyse crowd movements in stadiums, coaches, railway stations, airports, etc. Normally, abrupt movements of crowds happen in panic situations due to anomalies. Within this context, it is very interesting to provide the system with the ability to detect this kind of situations.

In order to complement the normality components-based approach, we described in the first chapter the option of detecting particular anomalous event to make appropriate decisions. In future works, abnormality components will be designed and combined with the analysis made by the normality components. In this way, the system will know that something abnormal is happening, being able to identify the particular anomalous situation. This information is very useful for top layers where critical decisions are made and alarms are activated.

Other possible research line is related to fusion information in low levels. In this work, the output of each normality component is combined to obtain a global normality value. However, fusion processes in low levels can help to improve the understanding results. In this way, a component may deal with the information gathered by multiple sensors about a particular object. The management of redundant information may improve classification and understanding results when the data provided by a single source is not reliable.

On the other hand, the process of creating training sets for the learning algorithm proposed in this work should be automated to ease the deployment of the surveillance system in new environments. As a possible solution, we have started to design clustering algorithms to address this problem. A first approach c is proposed in [168].

Finally, it is also important to remark that the model proposed in this thesis can be applied to different fields. In this work the model has directly been applied for developing surveillance systems, but it can be adapted

to another kinds of fields such as industrial processes, communication networks, computer systems, etc. In other words, the model proposed in this work can be used in any kind of environment where normality is well-known and can be defined by means of a set of constraints.

Referencias

- [1] http://es.wikipedia.org/wiki/proyecto_hesperia.
- [2] <http://opencv.willowgarage.com/wiki/videosurveillance>.
- [3] <http://www.baesystems.com/>.
- [4] <http://www.co-friend.net>.
- [5] <http://www.cyc.com/>.
- [6] <http://www.detec.no/>.
- [7] <http://www.gotchanow.com/>.
- [8] <http://www.ipsotek.com/>.
- [9] <http://www.ksl.stanford.edu/software/ontolingua/>.
- [10] <http://www.objectvideo.com/>.
- [11] <http://www.zeroc.com/>.
- [12] B. Abreu, L. Botelho, A. Cavallaro, D. Douchamps, T. Ebrahimi, P. Figueiredo, B. Macq, B. Mory, L. Nunes, J. Orri, et al. Video-based multi-agent traffic surveillance system. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 457–462, 2000.
- [13] J. Aguilera, H. Wildenauer, M. Kampel, M. Borg, D. Thirde, and J. Ferryman. Evaluation of motion segmentation quality for aircraft activity surveillance. In *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 293–300, 2005.
- [14] J.F. Allen. Time and Time Again: The Many Ways to Represent Time. *International Journal of Intelligent Systems*, 6(4):341–355, 1991.
- [15] J.F. Allen and G. Ferguson. Actions and Events in Interval Temporal Logic. *Journal of Logic and Computation*, 4(5):531–579, 1994.
- [16] P. Angelov and X. Zhou. Evolving Fuzzy-Rule-Based Classifiers From Data Streams. *IEEE Transactions on Fuzzy Systems*, 16(6):1462–1475, 2008.
- [17] N. Anjum and A. Cavallaro. Multi-feature object trajectory clustering for video analysis. In *IEEE Transactions on Circuit and Systems for Video Technology, Special Issue on Event Analysis in Video*, volume 18, pages 1555–1564, 2008.

- [18] P. Avis. Surveillance and Canadian maritime domestic security. *Canadian Military Journal*, 4(1):9–15, 2003.
- [19] F. El Baf, T. Bouwmans, and B. Vachon. Fuzzy integral for moving object detection. In *IEEE International Conference on Fuzzy Systems*, pages 1729–1736, 2008.
- [20] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.
- [21] C. Bauckhage, M. Hanheide, S. Wrede, and G. Sagerer. A cognitive vision system for action recognition in office environments. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 827–833, 2004.
- [22] M.A. Baumberg. Learning Deformable Models for Tracking Human Motion. *Doctoral Dissertation. University of Leeds*, 1995.
- [23] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters. *Computer Vision and Pattern Recognition*, pages 495–501, 1997.
- [24] D. Biliotti, G. Antonini, and J.P. Thiran. Multi-layer hierarchical clustering of pedestrian trajectories for automatic counting of people in video sequences. In *IEEE Workshop on motion and video computing*, volume II, pages 50–57, 2005.
- [25] P. Blauensteiner and M. Kampel. Visual Surveillance of an Airports Apron-An Overview of the AVITRACK Project. In *Proceedings of the Workshop of the Austrian Association for Pattern Recognition*, pages 213–220, 2004.
- [26] D. Bloisi, L. Iocchi, P. Remagnino, and N.D. Monekosso. ARGOS—A Video Surveillance System for Boat Traffic Monitoring In Venice. *To appear in International Journal of Pattern Recognition and Artificial Intelligence*, 2009.
- [27] A.F. Bobick and A.D. Wilson. A state-based approach to the representation and recognition of gesture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(12):1325–1337, 1997.
- [28] P.P. Bonissone and K. Decker. Selecting uncertainty calculi and granularity: An experiment in trading-off precision and complexity. *Uncertainty in Artificial Intelligence*, pages 217–248, 1986.
- [29] M. Borg, D. Thirde, J. Ferryman, F. Fusier, V. Valentin, F. Bremond, and M. Thonnat. Video Surveillance for Aircraft Activity Monitoring. *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 16–21, 2005.

- [30] J.E. Boyd, J. Meloche, and Y. Vardi. Statistical tracking in video traffic surveillance. In *7th International Conference on Computer Vision, Kerkyra, Greece*, volume 1, pages 163–168, 1999.
- [31] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 994–999, 1997.
- [32] F. Bremond. *Scene Understanding: perception, multi-sensor fusion, spatio-temporal reasoning and activity recognition*. Habilitation a diriger des Recherches, University of Nice-Sophia Antipolis, 2007.
- [33] F. Bremond and G. Medioni. Scenario recognition in airborne video imagery. *DARPA Image Understanding Workshop*, pages 211–216, 1998.
- [34] T. Brodsky, R. Cohen-Solal, S. Gutta, D. Lyons, V. Philomin, and M. Trajkovic. Visual Surveillance in Retail Stores and in the Home. *Video-Based Surveillance Systems: Computer Vision and Distributed Processing*, Kluwer Academic Publishers, pages 51–61, 2002.
- [35] H.H. Bui, S. Venkatesh, and G. West. Tracking and surveillance in wide-area spatial environments using the abstract hidden markov model. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1):177–195, 2001.
- [36] H. Buxton. Learning and understanding dynamic scene activity: a review. *Image and Vision Computing*, 21(1):125–136, 2004.
- [37] H. Buxton and S. Gong. Advanced Visual Surveillance using Bayesian Networks. In *International Conference on Computer Vision*, pages 111–123, 1995.
- [38] H. Buxton and S. Gong. Visual surveillance in a dynamic and uncertain world. *Artificial Intelligence*, 78(1-2):431–459, 1995.
- [39] N. Carter, D. Young, and J. Ferryman. A combined bayesian markovian approach for behaviour recognition. In *Proceedings of the 18th International Conference on Pattern Recognition*, pages 761–764, 2006.
- [40] J.L. Castro, J.J. Castro-Schez, and J.M. Zurita. Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems. *Fuzzy Sets and Systems*, 101(3):331–342, 1999.
- [41] F.W. Cathey and D.J. Dailey. A novel technique to dynamically measure vehicle speed using uncalibrated roadway cameras. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 777–782, 2005.
- [42] B. Chandrasekaran, J.R. Josephson, and V.R. Benjamins. What Are Ontologies, and Why Do We Need Them?. *IEEE Intelligent Systems*, 14(1):20–26, 1999.

- [43] B. Chen, H.H. Cheng, and J. Palen. Integrating mobile agent technology with multi-agent systems for distributed traffic detection and management systems. *Transportation Research Part C*, 17(1):1–10, 2009.
- [44] K. M. Cheung, T. Kanade, J. Y. Bouguet, and M. Holler. A real time system for robust 3D voxel reconstruction of human motions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 714–720, 2000.
- [45] Y. Cho and J. Rice. Estimating velocity fields on a freeway from low-resolution videos. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):463–469, 2006.
- [46] C.Y. Chong, S. Mori, W.H. Barker, K.C. Chang, A. Booz, H. Inc., and C.A. San Francisco. Architectures and algorithms for track association and fusion. *IEEE Aerospace and Electronic Systems Magazine*, 15(1):5–13, 2000.
- [47] H.I. Christensen, J. Matas, and J. Kittler. Using grammars for scene interpretation. *International Conference on Image Processing*, 1:793–796, 1996.
- [48] E. Clementini and P. Di Felice. Approximate topological relations. *International Journal of Approximate Reasoning*, 16:173–204, 1997.
- [49] F. Coenen and V. Pepijn. A generic ontology for spatial reasoning. *Proceedings of ES98, the Eighteenth Annual International Conference of the British Computer Society Specialist Group on Expert Systems, Cambridge UK, December 14th-16th*, pages 44–57, 1998.
- [50] C.J. Cohen, F. Morelli, and K.A. Scott. A surveillance system for the recognition of intent within individuals and crowds. In *Proceedings of IEEE Conference on Technologies for Homeland Security*, pages 559–565, 2008.
- [51] A.G. Cohn and B. Bennett. Qualitative spatial representation and reasoning with regions connection calculus. *GeoInformatica*, 1(3):275–316, 1997.
- [52] R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa. A system for video surveillance and monitoring. *Technical report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University*, 2000.
- [53] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.
- [54] M. Cristani, M. Bicego, and V. Murino. Audio-visual event recognition in surveillance video sequences. *IEEE transactions on multimedia*, 9(2):257–267, 2007.

- [55] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani. Probabilistic posture classification for human-behavior analysis. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 35(1):42–54, 2005.
- [56] F. Cupillard, A. Avanzi, F. Bremond, and M. Thonnat. Video understanding for metro surveillance. In *IEEE International Conference on Networking, Sensing and Control*, volume 1, pages 186–191, 2004.
- [57] F. Cupillard, F. Brémond, and M. Thonnat. Group behavior recognition with multiple cameras. In *Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision*, pages 177–183, 2002.
- [58] R. Cutler and L.S. Davis. Robust Real-Time Periodic Motion Detection, Analysis, and Applications. *IEEE Computer Society*, 22(8):781–796, 2000.
- [59] N. Cvejic, S.G. Nikolov, H.D. Knowles, A. Loza, A. Achim, D.R. Bull, and C.N. Canagarajah. The effect of pixel-level fusion on object tracking in multi-sensor surveillance video. In *Workshop on Image Registration and Fusion at the IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, volume 23, pages 1–7, 2007.
- [60] D. Damen and D. Hogg. Detecting Carried Objects in Short Video Sequences. In *Proceedings of the 10th European Conference on Computer Vision: Part III*, pages 154–167. Springer, 2008.
- [61] H. Dee and D. Hogg. Detecting inexplicable behaviour. In *British Machine Vision Conference*, pages 477–486, 2004.
- [62] H. Dee and D. Hogg. Navigational strategies and surveillance. In *Proceedings of the IEEE International Workshop on Visual Surveillance*, pages 73–81, 2005.
- [63] H. Dee and S.A. Velastin. How close are we to solving the problem of automated visual surveillance?. *Machine Vision and Applications*, 19(5):329–343, 2008.
- [64] A.K. Dey. Understanding and using context. *Personal and ubiquitous computing*, 5(1):4–7, 2001.
- [65] S.L. Dockstader and A.M. Tekalp. Multiple camera tracking of interacting and occluded human motion. *Proceedings of the IEEE*, 89(10):1441–1455, 2001.
- [66] D. Driankov, H. Hellendoorn, and M. Reinfrank. An Introduction to Fuzzy Control. *Springer Verlag*, 1996.
- [67] D. DuBois and H.M. Prade. *Fuzzy sets and systems: theory and applications*. Academic press, New York, 1980.

- [68] I. Duntsch, H. Wang, and S. McCloskey. A relation-algebraic approach to the region-connection calculus. *Theoretical Computer Science*, 255(1-2):173–204, 2001.
- [69] D. Dvorak and B. Kuipers. Model-based monitoring of dynamic systems. In *Proceedings of the 11th International Joint Conferences on Artificial Intelligence*, pages 1238–1243, 1989.
- [70] M.J. Egenhofer and J.R. Herring. Categorizing binary topological relations between regions, lines and points in geographic databases. *Technical report, Department of Surveying Engineering, University of Maine*, 1991.
- [71] R. J. Evans and R. G. Porges. Video Motion Anomaly Detection for Military Applications. In *3rd EMRS DTC Technical Conference, Edinburgh*, 2006.
- [72] T. Fawcett. ROC graphs: Notes and practical considerations for researchers. *Machine Learning*, 31, 2004.
- [73] S. Fejes and L. Davis. Detection of independent motion using directional motion estimation. *Computer Vision and Image Understanding*, 74(2):101–120, 1999.
- [74] H.M. Feng, C.Y. Chen, and C.Y. Wu. Vision-based fuzzy 8051 surveillance systems design. In *Proceedings of the 13th International Conference on Parallel and Distributed Systems*, volume 2, pages 1–2, 2007.
- [75] M. Fernandez, A. Gomez-Perez, and N. Juristo. METHONTOLOGY: From Ontological Art Towards Ontological Engineering. *Proceedings of the AAAI'97 Spring Symposium Series on Ontological Engineering*, pages 33–40, 1997.
- [76] J. Ferryman, M. Borg, D. Thirde, F. Fusier, V. Valentin, F. Bremond, M. Thonnat, J. Aguilera, and M. Kampel. Automated Scene Understanding for Airport Aprons. *Australian Joint Conference on Artificial Intelligence*, pages 593–503.
- [77] J. Ferryman, S.J Maybank, and A.D. Worrall. Visual Surveillance for Moving Vehicles. *International Journal of Computer Vision*, 37(2):187–197, 2000.
- [78] G. L. Foresti, C. Micheloni, and L. Snidaro. Event Classification for Automatic Visual-based Surveillance of Parking Lots. In *17th International Conference on Pattern Recognition*, volume 3, pages 314–317, 2004.
- [79] G. L. Foresti and L. Snidaro. A distributed sensor network for video surveillance of outdoor environments. In *Proceedings of International Conference on Image Processing*, volume 1, pages 22–25, 2002.

- [80] Foundation for Intelligent Physical Agents FIPA. *FIPA Agent Management Specification*. Foundation for Intelligent Physical Agents FIPA, Geneva, Switzerland, 2004.
- [81] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2):131–163, 1997.
- [82] F. Fusier, V. Valentin, F. Bremond, M. Thonnat, M. Borg, D. Thirde, and J. Ferryman. Video understanding for complex activity recognition. *Machine Vision and Applications*, 18(3):167–188, 2007.
- [83] M.P. Gardner. Highway Traffic Monitoring. *Transportation Research Board. Transportation in the New Millennium*, pages 5–10, 2000.
- [84] B. Georis, M. Maziere, F. Bremond, and M. Thonnat. A video interpretation platform applied to bank agency monitoring. In *Proceedings of Workshop on Intelligent Distributed Surveillance Systems*, pages 46–50, 2004.
- [85] A. Gomez-Perez. Ontological engineering: A state of the art. *Expert Update*, 2(3):33–43, 1999.
- [86] S. Gong and T. Xiang. Recognition of group activities using dynamic probabilistic networks. In *IEEE International Conference on Computer Vision*, pages 742–749, 2003.
- [87] K. C. Gross and R. M. Singer. Expert system for testing industrial processes and determining sensor status, June 2 1998. US Patent 5,761,090.
- [88] T.R. Gruber. Ontolingua: A Mechanism to Support Portable Ontologies (Technical Report). *Knowledge Systems Laboratory, Stanford University*, 1992.
- [89] T.R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43(5):907–928, 1995.
- [90] D.L. Hall and J. Llinas. An introduction to multisensor fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.
- [91] H. Hamacher, H. Leberling, and H.J. Zimmermann. Sensitivity analysis in fuzzy linear programming. *Fuzzy Sets and Systems*, 1(4):269–281, 1978.
- [92] I. Haritaoglu, D. Harwood, and L.S. Davis. w^4 : Real-time surveillance of people and their activities. *Patter Analysis and Machine Intelligence*, 22(8):809–830, 2000.
- [93] D. Heckerman. A Tutorial on Learning with Bayesian Networks. *Learning in graphical models, Kluwer, Dordrecht, Netherlands*, 1999.

- [94] R.J. Howarth and H. Buxton. Conceptual descriptions from monitoring and watching image sequences. *Image and Vision Computing*, 18(2):105–135, 2000.
- [95] C. Hsiao, C. Lin, and M. Cassidy. Application of Fuzzy Logic and Neural Networks to Automatically Detect Freeway Traffic Incidents. *Journal of Transportation Engineering*, 120(5):753–772, 1994.
- [96] M. Hsiao, G. West, S. Venkatesh, and M. Kumar. Using dynamic time warping for online temporal fusion in multisensor systems. *Information Fusion*, 9(3):370 – 388, 2008.
- [97] J.W. Hsieh and Y.T. Hsu. Boosted string representation and its application to video surveillance. *Pattern Recognition*, 41(10):3078–3091, 2008.
- [98] W. Hu, D. Xie, T. Tan, and S. Maybank. Learning activity patterns using fuzzy self-organizing neural network. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(3):1618–1626, 2004.
- [99] C. Hudelot and M. Thonnat. A cognitive vision platform for automatic recognition of natural complex objects. In *International Conference on Tools with Artificial Intelligence*, pages 398–405, 2003.
- [100] G. Iovane, G. Ingenito, and M. Leone. Motion tracking using fuzzy logic and consistent labeling for multiple objects in multiple cameras vision. *Journal of Discrete Mathematical Sciences & Cryptography*, 12(1):1–42, 2009.
- [101] Y. Ivanov, C. Stauffer, A. Bobick, and W. Grimson. Video surveillance of interactions. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 97–118, 1999.
- [102] O. Javed and M. Shah. Tracking and object classification for automated surveillance. *European Conference on Computer Vision*, 4:343–357, 2002.
- [103] E.N. Johnson, A.J. Calise, Y. Watanabe, J. Ha, and J.C. Neidhoefer. Real-time vision-based relative aircraft navigation. *AIAA Journal of Aerospace Computing, Information, and Communication*, 4(4):707–738, 2007.
- [104] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, 14(8):609–615, 1996.
- [105] I. Junejo, O. Javed, and M. Shah. Multi feature path modeling for video surveillance. In *International Conference on Pattern Recognition*, pages 716–719, 2004.
- [106] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

- [107] I.A. Karaulova, P.M. Hall, and A.D Marshall. A hierarchical model of dynamics for tracking people with a single video camera. *British Machine Vision Conference*, 1:352–361, 2000.
- [108] K.P. Karmann and A. Von Brandt. Moving object recognition using an adaptive background memory. *Time-Varying Image Processing and Moving Object Recognition*, 2:289–296, 1990.
- [109] L. Khoudour, J.P. Deparis, J.L. Bruyelle, F. Cabestaing, D. Aubert, S. Bouchafa, S.A. Velastin, M.A. Vincencio-Silva, and M. Wherett. Project cromatica. *Lecture notes in computer science*, Springer, pages 757–764, 1997.
- [110] M. Kilger. A shadow handler in a video-based real-time traffic monitoringsystem. *Proceedings of IEEE Applications of Computer Vision*, pages 11–18, 1992.
- [111] G.J. Klir and B. Yuan. *Fuzzy sets and fuzzy logic: theory and applications*. Prentice Hall Upper Saddle River, 1995.
- [112] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. *Proceedings of the 12th IAPR International Conference on Pattern Recognition and Computer Vision & Image Processing*, 4:3776–3781, 1994.
- [113] L. Lee, R. Romano, and G. Stein. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):758–767, 2000.
- [114] D.B. Lenat, R.V. Guha, K. Pittman, D. Pratt, and M. Shepherd. Cyc: toward programs with common sense. *Communications of the ACM*, 33(8):30–49, 1990.
- [115] T.W. Liao. Clustering of time series data - a survey. *Pattern Recognition*, 38(11):1857–1874, 2005.
- [116] L. Lin, H. Gong, L. Li, and L. Wang. Semantic event representation and recognition using syntactic attribute graph grammar. *Pattern Recognition Letters*, 30(2):180–186, 2009.
- [117] A. Lipton, H. Fujiyoshi, and R. Patil. Moving target classification and tracking from real-time video. *Fourth IEEE Workshop on Applications of Computer Vision*, pages 8–14, 1998.
- [118] W. Liu, F. Yamazaki, T. Vu, and Y. Maruyama. Speed detection of vehicles from aerial photographs. In *Proceeding of the 27th Asian Conference on Remote Sensing*, 2007.
- [119] B. Lo, J. Sun, and S. Velastin. Multisensor Fusion and Integration: Approaches, Applications, and Future Research Directions. *IEEE Sensors Journal*, 2(2):107–119, 2002.

- [120] B. Lo, J. Sun, and S. Velastin. Fusing visual and audio information in a distributed intelligent surveillance system for public transport systems. *Acta Automatica Sinica*, 29(3):393–407, 2003.
- [121] M. Lopez. Fuzzy management of surveillance sensors. *Proceedings of the 37th IEEE Conference on Decision and Control*, pages 245–250, 1998.
- [122] Y. Ma and M. Li. Detection for Abnormal Event Based on Trajectory Analysis and FSVM. *Lecture Notes in Computer Science, Springer*, 4681:1112–1120, 2007.
- [123] C. Maduro, K. Batista, P. Peixoto, and J. Batista. Estimation of vehicle velocity and traffic intensity using rectified images. In *Proceedings of the 15th IEEE International Conference on Image Processing*, pages 777–780, 2008.
- [124] D. Magee, C.J. Needham, P. Santos, A.G. Cohn, and D. Hogg. Autonomous learning for a cognitive agent using continuous models and inductive logic programming from audio-visual input. in *Proceedings of the AAAI workshop on Anchoring Symbols to Sensor Data*, pages 17–24, 2004.
- [125] D.R. Magee. Tracking multiple vehicles using foreground, background and motion models. *Image and Vision Computing*, 22(2):143–155, 2004.
- [126] D. Makris and T. Ellis. Path detection in video surveillance. *Image and Vision Computing*, 20(12):895–903, 2002.
- [127] D. Makris and T. Ellis. Learning semantic scene models from observing activity in visual surveillance. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(3):397–408, 2005.
- [128] R. Martinez-Tomas, M. Rincon, M. Bachiller, and J. Mira. On the correspondence between objects and events for the diagnosis of situations in visual surveillance task. *Pattern Recognition Letters*, 29(8):1117–1135, 2008.
- [129] S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. Tracking groups of people. *Computer Vision and Image Understanding*, 80(1):42–56, 2000.
- [130] C. Micheloni, G.L. Foresti, and L. Snidaro. A cooperative multicamera system for video-surveillance of parking lots. *IEEE Symposium on Intelligence Distributed Surveillance Systems*, pages 5–10, 2003.
- [131] H.B. Mitchell. *Multi-sensor data fusion: an introduction*. Springer, 2007.
- [132] A. Mohan, C. Papageorgiou, and T. Poggio. Example-Based Object Detection in Images by Components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4):349–361, 2001.

- [133] B.T. Morris and M. Trivedi. A Survey of Vision-Based Trajectory Learning and Analysis for Surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8):1114–1127, 2008.
- [134] V.Ñair and J.J. Clark. Automated visual surveillance using hidden markov models. *International Conference on Vision Interface*, pages 88–93, 2002.
- [135] C.J Needham, P.E Santos, D. Magee, V. Devin, D. Hogg, and A. G. Cohn. Protocols from perceptual observations. *Artificial Intelligence*, 167(1-2):103–136, 2005.
- [136] N.T. Nguyen, D.Q. Phung, S. Venkatesh, and H. Bui. Learning and Detecting Activities from Movement Trajectories Using the Hierarchical Hidden Markov Models. *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:955–960, 2005.
- [137] V.Ñiennattrakul and C.A. Ratanamahatana. On Clustering Multimedia Time Series Data Using K-Means and Dynamic Time Warping. In *Proceedings of the 2007 International Conference on Multimedia and Ubiquitous Engineering, IEEE Computer Society*, pages 733–738, 2007.
- [138] S.A Niyogi and E.H Adelson. Analyzing and recognizing walking figures in XYT. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 469–474, 1994.
- [139] H.Ñoor, S.H. Mirza, Y. Sheikh, A. Jain, and M. Shah. Model generation for video-based object recognition. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 715–718. ACM New York, NY, USA, 2006.
- [140] C.Ñwagboso. User focused surveillance systems integration for intelligent transport systems. *Advanced Video-Based Surveillance Systems, Kluwer Academic Publishers*, pages 8–12, 1999.
- [141] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. *Computer Vision and Pattern Recognition*, 97:193–199, 1997.
- [142] J. Owens, A. Hunter, and E. Fletcher. Novelty detection in video surveillance using hierarchical neural networks. *Lecture Notes in Computer Science, Springer*, pages 1249–1254, 2002.
- [143] H. Palaio, C. Maduro, K. Batista, and J. Batista. Ground plane velocity estimation embedding rectification on a particle filter multi-target tracking. In *IEEE International Conference on Robotics and Automation*, pages 825–830, 2009.
- [144] G. Park and S. S. Park. Moving objects spatiotemporal reasoning model for battlefield analysis. In *Proceedings of the 12th WSEAS international conference on Systems*, pages 722–733. World Scientific and Engineering Academy and Society, 2008.

- [145] M.A. Patricio, J. Carbo, O. Perez, J. Garcia, and J.M. Molina. Multi-agent framework in visual sensor networks. *EURASIP Journal on Advances in Signal Processing*, 2007(1):1–21, 2007.
- [146] C. Piciarelli and G.L. Foresti. On-line trajectory clustering for anomalous events detection. *Pattern Recognition Letters*, 27(15):1835–1842, 2006.
- [147] A. Pozzobon, G. Sciutto, and V. Recagno. Security in ports: the user requirements for surveillance system. *Advanced Video-Based Surveillance Systems*, Kluwer Academic Publishers, Boston, Mass, USA, 1999.
- [148] G. Pusiol, J. Patino, E. Corvee, F. Bremond, M. Thonnat, and S. Suresh. Optimizing Trajectories Clustering for Activity Recognition. In *Proceedings of the 1st International Workshop on Machine Learning for Vision-based Motion Analysis*, 2008.
- [149] D. Rawlinson, P. Chakravarty, and R. Jarvis. Distributed Visual Servoing of a Mobile Robot for Surveillance Applications. In *Proceedings of the Australasian Conference on Robotics and Automation*, 2004.
- [150] P. Remagnino, A.I. Shihab, and G.A. Jones. Distributed intelligence for multi-camera visual surveillance. *Pattern Recognition*, 37(4):675–689, 2004.
- [151] P. Remagnino, T. Tan, and K. Baker. Agent orientated annotation in model based visual surveillance. In *Proceedings of the Sixth International Conference on Computer Vision*, pages 857–862, 1998.
- [152] P. Remagnino, T. Tan, and K. Baker. Multi-agent visual surveillance of dynamic scenes. *Image and Vision Computing*, 16(8):529 – 532, 1998.
- [153] P. Remagnino, S.A. Velastin, G.L. Foresti, and M. Trivedi. Novel concepts and challenges for the next generation of video surveillance systems. *Machine Vision and Applications*, 18(3–4):135–137, 2007.
- [154] F.E. Retry, M.A. Cobb, D. All, R. Angryk, M. Paprzycki, S. Rahimi, L. Wen, and H. Yang. Fuzzy Spatial Relationships and Mobile Agent Technology in Geospatial Information Systems. *Applying Soft Computing in Defining Spatial Relations*, Springer, pages 123–155, 2002.
- [155] L. Rodriguez-Benitez, J. Moreno-Garcia, J.J. Castro-Schez, J. Albusac, and L. Jimenez-Linares. Automatic objects behaviour recognition from compressed video domain. *Image and Vision Computing*, 27(6):648–657, 2009.
- [156] K. Rohr. Towards model-based recognition of human movements in image sequences. *CVGIP: Image Understanding*, 59(1):94–115, 1994.

- [157] N. Ronetti and C. Dambra. Railway Station Surveillance: The Italian Case. *Multimedia Video-Based Surveillance Systems: Requirements, Issues and Solutions*, Kluwer Academic Publishers, pages 13–20, 2000.
- [158] N. Rota, M. Thonnat, and S.A. INRIA. Video sequence interpretation for visual surveillance. In *Proceedings of the third IEEE International Workshop on Visual Surveillance*, pages 59–68, 2000.
- [159] I. Saleemi, K. Shafique, and M. Shah. Probabilistic Modeling of Scene Dynamics for Applications in Visual Surveillance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1472–1485, 2009.
- [160] T. Schreck, J. Bernard, T. Von Landesberger, and J. Kohlhammer. Visual cluster analysis of trajectory data with interactive Kohonen maps. *Information Visualization*, 8(1):14–29, 2009.
- [161] B. Schweizer and A. Sklar. Associative functions and abstract semigroups. *Publ. Math. Debrecen*, 10(69-81):3, 1963.
- [162] A. Sekmen and F. Yao. Multi-Sensor Vision Data Fusion for Smart Airborne Surveillance, Storming Media, 2009.
- [163] M. Shah. Understanding human behavior from motion imagery. *Machine Vision and Applications*, 14(4):210–214, 2003.
- [164] M. Shmrat. Algorithm 112: Position of point relative to polygon. *Communications of the ACM*, 5(8):425, 1962.
- [165] N.T Siebel and S. Maybank. The advisor visual surveillance system. *Proceedings of the ECCV workshop Applications of Computer Vision*, pages 103–111, 2004.
- [166] G. J. D. Smith. Behind the screens: Examining constructions of deviance and informal practices among cctv control room operators in the uk. *Surveillance and Society*, 2(2/3):376–395, 2004.
- [167] L. Snidaro, G.L. Foresti, R. Niu, and P. Varshney. Sensor fusion for video surveillance. In *7th International Conference on Information Fusion*, pages 2049–2074, 2004.
- [168] C.J. Solana-Cipres, J. Albusac, J.J. Castro-Schez, and L. Rodriguez-Benitez. Automatic object labelling for monitored environments using clustering techniques. *Proceedings of the 3rd International Conference on Imaging for Crime Detection and Prevention (ICDP’09)*, 2009.
- [169] C.J. Solana-Cipres, L. Rodriguez-Benitez, J. Moreno-Garcia, and G. Fernandez-Escribano. Algoritmo de segmentacion de video en tiempo real sobre video comprimido h.264/avc. *XIV Congreso Español sobre Tecnologías y Lógica Fuzzy, ESTYLF’08*, pages 299–306, 2008.
- [170] M. Spirito, C.S. Regazzoni, and L. Marcenaro. Automatic detection of dangerous events for underground surveillance. In *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 195–200, 2005.

- [171] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:246–252, 1999.
- [172] J.G. Stell. Part and complement: Fundamental concepts in spatial relations. *Anal. of Mathematics and Artificial Intelligence*, 41(1):1–17, 2004.
- [173] N. Sumpter and A. Bulpitt. Learning spatio-temporal patterns for predicting object behaviour. *Image and Vision Computing*, 18(9):697–704, 2000.
- [174] S. Sutor, R. Reda, and A. KiwiSecurity. Multi sensor technologies augmenting video surveillance: Security and data fusion aspects. In *Proceedings of the 23rd International Symposium on Computer and Information Sciences*, pages 1–4, 2008.
- [175] E. Swears, A. Hoogs, and A.G. Perera. Learning motion patterns in surveillance video using hmm clustering. In *IEEE Workshop on Motion and video Computing (WMVC’08)*, pages 1–8, 2008.
- [176] K. Takahashi, S. Seki, E. Kojima, and R. Oka. Recognition of dexterous manipulations from time-varying images. *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 23–28, 1994.
- [177] Rawesak Tanawongsuwan and Aaron Bobick. Gait recognition from time-normalized joint-angle trajectories in the walking plane. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:726, 2001.
- [178] D. Tao, X. Li, X. Wu, and S. Maybank. Elapsed time in human gait recognition: A new approach. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2:177–180, 2006.
- [179] D. Tao, X. Li, X. Wu, and S. Maybank. Elapsed time in human gait recognition: A new approach. *General Tensor Discriminant Analysis and Gabor Features for Gait Recognition*, 29(10):1700–1715, 2007.
- [180] D. Tax. One-class classification; Concept-learning in the absence of counter-examples. *Phd Thesis, Delft University of Technology*, June 2001.
- [181] D. Tax and R. Duin. Combining one-class classifiers. *Lecture Notes in Computer Science, Springer*, pages 299–308, 2001.
- [182] D. Thirde, M. Borg, J.M. Ferryman, F. Fusier, V. Valentin, and F. Bremond M. Thonnat. A real-time scene understanding system for airport apron monitoring. In *Proceedings of the 4th IEEE International Conference on Computer Vision Systems*, page 26, 2006.

- [183] Chen T.P, H. Haussecker, A. Bovyrin, R. Belenov, K. Rodyushkin, A. Kuranov, and V. Eruhimov. Computer vision workload analysis: Case study of video surveillance systems. *Intel Technology Journal*, 9(2):109–118, 2005.
- [184] H. Tsutsui, J. Miura, and Y. Shirai. Optical flow-based person tracking by multiple cameras. *International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 91–96, 2001.
- [185] A. Utsumi, H. Mori, J. Ohya, and M. M. Yachida. Multiple-view-based tracking of multiple humans. In *Proceedings of the 14th International Conference on Pattern Recognition*, volume 1, pages 597–609.
- [186] M. Valera and S.A. Velastin. Intelligent distributed surveillance systems: a review. *Vision, Image and Signal Processing*, 152(2):192–204.
- [187] D. Vallejo, J. Albusac, J.J. Castro-Schez, C. Gonzalez, and L. Jimenez. A Multi-Agent Architecture for Supporting Distributed Normality-based Intelligent Surveillance. *Engineering Applications of Artificial Intelligence (en revision)*, 2009.
- [188] D. Vallejo, J. Albusac, C. Glez-Morcillo, and L. Jimenez. A service-oriented multiagent architecture for cognitive surveillance. In *Cooperative Information Agents*, volume 5180/2008, pages 101–115, 2008.
- [189] D. Vallejo, J. Albusac, L. Jimenez, C. Gonzalez, and J. Moreno. A cognitive surveillance system for detecting incorrect traffic behaviors. *Expert Systems With Applications*, 36(7):10503–10511, 2009.
- [190] D. Vallejo, J. Albusac, J.A. Mateos, C. Glez-Morcillo, and L. Jimenez. A Modern Approach to MultiAgent Development. *Journal of Systems and Software (Accepted)*, 2009.
- [191] S.A Velastin, L. Khoudour, B. Lo, J. Sun, and M. Vicencio-Silva. PRISMATICA: a multi-sensor surveillance system for public transport networks. In *Proceedings of the 12th IEEE International Conference on Road Transport Information and Control*, pages 19–25, 2004.
- [192] L. Wald. Definitions and terms of reference in data fusion. *International Archives of Photogrammetry and Remote Sensing*, 32(Part 7):651–654, 1999.
- [193] E.L Waltz and J. Llinas. *Multisensor data fusion*. Artech House, Inc. Norwood, MA, USA, 1990.
- [194] W. Wang and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man and Cybernetics*, 34(3):334–352, 2004.
- [195] X. Wang, X. Ma, and W. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models.

- IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):539–555, 2009.
- [196] X. Wang, K. Tieu, and E. Grimson. Learning semantic scene models by trajectory analysis. *ECCV, Part III, Lecture Notes in Computer Science*, 3953:110–123, 2006.
- [197] S. Weber. A general concept of fuzzy connectives, negations and implications based on t-norms and t-conorms. *Fuzzy sets and systems*, 11(2):115–134, 1983.
- [198] G. Weidl, A. L. Madsen, and S. Israelson. Applications of object-oriented Bayesian networks for condition monitoring, root cause analysis and decision support on operation of complex continuous processes. *Computers and Chemical Engineering*, 29(9):1996–2009, 2005.
- [199] G. Weiss. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 1999.
- [200] A.D. Wilson, A.F. Bobick, and J. Cassell. Temporal classification of natural gesture and application to video coding. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 948–954, 1997.
- [201] www.ecse.rpi.edu/Homepages/wrf/Research/ShortNotes/pn-poly.html. PNPOLY - Point Inclusion in Polygon Test. *Australasian Conference on Robotics and Automation (ACRA)*, 2008.
- [202] T. Xiang and S. Gong. Activity based surveillance video content modelling. *Pattern Recognition*, 41(7):2309–2326, 2008.
- [203] T. Xiang, M. Leung, and S. Cho. Expression recognition using fuzzy spatio-temporal modeling. *Pattern Recognition*, 41(1):204–216, 2008.
- [204] D. Xu, D. Tao, L. Zhang, X. Li, and H. Zhang. Human Gait Recognition With Matrix Representation. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(7):896–903, 2006.
- [205] R.R. Yager. On a general class of fuzzy connectives. *Fuzzy Sets and Systems*, 4(3):235–242, 1980.
- [206] R.R. Yager. Some procedures for selecting fuzzy set-theoretic operators. *International Journal of General Systems*, 8(2):115–124, 1982.
- [207] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE transactions on Systems, Man and Cybernetics*, 18(1):183–190, 1988.
- [208] R.R. Yager. Families of OWA operators. *Fuzzy sets and systems*, 59(2):125–148, 1993.

- [209] Y.D. Yu. Triangular norms and TNF-sigma-algebras. *Fuzzy Sets and Systems*, 16(3):251–264, 1985.
- [210] O. Brdiczka and P.C. Yuen, S. Zaidenberg, P. Reignier, and J.L. Crowley. Automatic acquisition of context models and its application to video surveillance. In *Proceedings of 18th International Conference on Pattern Recognition*, volume 1, pages 1175–1178, 2006.
- [211] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
- [212] L.A. Zadeh. From computing with numbers to computing with words. From manipulation of measurements to manipulation of perceptions. *Circuits and Systems I: Fundamental Theory and Applications*, 46(1):105–119, 1999.
- [213] W. Zajdel, A.T. Cemgil, and B. Krose. Dynamic bayesian networks for visual surveillance with distributed cameras. *Lecture Notes in Computer Science, Springer*, 4272:240–243, 2006.
- [214] F.B Zhan. Approximate analysis of binary topological relations between geographics regions with indeterminate boundaries. *Softcomputing*, 2:28–34, 1998.
- [215] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. Semi-Supervised Adapted HMMs for Unusual Event Detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:611–618, 2005.
- [216] W. Zhang, F. Chen, W. Xu, and E. Zhang. Real-time video intelligent surveillance system. In *IEEE International Conference on Multimedia and Expo*, pages 1021–1024, 2006.
- [217] Z. Zhang, A. Scanlon, W. Yin, L. Yu, P. Venetianer, and O. Inc. Video Surveillance using a Multi-Camera Tracking and Fusion System. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications-M2SFA2*, 2008.
- [218] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2(27):406–413, 2004.
- [219] H. Zhong, J. Shi, and M. Visontai. Detecting Unusual Activity in Video. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:819–826, 2004.

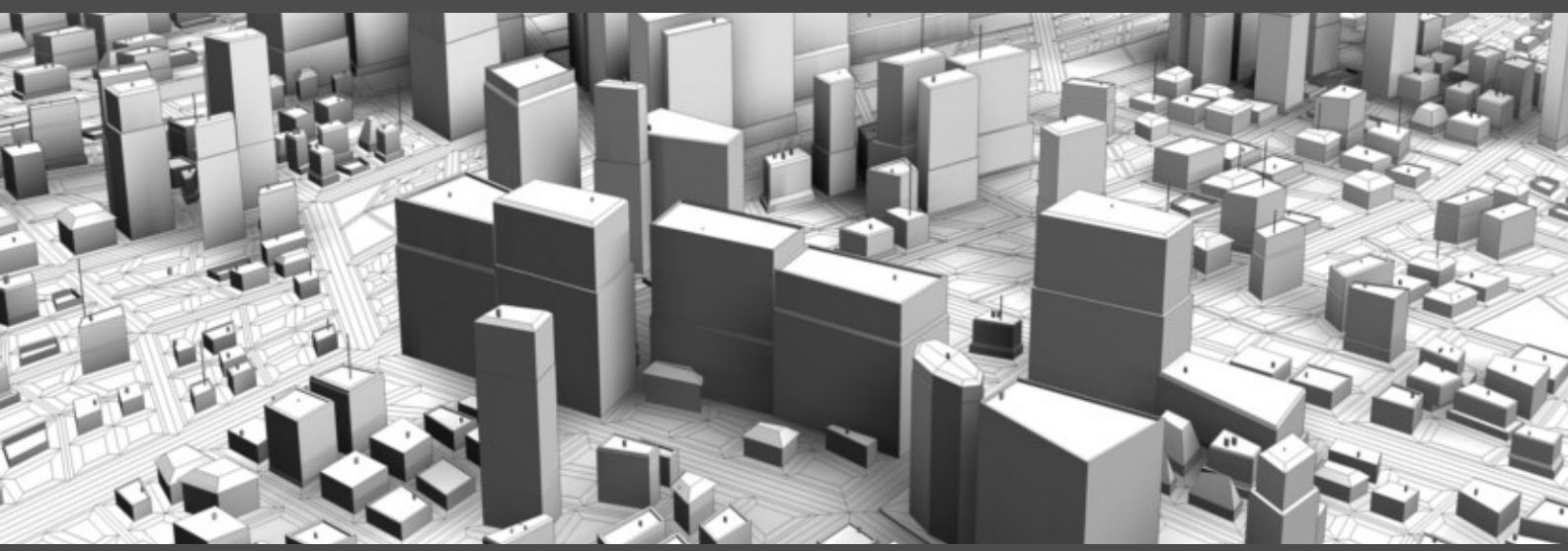


Anexo



A

Anexo





Herramientas Desarrolladas

En este Anexo se describen las herramientas de adquisición de conocimiento y monitorización que se han desarrollado y que forman parte del sistema de vigilancia inteligente propuesto en el presente trabajo.

A.1. Adquisición de conocimiento

Este tipo de herramientas facilitan la adquisición del conocimiento que los componentes de normalidad necesitan para llevar a cabo el análisis de eventos y conductas. Sin una base de conocimiento apropiada los componentes de análisis serían incapaces de identificar los eventos que suceden en el entorno observado y mucho menos diferenciar si éstos son normales, o por el contrario, anómalos. Como se comentó en capítulos anteriores, existen diferentes tipos de herramientas de adquisición de conocimiento como, por ejemplo, aquellas que permiten al experto introducir y modificar el conocimiento de forma manual, o bien, herramientas que adquieren dicho conocimiento de forma automática. En este trabajo se han puesto en práctica ambas modalidades, ya que se ha desarrollado una herramienta con la que el experto puede definir las zonas relevantes y las trayectorias normales del entorno observado en base a un conjunto de restricciones y, por otro lado, se ha implementado una máquina de aprendizaje genérica que permite la generación de reglas difusas, para ser empleadas posteriormente en procesos de clasificación. El uso exclusivo de algoritmos de aprendizaje automático no es aconsejable ya que, en muchas ocasiones, el conocimiento adquirido puede que no sea del todo correcto y debe ser depurado para evitar errores en el sistema. Por este motivo, es apropiado combinar estas técnicas de adquisición de conocimiento con herramientas que permitan al experto introducir nuevo conocimiento o modificar el existente. En las sucesivas secciones se describen cada una de

las herramientas desarrolladas.

A.1.1. Herramienta para la caracterización de entornos

Esta herramienta, desarrollada con el lenguaje de programación JAVA, ofrece las siguientes funcionalidades:

- Especificación del conjunto de zonas y la topología de la escena a partir de una imagen captada por una cámara de vigilancia. En la topología es donde se definen las conexiones existentes entre las diferentes zonas del entorno.
- Definición de clases de objetos a los que el sistema debe prestar especial atención. Por ejemplo, peatones y vehículos.
- Definición de intervalos de tiempo para ser empleados en las restricciones temporales, donde se especifica un instante de tiempo inicial y otro final.
- Definición de trayectorias normales. La definición de trayectorias implica la creación y asociación de restricciones de rol, espaciales y temporales.

La herramienta permite al usuario trabajar directamente sobre una imagen capturada por una cámara de seguridad y dibujar sobre ella las diferentes zonas (Figura A.1). Los trazos de las zonas se dibujan con el uso del ratón mientras se mantiene pulsado el botón izquierdo, o bien, con simples “clicks” del botón izquierdo (en este último caso se unen mediante una recta el último punto dibujado sobre la imagen y el nuevo, el cual corresponde con la posición en la que se encuentra el puntero en el momento de presionar el botón). Para que finalmente la zona pase a formar parte de la base de conocimiento asociada a la escena, el usuario debe especificar un identificador único. Además, la herramienta permite especificar si existe alguna conexión entre las zonas definidas y para ello se utilizan los identificadores. El conjunto de estas conexiones forman un grafo dirigido que constituye la topología de la escena.

Normalmente, el número de zonas y trayectorias a definir en la mayoría de las escenas no es demasiado elevado, si además se hace uso de una herramienta de adquisición de conocimiento, el tiempo necesario desde que se implanta una nueva cámara hasta que el componente se encuentra en perfecto funcionamiento es razonablemente pequeño. En aquellas escenas donde el número de zonas y trayectorias sea considerablemente grande, o las trayectorias puedan variar con cierta facilidad ante variaciones ambientales, es aconsejable el uso de algoritmos de aprendizaje automático en lugar de herramientas de adquisición de conocimiento.

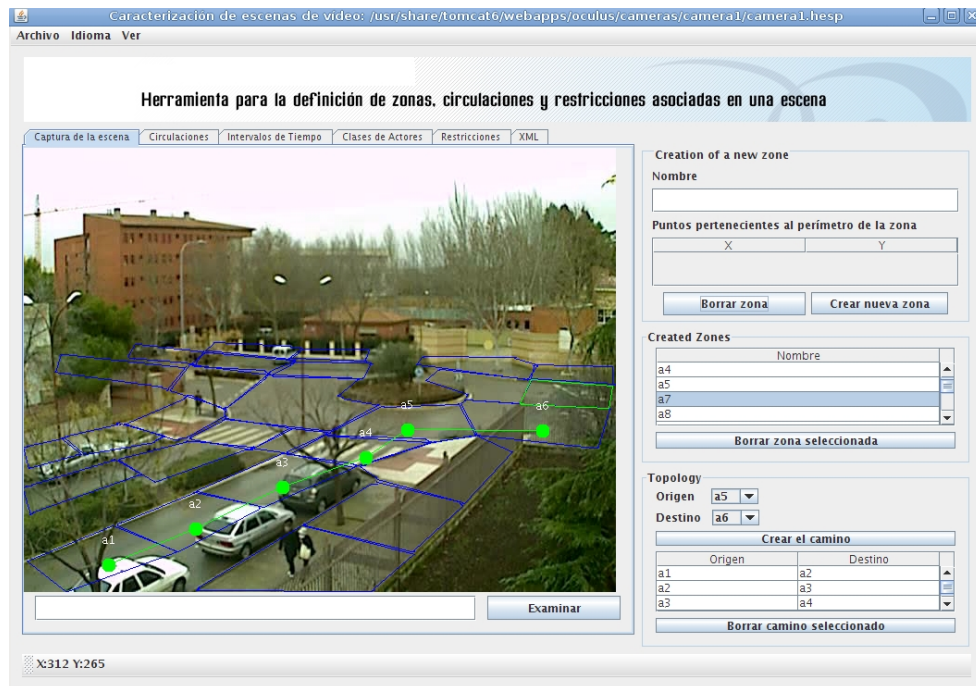


Figura A.1: Captura de la herramienta de adquisición de conocimiento donde se muestra la parte de la interfaz dedicada a la definición de zonas y especificación de la topología de la escena.

Por otro lado, si una trayectoria se define como una secuencia de zonas, es importante tener en cuenta dos aspectos relevantes a la hora de definir las:

- Si se definen zonas de gran tamaño, la secuencias de zonas que representan las trayectorias estarán formadas por un número reducido de elementos. Este hecho supone una menor carga computacional, ya que el sistema debe comprobar un número menor de zonas. Sin embargo, podría representar incorrectamente el movimiento de un objeto, ya que éste podría realizar múltiples cambios de sentido y variaciones en su movimiento sin necesidad de salir de la zona y no ser detectado por el sistema.
- Si por el contrario, las zonas son demasiado pequeñas, el sistema reconoce fielmente el camino recorrido por un objeto, sin embargo, el coste computacional es mucho mayor.

En base a los dos criterios anteriores, es decisión del experto que hace uso de la herramienta elegir el número y tamaño de zonas apropiado para cada escena. Normalmente se tiene en cuenta el tamaño medio de los objetos y se evita, en la mayor medida de los posible, cambios de dirección y desplazamientos significativos sin intersectar con las zonas vecinas.

Zonas

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<scene>

<image>/home/javier/prueba/v03t01/00000.jpg
</image>
  <region>
    <name>Entrada_Norte</name>
    <coordinate>
      <x>591</x>
      <y>102</y>
    </coordinate>
    ....
    <coordinate>
      <x>587</x>
      <y>102</y>
    </coordinate>
  </region>
```

Intervalos de tiempo

```
<interval>
<interval_name>horario_diurno</interval_name>
<interval_beginning>09:00:00:00:0000</interval_beginning>
<interval_end>14:00:00:00:0000</interval_end>
</interval>

<interval>
<interval_name>horario_nocturno</interval_name>
<interval_beginning>00:00:00:21:00:0000</interval_beginning>
<interval_end>00:00:00:21:00:0000</interval_end>
</interval>
....
```

Topología

```
<topology>
<path>
<beginning_path>Entrada_Norte</beginning_path>
<end_path>Carretera</end_path>
</path>
<path>
<beginning_path>Carretera</beginning_path>
<end_path>Jardin_Oeste</end_path>
</path>
.....
```

Clases de objetos

```
<types_actor>
  <actor>Car</actor>
  <actor>Motorbike</actor>
  <actor>Truck</actor>
  <actor>People</actor>
</types_actor>
```

Restricciones

```
<spatial_restriction>
<spatial_current_movement>C16</spatial_current_movement>
<allowed_region>Acceso_Edificio_Central</allowed_region>
<allowed_region>Jardin_Oeste</allowed_region>
</spatial_restriction>
<actor_restriction>
<actor_current_movement>C10</actor_current_movement>
<allowed_actor>People</allowed_actor>
</actor_restriction>
....
```

Figura A.2: Fragmentos de código XML, que representan parte del conocimiento de una escena, generados con la herramienta de adquisición de conocimiento

Toda la información sobre la escena generada con la herramienta de adquisición de conocimiento se almacena en un fichero XML, que utilizará el sistema para construir la base de conocimiento. La Figura A.2 muestra algunos ejemplos de fragmentos de código XML donde se almacena información de la escena relacionada con las zonas, topología, clases de objetos, intervalos de tiempo y restricciones.

A.1.2. Herramienta de aprendizaje automático

Esta herramienta, al igual que la anterior, también ha sido desarrollada haciendo uso del lenguaje de programación Java. Una de las principales ventajas y que a su vez ha motivado la elección de este lenguaje, es la gran portabilidad que ofrece ya que, al ser un lenguaje interpretado, cualquier sistema que disponga de una máquina virtual de JAVA puede interpretar aplicaciones desarrolladas en este lenguaje. La herramienta recibe como entrada un fichero de datos donde se representa el conjunto de entrenamiento a partir del cual se genera el conjunto de reglas difusas y, de manera optativa, el fichero donde se definen los dominios de definición para cada una de las variables. A continuación y a modo de ejemplo se muestra un ejemplo del fichero de entrada donde se especifican las variables y sus dominios de definición:

Definición de variables y dominios de definición

```

1  #XPOS
2  VL:-10,-10,100,150
3  L:100,150,200,250
4  C:200,250,350,400
5  R:350,400,500,550
6  VR:500,550,650,650
7  #YPOS
8  VU:-10,-10,70,100
9  U:70,100,190,210
10 C:190,210,280,300
11 D:280,300,390,410
12 VD:390,410,500,500
13 #Ancho-Alto
14 VS:0,0,0.1,0.4
15 S:0.1,0.4,0.8,0.95
16 E:0.8,0.95,1.05,1.15
17 L:1.05,1.15,2,2.5
18 VL:2,2.5,6,6
19 #Hspeed-Vspeed
20 VS:0,0,0.1,0.4
21 S:0.1,0.4,0.8,0.95
22 E:0.8,0.95,1.05,1.15
23 L:1.05,1.15,2,2.5
24 VL:2,2.5,6,6
25 #MOV
26 VS:0,0,0.1,0.3
27 S:0.1,0.3,1,1.2
28 SM:1,1.2,5,9
29 M:5,9,13,16
30 FM:13,16,19,23
31 F:19,23,30,34
32 VF:30,34,100,100
    
```

En el ejemplo anterior se puede apreciar como cada una de las variables se especifica anteponiendo el símbolo “#” y a continuación el conjunto de etiquetas lingüísticas que forma parte del dominio de cada una de ellas. Cada una de las etiquetas representa un conjunto difuso cuyos valores límite se muestran a continuación de cada etiqueta separado mediante comas. Como alternativa, la herramienta ofrece al usuario la posibilidad de indicar el número de variables y etiquetas en lugar de especificar un fichero de entrada con la definición. Para cada una de las variables es necesario especificar el valor máximo y mínimo que puede tomar, de tal forma que el sistema define los conjuntos en el rango *valor máximo - valor mínimo* de manera uniforme.

En cuanto al fichero de datos que representa el conjunto de entrenamiento existen dos posibles formatos:

- El primero de ellos especifica las clases y los elementos que pertenecen a cada una de las clases. Posteriormente el usuario puede especificar en la herramienta el porcentaje de ejemplos que se destina al proceso de aprendizaje y al proceso de clasificación para evaluar las reglas generadas (80 % de ejemplos para aprender y 20 %, como configuración por defecto). Tanto el conjunto de ejemplos de aprendizaje como el de

clasificación se genera de forma aleatoria.

- En la segunda alternativa se especifican directamente las clases a las que pueden pertenecer los ejemplos así como el grupo destinado al proceso de aprendizaje y el de clasificación. A continuación se muestra un ejemplo en el que se construye un conjunto de entrenamiento donde se especifica los ejemplos que se deben emplear en el proceso de aprendizaje y clasificación para cada una de las clases.

Conjunto de Entrenamiento

```
1  #Pedestrian
2  @Training
3  253,304,0.69,1,0
4  323,319,0.63,1,1.41
5  327,323,0.63,1,1
6  326,320,0.63,0.5,2.24
7  ...
8  @Test
9  334,317,0.67,1,1
10 334,318,0.6,1,0
11 334,320,0.6,0.5,2
12 ...
13 #Vehicle
14 @Training
15 157,243,1.87,2,0
16 158,243,2,1,9.22
17 161,244,2.07,3,0
18 163,245,2,1,4
19 ...
20 @Test
21 145,240,1.86,1,3.16
22 146,241,1.93,1,1
23 151,242,1.93,5,0
24 ...
```

Además, la herramienta permite ajustar otros parámetros importantes tales como la distancia máxima de amplificación permitida (ver Sección 3.4.1), número mínimo de veces que se debe repetir una regla para ser considerada y pasar al conjunto de reglas definitivas (aquellas que se repiten un número inferior al indicado, se pueden considerar como ruido producido en el proceso de aprendizaje) o la ampliación según el valor de varianza, entre otros. En la Figura A.3 se puede apreciar la parte de la interfaz donde se pueden configurar estos parámetros.

Una vez finalizado el proceso de aprendizaje, el usuario puede observar y analizar detenidamente el proceso de etiquetado en el que los valores numéricos de los ejemplos de entrada han sido sustituidos por etiquetas lingüísticas, formando así el conjunto de reglas iniciales. Tal como se describió en la Sección 3.4.1, los ejemplos con diferentes valores numéricos pero etiquetados de la misma forma, es decir, formados por las mismas etiquetas lingüísticas, representan la misma regla inicial. La herramienta tiene en cuenta el número de ejemplos que dan lugar a una regla inicial para ordenarlas de mayor a menor según el número de repeticiones. Aquellas

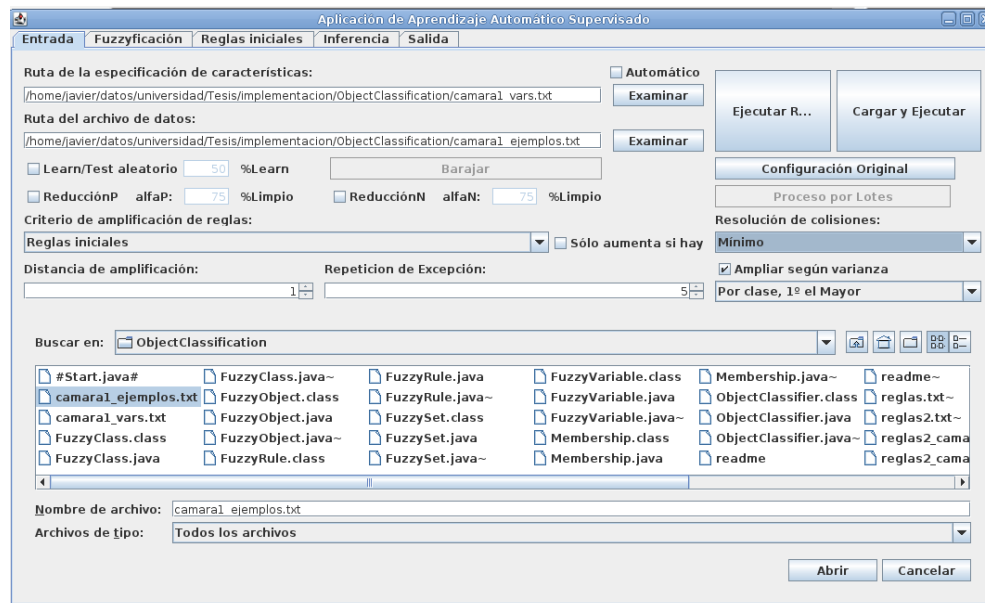


Figura A.3: Parte de la interfaz gráfica de la herramienta de aprendizaje automático donde se establecen los parámetros de configuración

reglas iniciales que se repiten con mayor frecuencia son las que se amplían en primer lugar. Además, la herramienta muestra el proceso de inferencia completo hasta que finalmente se obtienen el conjunto de reglas definitivas. Estas reglas se almacenan en un fichero de datos que podrá ser utilizado por cualquier sistema para incorporar y utilizar el conocimiento adquirido.

A.2. Herramientas de Monitorización

Las herramientas de monitorización son las que permiten al usuario observar el entorno vigilado y el análisis llevado a cabo por el sistema de seguridad formado por componentes de normalidad. El núcleo de razonamiento formado por los dos componentes de normalidad descritos en los capítulos 4 y 5, así como el módulo de fusión encargado de obtener un análisis global, es independiente de los componentes visuales encargados de mostrar la información. Es decir, se ha diseñado un sistema multicapa donde la forma en la que se procesan y se obtienen los datos es independiente de su representación. En este trabajo se han desarrollado dos herramientas de monitorización para mostrar los resultados: i) una herramienta de monitorización web que permite al usuario observar *frame* a *frame* los datos generados en el proceso de análisis, con el fin de comprobar el correcto funcionamiento del sistema; ii) y una herramienta que permite generar vídeos, en cuyos *frames* se añaden información relativa al análisis.

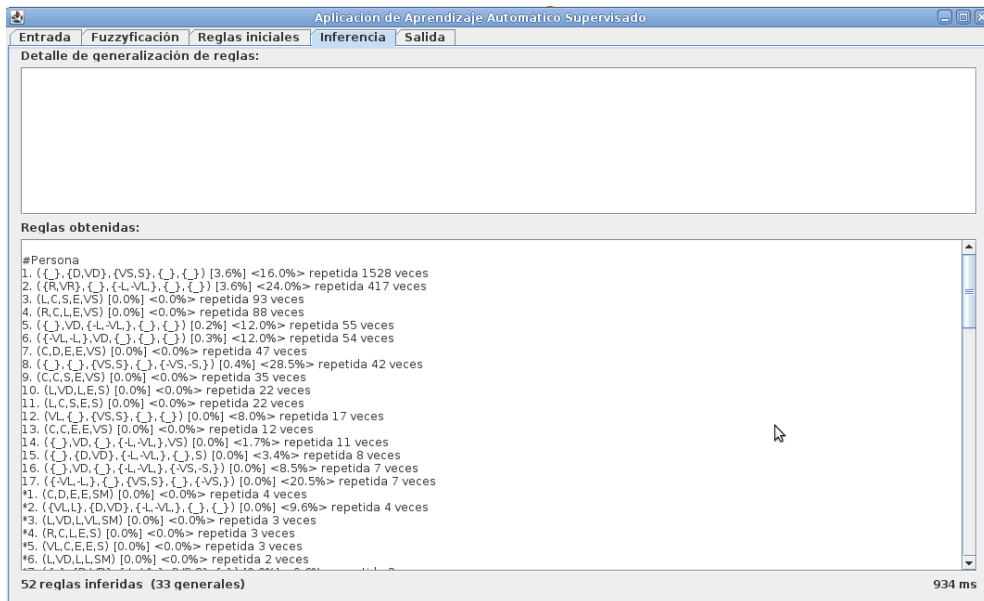


Figura A.4: Parte de la interfaz gráfica de la herramienta de aprendizaje automático donde se muestran las reglas obtenidas tras el proceso de aprendizaje.

A continuación se describe cada una de estas herramientas.

A.2.1. Herramienta web de monitorización

Esta herramienta, llamada OCULUS, no fue ideada únicamente para desempeñar las funciones básicas de monitorización, sino que también con la intención de proporcionar una herramienta que permitiera comprobar el correcto funcionamiento del núcleo encargado de realizar el análisis de eventos. Además, al haber optado por su desarrollo mediante tecnologías web, cabe destacar como gran ventaja su uso desde cualquier localización geográfica con acceso a internet. Tal como se comentó anteriormente, la capa encargada de mostrar los resultados (donde se encuentra OCULUS) y la capa donde se encuentra el núcleo de razonamiento son independientes, de tal forma que pueden existir diferentes medios o formas de visualización sin repercutir en la lógica del sistema. En la Figura A.5 se puede apreciar la interfaz de OCULUS en la que se pueden distinguir siete partes importantes:

1. Panel de monitores donde cada uno de ellos monitoriza una escena diferente. El usuario puede activarlos o desactivarlos simplemente pulsando sobre ellos.
2. Visualización del entorno vigilado en el que los objetos detectados por el sistema se marcan mediante una elipse.

3. Resultado del proceso de segmentación sobre el *frame* actual, en el que se marcan con píxeles de color blanco aquellas regiones que el sistema considera que pertenecen a un objeto en movimiento.
4. Histograma donde se reflejan las características de iluminación del *frame* actual.
5. Resultado del análisis global a partir de la información recopilada en el *frame* actual junto con la información obtenida en *frames* anteriores. Según el valor de normalidad obtenido, la valoración sobre el estado actual del entorno puede ser absolutamente normal, posiblemente normal, sospechosa, posiblemente anómala y absolutamente anómala.
6. Los controladores que se encuentran en la parte derecha de la interfaz permiten al usuario desplazarse o dirigirse a cualquiera de los *frames* del vídeo que está siendo analizado. En concreto, el usuario puede avanzar *frame* a *frame* (hacia delante o detrás) y observar detenidamente los resultados procedentes del núcleo de razonamiento, reproducir la escena de manera fluida o bien, dirigirse directamente a un *frame* a partir de su identificador. Bajo los controladores de reproducción del flujo de vídeo se encuentra la información de salida de los módulos de preprocesamiento, así como de los componentes de normalidad y el módulo encargado de realizar el análisis global.
7. Finalmente, en la parte inferior derecha se muestra la definición de zonas del entorno observado, realizada mediante el uso de las herramientas de adquisición de conocimiento. Esta especie o clase de mapa, permite al usuario comprobar si el sistema está estimando correctamente las zonas en las que se encuentra cada elemento detectado y las trayectorias que están siguiendo.

En cuanto a la salida de los preprocesadores, componentes de normalidad y módulo de análisis global el usuario puede comprobar la siguiente información para cada uno de los *frames*:

- Figura A.6 (a): Localización de cada objeto detectado, así como las características de las elipses que los envuelven, la herramienta también muestra el desplazamiento vertical, horizontal y global del objeto con respecto al último *frame*, la estimación de zonas en las que se puede encontrar cada objeto y el grado de pertenencia de cada uno de ellos a las clases peatón y vehículo.
- Figura A.6 (b): Salida del segundo componente de normalidad encargado de analizar la velocidad que alcanza cada objeto detectado por el sistema. Este módulo de visualización muestra el grado de normalidad asociado a cada objeto de acuerdo al concepto velocidad, así como las instancias de velocidad normal que podrían satisfacer cada uno de los objetos.

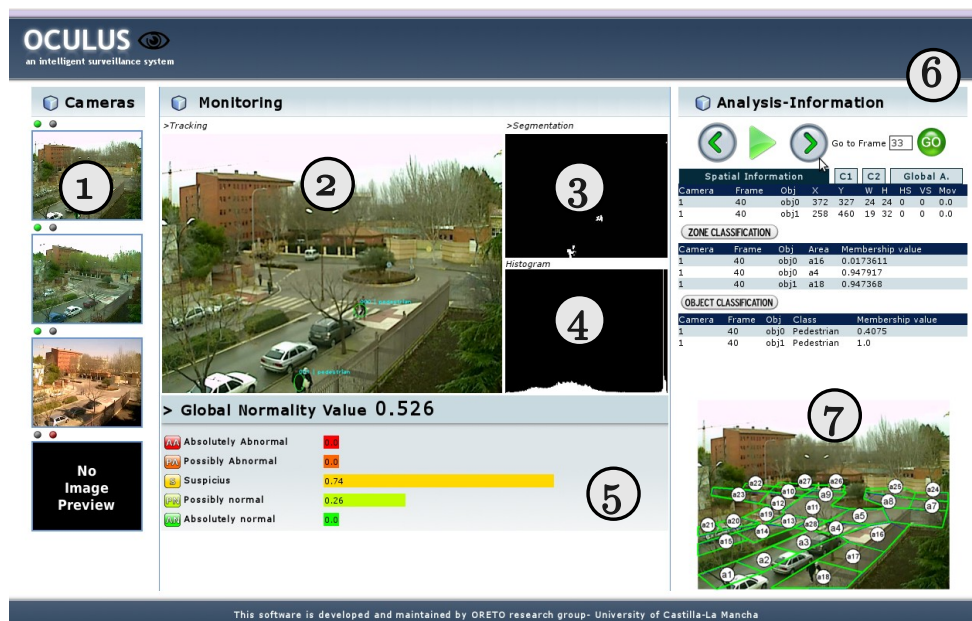


Figura A.5: Captura de la interfaz del sistema de monitorización OCULUS.

- Figura A.6 (c): Salida del primer componente de normalidad encargado de analizar las trayectorias que siguen los objetos detectados por el sistema. Al igual que en el componente anterior, la herramienta muestra el grado de normalidad asociado a cada objeto de acuerdo al concepto trayectoria normal, las posibles trayectorias que podrían estar siguiendo y el grado de satisfacción de cada una de ellas. Este grado de satisfacción asociado a cada trayectoria se calcula en base a el grado de cumplimiento de la restricción de rol (RS), restricción espacial (SS) y restricción Temporal (TS).
- Figura A.6 (d): En el último bloque se muestra el análisis global para cada uno de los objetos en donde se utiliza un operador OWA para combinar las salidas de los componentes de normalidad.

A.2.2. Herramienta para la generación de vídeos analizados

Cómo medio de visualización alternativo a la herramienta web descrita en la sección anterior, se ha desarrollado una herramienta que permite generar vídeos en los que quedan reflejados los resultados del análisis de normalidad. En este caso, la cantidad de información que se muestra con respecto a la herramienta anterior es inferior, ya que la inclusión excesiva de información en cada uno de los *frames* impediría observar correctamente cada uno de los elementos que se mueve en el entorno. La herramienta ha sido desarrollada con el lenguaje de programación Python y la librería PyGame que permite

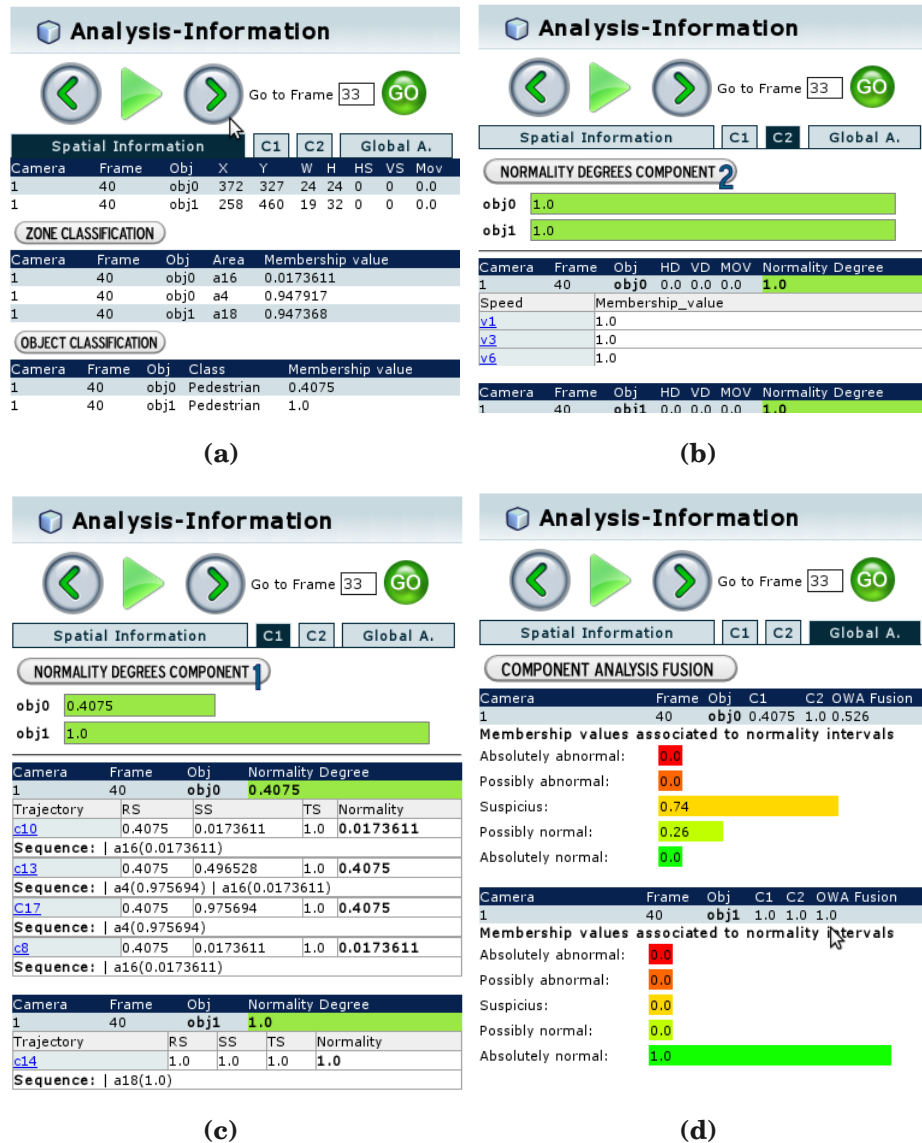


Figura A.6: Parte derecha de la herramienta de monitorización donde se muestra la información de salida de los componentes de normalidad y los módulos de procesamiento.

dibujar texto y cualquier forma geométrica sobre una imagen, en este caso, sobre la secuencia de imágenes que constituyen el vídeo a analizar.

A modo de ejemplo, en la Figura A.7 se muestra un *frame* de un vídeo generado con esta herramienta en el que se incluye información relativa al análisis de normalidad. En términos generales, la información que se muestra en los vídeos generados es la misma que se puede apreciar con

la herramienta web descrita en la sección anterior, aunque como se ha comentado anteriormente, de una forma simplificada. A continuación se describe de forma detallada la información que se muestra en cada uno de los *frames* de los vídeos generados:

- En la parte superior izquierda se muestra la pertenencia del valor de normalidad global a los intervalos: absolutamente normal, posiblemente normal, sospechoso, posiblemente anómalo y absolutamente anómalo. Cada una de las barras horizontales asociadas a cada intervalo tiene asignado un color y cuanto mayor sea su longitud, mayor es la pertenencia del valor global de normalidad a dicho intervalo.
- En la parte derecha se muestra un valor numérico que representa el grado de normalidad global. Cuando el sistema considera que la situación global no es normal, el color blanco de la fuente con la que está escrita el valor numérico se convierte en color rojo, con el objetivo de llamar la atención del usuario de la herramienta. Este no es el único indicador de situaciones anómalas sino que también se muestra un icono que representa un ojo en la parte inferior derecha, tal como muestra la Figura A.8.



Figura A.7: Ejemplo de un *frame* perteneciente a uno de los vídeos generados con la herramienta descrita en esta sección, en el que se incluye información relativa al análisis de normalidad.

- Finalmente, en las Figuras A.7 y A.8 se puede apreciar como cada uno de los objetos detectados por el sistema es marcado mediante una elipse. Al lado de cada una de ellas aparece la siguiente información: (a) identificador del objeto, (b) grado de normalidad asociado al objeto, (c) clases a las que pertenece el objeto y grados de pertenencia a cada una de ellas. A la derecha de cada elipse y bajo la información mencionada anteriormente, aparecen barras horizontales similares a las de la parte superior izquierda de la interfaz, que representan la pertenencia del valor de normalidad del comportamiento del objeto a los intervalos absolutamente anómalo (primera barra o barra de color rojo), posiblemente anómalo (segunda barra y color naranja), sospechoso (tercera barra y color amarillo), posiblemente normal (cuarta barra y color verde claro) y absolutamente normal (quinta barra y color verde oscuro).



Figura A.8: Ejemplo de un *frame* perteneciente a uno de los vídeos generados con la herramienta descrita en esta sección, en donde la situación global del entorno no es normal.

En definitiva, en cada uno de los *frames* que forman parte de la secuencia de vídeo se puede apreciar el análisis individual realizado para cada uno de los objetos detectados, así como el análisis global fruto de la combinación de los análisis particulares. Sin embargo, el nivel de detalle con el que se muestra la información no permite al usuario conocer la salida de cada uno de los componentes, las zonas en las que se podría encontrar cada objeto y, las posibles trayectorias y velocidades asociadas a cada uno de ellos. A pesar

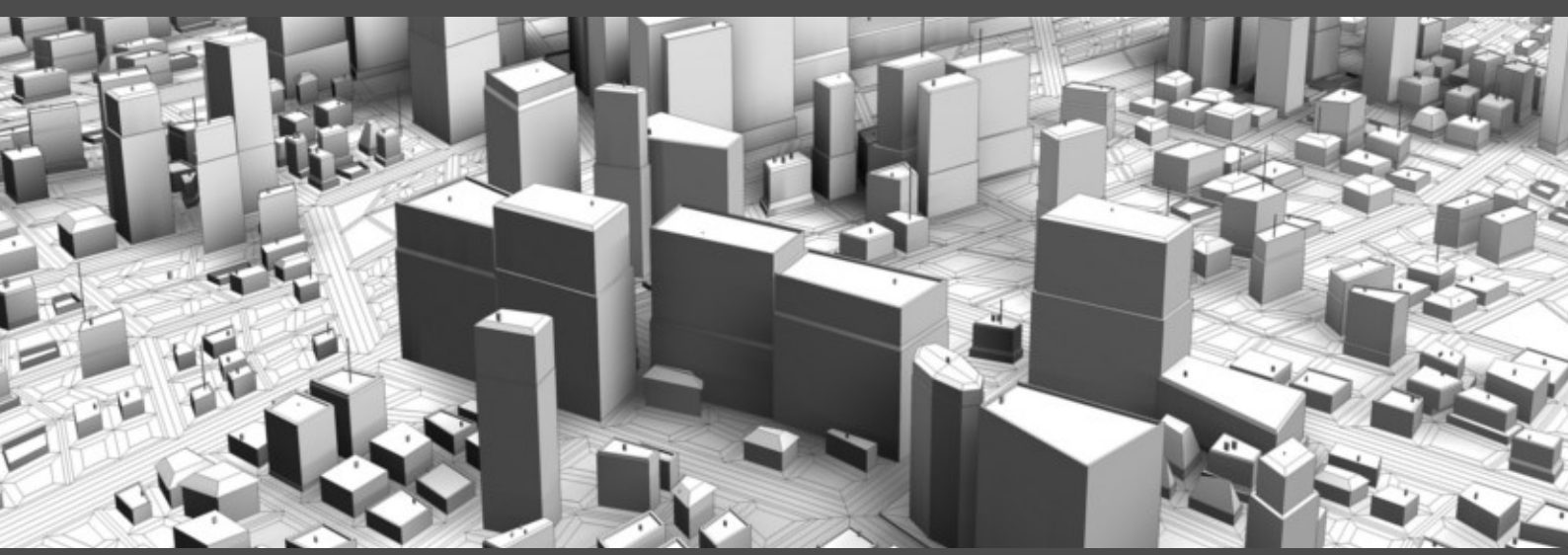
de la omisión de esta información se cumple el principal objetivo en tareas de vigilancia: mostrar de una forma clara si el comportamiento de cada objeto es el adecuado y el estado actual del entorno es normal.



Appendix

B

TR Appendix



B

A Scalable Approach based on Normality Components for Intelligent Surveillance

This appendix comprises a summary of the thesis work written in English. The thesis is organised as follows:

- **Chapter 1.** (*Introduction*). This chapter introduces the problem of intelligent surveillance, sets the main goals of the work and gives details of the structure of the document.
- **Chapter 2.** (*State of the Art*). This chapter positions the work in the context of relevant proposals and research areas.
- **Chapter 3.** (*Formal model based on Normality Components*). In this chapter, we present a formal model based on normality components to build scalable and flexible surveillance systems. These components need expert knowledge to be deployed in particular environments. An inductive learning algorithm is described in this chapter to automatically acquire this knowledge. The chapter concludes with a briefly introduction about how a multi-agent architecture can give support to the components deployed through formal model.
- **Chapter 4.** (*Normality Component for Trajectory Analysis*). Application of the formal model presented in chapter 3 in order to build a normality component for trajectory analysis.
- **Chapter 5.** (*Normality Component for Speed Analysis*). Application of the formal model presented in chapter 3 in order to build a normality component for speed analysis.
- **Chapter 6.** (*Results*). This chapter presents and discusses the results obtained from every one of the ten tests (each test is a scene to be

analysed). You can find the analysed videos on the CD attached to the documentation.

- **Chapter 7.** (*Conclusions and Future Work*). This chapter is written in English.
- **Appendix A.** Description of the knowledge acquisition and monitoring tools developed for this thesis.
- **Appendix B:** summary of this thesis (written in English).

A Scalable Approach based on Normality Components for Intelligent Surveillance

Abstract Since their first developments, traditional video surveillance systems have been designed to monitor environments. However, these systems have several limitations to automatically understand events and behaviours without human collaboration. In order to overcome this problem, intelligent surveillance systems arise as a possible solution. This kind of systems are not affected by negative factors such as fatigue or tiredness and they can be more effective than people when recognising certain kinds of events, such as the detection of suspicious or unattended objects. Intelligent surveillance refers to using Artificial Intelligence and Computer Vision techniques in order to improve traditional surveillance and process semantic information, obtained from low-level security devices. Normally these systems consist of a set of independent analysis modules that deal with particular problems, such as the trajectory analysis of pedestrian in parking lots, speed estimation of vehicles, gait or facial recognition, etc. However, most of them present a common problem: lack of flexibility and scalability to include new kinds of analysis and combine all of them in order to obtain a global interpretation. In this thesis, a formal model to define normal events and behaviours in monitored environments and to build scalable surveillance systems is presented. This model is based on the use of “normality components”, which are independent and reusable for environments with different characteristics and different kinds of objects. Each component specifies how an object should ideally behave according to a surveillance aspect, such as trajectory or velocity. The model also includes the fusion mechanisms required for combining the particular analysis made by each component. Finally, when a new component is designed making use of the proposed model, the system increases its abilities to detect new kind of abnormal events, and the normality of an object depends on a higher number of factors.

1 Introduction

Currently, the security of citizens and infrastructures in public and private environments is one of the main topics of interest for most governments and institutions. One representative example is London, where more than four million cameras have been deployed and connected to security control centres [14], where human operators supervise the video streams received in order to detect anomalous situations that might represent a danger for people. The main disadvantage of this kind of surveillance is the huge dependence on human supervisors, which usually involves an efficiency decrease after observing monitors for long periods of time, as stated in previous research [31]. Furthermore, there exist surveillance activities, such as the detection of unattended objects, where the automatic systems can be more effective than human surveillance [14]. Therefore, traditional video surveillance has several limitations to meet the security demands of monitored environments. In other words, the security systems should be driven by intelligence and proactiveness.

In this way, intelligent surveillance systems need to be developed in order to improve the robustness of traditional systems and reduce the workload of security staff [7], calling their attention only at particular moments. In the last two decades, the research community has proposed different algorithms, techniques and models to develop the named second and third generation of surveillance systems [32]. Second generation systems combine CCTV (closed-circuit television) technology and Computer Vision and Artificial Intelligence methods, while third generation surveillance systems provide solutions for highly distributed and complex environments.

There exist different proposals of intelligent surveillance systems such as the analysis of trajectories of vehicles and pedestrians in parking lots, the access control in buildings, the interaction among human beings in underground railway stations, the activities carried out in indoor environments, etc. Most of these systems have normally been designed to solve particular problems in specific scenarios. They can be understood as a set of pieces that act separately to detect specific events in particular scenarios, without establishing relationships between them to get a global analysis. Moreover, these proposals are not usually reused in scenarios different from the one they were initially considered. Within this context, a proposal of a scalable and flexible general model for the development of surveillance systems is needed, which allows to deploy instances and is adaptable to most of monitored environments. This model should also facilitate the analysis of different events of interest depending on the surveillance requirements, increasing the capability of the system when new aspects need to be monitored. In short, an advance surveillance system should be able to include new analysis modules that can be customised for the scenario where they will be deployed, determining the normality or abnormality of such a scenario at every moment.

The discussed limitations motivate the present work, where a formal model for the design and normality analysis of surveillance environments is proposed. This model is inspired by the component-based software development,

the multi-layered design and the division of complex problems into subproblems. According to this model, an intelligent surveillance system consists of a set of normality components and a set of methods to combine their output. A normality component defines how an object should ideally behave according to the monitored events of interest, considering a non-normal situation as suspicious or anomalous. These components can be easily integrated into the surveillance system and customised depending on the requirements and security level desired. In addition, knowledge acquisition tools and machine learning techniques are used to configure and deploy the components.

The output of the components designed by means of the proposed model are highly descriptive so that the security expert responsible for interacting with the surveillance system can easily check whether the system works correctly and identifies anomalous or suspicious situations. This feedback, close to natural language, facilitates the system maintenance and the reuse of knowledge in environments with similar features. On the other hand, the security systems should provide responses in a short time as possible to reduce the hypothetical damage caused by the occurrence of anomalous situations (e.g. crowd detection). This is the reason why the systems developed through the proposed model must provide a good balance between response time and robustness.

The rest of the chapter is organised as follows: the next section discusses previous work and their shortcomings. Section 3 describes a formal model to build scalable and flexible surveillance systems. Section 4, the model previously described is applied to design a new normality component, which deals with the trajectory analysis problem. The experimental results are presented and discussed in Section 5. The final section presents conclusions and future works.

2 Previous Work

In the last decade several research projects have been developed in order to achieve advances in the area of intelligent surveillance. Some of the problems addressed in these projects are moving object detection, tracking, object classification (usually between pedestrians and vehicles), automatic learning of activity models to understand events and behaviours, or support for decision making.

One relevant research project was CROMATICA (*Crowd Monitoring with Telematic and Communication Assistance*) [19], conceived with the goal of improving the surveillance of passengers in public transport. This project combined video analysis-based technologies and wireless data transfer [32]. Such a project was later extended with PRISMATICA (*Pro-active Integrated Systems for Security Management by Technological Institutional and Communication Assistance*) [33], where the social, ethic, organisational and technical aspects of surveillance in public transport were especially taken into account [32]. Regarding the technical advances, the system had a distributed nature and allowed the communication between heterogeneous devices by means of

the middleware CORBA. Some of the anomalous events detected by the system were people staying at forbidden areas, intrusions (access to restricted areas), anomalous length of queues, or people walking in the wrong direction. In these systems, the analysis and interpretation of events and behaviours is carried out in indoor environments and mainly focused on people.

Another relevant research project was VSAM (*Video Surveillance and Monitoring*) [13], funded by U.S. Defence Department, concretely by DARPA (*Defence Advanced Research Projects Agency*). The system used the video streams gathered by the security cameras to classify moving objects as people, groups of people or vehicles. The layer responsible for identifying events was designed to analyse vehicle trajectories and people movements. Thanks to the study of the people walk and their poses, the system was able to determine if a person was walking, running or performing other activities.

W⁴ [16] was a system focused on analysing people behaviour from their poses in outdoor environments, ignoring the behaviour of elements such as vehicles. The system detected and determined the people figure by using the images obtained by security cameras and the information captured by infrared sensors. Once obtained the figure, a second module identified relevant parts of the human body (head, trunk, arms, and legs). The position of these parts was used to identify events of interest such as people running, walking with a bag, carrying objects, putting their hands up (which might involve an external thread), etc.

ADVISOR (*Annotated Digital Video for Intelligent Surveillance and Optimised Retrieval*) [30] was another European project developed with the goal of analysing people behaviour in subway stations. Like PRISMATICA, this system is composed of multiple cameras distributed around a station, connected to a central server that performs the image analysis. This module stores in a database the sequence of images that represent an anomalous situation. Each one of these sequences are described by means of textual information. Precisely, this information is later used by human operators to retrieve the previously identify abnormal events.

AVITRACK [5, 8, 4], Co-Friend [12] and ARGOS [6] are more recent research projects. The first two are related to CROMATICA and PRISMATICA while Co-Friend evolves from AVITRACK. Such projects aim to analyse people and vehicles behaviour at airports, especially in maintenance tasks such as load and unload of aeroplanes, refuel, single repairs, etc. On the other hand, ARGOS was conceived to deal with the control of sea traffic.

Most of previously cited works are composed of multiple analysis modules designed in an independent way without following a common scheme. These systems can be understood as a set of pieces that act separately to detect specific events in particular scenarios, without establishing relationships between them to get a global analysis. Nevertheless, the tendency when designing intelligent surveillance system should be towards the development of scalable systems that provide mechanisms to include new analysis modules. Furthermore, this integration of components should not affect the already deployed ones, making easy their combination to increase the ability for the system to perform a global analysis of the environment.

3. Formal Model to Build Scalable and Flexible Surveillance Systems | 253 |

Another existing limitation to overcome is the low flexibility when configuring the behaviour analysis applied to each class of object. A flexible surveillance system should give the possibility of easily specifying what analysis modules monitor each class of object, according to the characteristics of the environment and the surveillance demands. Thus, if a system is composed of two modules to analyse trajectories and speed, to choose which module is applied to monitor people and vehicles should be allowed, fact that may vary for different environment. For these reasons, and as stated in the previous section, the main motivation of this work is to define a model that allows to build scalable and flexible surveillance systems, which will be discussed in depth in the next sections.

3 Formal Model to Build Scalable and Flexible Surveillance Systems

3.1 General Overview

The complexity of surveillance systems can be reduced by adopting a multi-layered approach in the design and development phases, providing the system with high flexibility and modularity and making easy the maintenance and integration of new functionalities. The low-level layers process the information gathered by the environment sensors and generate spatio-temporal data associated to each one of the recognised objects of interest. The intermediate layers analyse the identified events and behaviours. Finally, the high-level layers monitor the results obtained by the previous layers and give support to the decision making process when a critical situations was detected.

In order to simplify the design and development of surveillance systems, the problem of monitoring an environment can be structured into a set of subproblems. This is possible since surveillance generally involves monitoring a scenario from different points of view, the elements that exist in such a scenario (e.g. moving or static objects), the aspects or events of interest to monitor (e.g. trajectories, objects speed, normality in a pedestrian crossing, etc) or even the physical scenarios that compose the whole environment (e.g. different rooms in an indoor environment).

Another relevant aspect when reducing the complexity of the design of these systems is the time. For instance, the period of time in which the surveillance is performed determines the functionality of the system and should be taken into account when carrying out the design of the system.

On the other hand, to reuse existing and well-tested designs and developments may facilitate the task of devising new surveillance systems. This idea leads us to the develop intelligent surveillance systems based on components that can be reused as customised modules that deal with different surveillance tasks. These components may be previously designed and developed by third parties and available in public or private repositories in order

to use them when designing a new surveillance system. The adoption of this approach implies that each component has a set of tools that allow its instantiation for particular problems or environments.

From a general point of view and according to the previously introduced ideas, the development of surveillance systems implies the analysis of the surveillance tasks carried out to address these problems. This analysis is made by means of the following questions:

- What objects or elements of the environments will be analysed?
- What aspects or events of interest will be monitored in relation to the previous objects or elements?
- What kind of sensors are available in the monitored environment? What new kind of sensors need to be deployed on the scenario to gather useful information to perform the surveillance?
- When these events of interest may take place?

Once these questions are answered, a study of the existing solutions that can be used or adapted must be carried out, taking into account the following issues:

- What processing mechanisms are there to process the data gathered by the deployed sensors? The main goal of these processing elements is to transform and adapt these data to generate the information used by the components that perform the different surveillance tasks.
- What existing components are there to carry out such tasks? If a component was already designed and developed to accomplish a surveillance task, it has to be customised for the environment where it will be deployed. On the contrary, the component must be designed and developed.

In this way, a surveillance component is associated to a scenario element, to an aspect or event of interest to monitor, to a sub-environment, and to a particular time interval. Generally, these facts imply that the perceptual or low-level layer gets the information from the environment (e.g. objects of the scenario, their location, physical features such as size or orientation, time interval of the monitoring, etc). The intermediate layer processes this information and performs an analysis about the environment state to determine whether the studied situations are normal or abnormal. To do that, this layer makes use of a model that defines in a general way the normality of objects according to a particular aspect, generating the named normality components which are independent of any environment. Within this context, the most undesired situation that can take place under this approach is an anomalous situation that was not previously defined and considered by a surveillance system. However, the system is aware of something suspicious or abnormal is happening at that time.

These normality components defined in a general way must be instantiated for particular environments. To instantiate a component under the approach discussed in this work implies customising it by means of knowledge acquisition tools and/or machine learning algorithms. The use of this last kind

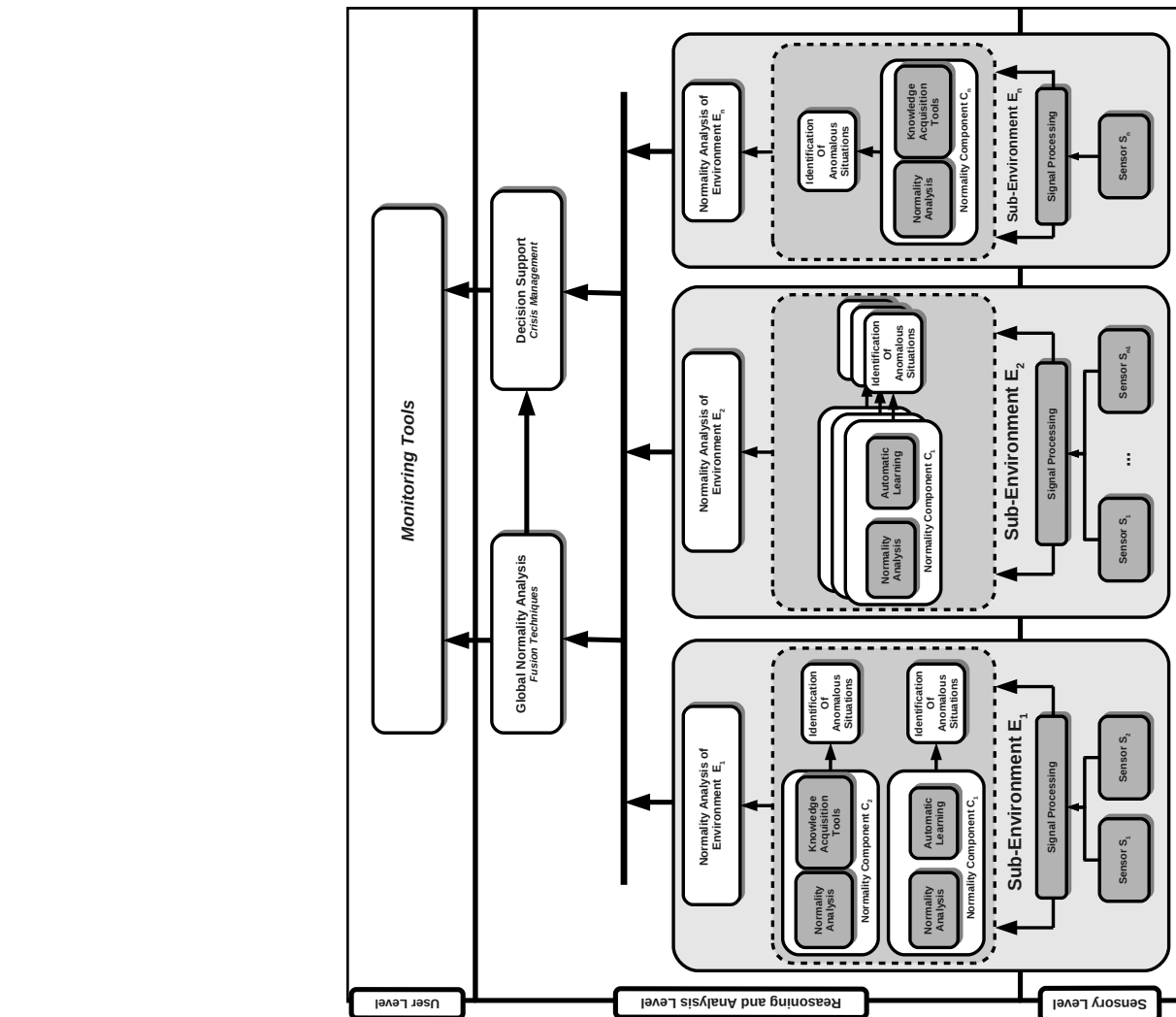


Fig. 1 General scheme of the proposed scalable surveillance system.

of algorithms is advisable in dynamic environments where the environmental conditions may easily change or the number of required instances is too high.

When a normality component determines that an object does not behave in a correct way according to the monitored aspect, the identification module of abnormal situations is activated. The combination of multiple normality analysis performed by each component and the output of the identification modules of anomalies constitutes the global normality analysis for a sub-environment. Similarly, the global analysis of the whole scenario is the result of applying multiple fusion techniques to the different analysis performed in each one of the monitored sub-environments (if the whole scenario was previously divided to reduce complexity). The surveillance system makes use

of this information to give support to the security personnel in the process of decision making. Finally, the upper layer is also composed of the monitoring tools used by the security staff to analyse the results provided by the rest of layers.

The design of surveillance systems based on the previously discussed ideas, that is, layers, normality components and sub-environments, offers the following advantages:

- **Reuse**, the development of normality components in a general way, independently of the monitored environment, facilitates the reuse of expert knowledge.
- **Reduction of the development cycle**, as a consequence of the previous advantage.
- **Quality improvement**, the reuse of previous experience avoid making mistakes.
- **Reliability**, if a component has been exhaustively tested and evaluated in multiple environments, debugging possible malfunctioning, its inclusion into a new surveillance system guarantees a high probability of success.
- **Flexibility**, the combination of components allows to build different kinds of surveillance systems depending on the security requirements. These components represents the *ingredients* of the application that are aggregated to reach a particular goal; in this case, the analysis of specific situations.

In the next sections, the concept of normality component that supports the ideas discussed in the current section is formalised, establishing how to perform the normality analysis of environments depending on these components.

3.2 Normality in monitored environments

The normality analysis in an environment is carried out by using the normality components responsible for the monitoring of the required aspects. Each one of these components is understood as a black box that receives information from the environment and, depending on the defined constraints (spatial, temporal, etc.) that checks the normality, provides an output that denotes the normality of the environment given in a list of grades (e.g. normal, suspicious, abnormal...). This is due to the fact that a situation is considered as normal or not depending on the degree of satisfaction of such constraints.

This approach leads us to Fuzzy Logic [36, 37], a mathematical model widely used to deal with the uncertainty and vagueness of real world problems, as the mechanism for representing these normality constraints. On the other hand, this model is also useful to work on the data gathered by the low-level layers, which are usually imprecise.

3. Formal Model to Build Scalable and Flexible Surveillance Systems | 257 |

3.2.1 The surveillance problem: *divide and conquer*

By means of the *divide and conquer* technique, the surveillance problem can be defined as follows:

Definition 1. The surveillance of an environment E is the understanding of the different perceptions obtained from the sensors deployed on the monitored environment $\{S_1, S_2, \dots, S_n\}$. Each perception of the global environment is considered, at the same time, as an environment E_i . Thus, the global normality definition in the environment E is composed of the particular normality definitions for each sub-environment E_i . Therefore, the surveillance problem P in an environment E is defined as the monitoring of multiple sub-environments, which are simpler and developed from the perceptions of the sensors deployed on the environment:

$$P = \{E_1, E_2, \dots, E_n\} \quad (1)$$

In this way, the complexity of defining the normality is reduced since the number of situations that can take place in a sub-environment E_i is lower than in the global environment E .

Example 1. The surveillance of a shopping center E , where multiple security cameras are deployed, may be carry out depending on the analysis of the situations that happen in those parts of the shopping center E_i monitored from some of the deployed cameras. The global normality analysis of E is obtained by aggregating the individual analysis of each E_i .

3.2.2 Subproblem of surveillance: monitored environments

Definition 2. A monitored sub-environment E is defined as a 4-tuple composed of the following elements:

$$E = \langle V; O; C; O \times C \rangle \quad (2)$$

where

- V is the set of input variables used to perform the required surveillance tasks and to inform about the objects features and the current state of the environment. On the other hand, the values of these variables can be directly gathered by the sensors or generated by preprocessing modules. Furthermore, such values may be precise or, on the contrary, provided with imprecision and vagueness.
- O is the set of classes of monitored objects in the sub-environment, whose behaviour must be analysed (e.g. people, group of people, car, truck, bicycle, etc).
- C refers to the set of monitored aspects in the sub-environment, denoted as *concepts* from now on.

- $O \times C$ determines the concepts that must be used to analyse the normality of each class of object. Depending on the class of each object, different concepts will be used to determine if their behaviour is normal or not.

Example 2. Let E_i be a part of the environment monitored by a security camera, where there exist traditional urban elements such as gardens, pavements, pedestrian crossings, traffic lights, roads, etc. The classes of objects that may appear are $O = \{pedestrian, vehicle\}$. The concepts to monitor in E_i are $C = \{trajectories, speed, pedestrian\_crossings\}$. That is, the trajectory, speed and behaviour regarding pedestrian normality crossings normality will be analysed. Some examples of the variables V needed to perform the surveillance may be as follows: object location, object size, time, list of key regions of the environment, position of relevant static elements, etc. The concepts proposed to analyse the behaviour of each class of object is given by $O \times C = \{(pedestrian, trajectories), (pedestrian, pedestrian crossing), (vehicle, trajectories), (vehicle, speed), (vehicle, pedestrian crossing)\}$. According to this example, a pedestrian behaves in a normal way if he/she follows normal trajectories and crosses the road through the pedestrian crossing. In the case of vehicles, their speed in each time is also monitored.

3.2.3 Surveillance based on the analysis of concepts

As previously discussed, the surveillance of an environment is carried out by means of concepts so that there exists one concept per each one of the events of interest to monitor.

Definition 3. A concept c_i ($c_i \in C$) is defined as a 3-tuple composed of the following elements:

$$c_i = \langle V_i; DDV_i; \Phi_i \rangle \quad (3)$$

where V_i is the set of input variables used to define the concept c_i so that $V_i \subseteq V$. On the other hand, DDV_i is the set of definition domains of the variables that belong to V_i . Therefore, if $V_i = \{v_{1i}, v_{2i}, \dots, v_{ni}\}$, then DDV_i is defined as $DDV_i = \{DDV_{1i}, DDV_{2i}, \dots, DDV_{ni}\}$, where DDV_{ji} is the definition domain of the variable v_{ji} . The definition domain of a variable specifies the possible values that can take. Finally, Φ_i is the set of constraints used to complete the definition of the concept c_i , according to the elements of V_i ($\Phi_i = \{\mu_{1i}, \mu_{2i}, \dots, \mu_{ki}\}$). The normality analysis of c_i depends on how the constraints associated to c_i are met.

3.2.4 Normality constraint

Definition 4. A normality constraint, associated to a concept c_i is defined as a fuzzy set X_i over the domain $\mathcal{P}(V_i)$, with an associated membership function μ_{X_i} :

$$\mu_{X_i} : \mathcal{P}(V_i) \longrightarrow [0, 1] \quad (4)$$

3. Formal Model to Build Scalable and Flexible Surveillance Systems | 259 |

where 1 represents the maximum degree of satisfaction of the constraint and 0 the minimum. The rest of values represent intermediate degrees of normality.

Sometimes, defining several constraints by means of crisp sets (the object meets the constraint or not) is more suitable and practical. In such a case, the membership function is as follows:

$$\mu_{X_i}(x) = \begin{cases} 1 & \text{if } x \in X_i; \\ 0 & \text{if } x \notin X_i; \end{cases}$$

The constraints definition can be done by using simple constraints through a set of operations generating a new constraint.

Operations between constraints

Let A and B be two normality constraints defined over the domain $\mathcal{P}(V_i)$ so that:

1. The **union** ($A \cup B$) of constraints is a new constraint that is met if and only if A or B are satisfied. The membership function of $A \cup B$ is defined as $\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$.
2. The **intersection** ($A \cap B$) of constraints is a new constraint that is met if and only if A and B are simultaneously satisfied. The membership function of $A \cap B$ is defined as $\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$.
3. The **complement** ($\bar{A}$) is a constraint that is met if and only if A is not satisfied. The membership function of $\bar{A}$ is defined as $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$.

Properties of normality constraints

The properties associated to normality constraints are exactly the same that the properties of fuzzy sets. Let A, B and C be normality constraints defined over the domain $\mathcal{P}(V_i)$:

1. **Idempotent:** $A \cup A = A$; $A \cap A = A$
2. **Commutative:** $A \cup B = B \cup A$; $A \cap B = B \cap A$
3. **Associative:** $A \cup (B \cup C) = (A \cup B) \cup C$; $A \cap (B \cap C) = (A \cap B) \cap C$
4. **Distributive:** $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$; $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
5. **Doble negation:** $\neg(\neg A) = A$
6. **Transitive:** Si $A \subset B \wedge B \subset C \rightarrow A \subset C$
7. **Limit condition:** $A \cup \emptyset = A$; $A \cap \emptyset = \emptyset$; $A \cup \mathcal{P}(V_i) = \mathcal{P}(V_i)$; $A \cap \mathcal{P}(V_i) = A$

The concept definition determines the framework that establishes the general rules about how to carry out the associated surveillance task. However, the process of making instances from such a concept needs to be applied for particular environments. For example, in the case of the trajectory concept, such a definition establishes the mechanisms required to analyse normal trajectories, but without instantiating them because they depend on the particular environment to monitor.

3.2.5 Concept instance

The next step after defining a concept and its constraint in a general way is to make instances of such a concept for particular environments.

Definition 5. An instance y of a concept c_i in an environment E_j ($E_j \in P$), denoted as c_{iy}^j , is defined as follows:

$$c_{iy}^j = \langle V_i; DDV_i; \tilde{\Phi}_i = \{\tilde{\mu}_{1i}, \tilde{\mu}_{2i}, \dots, \tilde{\mu}_{zi}\} \rangle \quad (5)$$

where $\tilde{\Phi}_i$ is the set of particularised constraints of the set Φ_i , that is, each $\tilde{\mu}_{xi} \in \tilde{\Phi}_i$ represents the particularisation of $\mu_{xi} \in \Phi_i$. It is verified that $|\Phi_i| \geq |\tilde{\Phi}_i|$.

Example 3. If c_i is the trajectory concept, an instance of c_i represents a normal trajectory within the environment E_j .

3.2.6 Normality constraint instance

A normality constraint instance is used to adapt the general definition of a kind of analysis based on a concept to a specific environment.

Definition 6. A normality constraint instance is a fuzzy set defined over $\mathcal{P}(DDV_i)$ with an associated membership function ($\tilde{\mu}_{X_i}$):

$$\tilde{\mu}_{X_i} : \mathcal{P}(DDV_i) \longrightarrow [0, 1] \quad (6)$$

so that if $v_{ki} \in \mathcal{P}(V_i)$ is employed to define μ_{X_i} , then the values of v_{ki} defined over $DDV_{ki} \in \mathcal{P}(DDV_i)$ are used to make the instance $\tilde{\mu}_{X_i}$.

Example 4. If μ_{X_i} represents a constraint that checks if a moving object follows the sequence of regions of a trajectory, $\tilde{\mu}_{xi}$ is the constraint employed to check if the moving object follows a particular sequence of regions within the monitored environment.

3.2.7 Degree of normality associated to an instance

Each monitored object has associated a degree of normality that establishes how normal its behaviour is according to each attached concept to it, represented by the deployed instances of such concepts within the environment.

Definition 7. The degree of normality of an object obj within an environment E_j ($E_j \in P$) regarding an instance y of the concept c_i (c_{iy}^j), denoted as $N_{c_{iy}^j}(obj)$, is calculated from the values obtained for each $\tilde{\mu}_{xi}$:

$$N_{c_{iy}^j}(obj) = \bigwedge_{x=1}^{|\tilde{\Phi}_i|} \tilde{\mu}_{xi}^y \quad (7)$$

being $\bigwedge$ a t-norm, such as the t-norm that calculates the minimum value.

3. Formal Model to Build Scalable and Flexible Surveillance Systems | 261 |

This particular t-norm, that is, the minimum, is suitable for surveillance systems and, particularly, for the model proposed in this work since in this kind of systems the violations of some of the constraints that define the normality in the environment are needed to be detected. In other words, if some of the constraints are not satisfied, the normality degree $N_{c_{iy}^j}(obj)$ will be low. The fact of using other t-norms implies obtaining lower values than this minimum, since a relevant property of the t-norm operators is that any of them is always lower or equal to the value obtained by the operator that calculates the minimum. Therefore, this would imply a stricter surveillance and, as a result of that, a high number of alarm activations. For instance, if the product t-norm ($T(a,b) = a \cdot b$) is used and c_{iy}^j is defined through two constraints that are satisfied with a value of 0.6, the degree of normality $N_{c_{iy}^j}(obj)$ will be $0.6 \cdot 0.6 = 0.36$, fact that does not make sense since most of the constraints were satisfied with a higher degree (0.6). On the other hand, with the minimum t-norm this degree would be $\min(0.6, 0.6) = 0.6$, which is a value that represents in a better way the degree of satisfaction of the constraints that define the instance.

Example 5. If c_i is the normal trajectory concept and c_{il}^j is a particular normal trajectory instantiated within the environment E_j from a set of constraints μ_{X_i} and their particularisations ($\tilde{\mu}_{X_i}^1$), then $N_{c_{il}^j}$ determines the degree of normality of the object obj regarding the trajectory c_{il}^j . A high value of $N_{c_{il}^j}$ means that obj follows the trajectory c_{il}^j in a suitable way. On the other hand, a low value represents the contrary case or that obj does not follow such a trajectory at all. In other words, a low value of $N_{c_{il}^j}$ means that one or more constraints of the instance c_{il}^j are not met.

3.2.8 Degree of normality associated to a concept

The next step After calculating the degree of normality of an object for each instance of a particular concept is to calculate the normality of such an object according to all the defined instances of such a concept. In this way, it is possible to study the general behaviour of an object according to a concept.

Definition 8. The degree of normality of an object obj within an environment E_j ($E_j \in P$) according to a concept c_i , denoted as $N_{c_i^j}(obj)$, is calculated as follows:

$$N_{c_i^j}(obj) = \bigvee_{y=1}^w N_{c_{iy}^j}(obj) \quad (8)$$

where w is the number of instances of the concept c_i and $\bigvee$ is a t-conorm operator, for instance the maximum t-conorm.

In the same way that $\min(a,b)$, which represents the maximum value that can be obtained by applying a t-norm, $\max(a,b)$ represents the minimum

value obtained by applying a t-conorm operator, that is, the value calculated with any other t-conorm operator will be higher than the calculated by max. This fact also justifies its use on the proposed model so that an object obj behaves in a normal way according to a concept c_i within an environment E_j ($N_{c_i}^j(obj)$) if it satisfies the constraints of some of the deployed instances for c_i .

On the other hand, considering the application of the t-norm and t-conorm operators, the normality analysis of an object obj , according to c_i in E_j , is the result of applying an AND-OR fuzzy network over a set of constraints:

$$\begin{array}{c}
 c_{i1}^j = \tilde{\mu}_{1i}^1 \wedge \tilde{\mu}_{2i}^1 \wedge \dots \tilde{\mu}_{ni}^1 = N_{c_{i1}}^j(obj) \\
 \quad \quad \quad \vee \\
 c_{i2}^j = \tilde{\mu}_{1i}^2 \wedge \tilde{\mu}_{2i}^2 \wedge \dots \tilde{\mu}_{ni}^2 = N_{c_{i2}}^j(obj) \\
 \quad \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \\
 \quad \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \quad \quad \vee \\
 \quad \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \\
 c_{iw}^j = \tilde{\mu}_{1i}^w \wedge \tilde{\mu}_{2i}^w \wedge \dots \tilde{\mu}_{ni}^w = N_{c_{iw}}^j(obj) \\
 \hline
 N_{c_i}^j(obj)
 \end{array}$$

Example 6. If c_i is the trajectory concept and c_{ik}^j represents the normal trajectories defined through the instances of the variables and constraints in E_j , then the degree of normality $N_{c_i}^j(obj)$ of each object according to the concept is calculated by applying a t-conorm operator over the values $N_{c_{ik}}^j(obj)$ obtained for each one of the instances. Let the following values be:

$$\begin{array}{c}
 c_{i1}^j = \tilde{\mu}_{1i}^1 \wedge \tilde{\mu}_{2i}^1 \wedge \dots \tilde{\mu}_{ni}^1 = N_{c_{i1}}^j(obj) = 0.2 \\
 \quad \quad \quad \vee \\
 c_{i2}^j = \tilde{\mu}_{1i}^2 \wedge \tilde{\mu}_{2i}^2 \wedge \dots \tilde{\mu}_{ni}^2 = N_{c_{i2}}^j(obj) = \mathbf{0.8} \\
 \quad \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \\
 \quad \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \quad \quad \vee \\
 \quad \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \quad \quad \cdot \\
 c_{iw}^j = \tilde{\mu}_{1i}^w \wedge \tilde{\mu}_{2i}^w \wedge \dots \tilde{\mu}_{ni}^w = N_{c_{iw}}^j(obj) = 0.0 \\
 \hline
 N_{c_i}^j(obj) = 0.8
 \end{array}$$

In this case, the object satisfies remarkably the constraints defined for the instance c_{i2}^j of the concept c_i , since the value calculated for $N_{c_{i2}}^j(obj)$ is 0.8. In other words, the object follows a normal trajectory, which is the trajectory instantiated in c_{i2} .

In short, the degree of normality of an object associated to a concept c_i , $N_{c_i}^j(obj)$, is a numerical value that belongs to the interval $[0, 1]$, which is a representative sign of the object behaviour regarding a concept or surveillance aspect. High values of this parameter represent normal situations within the

3. Formal Model to Build Scalable and Flexible Surveillance Systems | 263 |

monitored environment while low ones represent suspicious or abnormal situations.

3.2.9 Normal behaviour within an environment according to a concept

The final goal to reach after the analysis of a situation is to activate a set of alarms or to draw the attention of the security personnel when such a situation does not meet the limits of normality. That is, after having calculated the degree of normality $N_{c_i}^j(obj)$, the model needs a mechanism to decide whether the object behaves normally or not, depending on the calculated degree of normality. This can be addressed by using an alpha threshold defined within the range $[0, 1]$ over the degree of normality, fact that implies that a situation changes from normal to abnormal abruptly. For this reason, the normality is considered as a linguistic variable V_N that takes a set of values over the domain definition $DDV_{V_N} = \{AA, PA, SB, PN, AN\}$ (see Figure 2). In this way, the object behaviour can be absolutely abnormal (AA), possibly abnormal (PA), suspicious (SB), possibly normal (PN), and absolutely normal (AN); so that a behaviour can belong to more than one set at the same time.

The definition of each value of the domain DDV_{V_N} depends on the features of the environment to monitor, the desired security level and the criterion of the expert in charge of setting up the configuration parameters, which must be adapted to the system behaviour. Figure 2 graphically shows how the values assigned to the sets that represent anomalous situations are not high, which implies that the system is not very strict, avoiding in this way a frequent (and possible unnecessary) alarm activation.

In this way, every time that the degree of normality of an object behaviour is calculated according to a concept, $N_{c_i}^j(obj)$, the membership of this value to the fuzzy sets that establish the definition domain DDV_{V_N} of the normality variable (V_N) is studied, determining the normality of the analysed situation. The alarm activation will rely on upper layers, which will perform the required actions depending on the normality values received.

The next section discusses how to aggregate the output of different normality components to get a global analysis and determine if the behaviour of the objects is normal within a particular environment, taking into account multiple surveillance concepts.

3.3 Global normality analysis by aggregating independent analysis

The normality of an object does not exclusively depend on an concept. In fact, a global evaluation of the normality of all the monitored concepts within the environment must be carried out. Therefore, a mechanism for combining multiple analysis is needed to get a global value that represents the normality of the object behaviour in a general way.

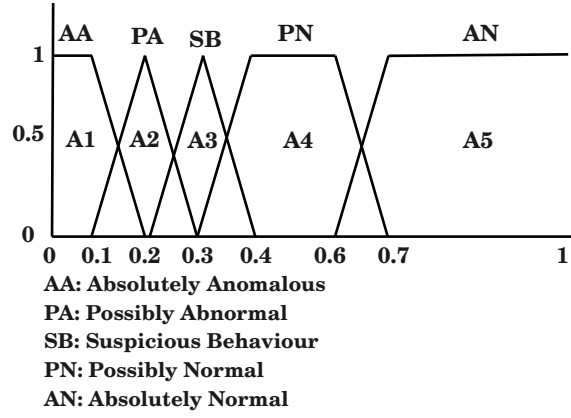


Fig. 2 Definition domain DDV_{V_N} of the normality variable V_N .

A possible approach consists in combining the output values of the normality components by using t-norms or t-conorms. For instance, the minimum t-norm establishes that the global normality value corresponds to the value of the analysis of the lowest degree of normality. On the other hand, if the maximum t-conorm is used, then the result will come determined by the component that provides the degree of normality with the highest value. Both techniques do not reflect the real process when combining multiple degrees of normality. For this reason, the OWA (Ordered Weighted Averaging) operators [34] are proposed to address this problem due to their flexibility.

Formally, an OWA operator is represented as a function $F : R^n \rightarrow R$ associated to a vector of weights W of length n ; $W = [w_1, w_2, \dots, w_n]$, where each $w_i \in [0, 1]$ and $\sum_{i=1}^n w_i = 1$.

$$OWA(a_1, a_2, \dots, a_n) = \sum_{j=1}^n w_j \cdot b_j \quad (9)$$

$$OWA(a_1, a_2, \dots, a_n) = [w_1, w_2, \dots, w_n] \begin{bmatrix} b_1 \\ \cdot \\ \cdot \\ b_n \end{bmatrix}$$

where $(a_1, a_2, \dots, a_n)$ represent the set of initial values or criteria used by the operator to make a decision, and $(b_1, b_2, \dots, b_n)$ represent the ordered set associated to $(a_1, a_2, \dots, a_n)$, being b_j the j -th highest value of such a set. Furthermore, the values of the weights w_i belonging to the vector W are linked to positions and not to particular values of the original set.

In the model devised in this work, the OWA operator is used to aggregate the normality values calculated by each component $(N_{c_1}^j, N_{c_2}^j, \dots, N_{c_n}^j)$.

One of the key characteristics of this set of aggregation operators is the flexibility to vary their behaviour depending on the values assigned to the vector of weights W . Such a vector determines the behaviour of the operator,

3. Formal Model to Build Scalable and Flexible Surveillance Systems | 265 |

which may tend to behave as union or intersection operators. In fact, this behaviour can be customised to reflect the minimum t-norm or the maximum t-conorm. Figure 3 graphically shows the relationships between the values obtained by the operators t-norm, OWA and t-conorm, respectively.

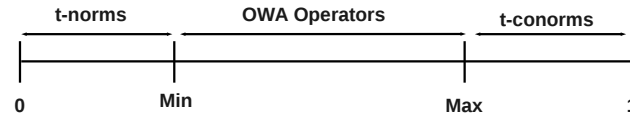


Fig. 3 Visual comparison between the range of values obtained by t-norm, t-conorm and OWA operators.

In order to analyse the behaviour of the OWA operator that deals with the minimum t-norm and the maximum t-conorm, R. Yager proposed the following formula [34, 35]:

$$orness(W) = (1/n - 1) \sum_{i=1}^n ((n-1) \cdot w_i) \quad (10)$$

After having initialised the vector W , closer values of the orness measure to 1 reflect that the OWA operator is close to the maximum; while closer values to 0 reflect the proximity to the minimum.

The particular values of the vector of weights W depend on the desired surveillance level. In this way, an orness value close to 1 represents a soft surveillance so that a situation will be considered normal while there is a normal behaviour. On the other hand, values close to 0 reflect a hard surveillance so that if there is a single anomalous behaviour, then the situation will be considered as abnormal. Values close to 0.5 reflect intermediate cases.

Table 1 shows three configurations of W and how they affect to a common monitored situation given by the output of three normality components $N_{c_1}^j(obj)$, $N_{c_2}^j(obj)$ and $N_{c_3}^j(obj)$, reflecting the minimum, the maximum and a weighted measure.

Criteria	Value	Minimum		Maximum		Average	
		Weight	Evaluation	Weight	Evaluation	Weight	Evaluation
$N_{c_1}^j(obj)$	0.95	0	0	1	0.95	0.33	0.31
$N_{c_2}^j(obj)$	0.8	0	0	0	0	0.33	0.264
$N_{c_3}^j(obj)$	0.2	1	0.2	0	0	0.33	0.066
TOTAL			0.2		0.95		0.64

Table 1 Multiple configurations of the OWA operator.

After having obtained the global normality value by applying the OWA operator ($OWA(N_{c_1}^j, N_{c_2}^j, \dots, N_{c_n}^j)$), the final normality value $N^j(obj_k)$ associated to an object obj_k in a particular environment E is given by the degree of membership of this value to a fuzzy sets. These sets define the values *possibly normal*

behaviour (PN) and *absolutely normal behaviour* (AN) of the definition domain DDV_N of the *normality* variable V_N (see Figure 2), determining in this way the normality of the analysed situation:

$$N^j(obj_k) = \mu_{PN}(OWA(N_{c_1}^j, N_{c_2}^j, \dots, N_{c_n}^j)) + \mu_{AN}(OWA(N_{c_1}^j, N_{c_2}^j, \dots, N_{c_n}^j)) \quad (11)$$

where $\mu_{PN}(OWA(N_{c_1}^j, N_{c_2}^j, \dots, N_{c_n}^j))$ and $\mu_{AN}(OWA(N_{c_1}^j, N_{c_2}^j, \dots, N_{c_n}^j))$ establish the membership of the output value of the OWA operator to the sets PN and AN , respectively.

The global normality value within an environment E_j considering the activity of all the moving objects, denoted as GN^j , is calculated as the minimum of the normality value N^j calculated for each object obj_k :

$$GN^j = \bigwedge_{k=1}^{|\mathcal{O}|} N^j(obj_k) \quad (12)$$

where $\mathcal{O}$ is the set of monitored objects at a particular time, $|\mathcal{O}|$ is the number of monitored objects and $N^j(obj_k)$ is the normality value calculated for each object $obj_k \in \mathcal{O}$.

The use of the minimum is justified because when an object does not behave normally, the global normality degree GN^j in the environment E_j must be a low value.

4 Model application: trajectory analysis

This section discusses the model application to monitor the trajectories and speed of moving objects in a particular scenario. These analyses are performed by two independent normality components, but the global normality is evaluated by taking into account both of them. This aggregation is carried out by using the approach described in Section 3.3.

The analysis of trajectories and speed, using the visual information gathered by security cameras, represent two problems that have been widely studied by other researchers. The use of a formal model to deal with them provides several advantages over other approaches. In the case of trajectory analysis, most authors focus their work on analysing the spatial information to recognise the paths of moving objects [18, 22, 23, 25]. However, these definitions might not be enough in the surveillance field, since additional constraints need to be checked very often. Most of existing proposals make use of algorithms for learning normal trajectories, which usually match with those frequently followed by most of moving objects. Within this context, it is important to take into account that a trajectory often followed by a vehicle may not be normal for a pedestrian or, on the other hand, normal trajectories at a particular time interval may be abnormal at a different interval. Therefore, the trajectory analysis based on the proposed model provides richer and more scalable definitions because new kinds of constraints are possible to

be included when they are needed (or even removed unnecessary constraints previously added).

On the other hand, the second component analyses the speed of moving objects by using the 2D images provided by a security camera without knowing the camera configuration parameters. The main goal of this component is to classify the real speed of objects as normal or anomalous. Most of the proposed methods estimate the speed and require a previous calibration process that needs relevant information of the camera, such as the height or the zoom level. Afterwards, they use transformation and projection operations to get a real 3D representation of the position and the object movement [9, 11, 20, 21, 24]. The definition of the speed concept in the proposed model is made by taking into account how people work when they analyse a video, that is, they do not need to know the exact values of position and speed of moving objects to infer that they move at a fast speed. On the contrary, the security staff only analyse the object displacement using the closest static objects as reference. The output of this component is therefore obtained depending on the analysis that determines whether an object moves at a normal speed regarding such a displacement.

To do that, horizontal and vertical fuzzy partitions of 2D images are done. This information is used by an inductive learning algorithm to generate a set of constraints that establish how the horizontal, vertical and global displacements of each kind of object in each region that composes the fuzzy partition are performed. An example of a generated constraint is as follows:
The speed of vehicles in far regions from the camera is normal if the horizontal displacement is small.

A more detailed discussion on the definition of this concept is given in [2]. Next, the application of the proposed formal model for the definition and analysis of trajectories is described in depth. Nevertheless, Section 5 will discuss the results obtained by the two normality components previously introduced.

4.1 Normal trajectory concept

A trajectory is defined by means of three different kinds of constraints: i) role constraints, that specify what kind of objects are allowed to follow a trajectory; ii) spatial constraints, that determine the sequence of regions where moving objects must move (possibly with an associated order) and, finally; iii) temporal constraints, that refer to the maximum period of time or interval allowed to follow a trajectory.

In this work, the regions are represented by means of polygons graphically defined by means of a knowledge acquisition tool over a 2D image in order to establish trajectories. In this way, a single sequence of regions may represent a trajectory that comprises multiple paths whose points are close each other (see Figure 4). On the other hand, two similar paths are uncommon to correspond to a single trajectory.

In order to define trajectories, the following information is required:

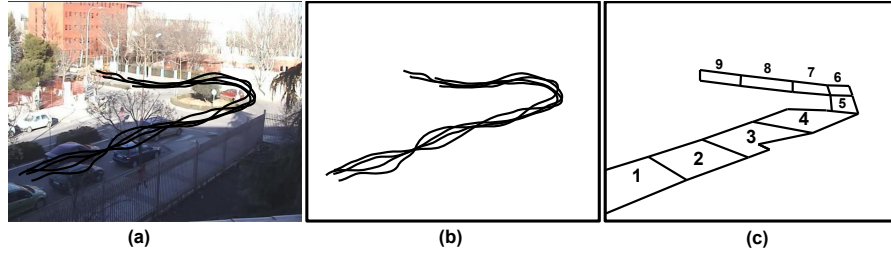


Fig. 4 (a) Group of trajectories followed by vehicles and marked by means of continuous lines over the scene. (b) Group of trajectories without the background image that represents the scene. (c) Example of sequence of zones or regions used to define the previous group of trajectories.

- Set of relevant regions of the monitored environment.
- Regions where every kind of object can be located at a particular time.
- Sequence of covered regions by an object, temporally ordered, since such an object is first identified.
- Kinds of monitored objects, determined by means of physical features such as height, width, horizontal and vertical displacements, etc.
- Identifiers associated to each moving object by the tracking process.
- Temporal reference to the current moment.
- Time spent to carry out each one of the defined trajectories.

This information justifies the choice of variables for the definition of the trajectory concept discussed in the next section.

4.2 Set of variables for the trajectory definition

As previously specified in the model formalisation, a concept is defined by means of a 3-tuple $c_i = \langle V_i; DDV_i; \Phi_i \rangle$ composed of three elements, where V_i is the set of variables needed to represent the concept c_i , DDV_i is the definition domain of each variable belonging to V_i and, finally, the concept definition is completed through the association of a set of constraints Φ_i . Next, the variables used in the trajectory component are described by taking into account this information.

(a) The system must be aware of the division of the environment into regions, and the regions where a moving object may be located. Within this context, the system needs an internal representation of the more relevant regions in order to establish a relation between the objects position and the previously defined regions.

(b) Sequence of temporally-ordered covered areas by an object. Table 3 completes the set of variables of the environment regions and the position of each object. Each $\mu_a(obj)$ represents the intersection of an object obj with a particular area a . This information is not generated by the component but obtained from the low-level modules, which take the data provided by the segmentation and tracking processes to perform this task. The main reason

4. Model application: trajectory analysis

| 269 |

Variable	DDV_A	Description
A	$\{a_1, a_2, \dots, a_n\}$	Set of areas belonging to a monitored environment. Each a_i is represented through a polygon, defined by means of a set of ordered points.

Table 2 Set of regions A that represent the regions of activity in a monitored environment.

to calculate the regions where an object is located in this low-level modules is because multiple modules or normality components may need it. On the other hand, the systems maintains a register $\ell(obj)$ for each object obj , where the set of covered regions by every object is stored to establish similarities with normal trajectories.

Variable (V_i)	DDV_i	Description
$\mu_a(obj)$	$[0, 1]$	Imprecise information received from low-level layers about the possibility for an object obj to be located over an area a , where 1 represents that the object is totally located over it and 0 the contrary case.
$\ell(obj)$	$\{\mu_a(obj)\}^+$	List of $\mu_a(obj) \mu_a(obj) > 0$, updated until the current time.

Table 3 Variable used, together with the set of areas A , to represent the objects position on the scene.

(c) Object membership to one or multiple classes (see Table 4). Each $\mu_c(obj)$ represents the degree of membership of an object obj to the class c within the interval $[0, 1]$, where 1 is the maximum membership value and 0 the minimum. The degree of membership of an object to a class is also calculated by low-level layers that use the spatial information of the segmentation and tracking processes. Therefore, both the regions where an object is located and its classes are the input of the component that analyses trajectories.

Variable (V_i)	DDV_i	Description
$\mu_c(obj)$	$[0, 1]$	Imprecise information of the object membership to a class c , $\forall c \in O, \exists \mu_c(obj)$.

Table 4 Membership value of an object obj to a class c .

(d) Temporal references. The system maintains a temporal reference to the current moment in which the analysis is being performed and, at the same time, it allows the maximum duration and temporal intervals of the trajectories (see Table 5).

Finally, Table 6 shows the rest of variables, which are mainly used to define constraints.

Variable	DDV	Description
t_c	$[1, 31] \cup \{*\} \times [1, 12] \cup \{*\} \times [1900 - 9999] \cup \{*\} \times [0, 24] \cup \{*\} \times [0, 59] \cup \{*\} \times [0, 59] \cup \{*\}$	Absolute reference to the current moment. The format is $DD/MM/YY - hh:mm:ss$. The symbol * is used as a wild card.
t_{max}	$t_{max} \in [0, 9999]$	t_{max} is the maximum allowed duration for a trajectory, measured in seconds. This parameter is often used to define temporal constraints.
$Int_j < t_s, t_e >$	$DDV_{t_c} \times DDV_{t_c}$	Time interval definition through initial (t_s) and final (t_e) moments. This variable is also used to define temporal constraints.

Table 5 Set of temporal constraints associated to trajectories.

Variable	DDV	Description
Υ	$\emptyset$	Υ represents the set of classes/roles that are allowed to follow a certain trajectory. Each trajectory has associated Υ , used to define role constraints.
Ψ	$DDV_{\Psi} \subseteq DDV_A$	Ψ represents the sequence of areas that must be covered to perform a certain trajectory. Each trajectory has associated a sequence Ψ , used to define spatial constraints.
$\ell_{\Psi}(obj)$	DDV_{Ψ}	Maximum membership value of object obj to each of the regions covered of Ψ until the current time. If an object does not satisfy the order imposed by Ψ (the object is not located on a region of the sequence Ψ or the order has been broken), the list of covered regions $\ell_{\Psi}(obj)$ becomes empty and is initialised to a void value: $\ell_{\Psi}(obj) = \emptyset$

Table 6 Variables Υ , Ψ and $\ell_{\Psi}(obj)$ to define role and spatial constraints, respectively.

4.3 Preprocessing modules

The preprocessing modules are responsible for providing the values of the variables used by the normality components in order to perform the analysis. Concretely, the component of normal trajectories needs three preprocessors whose main functions are as follows: i) tracking of objects, ii) estimation of object position (regions where it is located) and iii) object classification.

The first one is especially important for the system to know the objects that appear on the scene. In fact, the intermediate layers will not be able to analyse the behaviours of those objects that were not previously recognised. The design and development of tracking algorithms is not one of the main goals of this work but needed to carry out the proposed analysis. For this reason, OpenCV and the library *Blob Tracker Facility* [1] have been used to perform this task. The application **blobtrack.cpp**, implemented in C++, makes use of these libraries to segment and track objects from the video stream gathered by the security cameras. A detailed description of the whole

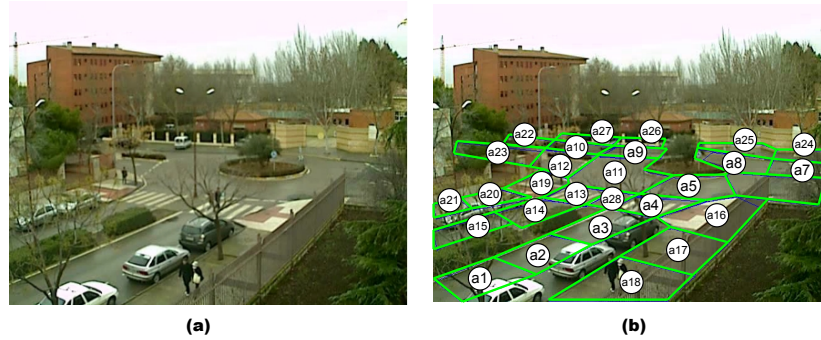


Fig. 5 (a) Scene monitored from a security camera. (b) Scene division into areas or regions.

process is discussed in [10]. The original application has been customised to extend its functionality and get more spatial data, providing information about the horizontal and vertical displacement of each object *obj* between consecutive frames, calculated as the difference between the central point of the ellipse that wraps the monitored object. On the other hand, the source code has been modified to generate a persistent log used by the preprocessing modules and the normality components.

The second preprocessor makes use of the scene knowledge and the information generated by the first preprocessor to estimate the regions where an object is located. That is, the preprocessor estimates the object location from the following information:

- Set of areas (*A*), defined over the scenario to monitor. Figure 5.b shows an example of scene division into multiple regions.
- Spatial information of each object, provided by the segmentation algorithm. Particularly, the coordinate of the ellipse central point that frames the object, the object height and the object width.

The membership value of an object to a particular area ($\mu_a(obj)$) is the result of dividing the number of points that represent the object base over the area *a* by the total number of points of the object base. The object base over an area matches up with the lower half of the ellipse that wraps the object. The points that form this base are those that belong to the rectangle which upper left vertex is calculated as $(x - width/2, y)$ and which inferior right vertex is $(x + width/2, y + height/2)$, being *x* and *y* the coordinates of the ellipse central point. To calculate if the point (x, y) belongs to a polygon defined from a set of points, W. Randolph Franklin's algorithm [26] is used, which is based on a previous algorithm [29].

Finally, the third preprocessor allows to establish the class or classes of a detected object. To do that, the output generated by the first preprocessor and a set of IF-THEN fuzzy rules automatically acquired by the inductive learning algorithm [2] are used. These rules, such as **IF** $v_0 \text{ es } ZD_0 \wedge \dots \wedge v_n \text{ es } ZD_n$ **THEN** y_j , are characterised by the antecedents composed of a set of variables V_i that take a subset of values $ZD_i \subset DDV_{v_i}$, and a consequent y_j that represents the object class if the rule is fired.

Currently, the set of variables V employed to classify objects differs from [2] and is composed of the following variables:

- **Horizontal v_1 and vertical v_2 location.** The position and distance regarding the camera affect to the object dimensions.
- **Width-height (W/H) aspect, v_3 .** This value is obtained by dividing the ellipse width by the ellipse height. To study the objects dimensions instead of their size improves the classification results. This is due to the fact that the segmentation algorithms are not perfect and, therefore, the estimated ellipse does not completely wrap the identified objects most of the times. However, it is possible to maintain the suitable dimensions of the class they belong.
- **Horizontal and vertical displacements aspect, v_4 .** This aspect is obtained by dividing by the variables that contain the horizontal and vertical displacement values of the object. In some scenarios, these variables may be a relevant discriminator between object classes. In the scene of Figure 5, the camera location and the routes where the vehicles drive imply that their horizontal displacements are normally larger than the vertical ones.
- **Global displacement (MOV), v_5 .** This value is calculated as the euclidean distance between the ellipse central point that wraps the object between consecutive frames. This variable is also interesting to distinguish object classes since there are displacements that are not real for certain kinds of objects. The length of the displacements varies depending on the region of the 2D image where they take place, that is, depending on the distance between the object and the camera. Obviously, the length of the displacements in regions that are far from the camera are much smaller than those produced in close regions, fact that does not imply a shorter distance on the real scene.

The preprocessor also maintains a persistent register that stores how each object was classified at every moment. Depending on this information and the fired rules, the membership values to each one of the classes are calculated. Algorithm 1 specifies the steps to be followed in order to determine the membership to the set of classes O . Figure 6 shows an example that describes this classification process.

4.4 Constraint definition

The last step to complete the normality definition of the trajectory concept consists in the specification of a set of general constraints Φ_i , included in the tuple $\langle V_i, DDV_i, \Phi_i \rangle$. According to the formal model, the satisfaction of these constraints allows the surveillance system to infer if the moving objects follow trajectories in a correct way, together with the degree of normality associated to the concept. A high value of this parameter implies that a monitored object follows in a suitable way one of the normal trajectories defined for the monitored environment. The defined constraints are as follows:

4. Model application: trajectory analysis

| 273 |

Algorithm 1 Object classification algorithm

Require: Detected object in the current frame. The values of the variables $(v_1, v_2, v_3, v_4, v_5)$ are $(x_1, x_2, x_3, x_4, x_5)$.

Ensure: Degrees of membership to the classes pedestrian y_1 and vehicle y_2 .

1.- Calculate the satisfaction degree of each rule.

2.- **FOR EACH** class $o_i \in O$:

A) The value of the current membership is calculated without taking into account previous classifications. If multiple rules of the same class are fired, the value used will be the one with the higher degree of satisfaction of such rules. If no rule is fired, the membership value is 0.

B) The final membership value to each one of the classes is calculated as the arithmetic average of the previous membership values and the current one.

END FOR

Time	t_1	t_2	t_3	t_4	t_5
Ellipse	 obj1	 obj1	 obj1	 obj1	 obj1
Classification	 1.0  0.0	 1.0 = (1.0+1.0)/2 = 1.0  0.0 = (0+0)/2 = 0.0	 0.8 = (1+1+0.8)/3 = 0.93  0.25 = (0+0+0.25)/3 = 0.08	 0.5 = (1+1+0.8+0.5)/4 = 0.825  0.5 = (0.5+0.25)/4 = 0.18	 1.0 = (1+1+0.8+0.5+1)/5 = 0.86  0.15 = (0.5+0.25)/5 = 0.15

Fig. 6 Object classification example. The first row represents the time, where $t_i < t_{i+1}$. The second row shows the form and size of the ellipse that wraps the object in each t_i . Finally, the last row shows the object classification. If the image that represents the class has a numeric value associated to the right, it means that the object was classified in t_i with such a degree of membership. However, the final membership value is calculated as the arithmetic average of the previous membership values and the current one.

Role constraint μ_{11} . Each trajectory is associated with a list Υ that specifies the kinds of objects that are allowed to follow it. Only those objects that belong to a class of Υ satisfy this constraint. For instance, those objects classified as pedestrians will not meet the role constraint of a trajectory defined for vehicles. The membership function μ_{11} of this constraint is defined in the following way:

$$\mu_{11}(obj) = \mu_{c_k}(obj) \iff \forall c \neq c_k; c, c_k \in \Upsilon; \mu_{c_k}(obj) \geq \mu_c(obj)$$

So that $\mu_c(obj)$ represents the membership value of the object obj to the class c , and $\mu_{c_k}(obj)$ the maximum membership value to one class in Υ . For example, if $\Upsilon = \{\text{Person}, \text{Group of People}\}$, and the object obj has been classified as Person = 0.3 and Group of People = 0.7, then the degree of satisfaction of the constraint $\mu_{11}(obj)$ is 0.7, that is, $\max(\mu_{\text{Person}}(obj), \mu_{\text{GroupOfPeople}}(obj))$.

Spatial constraint μ_{21} . The spatial constraint included in the trajectory component checks if an object follows the order of the sequence of regions de-

defined for the trajectory (Ψ). Every time that the component receives spation information from low-level layers, it calculates the membership value to each one of such regions depending on the object position. Secondly, each $\ell(obj)$ (see Table 3) and $\ell_\Psi(obj)$ are updated only if some of the maximum values stored in such a sequence is exceeded (see Table 6). For each one of the trajectories associated to an object, a list $\ell_\Psi(obj)$ is managed in order to store the maximum membership values to regions of Ψ covered by the object. In other words, if the maximum membership value of an object to an area a_k is $\mu_{a_k}(obj)=0.8$, being a_k an area belonging to Ψ and the current membership is $\mu_{a_k}(obj)=0.3$, then the value of $\mu_{a_k}(obj)$ in $\ell_\Psi(obj)$ is not updated. In this way, extremely low values in transitions between regions are avoided.

When an object moves from one region to another, the membership value to the previous region may be really low, fact that may generate the violation of the spatial constraint. The degree of satisfaction of the spatial constraint is calculated as the arithmetic average of all $\mu_{a_k}(obj) \in \ell_\Psi(obj)$. As shown in Table 6, if the object violates the order relation defined in Ψ , the value of $\ell_\Psi(obj)$ will be $\emptyset$ and, therefore, the degree of satisfaction of such a constraint will be 0. The membership function μ_{21} of this constraint is defined as follows:

$$\mu_{21}(obj) = \begin{cases} \frac{\sum_k^{\ell(obj)} \mu_{a_k}(obj)}{|\ell(obj)|} & \text{if } |\ell_\Psi(obj)| > 0; \\ 0 & \text{in other case;} \end{cases}$$

Temporal constraints μ_{31} and μ_{41} . The temporal constraints allow to specify when a particular trajectory is allowed to be followed. The first kind of temporal constraint (μ_{31}) is used to check if a certain time limit t_{max} is exceeded, that is, an object follows a certain trajectory in a correct way if the trajectory is completed in a time shorter than t_{max} . The membership function μ_{31} of the constraint is defined in the following way:

$$\mu_{31}(t_{max}, t_c, t_b) = \begin{cases} 1 & \text{if } (t_{max} = \emptyset) \vee (t_{max} \geq (t_c - t_b)) \\ 0 & \text{in other case;} \end{cases}$$

where t_c represents the current time and t_b the time when the object began the trajectory.

The second kind of temporal constraint (μ_{41}) is used to check if an object follows a certain trajectory within the suitable time interval. To do that, five temporal relationships that relate moments and intervals, based on the Allen's Interval Algebra [3] and the work developed by Lin [17], have been defined. Table 7 shows these relationships.

Relationship (t_c, Int_j)	Logic definition
Before	$t_c < start(Int_j)$
After	$end(Int_j) < t_c$
During	$start(Int_j) < t_c < end(Int_j)$
Start	$start(Int_j) = t_c$
Finish	$end(Int_j) = t_c$

Table 7 Temporal relationships between moments and time intervals.

4. Model application: trajectory analysis

| 275 |

Formally, the constraint μ_{41} is defined as follows:

$$\mu_{41}(Int_j, t_c) = \begin{cases} 1 & \text{if } (Int_j = \emptyset) \vee Start(t_c, Int_j) \vee During(t_c, Int_j) \\ & \vee Finish(t_c, Int_j) \\ 0 & \text{in other case;} \end{cases}$$

An object satisfies a temporal constraint with an associated interval Int_j if the current moment t_c belongs to such an interval. This fact takes place when some of the following conditions is met: $Start(t_c, Int_j)$, $During(t_c, Int_j)$ or $Finish(t_c, Int_j)$.

The next step after defining the concept in a general way and developing the normality component is to make instances for particular environments. Tables 8 and 9 show the set of particular instances for the scene of Figure 5. Each one of the rows represents an instance of the normal trajectory concept, while each one of the columns reflects a particularisation of the defined constraints.

$c_{1,y}^1$	$\tilde{\mu}_{11}$	$\tilde{\mu}_{21}$	$\tilde{\mu}_{31}$	$\tilde{\mu}_{41}$
$c_{1,1}^1$	$\mathcal{I} = \{\text{vehicle}\}$	$\Psi = \{a_1, a_2, a_3, a_4, a_5, a_6\}$	150sg	$\emptyset$
$c_{1,2}^1$	$\mathcal{I} = \{\text{vehicle}\}$	$\Psi = \{a_1, a_2, a_3, a_4, a_5, a_6, a_8, a_9, a_{10}, a_{22}\}$	150sg	$\emptyset$
$c_{1,3}^1$	$\mathcal{I} = \{\text{vehicle}\}$	$\Psi = \{a_1, a_2, a_3, a_4, a_5, a_6, a_8, a_9, a_{11}, a_{13}, a_{14}, a_{15}\}$	150sg	$\emptyset$
$c_{1,4}^1$	$\mathcal{I} = \{\text{vehicle}\}$	$\Psi = \{a_7, a_8, a_9, a_{10}, a_{22}\}$	150sg	$\emptyset$
$c_{1,5}^1$	$\mathcal{I} = \{\text{vehicle}\}$	$\Psi = \{a_7, a_8, a_9, a_{11}, a_{13}, a_{14}, a_{15}\}$	150sg	$\emptyset$
$c_{1,6}^1$	$\mathcal{I} = \{\text{vehicle}\}$	$\Psi = \{a_7, a_8, a_9, a_{11}, a_5, a_6\}$	150sg	$\emptyset$
$c_{1,7}^1$	$\mathcal{I} = \{\text{vehicle}\}$	$\Psi = \{a_{23}, a_{12}, a_{11}, a_5, a_6\}$	150sg	$\emptyset$
$c_{1,8}^1$	$\mathcal{I} = \{\text{vehicle}\}$	$\Psi = \{a_{23}, a_{12}, a_{11}, a_{13}, a_{14}, a_{15}\}$	150sg	$\emptyset$
$c_{1,9}^1$	$\mathcal{I} = \{\text{vehicle}\}$	$\Psi = \{a_{23}, a_{12}, a_9, a_{11}, a_5, a_6, a_8, a_9, a_{10}, a_{22}\}$	150sg	$\emptyset$

Table 8 Set of instances of the normal trajectory concept for vehicles.

$c_{1,y}^1$	$\tilde{\mu}_{11}$	$\tilde{\mu}_{21}$	$\tilde{\mu}_{31}$	$\tilde{\mu}_{41}$
$c_{1,10}^1$	$\mathcal{I} = \{\text{pedestrian}\}$	$\Psi = \{a_{29}, a_{30}, a_{31}\}$	$\emptyset$	[HU]
$c_{1,11}^1$	$\mathcal{I} = \{\text{pedestrian}\}$	$\Psi = \{a_{31}, a_{30}, a_{29}\}$	$\emptyset$	[HU]
$c_{1,12}^1$	$\mathcal{I} = \{\text{pedestrian}\}$	$\Psi = \{a_{18}, a_{17}, a_{16}\}$	$\emptyset$	$\emptyset$
$c_{1,13}^1$	$\mathcal{I} = \{\text{pedestrian}\}$	$\Psi = \{a_{16}, a_{17}, a_{18}\}$	$\emptyset$	$\emptyset$
$c_{1,14}^1$	$\mathcal{I} = \{\text{pedestrian}\}$	$\Psi = \{a_{16}, a_4, a_{28}, a_{13}, a_{19}\}$	$\emptyset$	$\emptyset$
$c_{1,15}^1$	$\mathcal{I} = \{\text{pedestrian}\}$	$\Psi = \{a_{19}, a_{113}, a_{28}, a_4, a_{16}\}$	$\emptyset$	$\emptyset$
$c_{1,16}^1$	$\mathcal{I} = \{\text{pedestrian}\}$	$\Psi = \{a_{19}, a_{20}, a_{21}\}$	$\emptyset$	$\emptyset$
$c_{1,17}^1$	$\mathcal{I} = \{\text{pedestrian}\}$	$\Psi = \{a_{21}, a_{20}, a_{19}\}$	$\emptyset$	$\emptyset$
$c_{1,18}^1$	$\mathcal{I} = \{\text{pedestrian}\}$	$\Psi = \{a_{24}, a_{25}\}$	$\emptyset$	$\emptyset$
$c_{1,19}^1$	$\mathcal{I} = \{\text{pedestrian}\}$	$\Psi = \{a_{25}, a_{24}\}$	$\emptyset$	$\emptyset$
$c_{1,20}^1$	$\mathcal{I} = \{\text{pedestrian}\}$	$\Psi = \{a_{26}, a_{27}\}$	$\emptyset$	$\emptyset$
$c_{1,21}^1$	$\mathcal{I} = \{\text{pedestrian}\}$	$\Psi = \{a_{27}, a_{26}\}$	$\emptyset$	$\emptyset$

Table 9 Set of instances of the normal trajectory concept for pedestrians. HU is the identifier of the time interval defined for the open hours at University.

5 Experimental results

The scenario chosen for validating and evaluating the model and the designed normality components is graphically shown in Figure 5. It is an urban traffic environment where both vehicles and pedestrians are submitted to traffic laws, such as correct trajectories or suitable speed for each stretch. This environment is monitored from a security camera located in the ORETO research group at the University of Castilla-La Mancha. The process of experimental validation consists in the monitoring of ten video scenes with a duration between 2'30" and 3'. One of the premises was to select visual material that covered multiple situations, with different illumination conditions and image resolutions. Figures 7 and 8 show the degree of illumination of each one of the scenes employed to test the system. Scenes 7 to 10 have a lower resolution.



Fig. 7 Screenshots of the scenes employed in the tests.

The main goal of the conducted experiments is to evaluate the proposed methods to classify objects and analyse the normality of trajectories and speed. For this reason, those moving objects that are not detected by the tracking algorithm, being global errors of the surveillance systems, were not reflected on the results discussed in this work. In other words, the normality components cannot analyse the behaviour of undetected objects and, therefore, these errors are due to the low-level processing of spatio-temporal information.

Each one of the detected objects in a single frame is considered as a situation to be analysed. For each one of these situations, the system checks if the classification, the regions calculation, trajectory analysis, speed analysis, and global analysis are normal. The matches and errors of the normality components and the global analysis are classified into true positives (TP), true negatives (VN), false positives (FP), and false negatives (FN); each one of them representing the following situations:

- **True positive (TP):** normal situation correctly classified as normal.

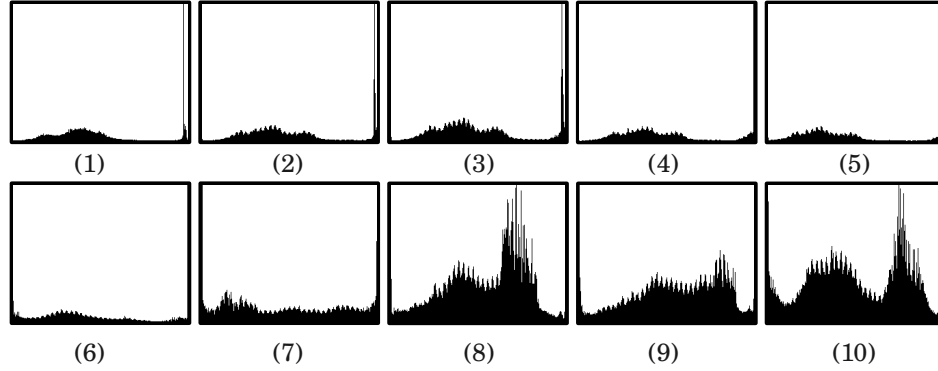


Fig. 8 Set of histograms that represent the brightness and contrast values of each one of the test scenes. The higher the values in the Y axis, the higher the illumination of the scene. The first six scenes correspond to a cloudy day where the objects lack of marked shadows while the remaining four have better illumination conditions.

- **True negative (TN):** anomalous situation correctly classified as anomalous.
- **False positive (FP):** anomalous situation incorrectly classified as normal.
- **False negative (FN):** normal situation incorrectly classified as anomalous.

Furthermore, for each one of the evaluated scenes, the relationships between these parameters have been established by means of the following coefficients [15]:

- Sensitivity or true positive rate (TPR): $TPR = TP / (TP + FN)$, that is, the hit rate.
- False negative rate (FPR): $FPR = FP / (FP + TN)$.
- Precision or accuracy (ACC): $ACC = (TP + TN) / (P + N)$, where P is the number of positive cases and N the number of negative cases.
- Specificity (SPC) or true negative rate: $SPC = TN / N = TN / (FP + TN) = 1 - FPR$.
- Positive predictive value (PPV): $PPV = TP / (TP + FP)$.
- Negative predictive value (NPV): $NPV = TN / (TN + FN)$.
- False discovery rate (FDR): $FDR = FP / (FP + TP)$.

Next, the results obtained taking into account the previous criteria are discussed. Particularly, Table 10 shows the success rate of the object classification process for each one of the ten test scenes. The three first causes of errors are due to the classification algorithm developed in this work, while the fourth one is due to the computer vision algorithms used in low-level layers, hardly avoided by the classification process. As can be appreciated, the classification method gets good results; concretely, 95% success rate as the average value, and 91% in the worst case.

Table 11 shows the results obtained in the estimated process of regions where detected objects are located, taking into account the intersection between their support or base points and the polygons that represent the re-

Object classification					
Scene	Success	Errors			
		Wrong rule	Rules not fired	Historic	Ellipse change
1	7176 (95%)	287	1	35	0
2	6760 (96%)	188	1	56	0
3	6657 (95%)	227	28	49	30
4	11065 (99%)	84	4	0	19
5	1397 (99%)	0	1	0	0
6	2242 (93%)	153	1	0	0
7	7498 (95%)	345	0	15	8
8	2866 (97%)	19	4	0	64
9	6115 (91%)	359	0	123	92
10	1887 (98%)	3	1	0	27
Average		95,8 %			

Table 10 Results obtained in the object classification process of pedestrians and vehicles. More common error causes are a) wrong fired rule; b) no rule fired, being the system unable to classify the object; c) historic, the system takes into account previous classifications of the object, making errors in current classifications if the previous ones were incorrect; d) ellipse change between two objects that cross at a point (this fact is critical when the objects belong to different classes).

gions. This process mainly depends on the tracking algorithm robustness and the association of the ellipses that wrap the objects. If the ellipses do not wrap most of the support points of objects, the error rate in the classification process will be high. This class of errors imply the violation of spatial constraints, generating false positives and false negatives.

Estimation of objects location (region calculation)			
Scene	Success	Errors	
		Wrong ellipse	Object with multiple ellipses
1	7069 (94%)	238	192
2	7001 (99%)	4	0
3	6920 (98%)	71	0
4	11130 (99%)	23	16
5	1398 (100%)	0	0
6	2396 (100%)	0	0
7	7732 (98%)	137	0
8	2910 (98%)	26	17
9	6654 (99%)	35	0
10	1906 (99%)	12	0
Average		98,4 %	

Table 11 Results obtained in the estimation of regions where the detected objects are located.

Table 12 shows the results obtained in the normality analysis of the trajectory concept. The average success rate in this case is 96,4%. Some of the errors made are due to errors in previous processes, such as object misclassification or wrong region estimation. These errors occasionally cause the violation of the constraints of the trajectory concept so that the monitored

5. Experimental results

| 279 |

object does not have associated normal trajectories, generating false positives or false negatives. The errors made by the preprocessors that provide the information to the normality components do not always imply errors in such components. For instance, a vehicle classified as a pedestrian located over a pedestrian crossing; or a pedestrian located over an allowed area for pedestrians, if the system considers that the pedestrian is located over another allowed area for pedestrians. In this last situation and although the region estimation by the system was not correct, it will infer a correct result. For this reason, the number of errors when classifying objects or estimating regions may be higher than in the trajectory analysis.

The output of the normality components for each analysed frame is a numeric value used to study the membership to different normality intervals: *absolutely normal*, *possibly normal*, *suspicious*, *possibly abnormal*, and *absolutely abnormal*. Any normal situation with an exclusive membership to the *absolutely normal* or *possibly normal* intervals is considered as a success of the trajectory component and, in the same way, any anomalous situation with an exclusive membership to the *absolutely abnormal* or *possibly abnormal* is also considered as a success. The contrary cases imply false positives and false negatives, which are considered as errors. Regarding the *suspicious* interval, an error is considered in the following cases: i) a normal situation where the membership to such interval is higher than to the *possibly normal* interval, and ii) an anomalous situation where the membership to the *suspicious* interval is higher than to the *possibly abnormal* interval. To consider suspicious situations as critical or not should rely on the layers responsible for the decision making process, depending on the desired security level. The more strict the system, the higher the probability of alarm activation and the lower the probability of undetected anomalous situations.

Trajectory analysis (Component 1)							
Scene	Number of frames	Number of situations		Success	Errors		
		Normal	Abnormal		C1	C2	C3
1	3943	7224	275	7093 (94%)	184	222	0
2	3000	7142	137	6847 (97%)	154	4	0
3	3000	6751	240	6731 (96%)	209	21	30
4	3250	11051	121	11026 (99%)	88	39	19
5	995	1398	0	1397 (99%)	1	0	0
6	1471	2343	53	2242 (93%)	154	0	0
7	5233	7831	35	7590 (96%)	134	134	8
8	1906	2738	215	2920 (98%)	7	26	0
9	2165	6390	299	6420 (95%)	234	35	0
10	772	1843	75	1875 (97%)	4	12	27
Total				96,4 %			

Table 12 Results obtained by the trajectory analysis component. Labels C1 (object misclassification), C2 (wrong position of the ellipse) and C3 (ellipse swap between objects) refer to error causes.

Table 13 shows the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) for the trajectory component and the relationships between them. These parameters are used to measure the quality of the proposed methods in this work in order to appreciate the number of normal situations classified as normal, the number of detected anomalous situations or the number of wrong alarm activations. Particularly, the results shown in this table are satisfactory since the number of false positives and false negatives is low in relation to the total number of analysed situations. More critical situations are those anomalous situations classified as normal or, in other words, the number of false positives (FP). This table also shows that the value of this parameter is really low in the ten test scenes.

Coefficient calculation for the trajectory component											
Scene	TP	TN	FP	FN	TPR	FPR	ACC	SPC	PPV	NPV	FDR
1	6928	165	71	335	0.95	0.30	0.94	0.699	0.98	0.330	0.01
2	6847	0	0	158	0.98	-	0.977	-	1	0	0
3	6684	47	5	255	0.963	0.096	0.963	0.904	0.999	0.156	0
4	11026	0	4	142	0.987	1	0.987	0	1	0	0
5	1397	0	0	1	0.99	-	0.99	-	1	0	0
6	2228	14	0	154	0.935	0	0.936	1	1	0.083	0
7	7590	0	0	276	0.965	-	0.965	-	1	0	0
8	2766	154	26	7	0.997	0.144	0.989	0.856	0.991	0.957	0.009
9	6181	239	0	269	0.958	0	0.96	1	1	0.470	0
10	1875	0	0	43	0.978	-	0.978	-	1	0	0

Table 13 Existing relationships between true/false positives and true/false negatives in the trajectory component.

Table 14 shows the results obtained in the speed classification process from the normality point of view, calculated by the second normality component developed in this work. In this case, the average success rate is 98,7%. In the conducted experiments, both vehicles and pedestrians use to drive along and walk around, respectively, at a normal speed. If this is not the case, most of anomalous behaviours were detected.

Since this normality component is based on the displacement of the ellipse that wraps the monitored object, the partial occlusions are important to be taken into account, which might cause abrupt changes in the ellipse position and, therefore, in the speed estimation. In other words, while a partial occlusion is taking place, the ellipse maintains the visible part of the object and might change the position abruptly when the previous invisible part becomes visible. In the test environment, this phenomenon happens three times (see Figure 5): i) lower left part by the tree branches, ii) upper central part by the bushes of the roundabout and iii) right central part by the fence of the building. The used learning algorithm deals with this problem since it learns that the displacements in this kind of regions may be higher, considering the object speed as normal.

Speed analysis (Component 2)								
Scene	Number of frames	Number of situations		Success	Errors			
		Normal	Abnormal		C1	C2	C3	C4
1	3943	7224	275	7478 (99%)	0	0	21	0
2	3000	7142	137	6923 (98%)	40	0	25	17
3	3000	6751	240	6842 (97%)	0	25	124	0
4	3250	11051	121	11095 (99%)	0	0	77	0
5	995	1398	0	1398 (100%)	0	0	0	0
6	1471	2343	53	2344 (97%)	13	39	0	0
7	5233	7831	35	7813 (99%)	0	18	35	0
8	1906	2738	215	2943 (99%)	0	0	10	0
9	2165	6390	299	6639 (99%)	0	15	10	25
10	772	1843	75	1918 (100%)	0	0	0	0
Total				98,7%				

Table 14 Results obtained by the normality component of speed analysis. Labels C1 (object misclassification), C2 (abrupt position change or wrong size ellipse), C3 (wrong fired rules), and C4 (ellipse swap between objects of different class) refer to error causes.

Table 15 shows the existing relationship between TP, TN, FP, and FN. The good results are a consequence of the reduced number of false positives and false negatives and the high number of true positives and true negatives.

Coefficient calculation for the component 2											
Scene	TP	TN	FP	FN	TPR	FPR	ACC	SPC	PPV	NPV	FDR
1	7460	18	21	0	1	0.53	0.99	0.462	0.99	1	0,002
2	6786	137	0	82	0,99	0	0,988	1	1	0,626	0
3	6641	201	39	110	0,984	0,163	0,979	0,838	0,994	0,646	0,005
4	10993	102	15	62	0,994	0,128	0,993	0,872	0,999	0,622	0,001
5	1398	0	0	0	1	-	1	-	1	-	0
6	2344	0	39	13	0,994	1	0,978	0	0,984	0	0,02
7	7788	25	10	43	0,995	0,286	0,995	0,714	0,999	0,368	0,001
8	2918	25	10	0	1	0,286	0,997	0,714	0,997	1	0
9	6589	50	10	40	0,994	0,167	0,99	0,83	0,998	0,556	0,001
10	1843	75	0	0	1	0	1	1	1	1	0

Table 15 Existing relationships between true/false positives and true/false negatives in the normality component of speed analysis.

Tables 16 and 17 show the results of the global analysis process, making use of the OWA aggregation operators to combine the output given by the two normality components and get a global degree of normality. The errors in this process can be due to the first component, the second or either both at the same time (this last situation is counted as a single error in the global analysis). Occasionally, the errors caused by one of the components do not produce an error in the global analysis due to the use of a vector of weights by the aggregation operator. For instance, a situation considered as suspicious

by the first component and absolutely normal by the second one might be considered as possibly normal by the global analysis module.

Global analysis: trajectories and speed							
Scene	Number of frames	Number of situations		Success	Errors		
		Normal	Abnormal		C1	C2	Both
1	3943	7224	275	7072 (94%)	406	21	0
2	3000	7142	137	6805 (97%)	118	42	40
3	3000	6751	240	6504 (93%)	260	149	0
4	3250	11051	121	10949 (98%)	146	77	0
5	995	1398	0	1397 (99%)	1	0	0
6	1471	2343	53	2190 (91%)	154	39	13
7	5233	7831	35	7537 (95%)	276	53	0
8	1906	2738	215	2910 (98%)	33	10	0
9	2165	6390	299	6370 (95%)	269	50	0
10	772	1843	75	1875 (97%)	43	0	0
Total				95,7%			

Table 16 Results obtained by the global module combining the analysis of the two normality components. Labels C1 and C2 refer to errors caused by the first and second components, respectively.

Coefficient calculation for the global analysis											
Sce.	TP	TN	FP	FN	TPR	FPR	ACC	SPC	PPV	NPV	FDR
1	6889	183	92	335	0.95	0.33	0.94	0.67	0.98	0,353	0.01
2	6668	137	0	200	0.97	0	0,971	1	1	0,407	0
3	6334	248	44	365	0,946	0,151	0,941	0,849	0,993	0,405	0,01
4	10847	102	19	204	0,982	0,157	0,98	0,843	0,998	0,333	0,001
5	1397	0	0	1	0,99	-	0,99	-	1	0	0
6	2189	14	39	154	0,934	0,736	0,919	0,264	0,982	0,083	0,02
7	7512	25	10	319	0,959	0,286	0,958	0,714	0,999	0,073	0,001
8	2731	179	36	7	0,997	0,167	0,985	0,833	0,987	0,962	0,01
9	6081	289	10	309	0,952	0,033	0,95	0,967	0,998	0,483	0,001
10	1830	75	0	43	0,977	0	0,978	1	1	0,636	0

Table 17 Existing relationships between true/false positives and true/false negatives in the global normality analysis process.

On the other hand, Figure 9 graphically shows the relationship between two of the parameters related to anomalous situations, TN and FP. The number of true negatives (TN) represents the number of anomalous situations detected and correctly understood by the system, while the number of false positives (FP) refers to the anomalous situations unnoticed by the system. The higher the number of false positives and the lower the number of true negatives, then the lower the quality and the efficiency of the methods employed to understand events and situations. The diagrams of Figure 9 reflect

5. Experimental results

| 283 |

that the proposed methods in this work do not only detect the normal situations correctly but also the anomalous situations that rarely take place.

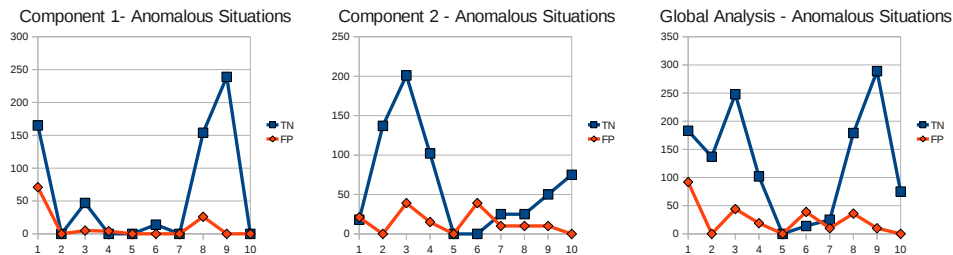


Fig. 9 Relationships between the anomalous situations detected by the normality components and the global analysis. True negatives (TN) represent detected anomalous situations, while false positives (FP) represent anomalous situations understood as normal.

Tables 18 and 19 show the total and average times spent by each one of the processes that compose the surveillance system. The region estimation where an object is located is the most time-consuming task. Even so, the time spent in the worst case is 17 milliseconds, approximately, which allows the system to quickly respond to any event. New versions of the algorithms will be proposed in future works to reduce this time.

Time (ms.)											
Sce.	Elem.	Object classification		Object location		Trajectory analysis		Speed analysis		Global analysis	
		T	M	T	M	T	M	T	M	T	M
1	7499	255,86	0,03	58020,09	7,73	113,54	0,02	87,57	0,01	78,75	0,01
2	7005	198,65	0,02	52288,12	7,46	141,93	0,02	78,2	0,01	61,29	0,008
3	6991	214,94	0,03	80829,99	11,56	235,01	0,03	96,84	0,01	79,1	0,01
4	11172	376,16	0,03	98661,42	8,83	272,14	0,02	99,06	0,01	80,21	0,007
5	1398	54,77	0,03	12375,41	8,85	74,33	0,05	14,96	0,01	50,77	0,03
6	2396	68,79	0,02	23938,75	10,02	125,02	0,05	25,31	0,01	39,79	0,01
7	7866	288,05	0,03	69862,03	8,88	183,93	0,02	82,99	0,01	85,68	0,01
8	2953	79,76	0,02	51176,20	17,33	133,56	0,04	26,44	0,008	37,92	0,01
9	6689	221,45	0,03	75399,88	11,27	176,43	0,02	48,33	0,007	73,26	0,01
10	1918	70,41	0,03	15376,81	8,01	63,7	0,03	29,93	0,01	37,88	0,01

Table 18 Time spent by the system processes, measured in milliseconds. Sce. is scene, elem. are elements to analyse, T is the total time spent, and M is the average time.

The results obtained prove that the design of components by means of the formal model discussed in this work is feasible, gives a high performance, offers response times really short and, finally, allows to represent knowledge with high interpretability. Furthermore, the two components developed have been combined with success, thanks to the use of OWA operators, to get

Time spent by the learning process			
Process	Examples of the training set	Time spent in learning (ms)	Number of fired rules
Learning of rules for object classification	4880	1921	56
Learning of rules for speed classification	2680	1610	15

Table 19 Time spent by the learning process of fuzzy rules for object and speed classification.

a global evaluation of the objects behaviour, which is normal if they follow one normal trajectory at a suitable speed. Although the success rates are often high (between 91% and 99%), it is important to take into account that the duration of the test videos ranges from 2'30" to 3'. This fact implies to include future modifications is still needed to reduce the number of alarm activations, above all in longer analysis times.

On the other hand, the goodness of the obtained results is also due to the system robustness against errors made by low-level layers. To conclude this section, some of these errors and the solutions adopted to reduce the number of false positives and false negatives in the event and behaviour understanding are discussed:

- The object classifier maintains a persistent register about the different classifications of objects since they were first detected until they disappear, that is, the classification of objects does not only depend on the last analysed situation. In this way, if an object was correctly classified during a long period of time, a misclassification at particular moments do not cause classification errors.
- The object classification is based on the object dimensions, movements and regions where it is located without taking into account their size. Therefore, although the ellipse that wraps an object in the tracking process is not perfect, the classification algorithm will work correctly.
- The developed surveillance system also manages a persistent register of the regions covered by each object together with the maximum degree of membership to each one of them. In this way, occasional segmentation errors and wrong ellipses do not affect to the degree of satisfaction of spatial constraints. In other words, the system believes that the object follows the sequence of regions that form the normal trajectory in a correct way although there are certain deviations at particular moments.
- Abrupt errors in the ellipse positioning in case of partial occlusions produced by static environmental elements do not cause error in the speed estimation. This is due to the proposed learning algorithm is able to learn the regions where this phenomenon takes place.
- The regions that compose the environment are classified as entry/exit areas where objects appear and disappear and intermediate areas. The occlusions might cause that the system loses the reference of an object and assigns a new identifier to a previously detected object. The trajectory definition through the proposed formal model allows to define trajectories

whose sequence of regions is only composed of intermediate areas. In this way, when an object is detected again on an intermediate area due to an occlusion, the system will be able to keep on assigning normal trajectories to such an object.

- The global normality value for a particular object depends on the combination of the analysis carried out by each one of the normality components. In this way, although the evaluation of one component is not right, the final normality degree might be correct.

6 Conclusions

Since their first developments, traditional video surveillance systems have been designed to monitor environments. However, these systems have several limitations to satisfy the security demands posed by the society. This need together with the arrival of new technologies and the price reduction of dedicated hardware for security represent some of the relevant reasons why intelligent surveillance is currently one of the hot research topics.

One of the main challenges in this field is to provide security expert systems with the autonomy and ability required to automatically understand events and analyse behaviours in order to improve the productivity and effectiveness in surveillance tasks. In complex environments where multiple situations take place at the same time, human agents are restricted to deal with all them. On the other hand, artificial expert systems do not have these limitations due to their processing capabilities. Furthermore, artificial systems are not affected by factors such as fatigue or tiredness and they can be more effective than people when recognising certain kinds of events, such as the detection of suspicious or unattended objects.

In the last few years, multiple second and third generation surveillance systems have been developed both in the commercial and academic fields. Most of them have been designed to solve particular problems in specific scenarios. These systems provide several advantages over the traditional ones, but two goals need to be reached in order to maintain this progress. i) First, the design of scalable surveillance systems that allow to include new modules to increase the analysis capacity and to aggregate their output to get a global view of the environment state. ii) Secondly, there exists a need for a higher flexibility to vary the artificial system behaviour depending on the requirements of the analysed environment and the kinds of monitored objects. The way of monitoring the behaviour of a particular object may change in relation to the rest of objects. Depending on the kind of objects that appear in the scenario and the security level established, the system must provide the flexibility to configure the existing analysis modules so that the system designed in a global way can be adapted to particular environments.

This work proposes a formal model for the design of scalable and flexible surveillance systems. This model is based on the normality definition of events and behaviours according to a set of independent components, which can be instantiated in particular monitored environments. These modules,

denoted as *normality components*, specify how each kind of object should ideally behave depending on the aspect to watch, such as the trajectory that follows or the monitored speed. This model also allows to specify the components that are employed for each kind of object in each environment, increasing the flexibility. In addition, the output of these components are combined by means of aggregation operators to get a global view of the behaviour of each single object and the whole environment. The integration of new normality components does not imply modifications on the rest, increasing the analysis capacity of the artificial system.

Two normality components, trajectories and speed analysis, have been defined by means of this model. Both make use of the spatial information gathered by the security cameras, previously processed by modules of low-level layers. Although the analysis of trajectories and speed have been widely studied by other researchers, most of the existing approaches are based on the analysis of spatial and temporal information. This information may be enough to recognise the route followed by objects or to estimate their speed, but it is not enough for surveillance systems that need to deal with a higher number of factors.

The proposed model defines the normality of a concept using a set of fuzzy constraints. This fact allows to develop more complete and suitable definitions for surveillance systems. In this way, the system also provides flexibility in the internal analysis performed by the normality components, since multiple constraints can be easily added or removed. The use of fuzzy logic not only deals with the uncertainty and vagueness of low-level information, but also allows us to represent the expert knowledge used by the surveillance system with a high interpretability by means of linguistic labels, providing it with the expressiveness required to justify the decision making. Precisely, in order to facilitate the knowledge acquisition and the deployment of component instances in particular environments, different knowledge acquisition tools and machine learning algorithms have been associated to the normality components.

Finally, the experimental results prove the feasibility of the proposed model and the designed normality components, both in relation to the events and behaviours understanding and the time response to provide such results, two key aspects in surveillance systems. However, to improve the system by reducing the number of false alarms is needed in order to minimise the human dependence. One of these improvements may involve the use of fusion techniques in low-level layers, in addition to the output of the normality components. In this way, a component may deal with the information gathered by multiple sensors about a particular object.

The management of redundant information may improve the classification and understanding results when the data provided by a single source is not reliable. Another future research line is to develop new normality components that extend the analysis capabilities of the system. Finally, the development of components that analyse and detect specific anomalous situations will be also addressed so that their output can be combined with the normality modules to improve and complete the analysis of the environment state.

Acknowledgements This work has been founded by the Spanish Ministry of Industry (MITYC) under Research Project HESPERIA (CENIT Project), and the Council of Science and Technology of Castilla-La Mancha under projects PII2I09-0052-3440 and PII1C09-0137-6488.

References

1. <http://opencv.willowgarage.com/wiki/VideoSurveillance>
2. Albusac J, Castro-Schez J.J, Lopez-Lopez L.M, Vallejo D, Jimenez-Linares L (2009) A supervised learning approach to automate the acquisition of knowledge in surveillance systems. *Signal Processing* 89(12):2400-2414.
3. Allen J.F, Ferguson G (1994) Actions and Events in Interval Temporal Logic. *Journal of Logic and Computation* 4(5):531-579.
4. Aguilera J, Wildenauer H, Kampel M, Borg M, Thirde D, Ferryman J (2005) Evaluation of motion segmentation quality for aircraft activity surveillance. *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 293-300.
5. Blauensteiner P, Kampel M (2004) Visual Surveillance of an Airports Apron-An Overview of the AVTRACK Project. *Digital Imaging in Media and Education, Annual Workshop of AAPR*.
6. Bloisi D, Iocchi L, Remagnino P, Monekosso N.D (2009) ARGOS-A Video Surveillance System for Boat Traffic Monitoring In Venice. To appear in *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*.
7. Buxton H (2004) Learning and understanding dynamic scene activity: a review. *Image and Vision Computing* 1(21):125-136
8. Carter N, Young D, Ferryman J (2006) A combined bayesian markovian approach for behaviour recognition. *Proceedings of the 18th International Conference on Pattern Recognition*, 761-764.
9. Cathey F.W, Dailey D.J (2005) A novel technique to dynamically measure vehicle speed using uncalibrated roadway cameras. *IEEE Intelligent Vehicles Symposium*, 777-782.
10. Chen T.P, Haussecker H, Bovyryn A, Belenov R, Rodyushkin K, Kuranov A, Eruhimov V (2005) Computer vision workload analysis: Case study of video surveillance systems. *Intel Technology Journal* 9(2):109-118.
11. Cho Y, Rice J (2006) Estimating velocity fields on a freeway from low-resolution videos. *IEEE Transactions on Intelligent Transportation Systems* 7(4):463-469.
12. Project Co-friend Official Site <http://www.co-friend.net>.
13. Collins R, Lipton A, Kanade T, Fujiyoshi H, Duggins D, Tsin Y, Tolliver D, Enomoto N, Hasegawa (2000) A System for Video Surveillance and Monitoring. Technical report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University.
14. Dee H.M, Velastin S.A (2008) How close are we to solving the problem of automated visual surveillance?. *Machine Vision and Applications* 5(19):329-343.
15. Fawcett, T (2003) Roc graphs: Notes and practical considerations for data mining researchers. Technical report hpl-2003-4, HP Laboratories, Palo Alto, CA, USA.
16. Haritaoglu I, Harwood D, Davis L.S (2000) W^4 : Real-Time Surveillance of People and Their Activities. *Pattern Analysis and Machine Intelligence*, 22(8):809-830.
17. Lin L, Gong H, Li L, Wang L (2009) Semantic event representation and recognition using syntactic attribute graph grammar. *Pattern Recognition Letters* 30(2):180-186.
18. Johnson N, Hogg D (1996) Learning the distribution of object trajectories for event recognition. *Image and Vision Computing* 14(8):609-615.
19. Khoudour D, Deparis J.P, Bruyelle J.L, Cabestaing F, Aubert D, Bouchafa S, Velastin S.A, Vincencio-Silva M.A and Wherett M (1997) Project Cromatica, *Lecture Notes in Computer Science*, Springer, pp. 157-764
20. Maduro C, Batista K, Peixoto P, Batista J (2008) Estimation of vehicle velocity and traffic intensity using rectified images. In *Proceedings of the 15th IEEE International Conference on Image Processing (ICIP'08)*, pp. 777-780.

21. Magee D (2004) Tracking multiple vehicles using foreground, background and motion models. *Image and Vision Computing* 22(2):143–155.
22. Makris D, Ellis T (2005) Learning semantic scene models from observing activity in visual surveillance. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 35(3):397–408.
23. Morris B.T, Trivedi M (2008) A Survey of Vision-Based Trajectory Learning and Analysis for Surveillance 18(8):1114–1127.
24. Palaio H, Maduro C, Batista K, Batista J (2009) Ground plane velocity estimation embedding rectification on a particle filter multi-target tracking. *IEEE International Conference on Robotics and Automation, ICRA*.
25. Piciarelli C, Foresti G.L (2006) On-line trajectory clustering for anomalous events detection 27(15):1835–1842.
26. Franklin W.R (2006) Pnpoly - point inclusion in polygon test. Available online: <http://www.ecse.rpi.edu/Homepages/wrf/Research/Short Notes/pnpoly.html>
27. Remagnino P, Jones G.A (2007) Novel concepts and challenges for the next generation of video surveillance systems. *Mach. Vis. Appl.* 3(18):135–137.
28. Remagnino P, Velastin S.A, Foresti G.L, Trivedi M (2007) Novel concepts and challenges for the next generation of video surveillance systems. *Mach. Vis. Appl.* 3(18):135–137.
29. Shimrat M (1962) Algorithm 112: Position of point relative to polygon. *Communications of the ACM* 5(8):425
30. Siebel N. T, Maybank S (2004) The advisor visual surveillance system. *Proceedings of the ECCV workshop Applications of Computer Vision*, pp. 103–111.
31. Smith G (2004) Behind the screens: Examining constructions of deviance and informal practices among cctv control room operators in the uk. *Surveillance and Society* 2(2): 376–395
32. Valera M, Velastin S.A (2005) Intelligent distributed surveillance systems: a review. *Vision, Image and Signal Processing* 2(152):192–204.
33. Velastin S.A, Khoudour L, Lo B, Sun J, Vicencio-Silva M (2004) PRISMATICA: a multi-sensor surveillance system for public transport networks. *12th IEE International Conference on Road Transport Information and Control*, pp. 19–25.
34. Yager R.R (1988) On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE transactions on Systems, Man and Cybernetics*, 18(1):183–190.
35. Yager R.R (1993) Families of OWA operators. *Fuzzy sets and systems*, 59(2):125–148.
36. Zadeh L.A (1965) Fuzzy Sets. *Information and Control* 8:338–353.
37. Zadeh L.A (1999) From computing with numbers to computing with words. From manipulation of measurements to manipulation of perceptions. *Circuits and Systems I: Fundamental Theory and Applications*, 46(1):105–119.

Grupo de Investigación ORETO
Universidad de Castilla-La Mancha

Ciudad Real, 1 de Diciembre de 2009

