

**TESIS DE MÁSTER**  
(Máster en Tecnologías Informáticas Avanzadas)

**Vigilancia Inteligente: Modelado de Entornos Reales e Interpretación de Conductas para la Seguridad.**

9 de Julio del 2008

**Autor**

Javier Alonso Albusac Jiménez  
(Ingeniero en Informática)

**Director**

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



Universidad de  
Castilla-La Mancha



Escuela Superior  
de Informática





© Javier A. Albusac Jiménez. Se permite la copia y la 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 compuestas con Gimp y OpenOffice.

# Resumen

Los numerosos atentados llevados a cabo por grupos terroristas en diversos países del mundo, así como el incremento de vandalismo tanto en infraestructuras públicas como privadas, son algunos de los principales temas de preocupación en el panorama actual. Tanto es así, que un gran número de gobiernos e instituciones han tomado la decisión de reforzar las medidas de seguridad como posible solución.

Durante muchos años, los sistemas de seguridad de primera generación, los cuales permitían la monitorización de los puntos más vulnerables y estratégicos de un determinado entorno, han sido implantados en multitud de lugares y han ayudado a la detección de posibles intrusiones y acciones malintencionadas. El talón de Aquiles de estos sistemas es la dependencia absoluta de la actividad humana. Factores implícitos al hombre, como la fatiga acumulada tras varias horas de trabajo o la dificultad de observar varios monitores al mismo tiempo, reducen considerablemente la probabilidad de detectar todas las situaciones anómalas. En los últimos quince años, la evolución constante de la tecnología, así como el abaratamiento del hardware, han impulsado el interés de numerosos autores por realizar propuestas de nuevos sistemas de seguridad más evolucionados, los cuales pueden trabajar de forma semiautomática y tienen capacidad para tomar decisiones por sí mismos.

El tema de investigación principal de la Tesis de Máster se centra en cómo representar el conocimiento de un dominio y cómo detectar las posibles anomalías mediante el análisis de los datos proporcionados por el nivel de sensorización. Con conocimiento de un dominio nos referimos a los elementos que pueden participar en un entorno del mundo real y las posibles relaciones que pueden existir entre ellos. En el presente trabajo se realiza un estudio del estado del arte sobre los sistemas de vigilancia, principalmente en los campos de Visión Cognitiva e Inteligencia Artificial. Además, como aportación se propone un nuevo modelo para representar formalmente la normalidad de un entorno vigilado, que servirá para establecer las bases de posibles líneas de trabajo futuro.

# Índice general

<b>Índice de figuras</b>	<b>V</b>
<b>Índice de tablas</b>	<b>VII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Planteamiento general . . . . .	1
1.2. Objetivos de la Tesis de Máster . . . . .	5
1.3. Marco de trabajo . . . . .	5
1.4. Estructura del documento . . . . .	5
<b>2. Estado del arte</b>	<b>7</b>
2.1. Sistemas de vigilancia inteligente . . . . .	7
2.1.1. Evolución de los sistemas de vigilancia . . . . .	7
2.1.2. Principales fases de un sistema de vigilancia inteligente . . . . .	12
2.1.3. Aplicaciones . . . . .	17
2.2. Modelado de entornos reales . . . . .	18
2.2.1. Ontologías . . . . .	18
2.2.2. Lógica de primer orden: cálculo de situaciones y cálculo de eventos	21
2.2.3. Relaciones espaciales . . . . .	24
2.2.4. Relaciones temporales: álgebra de Allen . . . . .	26
2.3. Análisis e interpretación de comportamientos . . . . .	29
2.3.1. Dynamic time warping (DTW) . . . . .	29
2.3.2. Máquinas de estados finitos . . . . .	30
2.3.3. Gramáticas libres de contexto . . . . .	30
2.3.4. Modelos ocultos de Markov . . . . .	31

2.3.5. Redes Neuronales . . . . .	31
2.3.6. Modelos difusos . . . . .	32
2.4. Principales problemas de los sistemas de vigilancia de 3ª generación . . . . .	33
<b>3. Representación y análisis de la normalidad en entornos vigilados</b>	<b>35</b>
3.1. Análisis de normalidad vs anormalidad . . . . .	35
3.2. Primera aproximación: Intelligent Traffic Control . . . . .	36
3.3. Modelo para la representación y formalización de la normalidad . . . . .	39
3.4. Caso de estudio: definición de trayectorias normales . . . . .	42
3.4.1. Normalidad del concepto <i>trayectoria normal</i> . . . . .	43
3.4.2. Normalidad del concepto <i>trayectoria normal</i> en el entorno $E_1$ . . . . .	48
3.4.3. Análisis de normalidad de un objeto en movimiento . . . . .	49
3.5. Resultados experimentales . . . . .	49
<b>4. Conclusiones y trabajo futuro</b>	<b>56</b>
<b>Bibliografía</b>	<b>59</b>
<b>A. Resumen de las asignaturas cursadas en el Máster</b>	<b>67</b>
A.1. Metodologías y técnicas de investigación en informática . . . . .	67
A.2. Sistemas heterogéneos en red . . . . .	68
A.3. Sistemas distribuidos avanzados: grid e inteligencia ambiental . . . . .	69
A.4. Cognición y colaboración . . . . .	69
A.5. Procesos software y gestión del conocimiento . . . . .	70
A.6. Técnicas de softcomputing . . . . .	71
<b>B. Currículum Vitae</b>	<b>72</b>

# Índice de figuras

1.1. Niveles de un sistema cognitivo de vigilancia . . . . .	2
2.1. Etapas que podrían formar parte de la arquitectura de cualquier sistema de vigilancia inteligente. . . . .	12
2.2. Relaciones base del esquema RCC8 . . . . .	26
2.3. Ejemplo de la representación de eventos simples y sus relaciones mediante un grafo. . . . .	28
2.4. Ejemplo de propagación de restricciones tras añadir $e_1 < e_4$ y $e_4 < e_2$ . . . .	28
2.5. Relaciones temporales entre periodos de tiempo definidas en el álgebra de Allen. . . . .	29
3.1. Comunicación entre los agentes inteligentes que forman parte del sistema de análisis de situaciones de tráfico en un entorno urbano . . . . .	37
3.2. (a) Escenario urbano capturado por una cámara situada en el grupo de investigación ORETO de la Universidad de Castilla-La Mancha. (b) Definición de zonas en el entorno. . . . .	43
3.3. Ejemplo de una trayectoria normal ( $P_1$ ) definida a partir de un área de comienzo (SidewalkA) y un área de finalización (SidewalkB) . . . . .	45
3.4. Ejemplo de una secuencia de áreas recorridas por un objeto. Cada instante de tiempo $t$ puede cubrir varias áreas al mismo tiempo . . . . .	46
3.5. (a) La trayectoria normal se define mediante una zona de origen y destino. Las líneas discontinuas muestran algunos recorridos que serían reconocidos mediante esta definición de trayectoria normal; (b) Se amplía la definición de la trayectoria normal con la inclusión del paso de peatones como zona intermedia. Al igual que en (a), las líneas discontinuas muestran algunos recorridos válidos acordes a la nueva definición. . . . .	47

---

3.6. (a) Representación general de la asociación entre objetos y conceptos. Se realiza un análisis de normalidad por cada uno de los conceptos asociados al objeto. (b) Asociación del concepto <i>trayectoria normal</i> al objeto $obj_1$ en movimiento. (c) Comportamiento anormal de $obj_1$ debido a que el grado de satisfacción de $c_2$ no supera el valor umbral $\alpha_{c_2}$ . (d) Comportamiento anormal del objeto $obj_1$ debido a que no existe un $r_{1j}$ cuyo grado de satisfacción $SD_{r_{1j}} > 0$ . . . . .	50
3.7. Seguimiento de un coche en el entorno y asociación de trayectorias normales	51
3.8. Arquitectura conceptual del sistema OCULUS . . . . .	53



# Indice de tablas

2.1. Evolución de los sistemas de vigilancia. . . . .	12
3.1. Lista de variables utilizadas para definir las trayectorias normales. . . . .	44
3.2. Relaciones temporales entre instantes e intervalos de tiempo. . . . .	48
3.3. Conjunto de trayectorias normales definidas para el entorno $E_1$ . Cada $r_{ij}$ representa una trayectoria normal, la cual es recorrida por un objeto cuando éste satisface sus restricciones. $\tilde{f}_{18}$ no se incluye en la tabla porque, a diferencia del resto de restricciones, no puede ser definida por el experto, sino que se calcula en función de $d_i$ y $d_c$ . . . . .	49
3.4. Valores numéricos obtenidos en el proceso de seguimiento del objeto de la Figura 3.7 con un valor umbral $\alpha_{c_1} = 0,3$ . . . . .	52
3.5. Número de objetos en movimiento y resultados de clasificación por test. Cada test tiene una duración de 120 segundos y los resultados fueron obtenidos en el mes de Mayo. Durante este mes el sol sale a las 7:00 am y se oculta a las 9:30 pm. Esta información se hace explícita porque las condiciones de luz afectan en gran medida a los resultados. . . . .	54
3.6. Comportamientos normales y anómalos. El comportamiento de un objeto siempre es clasificado como normal o anormal independientemente del número de trayectorias que esté siguiendo . . . . .	54
3.7. Detalle de las causas de error en cada uno de los test . . . . .	54
3.8. Estadísticas relacionadas con la asociación de trayectorias normales y reconocimiento de las mismas. . . . .	55

---

# 1

## Introducción

---

### 1.1. Planteamiento general

Actualmente, el incremento de atentados llevados a cabo por diferentes grupos terroristas, los delitos cometidos por pequeñas bandas organizadas, o el aumento de vandalismo en las grandes ciudades, son algunas de las principales fuentes de preocupación e incertidumbre en la actualidad. Tanto es así, que un gran número de gobiernos e instituciones han tomado la decisión de reforzar las medidas de seguridad como posible solución. Por tanto, hoy en día es común observar cámaras de seguridad instaladas en bancos, casinos, comercios, grandes empresas, calles conflictivas, etc.

La instalación de dichas cámaras permite aumentar la seguridad en la zona y reducir el índice de delincuencia y degradación. En más de una ocasión, los sistemas de seguridad se han convertido en los aliados perfectos de los cuerpos de seguridad, ya que las grabaciones han servido para evitar delitos, o como indicio para conseguir pruebas en el caso de que éstos se hayan producido.

SISTEMAS  
DE VIGILANCIA  
INTELIGENTES

Gran parte de los sistemas implantados actualmente son una evolución de los sistemas de primera generación (formados principalmente por una sola cámara y un monitor). Dichos sistemas están formados por un conjunto de cámaras que envían la señal a una sede central donde un guardia de seguridad puede observar varios monitores. Según Tan Kok Kheng, vicepresidente de la división OEM de WPG Systems, - una de las principales distribuidoras de sistemas de vigilancia avanzados -, “Tras 20 minutos de vigilancia, la atención humana a los detalles del vídeo disminuye hasta niveles inaceptables y la videovigilancia deja de tener sentido. La videovigilancia tradicional ya no puede cumplir las, cada vez mayores, demandas del sector”. Las palabras del Sr. Tan ponen de manifiesto que el talón de Aquiles de los sistemas tradicionales es la dependencia absoluta de la actividad humana, donde factores como la fatiga producida tras varias horas de trabajo, o la dificultad de observar varios monitores al mismo tiempo, reducen considerablemente la probabilidad de detectar todas las situaciones anómalas

que pueden ocurrir en un mismo instante de tiempo. La solución a estas deficiencias pasa por la utilización de sistemas de vigilancia inteligentes capaces de interpretar lo que está ocurriendo. “La videovigilancia ya no puede ser simple y reactiva, necesita ser inteligente y proactiva”, enfatiza el Sr. Tan.

Entre los años 1990 y 2000 comienzan a aparecer frecuentemente publicaciones relacionadas con el análisis avanzado de vídeo e interpretación de escenas. Pero es en esta última década, cuando la comunidad científica muestra mayor interés en dicho campo, sobre todo con numerosas propuestas orientadas a la vigilancia y la seguridad. La aparición de nuevas tecnologías y el abaratamiento del hardware tienen gran parte de culpa del incremento de interés, ya que permiten hacer realidad las propuestas realizadas por los investigadores en este campo.

VISIÓN  
COGNITIVA

La mayoría de los algoritmos y métodos propuestos para el diseño de sistemas de vigilancia avanzados, pertenecen al campo de la Inteligencia Artificial (IA) y al de Visión por Computador; la combinación de ambos da lugar a lo que se denomina como Visión Cognitiva. En la Figura 1.1 se muestra una arquitectura piramidal dividida en cuatro niveles, en los cuales se podrían encuadrar la mayoría de publicaciones realizadas en el campo de la Visión Cognitiva. Los sistemas de visión tradicionales trabajarían con los niveles 0 y 1, mientras que los sistemas de visión cognitivos añadirían los niveles 2 y 3. Por tanto, la principal diferencia entre un sistema de visión cognitivo para la seguridad y un sistema de visión tradicional, es que el primero tiene mayor capacidad para comprender lo que ocurre en un entorno y por tanto, mayor capacidad para tomar decisiones en consecuencia.

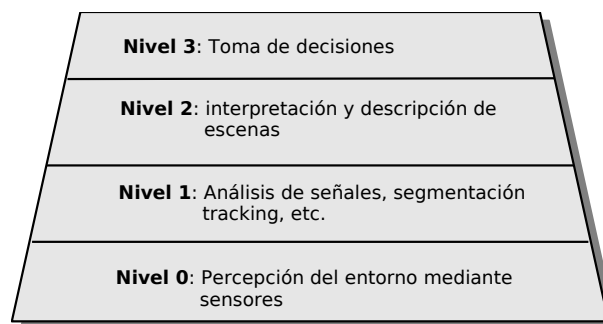


Figura 1.1: Niveles de un sistema cognitivo de vigilancia

A medida que avanzamos en los niveles de la arquitectura, el número de publicaciones relacionadas con la vigilancia y seguridad disminuye notablemente. Un posible motivo podría ser la dependencia de datos e información de cada nivel con su nivel inmediatamente inferior. Por ejemplo, para llevar a cabo un proceso de segmentación y tracking es necesario disponer de la señal de vídeo capturada por una cámara, o bien, para analizar una señal de audio es necesario que un micrófono la haya capturado previamente. El flujo de datos generado en este nivel, sirve de entrada al módulo encargado de analizar las conductas y determinar si son normales. Finalmente, el cuarto nivel necesita como entrada la interpretación y el análisis de normalidad para tomar decisiones (activación de alarmas, ejecución de planes de emergencia, etc.).

SIMULACIÓN

Algunos autores como Ingle et al. [53] y Zhen [110] han optado por simular los niveles inferiores, pudiendo trabajar directamente en los niveles superiores. Además,

en sistemas de seguridad las simulaciones son de gran utilidad ya que para probar cómo se comporta el sistema ante situaciones límite, no es necesario poner en peligro la integridad de las personas o dañar las infraestructuras.

Todos los niveles son necesarios para que un sistema de seguridad cognitivo funcione en perfecta armonía, pero quizás, el nivel 2 es el que comienza a establecer la diferencia con los sistemas tradicionales. Es decir, el segundo nivel disminuye la necesidad de actividad humana y, por tanto, aporta mayor autonomía al sistema (“*videovigilancia inteligente y proactiva*”).

REPRESENTACIÓN  
DEL  
CONOCIMIENTO

Por otra parte, para poder interpretar los eventos que ocurren en una determinada escena y analizar la normalidad de los mismos, es necesario disponer de conocimiento específico de dominio (base de conocimiento de un sistema experto). Según Russell et al. [90], la representación del conocimiento en dominios pequeños no supone un problema y es suficiente con encontrar un vocabulario consistente. En cambio, los dominios complejos requieren representaciones más generales y flexibles. Normalmente, en este caso se utiliza ingeniería ontológica para representar los conceptos abstractos, como pueden ser las acciones, el tiempo, objetos físicos y las creencias. Gruber [48] afirma que una ontología es una especificación explícita de una conceptualización, entendiendo conceptualización como una representación abstracta y simplificada de la porción de realidad que nos interesa.

La definición de ontologías no es la única forma de representar el conocimiento de un dominio complejo, H.I. Christensen en [25] y Yuri Ivanov en [54] optan por gramáticas libres de contexto tanto para la representación como para la interpretación de escenas. Russell [90] cumple el mismo objetivo mediante el uso de la lógica de primer orden y Ghanem [44] mediante redes de Petri, con las cuales establece fácilmente relaciones temporales entre los eventos que se producen en un escena.

Una vez formalizado el conocimiento de un dominio, el sistema puede identificar las entidades participantes a partir de sus características físicas y su comportamiento. El análisis del comportamiento es vital en los sistemas de seguridad avanzados, ya que no sólo permite determinar si lo que está ocurriendo es normal, sino que también permite reforzar la pertenencia de un objeto a una clase [72]. Por ejemplo, si un objeto es clasificado inicialmente como coche con un valor de creencia de 0.5 y como persona con un valor de 0.5, si posteriormente su comportamiento se asemeja más al de un coche, entonces se reforzaría la creencia de la pertenencia a la clase coche.

INCERTIDUMBRE  
Y VAGUEDAD

Además, estos tipos de sistemas suelen observar subdominios del mundo real no deterministas [35], por lo que tienen la dificultad añadida de convivir con la incertidumbre y la vaguedad [106]. La dificultad radica en el hecho de que este tipo de sistemas, en la mayoría de las ocasiones no puede afirmar con total certeza la clase de objetos o eventos que se están produciendo, por tanto, deben manejar varias alternativas al mismo tiempo. Dependiendo del tipo de vigilancia [72] que se lleve a cabo, vigilancia forense o vigilancia predictiva, existirá un mayor grado de incertidumbre o vaguedad. Cuando se realiza una vigilancia predictiva, el objetivo principal es anticiparse a posibles situaciones de riesgo antes de que ocurran. En este caso, la mayoría de autores tienen preferencia por los modelos probabilistas, principalmente redes Bayesianas y múltiples variantes de los modelos ocultos de Markov. En cambio, si se realiza una vigilancia forense, es decir, se analizan los hechos una vez que han ocurrido, puede que la incertidumbre sea mínima y la vaguedad sea mayor. En este caso, los modelos difusos podrían ser más adecuados.

INTERPRETACIÓN  
Y  
RAZONAMIENTO

Como comentábamos al principio de la sección, cada una de las capas presentadas en la estructura piramidal (ver Figura 1.1), necesitaba los datos generados por las capas inferiores. Para ser más precisos, el nivel 2 necesitaba los datos fruto del trabajo realizado por los algoritmos de visión en el nivel 1. Generalmente, estos algoritmos proporcionan variables que indican la posición de un objeto, su forma, tamaño, velocidad, dirección, etc. La suma de todas estas variables da lugar al estado actual de un objeto en la escena. Para poder interpretar lo que ocurre en una escena, el motor de razonamiento del sistema hace uso de estas variables y estudia las relaciones espaciales [9] que existen entre los objetos y el entorno, así como entre los propios objetos [94, 37, 26, 28, 108, 36]. Un cambio en alguna de estas variables se traduce como un evento simple que se produce en la escena. Por otra parte, los eventos compuestos (actividades o acciones complejas) son una secuencia de eventos simples relacionados temporalmente [3, 4]. Por tanto, el análisis de normalidad se basa en el estudio de las relaciones espacio-temporales, los eventos simples y compuestos, y como éstos se parecen a las definiciones de normalidad o anormalidad realizadas previamente.

ANÁLISIS  
DE  
NORMALIDAD

ADQUISICIÓN  
DEL  
CONOCIMIENTO  
Y APRENDIZAJE

Las personas que mejor pueden definir qué tipos de elementos suelen participar en un dominio y cómo se deben comportar, son los expertos en dicho dominio. La situación ideal se daría, cuando el experto tuviera capacidad para formalizar y hacer explícito todo su conocimiento tácito. Pero esto no ocurre siempre así y es entonces cuando los ingenieros del conocimiento deben hacer uso de técnicas de adquisición de conocimiento.

Otro gran inconveniente al que se enfrentan los sistemas de vigilancia cognitivos es la dificultad de reutilizar conocimiento, ya que existe una fuerte dependencia con el entorno observado. Por ejemplo, cada lugar observado por una cámara tiene sus zonas con características y normas propias. Un objeto observado se comportará de forma normal siempre y cuando cumpla dichas normas. Esta dificultad repercute directamente en el proceso de implantación o ampliación del sistema, ya que la ubicación de nuevos sensores implicaría la ampliación de la base de conocimiento por parte del experto. En este caso, los algoritmos de aprendizaje semi-automáticos y automáticos juegan un papel fundamental para agilizar este proceso y eliminar la fuerte dependencia con el experto. Relacionados con este tema, existen algunos trabajos interesantes como el de Jiangung et al. [63], que utiliza algoritmos de *clustering* para aprender y clasificar automáticamente las trayectorias que se producen en una escena; y el de Makris et al. [71] que obtienen automáticamente la topología de una escena mediante el estudio de las trayectorias y los puntos de entrada y salida.

SISTEMAS  
MULTIAGENTE

En este contexto, los sistemas multi-agente suponen un punto de vista muy interesante a la hora de abordar el diseño de un sistema de vigilancia. De hecho, si se dota a los agentes de la autonomía y capacidad necesaria para tomar decisiones, dichos agentes serán capaces de analizar el entorno desde un punto de vista independiente, posibilitando la cooperación entre ellos para obtener una información más detallada sobre el entorno a vigilar. Por otra parte, la propia naturaleza distribuida de los sistemas multi-agente permitiría que un sistema de vigilancia fuera más escalable y más robusto frente a fallos potenciales [82].

## 1.2. Objetivos de la Tesis de Máster

La presente Tesis de Máster tiene dos objetivos principales. El primero de ellos es la elaboración de un estado del arte sobre los sistemas de vigilancia que incluya la evolución desde los primeros sistemas más rudimentarios hasta los más avanzados, donde la mejora y adición de nuevas funcionalidades suponen un reto interesante para la comunidad científica. Además, con el objetivo de ofrecer una visión general se incluirá un estudio detallado sobre las etapas que forman parte de este tipo de sistemas y se destacarán los principales trabajos encuadrados en cada una de ellas, prestando especial atención al modelado de entornos reales y al análisis e interpretación de comportamientos. Para finalizar el estado del arte, se expondrán los principales problemas que presentan los sistemas de vigilancia de última generación, los cuales pueden dar lugar a diferentes líneas de investigación.

Por otra parte, como se comentaba en el planteamiento general, uno de los principales inconvenientes de los sistemas de vigilancia actuales es la dependencia de la actividad humana; el segundo objetivo pretende reducir esta dependencia en la medida de lo posible. Para ello, se elaborará un modelo conceptual que trate la incertidumbre y permita definir formalmente las conductas normales o permitidas en un entorno del mundo real. A partir de estas definiciones, el sistema artificial será capaz de identificar las conductas que son normales, o por el contrario, aquellas que pueden implicar cierto grado de riesgo para el entorno. Para finalizar, se aplicará el modelo conceptual a un caso concreto: análisis de trayectorias realizadas por objetos móviles en espacios abiertos.

## 1.3. Marco de trabajo

El presente trabajo está encuadrado en el Proyecto de Investigación Hesperia: Homeland sEcurity: tecnologíaS Para la sEguridad integRal en espacios públicos e infraestructuras, financiado por el CDTI (organismo adscrito al Ministerio de Industria Turismo y Comercio) con código UCTR060182, 2006-2010.

## 1.4. Estructura del documento

El resto del trabajo está estructurado de la siguiente forma.

- **Capítulo 2.** En este capítulo se presenta el estado del arte con la evolución de los sistemas de vigilancia, las principales fases de las que constan y posibles aplicaciones. El capítulo continúa con el modelado de entornos reales y el análisis e interpretación de comportamientos. Para finalizar se muestran algunos de los problemas que presentan actualmente los sistemas de vigilancia de tercera generación.
- **Capítulo 3.** En este capítulo se expone el trabajo de investigación realizado durante el año académico actual. En concreto, se presenta un modelo conceptual para representar formalmente las conductas normales de un entorno real. Este modelo se utilizará posteriormente para definir y analizar las trayectorias normales en un

entorno vigilado por una cámara de seguridad. El capítulo finaliza con los resultados ofrecidos por una aplicación software donde se ha implementado el modelo mencionado anteriormente.

- **Capítulo 4.** Se exponen las conclusiones y las líneas de trabajo futuro que darán lugar al desarrollo de una Tesis Doctoral.
- **Anexo A.** Resumen de las asignaturas cursadas en el Máster.
- **Anexo B.** Currículum Vitae.

---

# 2

## Estado del arte

---

### 2.1. Sistemas de vigilancia inteligente

Esta sección está basada, principalmente, en la evolución de los sistemas de vigilancia descrita en [101, 23, 72, 89, 65], que engloba desde los primeros sistemas analógicos CCTV, hasta soluciones distribuidas más complejas con capacidad de interpretar eventos y comportamientos del mundo real. Estos últimos se encuentran en fase de desarrollo e investigación y existen numerosos autores que realizan propuestas encuadradas en cada una de las principales fases que componen un sistema de este tipo (Sección 2.1.2).

#### 2.1.1. Evolución de los sistemas de vigilancia

1ª GENERACIÓN

Valera y Velastin [101] 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 sala central.

Entre los problemas que presentan los sistemas CCTV destacamos 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 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, en consecuencia, falta de atención; 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.

2ª GENERACIÓN

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 dependencia que existe con la actividad humana, 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 resuelto y existe un gran número de líneas de investigación abiertas. 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 estos 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 ante la complejidad de definir completamente el dominio. Por ejemplo, en el caso de los sistemas de seguridad, muchos de los eventos anómalos que ocurren en el entorno son imprevisibles incluso hasta para el experto del dominio. Por esta razón, resulta interesante el uso de algoritmos de aprendizaje semi-automáticos y automáticos para complementar la base de conocimiento del sistema. Así, a medida que crece la base de conocimiento, el sistema mejora en eficiencia y en porcentajes de acierto en sus decisiones.

El segundo reto es el **tratamiento adecuado de la incertidumbre y la vaguedad** que existe en cualquier escenario real. 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 realizáramos un experimento en el cual se mostrara a cinco personas un vídeo y les pidiéramos que interpretaran lo que ocurre en él, seguramente obtendríamos algunas interpretaciones comunes y otras diferentes. Por tanto, un tratamiento adecuado de la incertidumbre y vaguedad de un escenario real, 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.

3ª GENERACIÓN

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 geográficamente por todo el entorno observado, los cuales transmiten 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 solidez, 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, supongamos un vehículo en movimiento en una zona de aparcamiento, donde se encuentran instaladas varias cámaras de seguridad. Supongamos también que la primera cámara tiene dificultades para capturar el vehículo debido a la ubicación de algunos objetos ambientales 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. Supongamos 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 2.1.1 resume las principales ventajas y problemas [101] a los que se enfrentan los sistemas de videovigilancia de cada generación.

<b>1ª Generación</b>	
Descripción	Sistemas de videovigilancia analógicos de circuito cerrado CCTV.
Ventajas	<ul style="list-style-type: none"> <li>■ Buen funcionamiento del sistema con baja tasa de errores.</li> <li>■ A pesar de sus limitaciones, son sistemas ampliamente utilizados, debido en gran parte a la madurez de la tecnología que emplean.</li> </ul>

Problemas	<ul style="list-style-type: none"> <li>■ Utilizan técnicas analógicas para el tratamiento, la distribución y el almacenamiento de las imágenes.</li> <li>■ Dependen demasiado de la actividad humana para detectar las anomalías que suceden en un entorno.</li> </ul>
Investigación actual	<ul style="list-style-type: none"> <li>■ Tratamiento, distribución y almacenamiento de imágenes mediante técnicas digitales.</li> <li>■ Algoritmos de compresión de vídeo.</li> <li>■ Recuperación eficiente de datos multimedia.</li> </ul>
<b>2ª Generación</b>	
Descripción	Interpretación automática de escenas reales.
Ventajas	<ul style="list-style-type: none"> <li>■ Incremento de la eficiencia de los sistemas de seguridad. Observación y análisis de un mayor número de situaciones al mismo tiempo.</li> <li>■ Reducción de la dependencia de la actividad humana para detectar situaciones anómalas.</li> </ul>
Problemas	<ul style="list-style-type: none"> <li>■ 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).</li> <li>■ Algunas situaciones anómalas imprevisibles podrían ser no detectadas.</li> <li>■ Soluciones no demasiado robustas que dan lugar a un número elevado de falsas alarmas.</li> </ul>

Investigación actual	<ul style="list-style-type: none"> <li>■ Necesidad de diseñar algoritmos eficientes que ofrezcan respuestas en un tiempo cercano al real.</li> <li>■ Representación de los elementos físicos de un entorno real y la relación que existe entre ellos.</li> <li>■ Reconocimiento de eventos y actividades.</li> <li>■ Distinción entre situaciones normales y anormales.</li> <li>■ Algoritmos de aprendizaje que amplían el conocimiento que tiene el sistema sobre el entorno.</li> <li>■ Anticipación a posibles acciones que podrían dañar el entorno.</li> <li>■ Toma de decisiones y gestión de crisis.</li> </ul>
<b>3ª Generación</b>	
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	<ul style="list-style-type: none"> <li>■ 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.</li> <li>■ Mayor consistencia ante posibles ataques, debido a la naturaleza distribuida que posee.</li> <li>■ Carga de trabajo descentralizada.</li> </ul>
Problemas	<ul style="list-style-type: none"> <li>■ Distribución de la información.</li> <li>■ Comunicación entre dispositivos heterogéneos.</li> <li>■ Relación e interpretación de la información que procede de múltiples fuentes.</li> </ul>

Investigación actual	<ul style="list-style-type: none"> <li>■ Los temas de investigación expuestos en la segunda generación también son apropiados para la tercera generación.</li> <li>■ Cada uno de los problemas que presentan los sistemas de tercera generación da lugar a una línea de investigación.</li> </ul>
----------------------	---

Tabla 2.1: Evolución de los sistemas de vigilancia.

### 2.1.2. Principales fases de un sistema de vigilancia inteligente

Normalmente, los sistemas de vigilancia de 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 otras capas. En [19, 101, 102] se realizan tres propuestas interesantes sobre las etapas o fases de las que debería constar un sistema de vigilancia inteligente. La Figura 2.1 unifica en un único esquema las fases propuestas por diferentes autores.

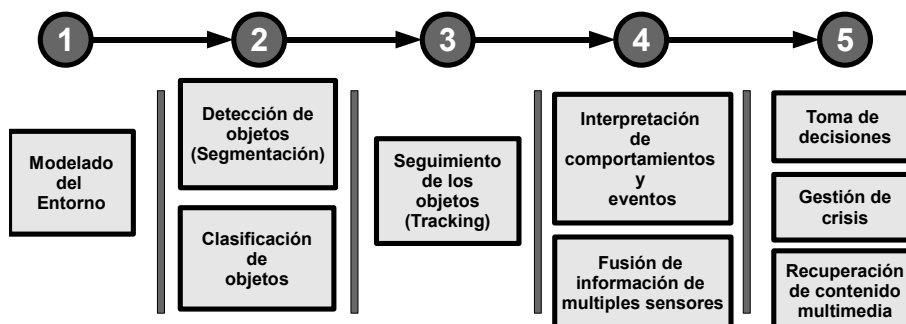


Figura 2.1: Etapas que podrían formar parte de la arquitectura de cualquier sistema de vigilancia inteligente.

MODELIZACIÓN La primera fase, o fase de modelización [102], consiste en definir los elementos que pueden participar en el entorno, así como sus propiedades y las principales relaciones que existen entre éstos y el entorno. Dicha modelización es necesaria, ya que cualquier sistema artificial es incapaz de reconocer por sí mismo los elementos que actúan en el entorno y que son capturados a través de los sensores. Según Gruber [48] y Guarino [49] las ontologías proporcionan los mecanismos necesarios para alcanzar el objetivo propuesto en esta fase de una manera formal y eficiente. Es decir, las ontologías permiten formalizar los elementos de un dominio, sus propiedades y relaciones.

SEGMENTACIÓN 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 instaladas. 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 mayoría de métodos de segmentación en vídeo utilizan información espacio-temporal y están basados principalmente en tres aproximaciones:

- Diferenciación de un elemento en movimiento con respecto a un fondo de imagen estático (**Background subtraction**). 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 del fondo lo más representativo posible para evitar el mayor número de errores. Algunos trabajos destacados en este campo son [51, 73, 93].
- Técnica de diferenciación temporal (**Temporal differencing**). Este método 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 [67] debido a sus capacidades adaptativas, sin embargo, tiene un rendimiento más pobre en la extracción de los píxeles relevantes de los objetos en movimiento.
- Flujo óptico (**Optical flow**). Este método [10, 38] extrae las regiones en movimiento a partir de las características que ofrecen los vectores de movimiento de los objetos a lo largo del tiempo. Los algoritmos de este tipo ofrecen como gran ventaja la detección de objetos incluso con cámaras en movimiento. Sin embargo, la mayoría de ellos son computacionalmente muy complejos [102] y difícilmente pueden ser aplicados directamente sobre flujo de vídeo en tiempo real. En [10] se discute con mayor profundidad los métodos de flujo óptico para la segmentación de objetos en escenas de vídeo.

CLASIFICACIÓN

Por otra parte, 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. Según Weiming et al. [102] actualmente existen dos categorías principales de 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 [41, 42, 73, 109]. 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 [29] 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 movimiento en cuatro clases: persona, vehículo, grupo de personas y, finalmente, aglomeración de formas no identificadas.

- **Clasificación basada en el movimiento.** Una alternativa al método anterior es la clasificación de objetos en función de los movimientos que éstos realicen [32, 66]. 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) 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.

Bremond [19] 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. [80] hacen uso de este tipo de características para detectar peatones en movimiento en entornos urbanos.

TRACKING

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. Weiming et al. [102] y Bremond [19] 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 [55].

- **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 [103] o derivados, los cuales son utilizados para eliminar el ruido o interferencias que se producen debido en gran parte a las condiciones de iluminación. Básicamente, esta técnica [59, 60] 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 abstraer 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* [11, 75]. 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 [58, 78, 87].

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 para tratar las oclusiones parciales y totales en escenas de vídeo, la mayoría de ellas se basan en el uso de múltiples cámaras [34, 99, 100].

INTERPRETACIÓN  
DE COMPORTA-  
MIENTOS

Continuando con la descripción de las etapas mostradas en la Figura 2.1, el siguiente paso a la clasificación y el proceso de *tracking*, es la interpretación y el reconocimiento de los comportamientos 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.

Por otra parte, el gran handicap en 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 alguien 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 esta capa es bastante inferior a las anteriores, donde las tecnologías y algoritmos empleados alcanzan un nivel mayor de madurez. Además, no sólo repercute en el número de aportaciones sino que también en el hecho de que no existe un consenso generalizado sobre los métodos y técnicas a utilizar. Así lo demuestra Weiming Hu en [102], enumerando un mayor número de categorías que en las etapas anteriores, como por ejemplo: *Dynamic time warping (DTW)*, Máquinas de estados finitos, Gramáticas libres de contexto, Modelos Ocultos de Markov, Redes Neuronales, etc. Estas categorías se verán con mayor detalle<sup>1</sup> en la sección 2.3.

<sup>1</sup>Las etapas de modelización de entornos e interpretación de comportamientos tienen su propia sección en el presente trabajo, por esta razón no se incluyen detalladamente las principales categorías y trabajos de estas etapas en la sección actual.



FUSIÓN DE IN-  
FORMACIÓN

Opcionalmente, la capa de razonamiento e interpretación de comportamientos puede incluir la posibilidad de fusionar la información [30, 92, 105] que proviene de múltiples sensores. Esta combinación de información puede proporcionar grandes beneficios como por ejemplo la eliminación de ruidos y distorsiones, tratamiento del problema de la oclusión [34], o incluso reforzar las creencias que se manejan en los procesos de esta etapa.

TOMA DE DECI-  
SIONES Y GES-  
TIÓN DE CRISIS

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 crisis. 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. Aún así, existen campos donde la automatización de la toma de decisiones está más extendida, como es el caso de las aplicaciones militares donde es necesario tomar una decisión rápidamente [57, 61], o en entornos industriales [33] donde la toma de decisiones es una tarea compleja.

RECUPERACIÓN  
DE DATOS  
MULTIMEDIA

Además, en esta última etapa podríamos encuadrar una actividad frecuentemente demandada 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 preguntas 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<sup>2</sup> que se han producido. Algunos trabajos importantes en este campo son el realizado por J. Black et al. [14] donde definen modelos de datos para soportar el almacenamiento en bases de datos de vigilancia, de diferentes niveles de abstracción de los datos obtenidos en el proceso de *tracking*. Hacid et al. [50] desarrollan un modelo de datos y un lenguaje de consulta basado en reglas para la recuperación de vídeo indexado. Stringa et al. [95] llevan a cabo previamente la clasificación de los objetos que aparecen en el vídeo y lo recuperan posteriormente en base a esta clasificación. Finalmente, cabe destacar el estándar MPEG-7 elaborado por el grupo MPEG de la Organización Internacional para la Estandarización ISO/IEC, el cual permite la representación de la información audiovisual para la descripción de contenidos de este tipo. Algunos trabajos orientados a la vigilancia en el que se utilice el estándar MPEG-7 son [7, 12, 31, 81].

<sup>2</sup>Consideramos como alarma cualquier evento no permitido que haya sido proporcionado directamente por cualquier sensor, o bien, como fruto de la interpretación realizada a partir de los datos que éste proporciona.

### 2.1.3. Aplicaciones

Con el transcurso de los años, el rango de entornos en el que se implantan sistemas de seguridad ha crecido considerablemente debido a la creciente demanda de seguridad y de soluciones más sofisticadas por parte de la sociedad. Ciudades como Londres han decidido instalar cámaras en las calles principales (decisión que ha creado cierta polémica en la población, ya que consideran que con la presencia de estas cámaras se atenta contra la intimidad), y más de 5000 de ellas aeropuertos.

Las nuevas propuestas para la mejora de los sistemas de seguridad se realizan tanto en el ámbito académico como en el comercial [101]. La principal diferencia entre unas y otras, es que las propuestas comerciales están bastante más 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 seguimiento de los mismos en la escena. Sin embargo, las propuestas que se realizan en el ámbito académico suelen ser algoritmos avanzados de gran complejidad, 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 entornos donde se instalan sistemas de seguridad comerciales y se realizan investigaciones desde el ámbito académico 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 ORION (ahora 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 [16, 98, 40].
- **Entornos marítimos.** Los sistemas de vigilancia también tienen cabida en entornos marítimos. Pozzobon et al. proponen en [83] un sistema de seguridad para la vigilancia en puertos de carga y descarga de mercancías. También Avis, P. detalla en [8] cuales son los sistemas de seguridad que utilizan en Canada para vigilar las costas.
- **Estaciones de tren.** Seguridad en estaciones de tren y vigilancia en vías de ferrocarril. Lai Lo et al. [68] 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 [79] propone diferentes medidas de seguridad que puede ser aplicadas en estaciones de transporte público, entre ellas las estaciones de ferrocarril. Ronetti et al [88], 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 [13, 17, 43, 62] 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 [46, 91].
- **Aplicaciones militares.** Los últimos avances en sistemas de vigilancia y toma de decisiones se suelen probar con frecuencia en primer lugar en aplicaciones militares [57, 61].
- Vigilancia en lugares públicos como **bancos** [70], **hogares** [21], **casinos**, **grandes almacenes** y **zonas de aparcamiento** [74, 54].

## 2.2. Modelado de entornos reales

La construcción de sistemas basados en conocimiento [27] implica la creación de un modelo de dominio particular (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. En las siguientes secciones veremos diferentes formas de representar y modelar el conocimiento de un dominio del mundo real, haciendo mayor hincapié en el modelado de relaciones espaciales y temporales necesarias para interpretar los eventos que suceden en el dominio.

*“La construcción de sistemas de conocimiento implica la creación de un modelo de dominio particular”.*  
*“Un modelo es una abstracción del dominio que estamos considerando”.*

### 2.2.1. 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 y es en esa situación cuando las ontologías cobran mayor sentido.

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 [47, 48] 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.

- 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. [24] 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 representar una ontología podemos seguir diferentes estrategias. Una de estas estrategias es el uso de metodologías que permiten construirlas de forma sistemática, como por ejemplo, las metodologías CYC [1] y METHONTOLOGY [39]. 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 [1], 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 dispone de actividades y guías para construir 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 descripción 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 para diseñar una nueva ontología. Entre los lenguajes más representativos podemos destacar por una parte Ontolingua [47] y CycL [64], y por otra parte los lenguajes más orientados a la representación de ontologías en la web semántica como RDF, DAM+OIL, OWL, etc.

*Ontolingua* [2] 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 [45]. 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 [45]. Por otro lado, CycL es un lenguaje declarativo con fuerza de expresividad (similar al cálculo de predicados de primer orden) que fue creado con el objetivo de representar y organizar grandes cantidades de conocimiento. A esto debemos añadir que se trata de un lenguaje con una sintaxis similar a la de LISP y está basado en marcos con herencia múltiple.

Entre los lenguajes diseñados para la definición de ontologías y ampliamente utilizados para la web semántica destacamos OWL porque es el más actual y porque, quizás, el resto han quedado un tanto obsoletos. OWL (Web Ontology Language), es un lenguaje de ontologías utilizado para representar el significado de los términos y las relaciones de un área de conocimiento. 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 el diseño de estos lenguajes no es demasiado compatible 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. Además es importante destacar que 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.

### 2.2.2. Lógica de primer orden: cálculo de situaciones y cálculo de eventos

La lógica de primer orden (LPO) o cálculo de predicados de primer orden es uno de los mecanismos más antiguos y más utilizados para representar y formalizar el conocimiento de un dominio del mundo real. Sin embargo, en algunas ocasiones tiene dificultades para representar aquellos elementos que son de carácter general y no son ciertos en todos los dominios, como por ejemplo, el manejo de la incertidumbre, la imprecisión y la vaguedad. En las siguientes secciones se verá brevemente como la lógica de primer orden puede utilizarse para definir categorías y objetos, acciones, situaciones, eventos, representación del tiempo, etc.

#### 2.2.2.1. Categorías y objetos

A pesar de que las interacciones en el mundo real ocurren entre objetos individuales, para representar el conocimiento es necesario organizar los objetos en categorías, ya que la mayor parte del proceso de razonamiento ocurre en este nivel [90]. En la lógica de primer orden existen dos posibles opciones para representar las categorías: predicados, como por ejemplo **Familias\_Españolas(f)**, o con objetos, por ejemplo **Familias\_Españolas** representaría el conjunto de todas las familias, en este caso para decir que una instancia pertenece a un conjunto podríamos representarlo de la siguiente forma: **Miembro (f, Familias\_Españolas)** que es equivalente a  $f \in \text{Familias\_Españolas}$  y  $\text{Familias\_Españolas} \subset \text{Familias}$ . Cuando decimos que el conjunto de las Familias\_Españolas está incluido en el conjunto más genérico de Familias estamos creando una jerarquía formada por categorías y subcategorías.

Cuando se define una jerarquía se obtiene de forma implícita herencia entre las categorías. La herencia es una forma de simplificar la representación del conocimiento. Veamos un ejemplo: “La pera es una fruta, la fruta es un alimento y los alimentos son comestibles” es lo mismo que decir que  $\text{Pera} \subset \text{Fruta} \subset \text{Alimentos}$ . Podemos deducir que la pera es comestible porque es una fruta y la fruta es un alimento comestible. Existen diferentes tipos de relaciones que se pueden dar entre las categorías y que dan lugar a la aparición de taxonomías: disjuntas, descomposición exhaustiva y partición.

- Dos categorías son disjuntas cuando no tienen ningún elemento en común. Por ejemplo, las categorías Hombre y Mujer. Una persona que sea hombre no puede ser mujer. Si lo expresamos de manera formal obtendríamos lo siguiente:

$$\text{Disjunto(s)} \Leftrightarrow \forall c_1, c_2, c_1 \in S \wedge c_2 \in S \wedge c_1 \neq c_2 \rightarrow \text{Interseccion}(c_1, c_2) = \{\}$$

- Con la *descomposición exhaustiva* limitamos la pertenencia de un individuo a un rango de clases. Por ejemplo, *DescomposicionExhaustiva*({Hombre, Mujer}, Personas) indicamos que las personas sólo pueden ser hombres o mujeres. Su representación formal sería la siguiente:

**DescomposicionExhaustiva(s,c)**  $\Leftrightarrow (\forall i, i \in S \Leftrightarrow \exists c_2, c_2 \in S \wedge i \in S)$

- Una partición es una descomposición exhaustiva disjunta. El ejemplo anterior sería una partición. Un ejemplo de no partición sería DescomposicióExhaustiva({Americanos, Canadienses, Mejicanos}, Norteamericanos) con la que indicamos que los norteamericanos sólo pueden ser Americanos, Canadienses o Mejicanos pero no es considerado partición ya que alguien puede tener doble nacionalidad. Su representación formal correspondería con: **Particion(s,c)**  $\Leftrightarrow$  **Disjunto(s)**  $\wedge$  **DescomposicionExhaustiva(s,c)**

### 2.2.2.2. Acciones, situaciones y eventos

Según Russel et al. [90] una de las desventajas de la lógica proposicional es “la necesidad de tener diferentes copias de la descripción de la acción para cada intervalo de tiempo en el cual la acción se podría llevar a cabo”. La solución que propone es el uso de **situaciones** en lugar de intervalos de tiempo concretos (t, t + 1, ...); donde una situación denota el estado resultante tras la ejecución de una acción. Esta aproximación se denomina **cálculo de situaciones** y maneja los siguientes conceptos:

- **Situaciones.** Una situación es un término lógico que expresa el estado de los diferentes elementos que componen una escena. Normalmente existe una situación inicial ( $S_0$ ) y una función **Resultado (a,s)** que ofrece la situación resultante después de realizar la acción **a** en la situación **s**.
- **Flujos.** Cuando hablamos de flujos nos referimos a funciones y predicados que varían de una situación a la siguiente. Un ejemplo de flujo podría ser: *Edad (Juan,  $S_0$ )*. La edad es una función cuya respuesta depende de la situación.
- Por contra, también podemos hablar de **funciones o predicados atemporales o eternos**, es decir, aquellos que permanecen invariables en el tiempo. Persona(Juan), Juan siempre será una persona durante su existencia, independientemente de la situación en la que se encuentre.

Podemos clasificar las acciones en dos categorías: *acciones simples y acciones compuestas* (secuencia de ejecución de acciones simples). En una secuencia las acciones que se encuentran en posiciones más adelantadas necesitarán los resultados obtenidos por las acciones anteriores, es decir, la acción **a + 1** necesitará el resultado de ejecutar la acción **a** en la situación **s**. Podemos utilizar la recursividad para el cálculo de acciones compuestas; para ello lo primero que necesitamos es un caso base. En este caso podría valer que el resultado de ejecutar una secuencia vacía de acciones sobre una situación ofrece como resultado la misma situación (no se realiza ninguna acción y por tanto la situación no varía).

- Caso base:

**Resultado ([], s) = s**

- Caso general:

**Resultado([a]seq, s) = Resultado (seq, Resultado (a,s))**

Ejecutar una secuencia no vacía implica ejecutar la primera acción y posteriormente ejecutar el resto sobre el resultado obtenido con la primera ejecución.

Cualquier agente inteligente diseñado para el cálculo de situaciones debe ser siempre capaz de obtener un resultado tras la realización de una secuencia de acciones en un escenario concreto (**Proyección**) y debe ser capaz de encontrar la secuencia apropiada para alcanzar los objetivos deseados (**Planificación**), es decir, el agente debe tener capacidad de proyección y planificación para la resolución de situaciones.

En el cálculo de situaciones podemos representar las acciones con el uso de dos axiomas: axioma de posibilidad y axioma de efecto. Por cada una de las acciones se analiza su viabilidad, es decir, si se puede realizar la acción teniendo en cuenta la situación actual y en caso afirmativo se determinan cuáles son los efectos que produce la ejecución de dicha acción. Los axiomas tienen la siguiente forma:

- AXIOMA DE POSIBILIDAD: **Precondiciones**  $\rightarrow$  **Posible(a,s)**.
- AXIOMA DE EFECTO: **Posible(a,s)**  $\rightarrow$  **Cambios que son el resultado de ejecutar la acción.**

### 2.2.2.3. Representación del tiempo y cálculo de eventos

La utilización del cálculo de situaciones es adecuado cuando las acciones que realizamos son acciones discretas e instantáneas. Por el contrario, cuando las acciones no se realizan en un instante de tiempo concreto, sino que necesitan de un intervalo de mayor duración, el cálculo de situaciones se convierte en una tarea tediosa. Ante esta situación existe una alternativa más eficaz denominada cálculo de eventos, que está basado en puntos en el tiempo en lugar de situaciones. Entendemos por evento, un conjunto amplio de acciones que se ejecutan en un intervalo de tiempo. Según Russell et al. [90] “El axioma del cálculo de eventos dice que un flujo es cierto en un punto concreto en el tiempo, si el flujo fue iniciado por un evento en un instante de tiempo anterior y no fue finalizado por la intervención de algún otro evento”.

Existen dos tipos de relaciones que determinan el valor de verdad de un flujo en el tiempo, estas relaciones son *Inicio* y *Terminación*; ambas relaciones trabajan de forma similar a la función *Resultado* en el cálculo de situaciones. *Inicio* ( $e, f, t$ ), indica que la ocurrencia del evento  $e$  en el tiempo  $t$  causa que el flujo  $f$  sea cierto. *Terminación*( $e, f, t$ ) significa que  $f$  deja de ser cierto. Russell también utiliza otras dos relaciones para expresar cuando un evento está ocurriendo y la interrupción de un evento realizada por otro diferente: *Sucedo* ( $e,t$ ), *Interrumpido* ( $f, t, t2$ ).

Formalmente, el axioma del cálculo de eventos queda expresado de la siguiente manera:

- $T(f, t2) \Leftrightarrow \exists e, t \text{ Sucede}(e,t) \wedge \text{Inicio}(e, f, t) \wedge (t < t2) \wedge \neg \text{Interrumpido}(f, t, t2) \Leftrightarrow \exists e, t1 \text{ Sucede}(e,t1) \wedge \text{Terminacion}(e, f, t1) \wedge (t < t1) \wedge (t1 < t2)$

Con el cálculo de eventos podemos expresar acciones que ocurren en un punto en el tiempo o bien en un intervalo. Para expresar que una alarma se activará exactamente a las 22 horas podríamos afirmar *Sucedo(Activar(Alarma), 22:00)*.



### 2.2.3. Relaciones espaciales

Las representación de los elementos de un entorno, junto con sus propiedades físicas y las relaciones espaciales que existen entre éstos y el propio entorno, son fundamentales para que un sistema artificial que interactúa con una fragmento 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. en su diseño de una ontología general para el razonamiento espacial [27] proponen una serie de propiedades que debe tener cualquier objeto en una ontología espacial. Estas propiedades son las siguientes.

- *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.

Por otro lado, existen dos familias principales de relaciones espaciales [94], las basadas en los modelos de intersección creados por Egenhofer et al., y los esquemas basados en el cálculo de regiones conectadas desarrollados por Cohn et al. 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 [37] 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 [37] 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, de las cuales, destacamos las realizadas por Clementini y Di Felice en [26] y Zhan en [108] 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 [28]. Los esquemas RCC también han sido extendidos a sistemas RCC15 y RCC23 [28], los cuales tienen en cuenta las envolturas convexas de las regiones implicadas. Düntsch et al. muestran en [36] 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 posibles relaciones que se pueden dar 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.

Finalmente, RCC8 al igual que RCC5 es un esquema basado en RCC. A diferencia del anterior, en este caso, DC, EC, PO, EQ, TPP, NTPP, TPPi, NTPPi forman el conjunto de relaciones base. La Figura 2.2 ilustra gráficamente estas relaciones.

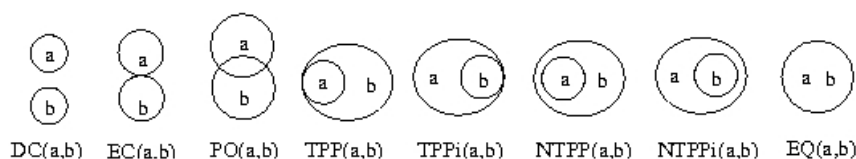


Figura 2.2: Relaciones base del esquema RCC8

### 2.2.4. Relaciones temporales: álgebra de Allen

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 es un problema de interés para múltiples disciplinas.

James F. Allen (Universidad de Rochester) propone 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 Inteligencia Artificial. El trabajo de 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. [6] 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 [6] 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.

En [5] 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. 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 [5]. 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) [5]. La Figura 2.3 muestra un ejemplo de un grafo en el que se representan tres eventos simples y las restricciones que existen entre éstos. La Figura 2.4 muestra un ejemplo de cómo varía el grafo anterior cuando se añaden nuevos eventos y nuevas restricciones.

Finalmente, el álgebra de intervalos de Allen está 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* [6]. 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)  $\forall i, j, k, l. Meets(i, j) \wedge Meets(j, k) \wedge Meets(k, l) \supset \exists m. Meets(i, m) \wedge Meets(m, l)$

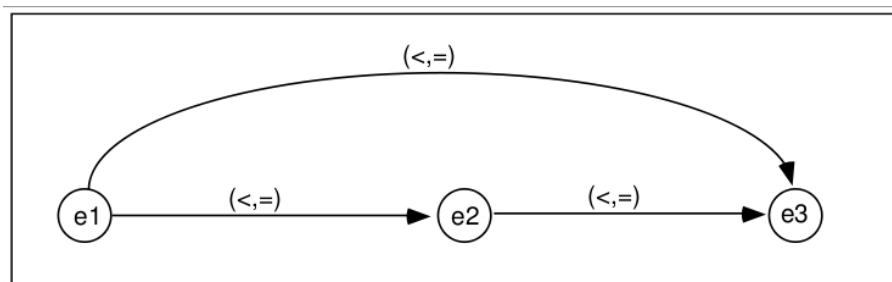


Figura 2.3: Ejemplo de la representación de eventos simples y sus relaciones mediante un grafo.

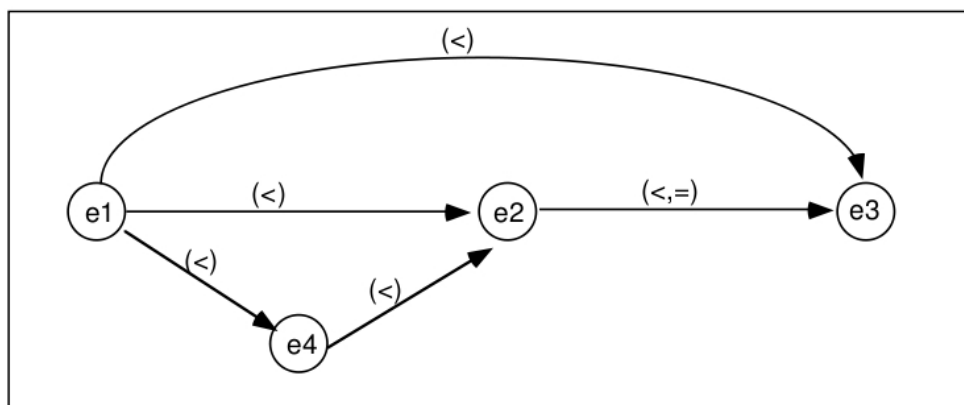


Figura 2.4: Ejemplo de propagación de restricciones tras añadir  $e_1 < e_4$  y  $e_4 < e_2$

- (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 2.5). Precisamente este rango de relaciones temporales es uno de los culpables 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 y cómo se deben producir los eventos en un entorno vigilado y que relaciones debe guardar con el resto para ser considerados normales.

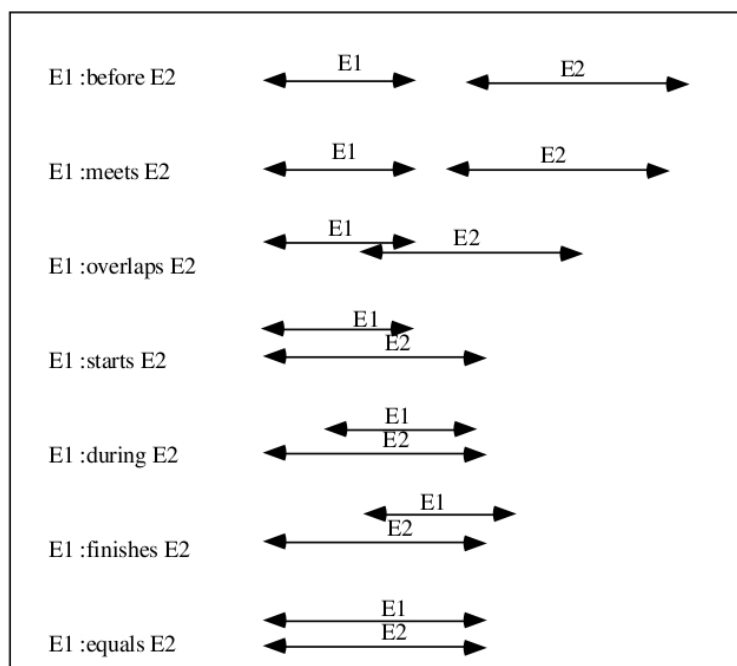


Figura 2.5: Relaciones temporales entre periodos de tiempo definidas en el álgebra de Allen.

## 2.3. Análisis e interpretación de comportamientos

En esta sección se presentan brevemente algunas técnicas<sup>3</sup> para el análisis y la interpretación de comportamientos en entornos vigilados. En esta capa no existe un consenso claro sobre las técnicas a emplear para el análisis e interpretación de comportamientos, como bien se puede apreciar en el número de técnicas descritas en esta sección. Sin embargo, existe un mayor acuerdo en el hecho de representar y analizar tanto el comportamiento de personas como de vehículos, que son los dos tipos de objetos móviles elegidos por la mayoría de autores. Otro aspecto en común, es el estudio y análisis de las trayectorias que siguen estos dos tipos de objetos ya que se adaptan bien a la mayoría de técnicas propuestas cuyo objetivo principal en el contexto de los sistemas de seguridad es el estudio de las relaciones espacio-temporales.

### 2.3.1. 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 Weiming et al. [102], la potente simplicidad concep-

<sup>3</sup>La lógica de primer orden es una de las técnicas más antiguas para representar el conocimiento de un dominio. Esta técnica también se puede utilizar para analizar e interpretar los comportamientos y, debido a que ha sido presentada en la sección 2.2.2, no se incluirá en esta sección.

tual 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. [97] que utiliza DTW para reconocer acciones realizadas con las manos por personas que son diestras, y el trabajo de Bobick et al. [15] que analiza una secuencia determinista de estados para reconocer gestos faciales.

### 2.3.2. 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. [84] utilizan una máquina de estados finitos para controlar las trayectorias de un robot en un entorno observado por una cámara. Ésta máquina de estados incorpora tres posibles comportamientos: evitar obstáculos, deambular y, finalmente, aproximarse al objetivo. Wilson et al. [104] 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). Finalmente, Bremond et al. [20] emplean autómatas finitos para reconocer comportamientos de vehículos, capturados desde cámaras aéreas.

### 2.3.3. Gramáticas libres de contexto

Una gramática libre de contexto se define como una cuadrupla  $G = (N, T, P, S)$  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 es el de Ivanov et al. [54], 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. [25] también proponen el uso de gramáticas libres de contexto para la interpretación de comportamientos en sistemas de vídeo vigilancia.

### 2.3.4. 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*. En realidad se puede considerar un HMM como una red bayesiana más simple.

El uso de los HMM para el reconocimiento de movimientos se basa principalmente en dos etapas [102]: (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.

En [76] se propone la construcción de un sistema de vigilancia automático mediante el uso de modelos ocultos de Markov. Este sistema dispone de un amplio repertorio de conductas normales para personas y estudia posibles desviaciones para detectar acciones sospechosas. Brand et al. [18] utilizan una variante de los modelos ocultos de Markov para reconocer acciones complejas en entornos vigilados. Otros trabajos destacados para el reconocimiento de comportamientos con HMM en el contexto de la vídeo vigilancia son [22, 77, 85].

Los HMM, a pesar de ser una de las técnicas más utilizadas de las presentadas en esta sección, tienen el gran inconveniente de que necesitan un largo proceso de entrenamiento para que funcionen correctamente en entornos complejos con un número elevado de variables. Además, otro gran inconveniente es la llamada *asunción Markoviana*, que consiste en que la probabilidad de transición de un determinado estado depende del estado actual y no tiene en cuenta los anteriores.

### 2.3.5. Redes Neuronales

Una red neuronal es un modelo matemático inspirado en el sistema nervioso de los seres humanos. Cada red está 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).

Normalmente, las redes neuronales en el contexto de los sistemas de seguridad ofrecen un mecanismo para aprender y clasificar los comportamientos de los objetos de la escena. Jhonson et al. [56] utilizan redes neuronales para aprender las trayectorias de los objetos y el reconocimiento de eventos. 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. [96] proponen el uso de redes neuronales para aprender relaciones espacio-temporales y utilizarlas posteriormente para la predicción de los comportamientos de los objetos.

### 2.3.6. 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 [106] 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 destaca la habilidad del cerebro humano para interpretar y resolver situaciones complicadas sin la necesidad de manejar valores numéricos [107]. 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.

Un trabajo destacado en este área es el de Petry et al. [86], en el cual se estudian las relaciones espaciales entre objetos y, entre objetos y regiones observadas por cámaras de vídeo. 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. Otros trabajos destacados con

modelos difusos son [69] donde se gestiona la información procedente de varios sensores, y [52] donde se combinan modelos difusos con redes neuronales para la detección de incidentes en escenas de tráfico.

## **2.4. Principales problemas de los sistemas de vigilancia de 3ª generación**

Como se ha podido apreciar en la Sección 2.1.2, los sistemas de vigilancia de última generación se enfrentan a multitud de dificultades y problemas en cada una de las etapas. Debido a que actualmente se considera que muchos de estos problemas no han sido resueltos, los sistemas de vigilancia de tercera generación no han alcanzado el nivel de madurez suficiente para consolidarse en el mercado. De hecho, muchas de las instituciones que deciden implantar un sistema de seguridad optan por soluciones más básicas pero que aseguran ciertas garantías de éxito. En esta sección nos centraremos en los principales problemas que se presentan en las capas de modelado de entornos reales e interpretación de escenas (análisis de eventos y conductas), por su relación directa con la temática principal del trabajo actual.

El primer problema surge de la necesidad de **ofrecer respuestas en un tiempo cercano al real**, es decir, un sistema de vigilancia no puede tardar demasiado tiempo en activar las alarmas desde que ocurrieron los sucesos anómalos (lo ideal es que el sistema tuviera capacidad para prevenir estos sucesos). Este problema no sólo afecta a la capa de interpretación de escenas, sino que también al resto. Recordemos que las capas estaban interrelacionadas y que cada una de ellas necesitaba el resultado del trabajo llevado a cabo por la capa subyacente, por tanto, el tiempo global será la suma de todos los tiempos invertidos en cada una de las capas. Muchos de los algoritmos propuestos en la literatura ofrecen grandes resultados en el proceso de interpretación de escenas reales, sin embargo, invierten demasiado tiempo en la obtención de los resultados. Por otro lado, existen algoritmos que ofrecen peores resultados pero, en cambio, ofrecen sus respuestas en menor tiempo. Por tanto, a partir de este problema surge la necesidad de buscar un equilibrio entre fiabilidad y tiempo de respuesta.

Un segundo problema es el hecho de que gran parte de las personas encargadas de diseñar un sistema de seguridad optan por **representar situaciones anormales muy concretas**. El problema de esta aproximación es que muchas de las situaciones anormales que pueden suceder en un entorno son imprevisibles, por tanto, cuando sucede una situación anómala que no ha sido definida a priori el sistema es incapaz de responder ante ésta. En la Sección 3.1 se describen diferentes alternativas y se propone una solución a este problema.

Otra de las grandes dificultades a las que se enfrentan estos tipos de sistemas es la **construcción de las bases de conocimiento** donde se modela el entorno observado. Normalmente 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 ampliar y modificar sobre la base de conocimiento. Por tanto, para poder eliminar esta dependencia es necesario dotar al sistema de capacidad para ampliar y modificar su base de conocimiento, es decir, reemplazar en la medida de lo posible al experto humano. En este contexto, los algoritmos de aprendizaje automáticos y semi-automáticos son una alternativa interesante para

solucionar este problema.

El cuarto problema guarda relación con la **escalabilidad** de los sistemas propuestos para analizar los comportamientos en los entornos observados. La mayoría de estos sistemas se centran en el estudio de un concepto (velocidad, trayectorias, localizaciones, etc) de una forma muy específica, de tal forma que dificulta la adición de un nuevo concepto al análisis. Además, este análisis suele ser el mismo para todas las clases de objetos y no varía en el tiempo. Por esta razón, sería interesante proponer arquitecturas flexibles que faciliten la inclusión de nuevos conceptos y permitan establecer diferentes configuraciones para el análisis acorde al tipo de objeto. Por ejemplo, para los vehículos en un parking se analizarían las velocidades y trayectorias, sin embargo, para las personas en una habitación interior quizás resulte más interesante estudiar únicamente las trayectorias. Este tipo de arquitecturas también ofrecerían una solución al primer problema, es decir, dependiendo de las condiciones y necesidades de cada lugar se puede optar por una configuración menos pesada para obtener resultados con mayor rapidez (cuanto menor sea el número de conceptos analizados menor será el tiempo necesario para obtener los resultados).

Finalmente, el último problema se ha comentado en parte en la sección 2.1.2 y está relacionado con la **fusión de información procedente de multitud de sensores**. Uno de los aspectos más interesantes en la fusión de información es la posibilidad de indicar al sistema artificial que partes de la información que procede de varios sensores se refieren a un mismo objeto. De esta forma se podría reforzar el proceso de razonamiento, eliminar parte del ruido procedente de las capas inferiores y eliminar problemas como la oclusión.

En el siguiente capítulo se describe el trabajo de investigación, el cual pretende ofrecer soluciones a algunos de los problemas expuestos en esta sección, en concreto, para el primero, segundo y cuarto problema. Estos problemas y los restantes también serán tratados en el desarrollo de una futura Tesis Doctoral como se indica en las líneas de trabajo futuro del capítulo 4.

---

# 3

## **Representación y análisis de la normalidad en entornos vigilados**

---

### **3.1. Análisis de normalidad vs anormalidad**

Como comentamos en las secciones anteriores, uno de los principales retos que existen actualmente en la comunidad científica es la interpretación de eventos y conductas que se producen en un escenario observado por múltiples sensores. En el caso concreto de los sistemas de vigilancia de tercera generación no es suficiente dicha interpretación, y es necesario dar un paso más allá con la diferenciación entre situaciones que son normales y aquellas que pueden implicar cierto peligro para el entorno. Por tanto, una de las decisiones más importantes en el diseño de un sistema de seguridad es cómo modelar las diferentes situaciones para poder realizar esta diferenciación. A continuación se describen tres posibles enfoques con sus principales ventajas e inconvenientes..

1. Definición de situaciones anormales con la colaboración de un experto.
2. Definición de situaciones normales con la colaboración de un experto.
3. Definición de situaciones normales y las anormales más comunes con la colaboración de un experto.

Generalmente, la mayoría suele optar por el primero de los tres enfoques anteriores, es decir, definir y tratar de identificar situaciones anómalas muy concretas. Este enfoque es apropiado en dominios que no son demasiado complejos y el rango de situaciones anómalas que podrían suceder es muy acotado. Sin embargo, en dominios más complejos existen multitud de situaciones anómalas que son imprevisibles incluso

para un experto en dicho dominio, y que el sistema sería incapaz de identificar ya que no están previamente definidas. Este es el principal inconveniente de este enfoque, la imposibilidad de responder ante situaciones que no han sido definidas en la base de conocimiento del sistema. Este hecho es aún más crítico en los sistemas de seguridad cuyo objetivo principal es la detección de situaciones anómalas para la activación de alarmas.

Por otro lado, el segundo enfoque podría ofrecer una solución al problema anterior, ya que la normalidad de un entorno, a diferencia de la anormalidad, es bien conocida. Así, si sucede alguna situación que no se corresponde con algunas de las situaciones normales definidas, el sistema podría cambiar a un estado de alerta. Esta aproximación es adecuada para detectar cuando se ha producido una situación que no es normal, sin embargo, no es capaz de detectar cuál es la situación anómala concreta que se ha producido y, por tanto, no puede darle un tratamiento específico (activación de alarmas específicas, toma de decisiones y planes de emergencia acordes a la situación que se ha producido, etc.).

Finalmente, el tercer enfoque es el más adecuado ya que ofrece soluciones a los problemas planteados por los dos anteriores. Éste se basa principalmente en la definición de la normalidad de un entorno para detectar todas aquellas situaciones que no son normales, y en la definición de las situaciones anormales más comunes para poder darle un tratamiento específico. De esta forma las posibles salidas que puede ofrecer el módulo de interpretación son las siguientes:

- Situación normal reconocida.
- Situación no normal y reconocida como una de las anormales definidas previamente.
- Situación no normal no reconocida.

La peor situación que podría ocurrir desde este enfoque es la tercera que aparece en la lista anterior, es decir, que ocurra una situación anómala que no haya sido definida previamente, pero al menos el sistema detectará que ésta no es normal y, por tanto, podrá permanecer en un estado de alerta.

**Enfoque:** *Modelado de la normalidad de un entorno y definición de situaciones anómalas más comunes para la detección de anomalías*

## 3.2. Primera aproximación: Intelligent Traffic Control

Como primera aproximación se decidió desarrollar un sencillo sistema<sup>1</sup> multi-agente que simula escenas de tráfico en un entorno urbano y realiza el análisis de las situaciones que ocurren en cada una de las simulaciones. Cada tipo de agente desempeña

<sup>1</sup>A partir de este trabajo se elaboró un artículo que actualmente se encuentra en estado de revisión. D. Vallejo, J. Albusac, L. Jimenez-Linares, C. Gonzalez-Morcillo. *A cognitive Surveillance System for Detecting Incorrect Traffic Behaviors*. Applied Artificial Intelligence International Journal.

una función y se comunican entre sí mediante el *middleware*<sup>2</sup> ZeroC ICE. La Figura 3.1 muestra los tipos de agentes que cooperan en el sistema y el flujo de trabajo que existe entre éstos.

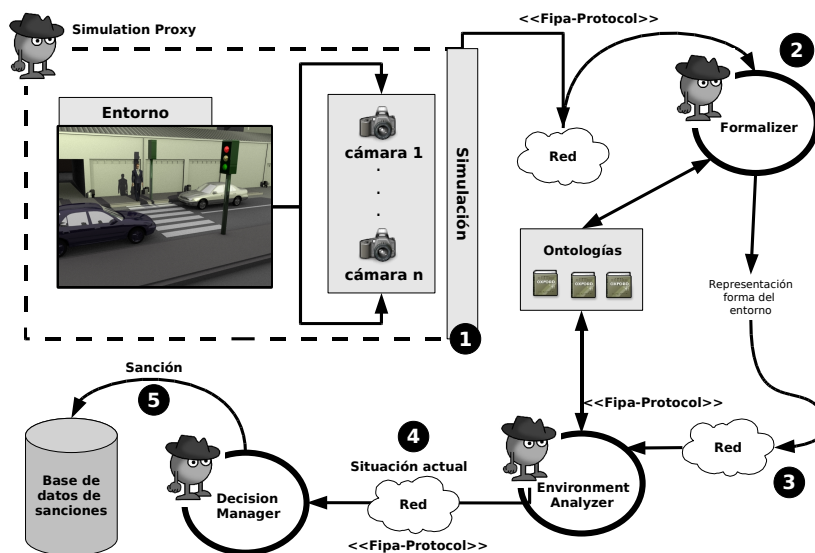


Figura 3.1: Comunicación entre los agentes inteligentes que forman parte del sistema de análisis de situaciones de tráfico en un entorno urbano

A continuación se enumeran los tipos de agentes y la principal función de cada uno de ellos.

- **SimulationProxy**. La misión principal de un agente del tipo *SimulationProxy* es la creación de una nueva simulación, siempre solicitada por un usuario desde la interfaz gráfica. En cada simulación, el agente establece una nueva configuración de vehículos y peatones en un entorno urbano (número de objetos, posición, velocidad, dimensiones, etc.), así como el estado de los elementos que regulan el tráfico (por ejemplo, los semáforos).
- **Formalizer**. Los agentes del tipo *Formalizer* se encargan de identificar los distintos elementos que conforman la escena a partir de las datos simulados por un agente del tipo *SimulationProxy*. Estos agentes hacen uso del motor de inferencia de Prolog y de la ontología definida para identificar dichos elementos.
- **Analyzer**. Los agentes del tipo **Analyzer** tienen como misión razonar para discernir sobre la normalidad o la anormalidad de la situación. Al igual que los agentes del tipo *Formalizer*, este tipo de agentes razonan a partir del motor de inferencia de Prolog y de la ontología previamente definida.

<sup>2</sup>Se puede entender un *middleware* como un software de conectividad que hace posible que aplicaciones distribuidas se puedan ejecutar sobre distintas plataformas heterogéneas

- **DecisionManager.** Por último, los agentes del tipo *DecisionManager* serán los encargados de tomar las acciones pertinentes en caso de que se haya detectado un comportamiento incorrecto por parte de algún conductor

Para esta primera aproximación, el entorno se modeló mediante la lógica de primer orden y el proceso de razonamiento se llevó a cabo haciendo uso del motor de inferencia del lenguaje declarativo Prolog, que tiene su equivalencia directa con la lógica de primer orden. La modelización incluye la definición de los elementos móviles (a los cuales el sistema debe prestar especial atención) a partir de un conjunto de propiedades y las normas que determinan la normalidad del entorno. Las siguientes líneas de código en lenguaje Prolog muestran algunos ejemplos de definición de elementos en función de propiedades como el tamaño o el número de ruedas.

#### Elementos de dominio

```

1  car(V):- height(V,medium), wheels(V,medium).
2  motorbike(V):- height(V,small), wheels(V,few).
3  truck(V):- height(V,tall), wheels(V,many).
4  vehicle(V):- car(V); motorbike(V); truck(V).

```

Algunos ejemplos de definiciones de normalidad teniendo en cuenta grupos de personas (G), un paso de peatones (pc) y el estado actual de un semáforo (green, amber, red), podrían ser los siguientes:

#### Ejemplos de definiciones de normalidad

```

1  normal(V,red):- semaphore(red), vehicle(V), /+ (is_on(V,pc)).
2  normal(V,green):- vehicle(V), (semaphore(green);semaphore(amber)).
3  normal(V,amber):- normal(V,green).
4  normal(V,off):- vehicle(V), semaphore(off).
5  normal(V,G,green):- group(G), normal(V,green), /+ is_on(G,pc),
6                      is_on(V,pc)).
7  normal(V,G,amber):- normal(V,G,green).
8  normal(V,G,off):- group(G), normal(V,off), /+ ((near_of(G,pc);
9                      is_on(G,pc)), is_on(V,pc)).

```

Para finalizar con los ejemplos, mostramos el proceso de transformación de una implicación lógica en una sentencia del Lenguaje Prolog.

1.  $\forall V[\text{height}(V, \text{medium}) \wedge \text{wheels}(V, \text{medium}) \rightarrow \text{car}(V)]$
2.  $\forall V[\text{car}(V) \vee \neg[\text{height}(V, \text{medium}) \wedge \text{wheels}(V, \text{medium})]]$
3.  $\forall V[\text{car}(V) \vee \neg\text{height}(V, \text{medium}) \vee \neg\text{wheels}(V, \text{medium})]$
4.  $\text{car}(V) : \neg\text{height}(V, \text{medium}), \text{wheels}(V, \text{medium})$

El diseño e implementación de este sistema ha servido como punto de partida y toma de contacto con el problema de la interpretación y el análisis de escenas. Sin embargo, el sistema como tal no ofrece mejoras con respecto a los sistemas existentes ya que presentan importantes carencias; algunas de éstas son las siguientes:

- El sistema establece una configuración de elementos en cada simulación sin tener en cuenta la incertidumbre, es decir, la clasificación y posición de los objetos es siempre correcta. Sin embargo, la realidad es totalmente distinta, ya que cualquier sistema que trabaje con la información procedente de múltiples sensores, maneja un alto grado de incertidumbre y vaguedad, que le impedirá afirmar con total certeza los hechos que se producen.
- El sistema analiza hechos que ocurren en instantes concretos, sin tener en cuenta las relaciones temporales que puedan existir entre éstos. Por tanto, el sistema no es capaz de determinar comportamientos complejos.
- La lógica de primer orden es un mecanismo que nos permite formalizar el conocimiento pero, quizás, no sea el más apropiado para representar la incertidumbre y las relaciones temporales.
- El sistema actual no es fácilmente escalable, es decir, no se adapta a todos los entornos y diferentes tipos de análisis.

Este trabajo dio paso a un segundo (Sección 3.3) en el que se intentan solventar las carencias presentadas anteriormente. Para ello, se ha definido un modelo de formalización genérico para representar la normalidad de un entorno. Además, a partir de dicho modelo se realizó el diseño de una arquitectura conceptual que es fácilmente escalable mediante los llamados “*componentes de normalidad*” (ver Sección 3.5).

### 3.3. Modelo para la representación y formalización de la normalidad

A pesar de que, como se comentó en la Sección 3.1, la aproximación más adecuada es la definición de normalidad y las situaciones anómalas más comunes, el presente trabajo se centra principalmente en la definición de normalidad. La representación de las situaciones anómalas más comunes y la combinación de su análisis con el de normalidad, forma parte de una de las líneas de trabajo futuro. A continuación, se presenta un modelo<sup>3</sup> formal para representar la normalidad en un entorno real.

**Definición 1.** Se define el problema de la vigilancia en un entorno observado como la interpretación y el análisis de las percepciones que se producen en el entorno. Dichas percepciones las proporcionan los sensores ubicados en el entorno vigilado.

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

Un conjunto de estas percepciones representa a su vez un entorno observado  $E_i$ , es decir, como es visto una porción del entorno global  $E$  por un conjunto de sensores. Cada  $E_i$  se define de la siguiente forma:

**Definición 2.** Un entorno observado  $E_i$  es definido como una 4-tupla formada por los siguientes elementos.

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

<sup>3</sup>A partir de este trabajo se elaboró un artículo titulado *Intelligent Surveillance based on Normality Analysis to Detect Abnormal Behaviors*, para el special issue *Video Analysis and Understanding for Surveillance Applications* de la revista *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*. Actualmente el artículo se encuentra en estado de revisión.



Donde

- $V$  representa el conjunto de variables de entrada utilizadas para analizar la normalidad. Los valores de estas variables podrían corresponder directamente con la salida de los sensores, o bien, podría ser necesario construirla en base a éstos (por ejemplo, variables difusas).
- $O$  es el conjunto de clases de objetos cuyo comportamiento es analizado para la interpretación de la escena (por ejemplo, peatones; vehículos: coche, moto, camión; animales).
- $C$  representa el conjunto de conceptos a partir de los cuales se realiza el análisis de normalidad (por ejemplo, circulaciones normales, velocidades normales, accesos permitidos a áreas restringidas, etc.)
- $O \times C$  indica qué conceptos del conjunto  $C$  deben ser utilizados para analizar la normalidad de un tipo de objeto perteneciente a  $O$ . Por ejemplo, para determinar si un objeto de la clase *vehículo* se está comportando de forma normal, se podría analizar tanto la trayectoria como la velocidad.

Una vez que el entorno, los objetos y los conceptos han sido definidos es necesario definir la normalidad de un concepto.

**Definición 3.** La normalidad de un concepto  $c_i$ , denotado como  $N_{c_i}$ , se construye en base a tres elementos:

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

Siendo  $V_i$  el conjunto de variables de entrada necesarias para analizar la normalidad del concepto  $c_i$ , verificándose que  $V_i \subseteq V$ . Además,  $DDV_i$  es el conjunto de dominios de definición para cada variable  $v_j$  perteneciente a  $V_i$ . Por tanto, si  $V_i = \{v_{1i}, v_{2i}, \dots, v_{ni}\}$ ,  $DDV_i$  será definido como  $DDV_i = \{DDV_{1i}, DDV_{2i}, \dots, DDV_{ni}\}$ . Finalmente,  $\Phi_i$  es el conjunto de restricciones utilizadas para analizar la normalidad de el concepto  $c_i$ , definidas a partir del conjunto de variables  $V_i$  ( $\Phi_i = \{f_{1i}, f_{2i}, \dots, f_{ki}\}$ ). Cada  $f_{ji}$  de  $\Phi_i$  se define de la siguiente forma.

**Definición 4.** Una restricción de normalidad  $f_{ij}$  se define como una función que toma como entrada un subconjunto de  $V_i$  y devuelve un valor perteneciente al intervalo  $[0, 1]$ , donde 1 representa el máximo valor de satisfacción de la restricción y 0 lo contrario. En realidad, las restricciones representan un conjunto de condiciones que los objetos deben satisfacer para considerar que su comportamiento es normal.

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

El procedimiento para definir la normalidad de un concepto  $c_i$  en un entorno siempre es el mismo. Sin embargo, la propia definición puede variar de un entorno a otro en función de los requisitos, y por este motivo es posible realizar particularizaciones de la definición de normalidad de un concepto de la siguiente forma.

**Definición 5.** La normalidad de un concepto  $c_i$  en un entorno  $E$ , denotado como  $E_{c_i}$ , se define como:

$$E_{c_i} = \langle N_{c_i}; \chi_i \rangle \quad (3.5)$$

donde  $\chi_i$  es el conjunto de restricciones que representa el análisis de normalidad en un entorno  $E$  ( $\chi_i = \bigcup_{y=1}^k r_{iy}$ ), y se definen a partir de instancias concretas de los elementos de  $\Phi_i$ , es decir:

$$r_{ij} = \bigcap_{x=1}^{|\Phi_i|} \tilde{f}_{ix} \quad (3.6)$$

Cada  $r_{ij}$  representa una particularización de los elementos de  $\Phi_i$ . El valor de satisfacción de cada restricción  $r_{ij}$  (denotado como  $SD_{r_{ij}}$ ) se calcula como el mínimo valor ofrecido por la Ecuación 3.6.

$$SD_{r_{ij}} = \min_{x=1}^{|\Phi_i|} \{\tilde{f}_{ix}\} \quad (3.7)$$

**Definición 6.** Una instancia de una restricción de normalidad  $f_{ij}$ , denotada como  $\tilde{f}_{ij}$ , establece las condiciones que deben satisfacer las variables especificadas en  $\mathcal{P}(V_i)$ , y que son empleadas para definir  $f_{ij}$ . El grado de satisfacción siempre pertenece a el intervalo  $[0, 1]$ :

$$\tilde{f}_{ij} : \mathcal{P}(DDV_i) \longrightarrow [0, 1] \quad (3.8)$$

Tal que si  $v_{ik} \in \mathcal{P}(V_i)$  define  $f_{ij}$ , entonces  $DDV_{ik} \in \mathcal{P}(DDV_i)$  se utiliza para instanciar  $\tilde{f}_{ij}$ .

**Definición 7.** La normalidad global en un entorno  $E$  de acuerdo a un conjunto de conceptos  $C$ , denotado como  $E_c$ , se define a partir de los análisis de normalidad para cada concepto en el entorno.

$$E_c = \bigcup_{i=0}^{|C|} E_{c_i} \quad (3.9)$$

El nivel de detalle en el análisis de normalidad de un entorno  $E$  ( $E = \langle V; O; C; O \times C \rangle$ ) se puede ampliar dando lugar a un nuevo entorno  $E'$  ( $E' = \langle V'; O'; C'; O' \times C' \rangle$ ). El entorno  $E'$  puede constar de nuevos conceptos ( $C \subseteq C'$ ), variables ( $V \subseteq V'$ ), y clases de objetos ( $O \subseteq O'$ ). La inclusión de un nuevo concepto  $C_i$  implica un nuevo análisis de normalidad para dicho concepto ( $E_{c_i}$ ), es decir,  $\forall C_i | C_i \in C' \text{ y } C_i \notin C \text{ entonces } \exists E'_{c_i}$ .

Una vez que se ha definido los conceptos y sus normalidades, es posible analizar el comportamiento de un objeto de acuerdo a estos conceptos.

**Definición 8.** La normalidad de un objeto en relación a un concepto  $c_i$  en un entorno  $E$ , denotado como  $NOE_{c_i}$ , es calculado como el valor máximo de todos los  $SD_{r_{iy}}$  de el concepto  $c_i$ .

$$NOE_{c_i} = \max_{y=1}^k \{SD_{r_{iy}}\} \quad (3.10)$$

Un objeto se comporta de forma adecuada en relación a un concepto  $c_i$  en un entorno  $E$ , cuando el valor máximo de todos los  $SD_{r_{iy}}$  no supera un valor límite  $\alpha_{c_i}$  (definido por un experto) asociado al concepto  $c_i$ . Se verifica que  $\alpha_{c_i} \in [0, 1]$ . Cuanto mayor sea el valor de  $\alpha_{c_i}$  más estricto será el sistema de vigilancia en su análisis de normalidad.

**Definición 9.** La normalidad de un objeto en un entorno  $E$ , denotado como  $NOE$ , se obtiene como el mínimo valor de todos los  $NOE_{c_i}$  (ver Ecuación 3.10).

$$NOE = \min_{i=1}^w \{NOE_{c_i}\} \quad (3.11)$$

Donde  $w$  es el número de conceptos utilizados para analizar el comportamiento del objeto (especificado en  $O \times C$ ). Un objeto se comporta adecuadamente en un entorno si  $\forall NOE_{c_i} \geq \alpha_{c_i}$ , es decir, un objeto no se comporta de forma normal, cuando su comportamiento no respeta la definición de normalidad dada para un concepto. Finalmente, quedaría definir la normalidad en todo un entorno teniendo en cuenta los objetos y sus comportamientos. Todas las situaciones en un entorno  $E$  son normales si no existe un  $NOE_{c_i}$  para un objeto, cuyo valor sea inferior a  $\alpha_{c_i}$ .

Por otra parte, es importante destacar la dificultad de determinar la clase o el tipo de un objeto móvil en una escena real, de hecho, en la mayoría de los casos el sistema es incapaz de afirmar con total certeza la clase a la que pertenece. Por esta razón, se ha optado por realizar en paralelo varios análisis de normalidad para un mismo objeto (Ecuación 3.11), uno por cada clase a la que pueda pertenecer el objeto. Así, si un objeto de la escena puede ser un coche o un peatón con valores de pertenencia de 0.8 y 0.2 respectivamente, se llevarían a cabo dos análisis independientes de normalidad. Esta decisión tiene sentido ya que la definición de normalidad para cada tipo de objeto es diferente, es decir, los comportamientos que son normales para una clase, puede que no los sean para otra diferente.

En la siguiente sección se expondrá un caso de estudio concreto donde se definirá, haciendo uso del formalismo presentado en esta sección, el concepto de *trayectoria normal* y su normalidad.

### 3.4. Caso de estudio: definición de trayectorias normales

El motivo por el cual se ha decidido estudiar las trayectorias es, principalmente, porque cualquier objeto con capacidad de movimiento suele tener un objetivo dentro del entorno y para alcanzar dicho objetivo es necesario dirigirse hacia un lugar determinado. Por esta razón, el análisis de trayectorias puede resultar interesante para la detección de posibles anomalías. En esta sección, se definirá el concepto *trayectoria normal* junto con su normalidad, tanto para peatones como para vehículos. La Figura 3.2 muestra el escenario elegido para la obtención de resultados y la definición de regiones de la escena. Dicho escenario es capturado por una cámara situada en el grupo de investigación ORETO de la Universidad de Castilla-La Mancha.

En este caso concreto, el problema de la vigilancia (Ec. (3.1)) se define como la interpretación y el análisis del entorno mostrado en la Figura 3.2.a y percibido por una cámara ( $P = \{E_1\}$ ). A su vez, el entorno observado  $E_1$  se define de acuerdo a la Ecuación 3.2, donde:

- $V$  es el conjunto de variables utilizadas para definir las trayectorias normales y analizar la normalidad (En la Tabla 3.1 se puede consultar la lista de variables junto con su especificación).
- $O = \{\text{peatón, vehículo}\}$

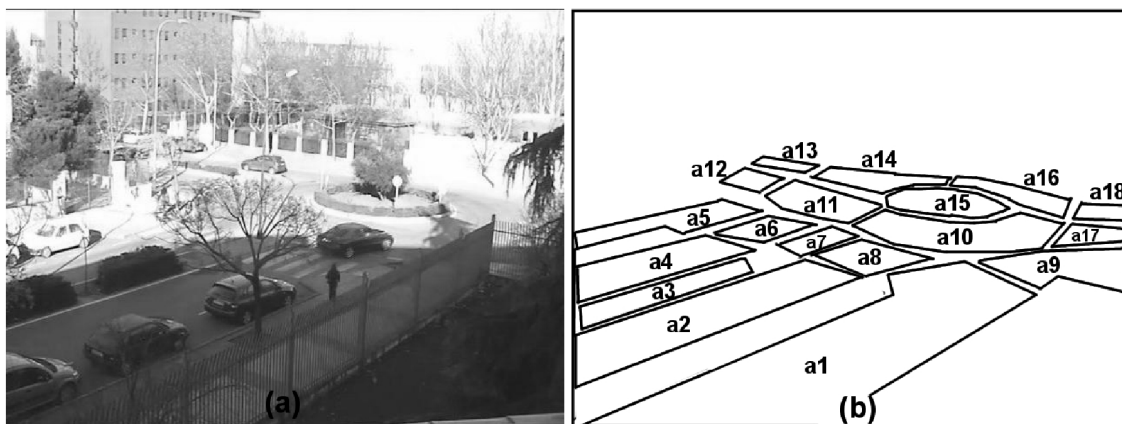


Figura 3.2: (a) Escenario urbano capturado por una cámara situada en el grupo de investigación ORETO de la Universidad de Castilla-La Mancha. (b) Definición de zonas en el entorno.

- $C = \{\text{trayectoria normal}\}$ . A partir de ahora, nos referiremos a el concepto *trayectoria normal* como  $c_1$ .
- $O \times C = \{\{\text{peatón}, c_1\}, \{\text{vehículo}, c_1\}\}$ . Es decir, el concepto *trayectoria normal* se utiliza para analizar la normalidad de los comportamientos de vehículos y peatones.

En las siguientes subsecciones se define la normalidad del concepto  $c_1$  en el entorno  $E_1$  de acuerdo a la Ecuación 3.3 y 3.5.

### 3.4.1. Normalidad del concepto *trayectoria normal*

La normalidad del concepto  $c_1$  ( $N_{c_1}$ ) se construye en base a  $V_1$ ,  $DDV_1$  (ambos definidos en la Tabla 3.1) y  $\Phi_1$ , que representa el conjunto de restricciones utilizadas para definir y analizar la normalidad del concepto  $c_1$ . Además, podemos ver una trayectoria normal como un patrón de comportamiento definido mediante un conjunto de restricciones (instancias de  $\Phi_1$ ); este patrón es reconocido cuando un objeto satisface sus restricciones. En concreto,  $\Phi_1$  está compuesto por ocho tipos de restricciones  $\Phi_1 = \{f_{11}, f_{12}, f_{13}, f_{14}, f_{15}, f_{16}, f_{17}, f_{18}\}$ , cada una de estas restricciones serán descritas en las sucesivas subsecciones.

#### 3.4.1.1. Restricciones de inicio

Cualquier trayectoria normal tiene siempre un área o región de comienzo ( $a_b$ ) o punto de partida, y un área donde finaliza ( $a_e$ ). La restricción de comienzo ( $f_{11}$ ) es una función que determina si un objeto ha comenzado a seguir una trayectoria normal (cada trayectoria normal tiene definida su propia restricción de comienzo, dependiendo siempre del punto de partida).

Variable	Dominio de Definición (DDV)	Descripción
$a$	$\{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}, a_{15}, a_{16}, a_{17}, a_{18}\}$	$a$ representa un área (zona) en el entorno (ver Fig. 3.2.b), en la que un objeto podría encontrarse. Un área se define a partir de un conjunto de puntos.
$pv_a$	$pv_a \in [0, 1]$	Valor de posibilidad de que un objeto se encuentre en el área $a$ .
$a_b$	$a_b \in DDV_a$	Área de comienzo para una trayectoria normal.
$a_e$	$a_e \in DDV_a$	Destino de una trayectoria normal.
$p = \langle x, y \rangle$	$DDV_x \cup DDV_y$	$p$ es un punto en la imagen capturada por una cámara.
$x$	$x \in [0, 640]$	Coordenada horizontal de un punto en la imagen.
$y$	$y \in [0, 480]$	Coordenada vertical de un punto en la imagen.
$C_{obj}$	$C_{obj} \in O$	$C_{obj}$ Clase a la que pertenece el objeto. Si un objeto tiene valores de pertenencia a varias clases, cada una de las opciones es tratada como un objeto individual. Es decir, si un objeto puede ser clasificado como persona o vehículo, cada caso es tratado particularmente con la creación de un objeto persona y un objeto vehículo.
$mv_{obj,c}$	$mv_{obj,c} \in [0, 1]$	$mv_{obj,c}$ valor de pertenencia de un objeto a la clase $c$ .
$\Upsilon$	$\Upsilon \subseteq O$	$\Upsilon$ representa el conjunto de clases de objeto que pueden seguir una determinada trayectoria.
$\ell_a$	$\{DDV_a \times DDV_{pv_a}\}^+$	Secuencia de áreas recorridas por un objeto. Cada área recorrida tiene asociada un valor de posibilidad $pv_a$ .
$\Psi$	$DDV_\Psi \subseteq DDV_z$	$\Psi$ indica la secuencia de zonas permitidas para una trayectoria normal.
$order$	$\{true, false\}$	La variable $order$ se utiliza en las restricciones temporales e indica si la secuencia de áreas permitidas deben recorrerse en orden.
$r_{ij}$	$DDV_{r_{ij}} = \chi_i$	Particularización de los elementos de $\Phi_i$ , la cual representa una trayectoria normal.
$\ell_{r_i}$	$DDV_{\ell_{r_i}} \subseteq \chi_i$	Lista de $r_{ij}$ asociados a un objeto, cuyo $SD_{r_{ij}} > 0$ .
$max_d$	$max_d \in [0, 9999]$	$max_d$ máxima duración permitida, expresada en segundos, para recorrer una trayectoria normal.
$t$	$[1, 31] \cup \{*\} \times [1, 12] \cup \{*\} \times [1900 - 9999] \cup \{*\} \times [0, 24] \cup \{*\} \times [0, 59] \cup \{*\} \times [0, 59] \cup \{*\}$	$t$ representa un instante de tiempo y su formato es: $DD/MM/YY - hh : mm : ss$ . El símbolo $*$ se utiliza como comodín..
$t_c$	$DDV_t$	Instante de tiempo actual (utilizado en las restricciones temporales).
$t_b$	$DDV_t$	Instante de tiempo en el cual se asoció una trayectoria normal a un objeto.
$Int_j = \langle t_s, t_e \rangle$	$DDV_t \times DDV_t$	Intervalo de tiempo definido a partir de dos instantes de tiempo (comienzo y final).

Tabla 3.1: Lista de variables utilizadas para definir las trayectorias normales.

$$f_{11}(a, pv_a, a_b) = \begin{cases} pv_a & \text{if } a = a_b; \\ 0 & \text{en caso contrario;} \end{cases}$$

### 3.4.1.2. Restricción de finalización

La definición de la restricción de finalización es similar a la de inicio, y determina si un objeto ha finalizado una trayectoria normal.

$$f_{12}(a, pv_a, a_e) = \begin{cases} pv_a & \text{if } a = a_e; \\ 0 & \text{en caso contrario;} \end{cases}$$

Las restricciones de inicio y finalización son las únicas que son obligatorias para definir una trayectoria normal. Es decir, toda trayectoria normal tiene un área de origen y un área final, mientras que el resto de restricciones complementan la definición de trayectoria normal. En la Figura 3.3 se muestra un ejemplo sencillo de trayectoria normal, cuyo origen se encuentra en el área *SidewalkA* y el final en *SidewalkB*.



Figura 3.3: Ejemplo de una trayectoria normal ( $P_1$ ) definida a partir de un área de comienzo (*SidewalkA*) y un área de finalización (*SidewalkB*)

### 3.4.1.3. Restricción de roles

Este tipo de restricción indica qué tipo de objetos pueden seguir una trayectoria normal específica. Por ejemplo, se podría ampliar la definición de la trayectoria normal de la Figura 3.3 añadiendo una restricción en la que la lista de objetos permitidos estaría formada únicamente por la clase “peatón”. Así, si un vehículo realizara esta trayectoria no cumpliría todas las restricciones, ya que los vehículos no pertenecen a la lista de roles permitidos.

$$f_{13}(c_{obj}, mv_{obj,c}, \Upsilon) = \begin{cases} mv_{obj,c} & \text{if } c_{obj} \in \Upsilon; \\ 0 & \text{en otro caso;} \end{cases}$$

## 3.4.1.4. Restricciones espaciales

Las restricciones espaciales, asociadas a una trayectoria normal, especifican las áreas intermedias entre el origen y el destino que debe recorrer un objeto. Cuando un experto define las trayectorias normales y les asocia restricciones espaciales puede especificar si se debe tener en cuenta el orden. En caso afirmativo, para que el sistema considere que el objeto está siguiendo una trayectoria normal, éste debe pasar por todas las zonas y en el orden especificado. En cambio, si el orden no se tiene en cuenta, el objeto no tiene por qué pasar por todas las zonas especificadas en la restricción y mucho menos en orden. En este caso la restricción indicaría las zonas por las que podría pasar sin implicar obligatoriedad.

$$f_{14}(\ell_a, \Psi, order) = \begin{cases} 0 & \text{if } (lastElement(\ell_a) \cap \Psi_i = \emptyset) \\ & \vee (order = true \wedge \neg Ordered(\Psi_i, \ell_a)); \\ min(\ell_a \cap \Psi) & \text{if } (order = false \wedge (lastElement(\ell_a) \cap \Psi_i \neq \emptyset)) \\ & \vee (order = true \wedge Ordered(\Psi_i, \ell_a)); \end{cases}$$

Donde  $\emptyset$  simboliza un valor nulo o no definido y  $\ell_a$  se refiere a la secuencia de zonas recorridas por un objeto en movimiento. La Figura 3.4 muestra un ejemplo de una secuencia de zonas recorridas, donde cada elemento de  $\ell_a$  contiene un conjunto de pares de áreas y valores de posibilidad asociados  $pv_a$ . La función *lastElement* devuelve el último elemento de  $\ell_a$ . La función *Ordered* indica si las áreas están siendo recorridas en el mismo orden que especifica  $\Psi$ . Finalmente,  $min(\ell_a \cap \Psi)$  es el mínimo valor de posibilidad  $pv_a$  entre todas las zonas recorridas, y que pertenecen a  $\ell_a$  y  $\Psi$ .

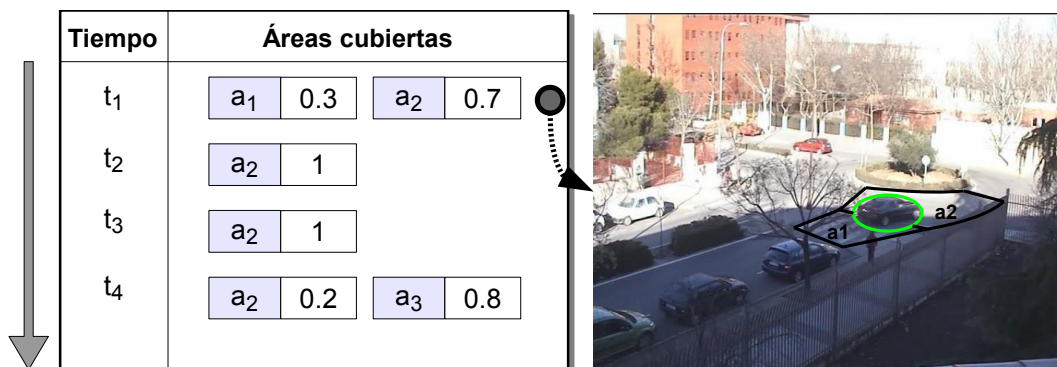


Figura 3.4: Ejemplo de una secuencia de áreas recorridas por un objeto. Cada instante de tiempo  $t$  puede cubrir varias áreas al mismo tiempo

La definición de la trayectoria normal  $P_1$  de la Figura 3.3, también podría ser ampliada con una restricción espacial, incluyendo en la lista de áreas permitidas el paso de peatones que existe entre las áreas de origen y destino. La Figura 3.5 muestra algunos ejemplos de recorridos válidos con y sin restricciones espaciales en la definición de la trayectoria normal.



Figura 3.5: (a) La trayectoria normal se define mediante una zona de origen y destino. Las líneas discontinuas muestran algunos recorridos que serían reconocidos mediante esta definición de trayectoria normal; (b) Se amplía la definición de la trayectoria normal con la inclusión del paso de peatones como zona intermedia. Al igual que en (a), las líneas discontinuas muestran algunos recorridos válidos acordes a la nueva definición.

### 3.4.1.5. Restricciones temporales

Las restricciones temporales indican **cuándo** es normal que se recorran las trayectorias definidas. Estas restricciones se dividen en tres tipos, el primero de ellos ( $f_{15}$ ) especifica si es necesario recorrer una trayectoria normal  $r_{ij}$  previamente a la actual.

$$f_{15}(r_{ij}, \ell_{r_i}) = \begin{cases} SD_{r_{ij}} & \text{if } SD_{r_{ij}} > 0 \wedge r_{ij} \in \ell_{r_i}; \\ 0 & \text{en caso contrario;} \end{cases}$$

Donde  $SD_{r_{ij}}$  es el grado de satisfacción con el que fue reconocida la trayectoria normal especificada en la restricción (ver Ec. 3.7). En segundo lugar, la restricción temporal  $f_{16}$  indica el valor máximo de duración permitido para recorrer una determinada trayectoria normal.

$$f_{16}(max_d, t_c, t_b) = \begin{cases} 1 & \text{if } (max_d = ) \vee (max_d \leq (t_c - t_b)) \\ 0 & \text{en caso contrario;} \end{cases}$$

Siendo  $t_c$  el instante de tiempo actual y  $t_b$  el instante de tiempo en el que el objeto comenzó a seguir la trayectoria actual. Finalmente, el tercer tipo de restricciones temporales ( $f_{17}$ ) determinan el intervalo de tiempo en el que es normal seguir la trayectoria. Las restricciones y las relaciones temporales que existen entre los eventos simples son vitales para representar e interpretar eventos compuestos (comportamientos complejos). Por este motivo, se han definido cinco relaciones temporales basadas en el álgebra de Allen [3, 4], las cuales se muestran en la Tabla 3.2. Estas relaciones se utilizan para comprobar si un objeto está siguiendo una trayectoria normal en el intervalo de tiempo apropiado.

$f_{17}$  se construye en base a las relaciones temporales especificadas en la Tabla 3.2.



Relación temporal ( $t_i, Int_j$ )	Definición lógica	Satisface la restricción
Antes	$t_i < comienzo(Int_j)$	NO
Después	$final(Int_j) < t_i$	NO
Durante	$comienzo(Int_j) \leq t_i \leq final(Int_j)$	SÍ
Al_comienzo	$comienzo(Int_j) = t_i$	SÍ
Al_final	$fin(Int_j) = t_i$	SÍ

Tabla 3.2: Relaciones temporales entre instantes e intervalos de tiempo.

$$f_{17}(Int_p, t_c) = \begin{cases} 1 & \text{if } (Int_p = ) \vee (During(t_c, Int_p) \vee Starts(t_c, Int_p) \\ & \vee (Finish(t_c, Int_p))) \\ 0 & \text{en caso contrario;} \end{cases}$$

#### 3.4.1.6. Restricciones de destino

Las restricciones de destino comprueban en cada instante si un objeto se está acercando al destino indicado en cada una de las trayectorias normales asociadas. Si la distancia con respecto al destino disminuye en cada instante la función devuelve el valor 1, en caso contrario, el valor decrece hasta cero a medida que el objeto se aleja. En otras palabras, la posibilidad de que un objeto esté siguiendo una determinada trayectoria normal decrece a medida que éste se aleje de su destino.

$$f_{18}(d_i, d_c) = \begin{cases} 1 & \text{if } d_c \leq d_i \\ 1 - \frac{d_c}{2 * d_i} & \text{if } d_i < d_c < 2 * d_i \\ 0 & \text{if } d_c \geq 2 * d_i; \end{cases}$$

Donde  $d_i$  especifica la distancia inicial entre el objeto y el destino de la trayectoria normal y  $d_c$  se refiere a la distancia actual. Cuando la distancia actual es el doble de la inicial, es una distancia lo suficientemente grande para considerar que el objeto no se dirige hacia el destino de la trayectoria normal y, por tanto, no está siguiendo dicha trayectoria.

#### 3.4.2. Normalidad del concepto *trayectoria normal* en el entorno

$E_1$

Teniendo en cuenta la Ecuación 3.5, la normalidad del concepto  $c_1$  en el entorno  $E_1$  depende de  $N_{c_1}$  (definida en la sección anterior) y  $\chi_1$ , que representa el conjunto de particularizaciones de  $\Phi_1$ . La Tabla 3.3 reúne las trayectorias normales definidas para el entorno  $E_1$ .

Un objeto recorre una trayectoria normal si existe un  $r_{ij}$  cuyo grado de satisfacción ( $SD_{r_{ij}}$ ) es mayor que un valor umbral  $\alpha_{c_1}$  definido por un experto ( $\alpha_{c_1} \in [0, 1]$ ). Además, cuando un objeto se mueve y modifica su estado, se vuelven a calcular todos los valores  $SD_{r_{ij}}$  de las trayectorias asociadas (las trayectorias que tiene asociadas un objeto, son las trayectorias que podría estar recorriendo).

$r_{ij}$	$f_{11}$	$f_{12}$	$f_{13}$	$f_{14}$	$f_{15}$	$f_{16}$	$f_{17}$
$r_{11}$	$a_b = \{a_2\}$	$a_e = \{a_{13}\}$	$\Upsilon = \{vehiculo\}$	order=true, $\Psi = \{a_8, a_{10}, a_{16}, a_{14}\}$	$\emptyset$	$\emptyset$	$\emptyset$
$r_{12}$	$a_b = \{a_2\}$	$a_e = \{a_{17}\}$	$\Upsilon = \{vehiculo\}$	order=true, $\Psi = \{a_8, a_{10}\}$	$\emptyset$	$\emptyset$	$\emptyset$
$r_{13}$	$a_b = \{a_2\}$	$a_e = \{a_4\}$	$\Upsilon = \{vehiculo\}$	order=true, $\Psi = \{a_8, a_{10}, a_{16}, a_{14}, a_{11}, a_6\}$	$\emptyset$	$\emptyset$	$\emptyset$
$r_{14}$	$a_b = \{a_1\}$	$a_e = \{a_7\}$	$\Upsilon = \{peaton\}$	$\Psi = \{a_8\}$	$\emptyset$	$\emptyset$	$\emptyset$
$r_{15}$	$a_b = \{a_7\}$	$a_e = \{a_5\}$	$\Upsilon = \{peaton\}$	$\Psi = \{a_6\}$	$r_{ij} = \{r_{14}\}$	$\emptyset$	$\emptyset$
$r_{16}$	$a_b = \{a_5\}$	$a_e = \{a_7\}$	$\Upsilon = \{peaton\}$	$\Psi = \{a_6\}$	$\emptyset$	$\emptyset$	$\emptyset$
$r_{17}$	$a_b = \{a_7\}$	$a_e = \{a_1\}$	$\Upsilon = \{peaton\}$	$\Psi = \{a_8\}$	$r_{ij} = \{a_6\}$	$\emptyset$	$\emptyset$
$r_{18}$	$a_b = \{a_{12}\}$	$a_e = \{a_4\}$	$\Upsilon = \{peaton\}$	order=true, $\Psi = \{a_6, a_{11}\}$	$\emptyset$	$\emptyset$	$\emptyset$
$r_{19}$	$a_b = \{a_{12}\}$	$a_e = \{a_{17}\}$	$\Upsilon = \{peaton\}$	order=true, $\Psi = \{a_{11}, a_{14}, a_{16}\}$	$\emptyset$	$\emptyset$	$\emptyset$
$r_{1,10}$	$a_b = \{a_{18}\}$	$a_e = \{a_{13}\}$	$\Upsilon = \{peaton\}$	order=true, $\Psi = \{a_{14}, a_{16}\}$	$\emptyset$	$\emptyset$	$\emptyset$

Tabla 3.3: Conjunto de trayectorias normales definidas para el entorno  $E_1$ . Cada  $r_{ij}$  representa una trayectoria normal, la cual es recorrida por un objeto cuando éste satisface sus restricciones.  $f_{18}$  no se incluye en la tabla porque, a diferencia del resto de restricciones, no puede ser definida por el experto, sino que se calcula en función de  $d_i$  y  $d_c$ .

### 3.4.3. Análisis de normalidad de un objeto en movimiento

El análisis de normalidad de los objetos en movimiento determina si éstos se comportan correctamente de acuerdo a los conceptos definidos en el entorno. Los conceptos que forman parte del análisis de normalidad son definidos en  $O \times C$  y dependen de la clase del objeto. La Figura 3.6.a ilustra de forma general la asociación entre objetos y conceptos, Fig. 3.6.b muestra la asociación del concepto *trayectoria normal* a un objeto y, finalmente, Fig. 3.6.c y 3.6.d muestra dos ejemplos en los que un objeto  $obj_1$  no se comporta de forma normal.

De acuerdo a la Ecuación 3.10, la normalidad de un objeto con respecto a el concepto  $c_1$  en el entorno  $E_1$ , denotado como  $NOE_{c_1}$ , se obtiene como el máximo valor de todos los  $SD_{r_{1j}}$ . Un objeto se comporta correctamente de acuerdo al concepto  $c_1$  en  $E_1$  cuando el máximo valor de todos los  $SD_{r_{1j}}$  es mayor que  $\alpha_{c_1}$ . Es decir, un objeto se comporta correctamente mientras esté recorriendo, al menos, una de las trayectorias normales. Para finalizar esta sección, la Figura 3.7 y la Tabla 3.4 muestra frame a frame el comportamiento de un objeto en movimiento y la asociación de trayectorias normales.

## 3.5. Resultados experimentales

Para poner en práctica el modelo conceptual propuesto en este trabajo, se ha implementado un prototipo de sistema de vigilancia inteligente (denominado OCULUS), que permite el análisis de normalidad en función de diferentes conceptos. En la Figura 3.8 se muestra la arquitectura conceptual de OCULUS.

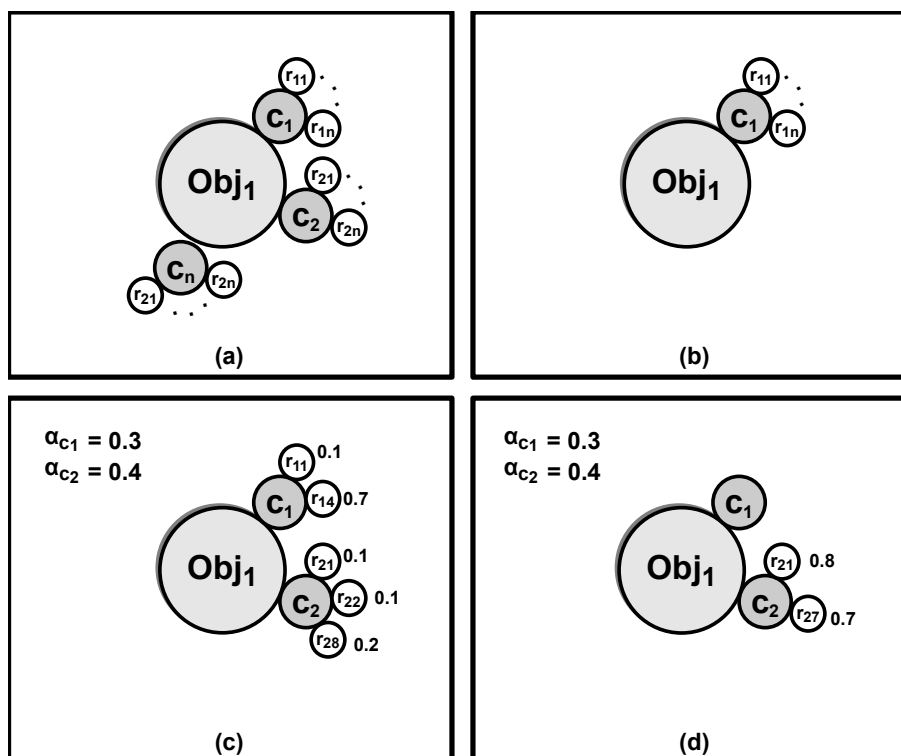


Figura 3.6: (a) Representación general de la asociación entre objetos y conceptos. Se realiza un análisis de normalidad por cada uno de los conceptos asociados al objeto. (b) Asociación del concepto *trayectoria normal* al objeto *obj1* en movimiento. (c) Comportamiento anormal de *obj1* debido a que el grado de satisfacción de  $c_2$  no supera el valor umbral  $\alpha_{c_2}$ . (d) Comportamiento anormal del objeto *obj1* debido a que no existe un  $r_{1j}$  cuyo grado de satisfacción  $SD_{r_{ij}} > 0$ .

En la capa inferior o capa perceptual se encuentran los sensores encargados de percibir el entorno y de enviar la información a las capas superiores. Si algo caracteriza a la información que procede directamente de los sensores es que ésta posee un alto grado de incertidumbre y vaguedad, aspecto que debe tener en cuenta la capa intermedia a la hora de intentar interpretar lo que ocurre. Además, la información de bajo nivel no siempre se envía como tal a las capas superiores, sino que podría ser tratada mediante diferentes algoritmos. Por ejemplo, en el caso del vídeo se ejecutan algoritmos de segmentación y de tracking.

Por otra parte, la capa conceptual o capa intermedia es la encargada de realizar la interpretación de comportamientos y eventos que suceden en el entorno. A partir de la información que procede de la capa inferior, se construye el conjunto de variables ( $V$ ) y el dominio de definición de cada una de ellas ( $DDV$ ), que serán utilizadas por los componentes de normalidad para analizar lo que ocurre en cada instante de tiempo. Llamamos componentes de normalidad (ver Figura 3.8) a aquellos componentes software independientes que analizan la normalidad de acuerdo a un concepto  $c_i$ , por tanto, por cada concepto  $c_i$  existirá un componente de normalidad. Cuando hablamos de componentes software independientes nos referimos a que cada uno de ellos puede tener su

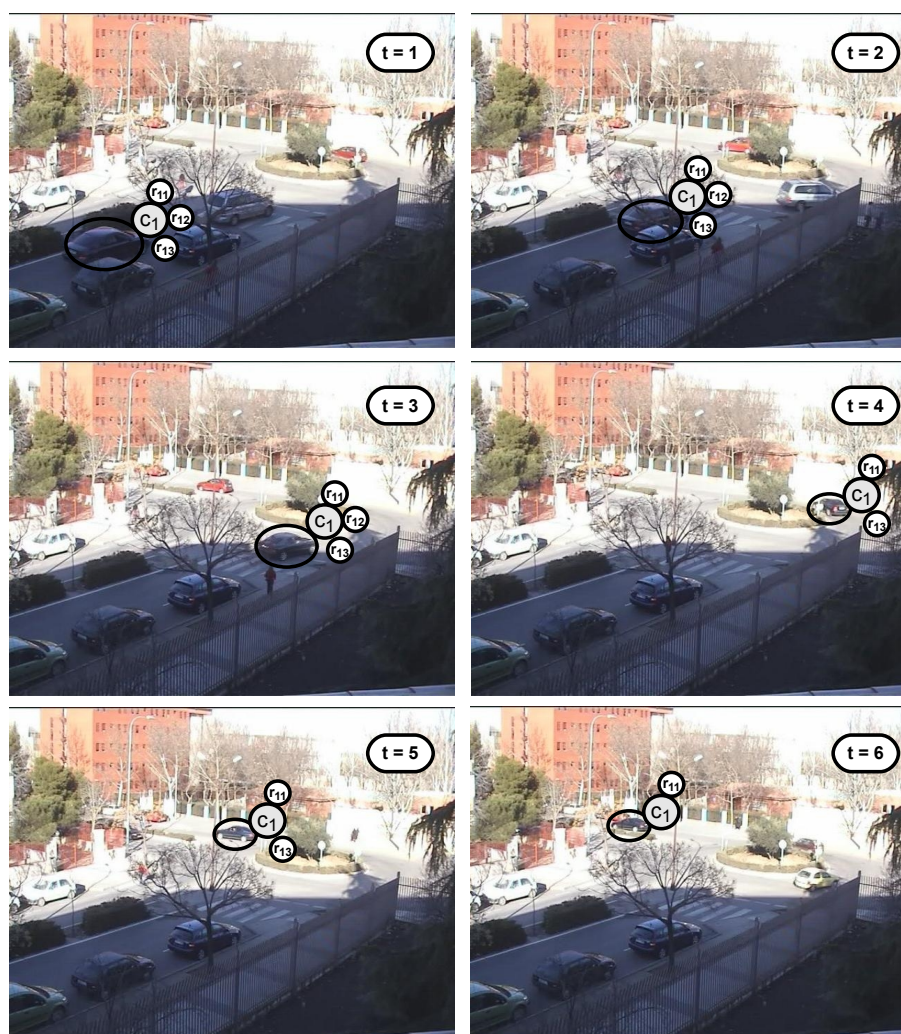


Figura 3.7: Seguimiento de un coche en el entorno y asociación de trayectorias normales

propio núcleo de razonamiento y estar implementado con tecnologías diferentes (esto es posible gracias al uso de un *middleware*).

Entre las grandes ventajas de este planteamiento podemos destacar dos de ellas:

- El sistema es fácilmente extensible. Si queremos ampliar el análisis de normalidad basta con definir un nuevo concepto y construir su correspondiente componente de normalidad.
- El sistema es de naturaleza distribuida. Gracias al uso del *middleware* y la descomposición en componentes independientes, cada uno de ellos puede estar localizado en una máquina diferente. Esta característica es vital en un sistema de seguridad, ya que al no ser un sistema centralizado éste podrá seguir funcionando correctamente aunque alguna de las partes quede dañada. Además, recordemos

$t$	obj	Clasificación	Ubicación	$r_{ij}$ asociados al objeto	$SD_{r_{ij}}$
1	obj <sub>1</sub>	$mv_{obj_1,car} = 1$	$pv_{a_2} = 0,8$ $pv_{a_3} = 0,2$	$r_{11} = \{\tilde{f}_{11} = 0,8, \tilde{f}_{13} = 1, \tilde{f}_{18} = 1\}$	0.8
				$r_{12} = \{\tilde{f}_{11} = 0,8, \tilde{f}_{13} = 1, \tilde{f}_{18} = 1\}$	0.8
				$r_{13} = \{\tilde{f}_{11} = 0,8, \tilde{f}_{13} = 1, \tilde{f}_{18} = 1\}$	0.8
2	obj <sub>1</sub>	$mv_{obj_1,car} = 1$	$pv_{a_2} = 1$	$r_{11} = \{\tilde{f}_{11} = 0,8, \tilde{f}_{13} = 1, \tilde{f}_{18} = 1\}$	0.8
				$r_{12} = \{\tilde{f}_{11} = 0,8, \tilde{f}_{13} = 1, \tilde{f}_{18} = 1\}$	0.8
				$r_{13} = \{\tilde{f}_{11} = 0,8, \tilde{f}_{13} = 1, \tilde{f}_{18} = 0,8\}$	0.8
3	obj <sub>1</sub>	$mv_{obj_1,car} = 1$	$pv_{a_8} = 0,4$ $pv_{a_{10}} = 0,6$	$r_{11} = \{\tilde{f}_{11} = 0,8, \tilde{f}_{13} = 1, \tilde{f}_{14} = 0,4, \tilde{f}_{18} = 1\}$	0.4
				$r_{12} = \{\tilde{f}_{11} = 0,8, \tilde{f}_{13} = 1, \tilde{f}_{14} = 0,4, \tilde{f}_{18} = 1\}$	0.4
				$r_{13} = \{\tilde{f}_{11} = 0,8, \tilde{f}_{13} = 1, \tilde{f}_{14} = 0,4, \tilde{f}_{18} = 0,68\}$	0.4
4	obj <sub>1</sub>	$mv_{obj_1,car} = 1$	$pv_{a_{16}} = 1$	$r_{11} = \{\tilde{f}_{11} = 0,8, \tilde{f}_{13} = 1, \tilde{f}_{14} = 0,4, \tilde{f}_{18} = 1\}$	0.4
				$r_{12} = \{\tilde{f}_{11} = 0,8, \tilde{f}_{13} = 1, \tilde{f}_{14} = 0, \tilde{f}_{18} = 0,7\}$	0
				$r_{13} = \{\tilde{f}_{11} = 0,8, \tilde{f}_{13} = 1, \tilde{f}_{14} = 0,4, \tilde{f}_{18} = 0,50\}$	0.4
5	obj <sub>1</sub>	$mv_{obj_1,car} = 1$	$pv_{a_{14}} = 1$	$r_{11} = \{\tilde{f}_{11} = 0,8, \tilde{f}_{13} = 1, \tilde{f}_{14} = 0,4, \tilde{f}_{18} = 1\}$	0.4
				$r_{13} = \{\tilde{f}_{11} = 0,8, \tilde{f}_{13} = 1, \tilde{f}_{14} = 0,4, \tilde{f}_{18} = 0,35\}$	0.35
6	obj <sub>1</sub>	$mv_{obj_1,car} = 1$	$pv_{a_{13}} = 1$	$r_{11} = \{\tilde{f}_{11} = 0,8, \tilde{f}_{12} = 1, \tilde{f}_{13} = 1, \tilde{f}_{14} = 0,4, \tilde{f}_{18} = 1\}$	0.4
				$r_{13} = \{\tilde{f}_{11} = 0,8, \tilde{f}_{13} = 1, \tilde{f}_{14} = 0, \tilde{f}_{18} = 0,20\}$	0

Tabla 3.4: Valores numéricos obtenidos en el proceso de seguimiento del objeto de la Figura 3.7 con un valor umbral  $\alpha_{c_1} = 0,3$

que otro de los requisitos fundamentales era ofrecer respuestas en un tiempo cercano al real, por lo que el hecho de disponer de varios núcleos de procesamiento hace más factible la consecución de este objetivo.

Actualmente, el sistema OCULUS cuenta con un componente de normalidad para analizar las trayectorias que siguen los objetos en el entorno observado. La formalización del concepto *trayectoria normal* y su normalidad se ha llevado a cabo mediante el formalismo presentado en este trabajo y el uso del lenguaje CLIPS. Además, no sólo se utiliza el lenguaje para la representación del conocimiento, sino que también su potente motor de inferencia para realizar el proceso de análisis e interpretación.

Para finalizar la descripción de la capa intermedia observemos la parte superior de la misma en la Figura 3.8, donde se encuentran los módulos de análisis de anomalías, el análisis global de normalidad y el módulo de predicción. El primero de ellos se ejecuta en paralelo con el análisis de normalidad, es decir, se intentan buscar correspondencias entre las conductas actuales y las conductas anormales más comunes que han sido definidas. Por otra parte, el segundo módulo analiza un análisis global de normalidad en función de los resultados ofrecidos por cada componente de normalidad (la situación actual será normal siempre y cuando cada uno de los componentes indique normalidad). El último de los módulos (módulo de predicción) es el encargado de intentar anticiparse a futuros sucesos. Este módulo juega un papel vital, ya que en un sistema de seguridad no sólo es importante determinar qué es lo que ha ocurrido sino que también intentar evitar aquellos sucesos que puedan ser perjudiciales para el entorno. Es importante recalcar que estos últimos módulos no han sido implementados aún y su creación forma parte de las líneas de trabajo futuro.

Finalmente, la última capa es la encargada de realizar la gestión de crisis y tomar las

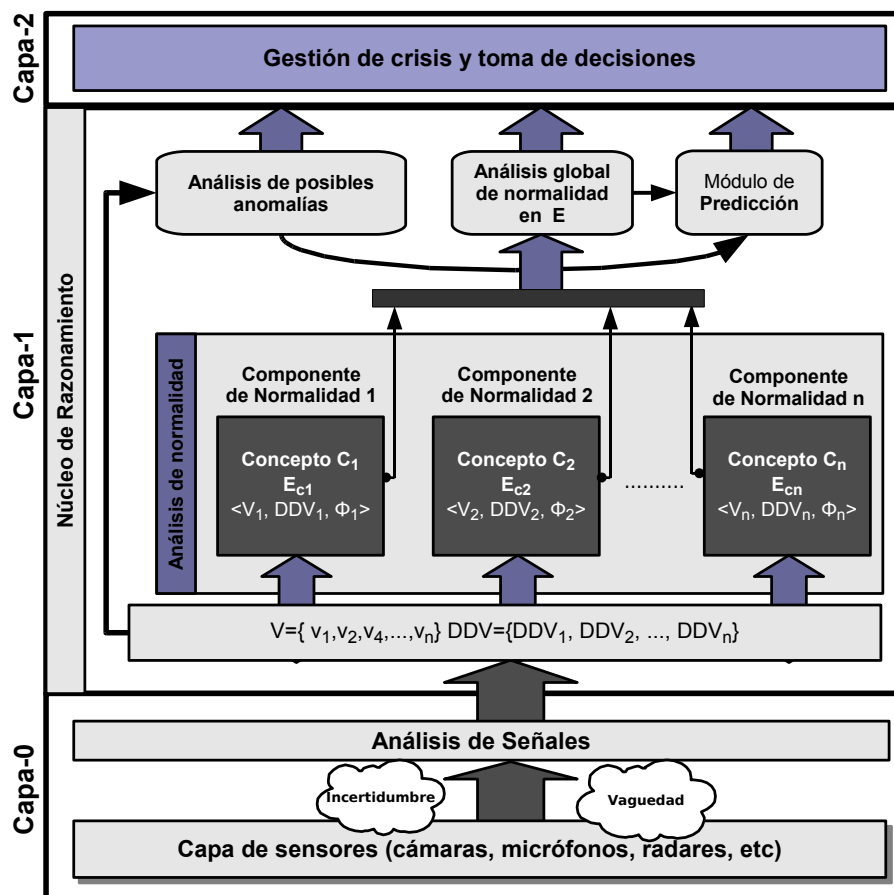


Figura 3.8: Arquitectura conceptual del sistema OCULUS

decisiones en función de los análisis realizados por la capa subyacente. Como entrada podría recibir la identificación de una o varias situaciones anormales reconocidas, análisis de normalidad negativo porque no se ha cumplido la normalidad de algún concepto y ninguna situación anormal reconocida, o bien, todas las situaciones reconocidas como normales. Además de estas entradas, la capa superior también tiene en cuenta las predicciones para tomar las decisiones.

A continuación se comentan algunos resultados obtenidos a partir de vídeos reales de dos minutos de duración y que fueron grabados en el mes de Mayo. La Tabla 3.5 resume los test realizados y en ella se especifica el intervalo de tiempo en el que fueron realizados, los diferentes tipos de objetos que estuvieron en movimiento durante ese intervalo, y como éstos fueron clasificados por el sistema OCULUS.

Por otra parte, en la Tabla 3.6 se indica si cada uno de los objetos que participan en los test se comportan de forma normal, o por el contrario, tiene conductas anómalas. Además, los valores numéricos de la segunda columna reflejan la eficiencia del sistema a la hora de clasificar estos comportamientos. Como bien indica la Tabla 3.7, la mayoría de los errores que se producen en la capa intermedia son debido a clasificaciones

Test	Intervalo de tiempo	Objetos		Porcentaje de acierto en clasificación	
		Peatón	Vehículo	Peatón	Vehículo
1	8:00-8:02	5	7	5/5(100%)	7/7(100%)
2	11:15-11:17	16	12	15/16(93%)	11/12(91%)
3	14:05-14:07	24	16	21/24 (87%)	16/16(100%)
4	18:20-18:22	12	18	11/12 (91%)	16/18(88%)
5	22:30-22:32	6	7	4/6 (66%)	4/7(57%)

Tabla 3.5: Número de objetos en movimiento y resultados de clasificación por test. Cada test tiene una duración de 120 segundos y los resultados fueron obtenidos en el mes de Mayo. Durante este mes el sol sale a las 7:00 am y se oculta a las 9:30 pm. Esta información se hace explícita porque las condiciones de luz afectan en gran medida a los resultados.

Test	Comportamiento de peatones		Comportamiento de vehículos	
	Normal	No normal	Normal	No normal
1	5/5(100%)	0/0	7/7(100%)	0/0
2	13/16(81%)	0/0	11/12(91%)	0/0
3	16/20(80%)	3/4(75%)	13/13 (100%)	3/3(100%)
4	7/8(87%)	4/4(100%)	14/16(87%)	2/2(100%)
5	3/6(50%)	0/0	4/7(57%)	0/0

Tabla 3.6: Comportamientos normales y anómalos. El comportamiento de un objeto siempre es clasificado como normal o anormal independientemente del número de trayectorias que esté siguiendo

incorrectas en las capas inferiores (recordemos que los comportamientos normales suelen variar de una clase de objeto a otra). Normalmente, las condiciones de iluminación, sombras, oclusiones y la perspectiva suelen ser las principales causas de clasificaciones incorrectas en el análisis de vídeo.

Para reconocer las trayectorias que realizan los objetos es necesario determinar en cada momento la zona o el área donde se encuentran. En este contexto, tanto la altura de la cámara como el ángulo son dos factores claves para determinar la ubicación de un objeto (ambos determinan la perspectiva). Al igual que sucede con la clasificación de los objetos, la localización incorrecta repercute directamente en los resultados finales, ya que el sistema interpretará trayectorias que realmente no están ocurriendo.

Finalmente, la Tabla 3.8 muestra estadísticas relacionadas con las asociaciones de trayectorias normales a los objetos. Es importante tener en cuenta que una trayectoria

Test	Clasificación de objetos	Fuentes de errores		
		Luces y sombras	Oclusión	Perspectiva
1	0	0	0	0
2	1	0	2	1
3	2	1	0	2
4	3	0	0	0
5	5	0	0	1

Tabla 3.7: Detalle de las causas de error en cada uno de los test

$r_{ij}$	Asociación de trayectorias normales		Trayectorias reconocidas	
	Peatón	Vehículo	Peatón	Vehículo
$r_{11}$	0	26	0	9
$r_{12}$	0	26	0	10
$r_{13}$	0	26	0	6
$r_{14}$	34	0	30	0
$r_{15}$	41	0	27	0
$r_{16}$	27	0	21	0
$r_{17}$	51	0	21	0
$r_{18}$	0	7	0	7
$r_{19}$	0	14	0	11
$r_{1,10}$	0	13	0	11

Tabla 3.8: Estadísticas relacionadas con la asociación de trayectorias normales y reconocimiento de las mismas.

asociada a un objeto no es totalmente reconocida hasta que el objeto llega al destino y las restricciones han sido satisfechas. Por tanto, habrá trayectorias que son asociadas en un principio a un objeto pero que no pasan al conjunto de trayectorias reconocidas al no cumplir las restricciones o no llegar al área de destino.



---

# 4

---

## Conclusiones y trabajo futuro

---

En la presente Tesis de Máster se ha elaborado un estudio del arte sobre los sistemas de vigilancia donde ha quedado de manifiesto la evolución desde los primeros sistemas más rudimentarios de circuito cerrado CCTV, hasta los sistemas de tercera generación distribuidos formados por multitud de tipos de sensores y con capacidad para interpretar lo que sucede en el entorno observado por dichos sensores. Son éstos últimos, conocidos también como sistemas cognitivos de seguridad avanzados o de tercera generación, los que se encuentran aún en fase de investigación y no han alcanzado un nivel de madurez aceptable. Además, con el objetivo de ofrecer un enfoque general y poner en contexto el trabajo de investigación, se han analizado las diferentes etapas que suelen formar parte de un sistema de este tipo y se han descrito algunos de los trabajos más importantes en cada una de ellas.

Como comentamos anteriormente en el capítulo 2, existe una fuerte dependencia entre las capas de un sistema cognitivo donde el resultado obtenido por cada una de las capas sirve como entrada para la capa inmediatamente superior, es decir, el éxito o fracaso en los resultados obtenidos por las capas inferiores repercute directamente en el trabajo de las capas superiores. Esta dependencia es uno de los principales motivos por los cuales la mayoría de investigadores optan por elegir las capas inferiores para centrar sus trabajos y son pocos los que deciden simular los datos ofrecidos por las capas inferiores para trabajar directamente en las superiores. De hecho, el nivel de madurez y número de trabajos propuestos en las capas inferiores es mucho mayor, y actualmente existen algoritmos muy eficientes para el tratamiento de señales, procesamiento de vídeo, segmentación, tracking, etc. No ocurre así con las capas superiores donde no abundan los trabajos orientados a la seguridad para representar y analizar comportamientos complejos. Por este motivo se ha decidido centrar el trabajo de investigación en el modelado de entornos reales y el análisis de comportamientos en entornos observados, simulando y obteniendo de forma manual los datos que deben ofrecer las capas de bajo nivel.

En estas capas, tanto en la de modelado y representación del conocimiento de un

entorno, así como la capa encargada de analizar los comportamientos, existen una serie de problemas (ver Sección 2.4) entre los que cabe destacar la necesidad de ofrecer respuestas en un tiempo cercano al real (velocidad de respuesta), reconocer el mayor número de situaciones anormales posibles y poder responder ante éstas, mecanismos para facilitar la construcción de la base de conocimiento y eliminar la dependencia con el experto, construcción de sistemas escalables para facilitar la ampliación y mejora de los análisis de situaciones y, finalmente, fusión de la información procedente de múltiples sensores para reforzar el proceso de razonamiento.

En concreto, en el presente trabajo se ha propuesto un modelo para representar y formalizar la normalidad de un entorno. Este modelo permite definir varios conceptos de un entorno (velocidad, trayectorias, tamaño, etc), la normalidad de éstos de forma general en base a una serie de restricciones, la normalidad de cada concepto en un entorno concreto con instanciaciones de estas restricciones, y la normalidad de los objetos en ese entorno en base a los conceptos definidos. Además, el modelo permite especificar cuáles son los conceptos que se deben emplear para analizar la normalidad de cada tipo de objeto, es decir, permite construir un análisis de normalidad a medida en función de la clase del objeto. El modelo propuesto en el presente trabajo ofrece una posible solución al segundo problema mencionado anteriormente, es decir, mediante la definición de las situaciones normales el sistema puede detectar una amplia mayoría de situaciones anormales y permitir que un sistema de vigilancia pueda cambiar su estado actual a un estado de alerta. Además, el modelo también ofrece una posible solución al primer problema, ya que la posibilidad de elaborar análisis de normalidad a medida permite buscar un equilibrio entre análisis más sofisticados con mayores requisitos de cómputo y análisis menos exigentes que ofrecen respuestas en un tiempo cercano al real. A su vez, la arquitectura conceptual propuesta en la Sección 3.5 y el concepto de *componente de normalidad* como componente software independiente facilitan la extensión del sistema con la inclusión de nuevos conceptos, ofreciendo una posible solución al problema de la escalabilidad.

Por otro lado, también se ha implementado un prototipo software que hace uso del modelo teórico presentado en este trabajo para definir las trayectorias normales en un entorno observado por una cámara. El prototipo no trabaja con vídeo grabado en tiempo real y los procesos de *tracking* y segmentación se hacen de forma manual (construcción manual de un archivo xml con toda la información de la escena en cada *frame* y posterior generación de eventos en el middleware ZeroC-ICE). El sistema trata la incertidumbre y analiza las trayectorias de los vehículos y peatones para determinar si éstas son normales.

En cuanto a posibles ampliaciones o mejoras del modelo presentado en este trabajo y líneas de trabajo futuro, podemos destacar las siguientes:

- Como se comentó en el capítulo 3, la mejor opción para modelar los comportamientos de un entorno real es definir todas las situaciones normales y las anómalas más conocidas. Sin embargo, a pesar de que la normalidad es bien conocida es posible que el experto olvide definir algunas de ellas en entornos muy complejos, y esto dará lugar a un número elevado de falsas alarmas (problema importante en los sistemas de seguridad). Este problema se puede solucionar si el experto amplía la base de conocimiento del sistema con nuevas definiciones, el inconveniente es que de esta forma no se mejora la autonomía del sistema artificial. El modelo original se podría mejorar si se asocian algoritmos de aprendizaje a cada componente de normalidad, de tal forma que el sistema podría aprender aquellas situaciones

---

normales sin definir y mejoraría su comportamiento a media que avanza el tiempo.

- En el presente trabajo se ha definido el concepto de trayectoria normal y su normalidad en un entorno concreto. Como trabajo futuro queda pendiente la definición de un mayor número de conceptos y su implementación mediante componentes de normalidad. Algunos de estos conceptos podrían ser, velocidades normales, relaciones de proximidad (hasta que punto es normal que un objeto este cerca de otro o cerca de una zona), estancias en zonas permitidas según el tipo del objeto, objetos abandonados, etc.
- Actualmente el sistema trabaja con vídeo diferido y simula el trabajo de las capas inferiores. Para ver como responde el sistema en una situación real se conectará la capa encargada de realizar el análisis de comportamientos con sistemas de *tracking* y segmentación que están siendo desarrollados por otros miembros del grupo de investigación ORETO.
- Para reforzar el proceso de razonamiento e interpretación de comportamientos se establecerán mecanismos para la fusión de información procedente de varios sensores. Cuanta más información posea el sistema de un objeto en movimiento, menor será la incertidumbre y mayor será la probabilidad de realizar una interpretación adecuada.
- Finalmente, se elaborará un módulo de predicción de comportamientos en base a los conceptos definidos en el sistema (que velocidad se prevé para un determinado objeto, que trayectoria podría seguir a continuación, etc). De esta forma, la capa superior encargada de tomar las decisiones no sólo recibirá como entradas el análisis de normalidad y la identificación de situaciones anormales más comunes, sino que también las situaciones que podrían ocurrir.

# Bibliografía

---

- [1] <http://www.cyc.com/>.
- [2] <http://www.ksl.stanford.edu/software/ontolingua/>.
- [3] J.F. Allen. An interval-based representation of temporal knowledge. In *Proceedings 7th IJCAI*, pages 221–226, 1981.
- [4] J.F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.
- [5] J.F. Allen. Time and Time Again: The Many Ways to Represent Time. *International Journal of Intelligent Systems*, 6(4):341–355, 1991.
- [6] J.F. ALLEN and G. FERGUSON. Actions and Events in Interval Temporal Logic. *Journal of Logic and Computation*, 4(5):531–579, 1994.
- [7] J. Annesley, J. Orwell, and J.P. Renno. Evaluation of MPEG7 color descriptors for visual surveillance retrieval. *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 105–112, 2005.
- [8] P. Avis. Surveillance and Canadian maritime domestic security. *Canadian Military Journal*, 4(1):9–15, 2003.
- [9] M. Bachiller, R. Martínez-Tomás, J. Mira, and Rincón M. Niveles de descripción para la interpretación de secuencias de vídeo en tareas de vigilancia. *CMPI-2006*, 2006.
- [10] JL Barron, DJ Fleet, and SS Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.
- [11] A.M. Baumberg. Learning Deformable Models for Tracking Human Motion. *University of Leeds (Tesis Doctoral)*, 1995.
- [12] WP Berriss, WC Price, and MZ Bober. The use of MPEG-7 for intelligent analysis and retrieval in video surveillance. *Intelligence Distributed Surveillance Systems, IEEE Symposium on (Ref. No. 2003/10062)*, page 8, 2003.
- [13] 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.

- [14] J. Black, T. Ellis, and P. Rosin. A novel method for video tracking performance evaluation. *Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, pages 125–132, 2003.
- [15] AF Bobick and AD Wilson. A state-based technique to the representation and recognition of gesture. *IEEE Trans. Pattern Anal. Machine Intell.*, 19:1325–1337, 1997.
- [16] M. Borg, D. Thirde, J. Ferryman, F. Fusier, V. Valentin, F. Bremond, M. Thonnat, O. Team, and I. Sophia-Antipolis. Video Surveillance for Aircraft Activity Monitoring. *Proc. of IEEE Conf. on AVSS*, 1:16–21, 2005.
- [17] J.E. Boyd, J. Meloche, and Y. Vardi. Statistical tracking in video traffic surveillance. *Proc. ICCV99, Corfu Greece, Sept.*, 1:163–168, 1999.
- [18] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. *Computer Vision and Pattern Recognition*, pages 994–999, 1997.
- [19] F. Bremond. *Scene Understanding: perception, multi-sensor fusion, spatio-temporal reasoning and activity recognition*. Tesis doctoral, University of Nice- Sophia Antipolis, 2007.
- [20] F. Bremond and G. Medioni. Scenario recognition in airborne video imagery. *DARPA Image Understanding Workshop 1998*, pages 211–216, 1998.
- [21] T. Brodsky, R.C.E. 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.
- [22] 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.
- [23] H. Buxton. Learning and understanding dynamic scene activity: a review. *Image and Vision Computing*, 21(1):125–136, 2003.
- [24] 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.
- [25] HI Christensen, J. Matas, and J. Kittler. Using grammars for scene interpretation. *Proceedings., International Conference on Image Processing*, vol. 1:793–796, 1996.
- [26] E. Clementini and P. Di Felice. Approximate topological relations. *International Journal of Approximate Reasoning*, 16:173–204, 1997.
- [27] 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.
- [28] A.G. Cohn and B. Bennett. Qualitative spatial representation and reasoning with regions connection calculus. *GeoInformática*, pages 275–316, 1997.

- [29] 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. *Proc. American Nuclear Society (ANS) Eighth International Topical Meeting on Robotics and Remote Systems*, 1999.
- [30] R.T. Collins, A.J. Lipton, H. Fujiyoshi, and T. Kanade. Algorithms for cooperative multisensor surveillance. *Proceedings of the IEEE*, 89(10):1456–1477, 2001.
- [31] R. Cucchiara. Multimedia surveillance systems. *Proceedings of the third ACM international workshop on Video surveillance & sensor networks*, pages 3–10, 2005.
- [32] R. Cutler and L.S. Davis. Robust Real-Time Periodic Motion Detection, Analysis, and Applications. *IEEE Computer Society*, 22(8):781–796, 2000.
- [33] V. De Keyser. Temporal decision making in complex environments. *Philosophical Transactions of the Royal Society of London, B*, 327:569–576, 1990.
- [34] 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.
- [35] F.J. Díez. *Introducción al razonamiento aproximado*. Dpto. Inteligencia Artificial. Razonamiento y Aprendizaje. Ingeniería Técnica de Sistemas de la UNED, 2005.
- [36] I. Düntsch, H. Wang, and S. McCloskey. A relation-algebraic approach to the region-connection calculus. *Theoretical Computer Science*, 255:173–204, 2001.
- [37] 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.
- [38] S. Fejes and L.S. Davis. Detection of independent motion using directional motion estimation. *Computer Vision and Image Understanding*, 74(2):101–120, 1999.
- [39] M. Fernandez, A. Gomez-Perez, and N. Juristo. METHONTOLOGY: From Ontological Art Towards Ontological Engineering. *Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering, Stanford, USA*, pages 33–40, 1997.
- [40] 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.
- [41] J.M. Ferryman, S.J. Maybank, and A.D. Worrall. Visual Surveillance for Moving Vehicles. *International Journal of Computer Vision*, 37(2):187–197, 2000.
- [42] F. Fusier, V. Valentin, F. Brémond, M. Thonnat, M. Borg, D. Thirde, and J. Ferryman. Video understanding for complex activity recognition. *Machine Vision and Applications*, 18(3):167–188, 2007.
- [43] M.P. Gardner. Highway Traffic Monitoring. *Transportation Research Board. Transportation in the New Millennium*, pages 5–10, 2000.
- [44] N. Ghanem, d. DeMenthon, D. Doermann, and L. David. Representation and recognition of events in surveillance video using petri nets. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'04)*, pages 97–118, 2004.

- [45] A. Gomez-Perez. Ontological engineering: A state of the art. *Expert Update*, 2(3):33–43, 1999.
- [46] K.C. Gross, R.M. Singer, S.W. Wegerich, and J.E. Mott. Industrial process surveillance system, January 30 2001. US Patent 6,181,975.
- [47] T.R. Gruber. Ontolingua: A Mechanism to Support Portable Ontologies. *Knowledge Systems Laboratory, Stanford University (Technical Report)*, 1992.
- [48] 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.
- [49] N. Guarino. Formal Ontology and Information Systems. *Proceedings of FOIS*, 98(1998):81–97, 1998.
- [50] M.S. Hacid, C. Declair, and J. Kouloumdjian. A Database Approach for Modeling and Querying Video Data. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, pages 729–750, 2000.
- [51] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: Real-Time Surveillance of People and Their Activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.
- [52] 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.
- [53] M.R. Ingle, B.D. Willians, and N. Sobhi. An advanced traffic management system simulator for intelligent vehicle-highway systems research. *Proceedings of the 1994 Winter Simulation Conference*, pages 1455–1460, 1994.
- [54] Y. Ivanov, C. Stauffer, A. Bobick, and WEL Grimson. Video surveillance of interactions. *Proc. of the CVPR'99 Workshop on Visual Surveillance*, pages 97–118, 1999.
- [55] O. Javed and M. Shah. Tracking and object classification for automated surveillance. *European Conference on Computer Vision*, 4:343–357, 2002.
- [56] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, 14(8):609–615, 1996.
- [57] G.L. Kaempf, G. Klein, M.L. Thorsden, and S. Wolf. Decision Making in Complex Naval Command-and-Control Environments. *Human Factors*, 38(2), 1996.
- [58] IA Karaulova, PM Hall, and AD Marshall. A hierarchical model of dynamics for tracking people with a single video camera. *British Machine Vision Conference*, 1:352–361, 2000.
- [59] 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.
- [60] M. Kilger. A shadow handler in a video-based real-time traffic monitoring system. *Applications of Computer Vision, Proceedings, 1992., IEEE Workshop on*, pages 11–18.

- [61] G. Klein. Decision making in complex military environments. *Fairborn, OH: Klein Associates Inc. Prepared under contract (Technical Report)*, vol. 66001, 1992.
- [62] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. *Pattern Recognition, Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, 4:3776–3781, 1994.
- [63] Jianguang L., L. Qifeng, T. Tieniu, and H. Weiming. Semantic interpretation of object activities in a surveillance system. *Proceedings of the 16th International Conference on Pattern Recognition. ICPR'02*, pages 777–780, 2002.
- [64] D.B. Lenat, RV Guha, K. Pittman, D. Pratt, and M. Shepherd. Cyc: toward programs with common sense. *Communications of the ACM*, 33(8):30–49, 1990.
- [65] L. Liang, G. Haifeng, .L Li, and W. Liang. Semantic event representation and recognition using syntactic attribute graph grammar. *Pattern Recognition Letters*, doi:10.1016/j.patrec.2008.02.023, 2008.
- [66] A.J. Lipton. Local application of optic flow to analyse rigid versus non-rigid motion. *Proc. Int. Conf. Computer Vision Workshop Frame-Rate Vision, Corfu, Greece*, 1999.
- [67] AJ Lipton, H. Fujiyoshi, and RS Patil. Moving target classification and tracking from real-time video. *Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop on*, pages 8–14, 1998.
- [68] B.P.L. Lo, J. Sun, and S.A. Velastin. Fusing visual and audio information in a distributed intelligent surveillance system for public transport systems. *Acta Automatica Sinica*, 29(3):393–407, 2003.
- [69] M. Lopez. Fuzzy management of surveillance sensors. *IEEE Conference on Decision and Control, 37th, Tampa, FL*, pages 245–250, 1998.
- [70] D. Makris and T. Ellis. Path detection in video surveillance. *Image and Vision Computing*, 20(12):895–903, 2002.
- [71] D. Makris and T. Ellis. Automatic learning of an activity-based semantic scene model. *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2003.*, pages 183–188, 2003.
- [72] R. Martínez-Tomás, M. Rincón, M. Bachiller, and Mira. J. On the correspondence between objects and events for the diagnosis of situations in visual surveillance task. *Pattern Recognition Letters*, doi:10.1016/j.patrec.2007.10.020, 2007.
- [73] S.J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. Tracking groups of people. *Computer Vision and Image Understanding*, 80(1):42–56, 2000.
- [74] C. Micheloni, GL Foresti, and L. Snidaro. A cooperative multicamera system for video-surveillance of parking lots. *Intelligence Distributed Surveillance Systems, IEE Symposium on (Ref. No. 2003/10062)*, pages 5–10, 2003.
- [75] 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.



- [76] V.Ñair and J.J. Clark. Automated visual surveillance using hidden markov models. *International Conference on Vision Interface*, pages 88–93, 2002.
- [77] NT. Nguyen, DQ. 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 (CVPR'05)*, 2:955–960, 2005.
- [78] SA. Niyogi and EH. Adelson. Analyzing and recognizing walking figures in XYT. *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 469–474, 1994.
- [79] C.Ñwagboso. User focused surveillance systems integration for intelligent transport systems. *Advanced Video-Based Surveillance Systems, Kluwer Academic Publishers*, pages 8–12, 1999.
- [80] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. *Proc. Computer Vision and Pattern Recognition*, 97:193–199, 1997.
- [81] BB Orten, M. Soysal, and AA Alatan. Person identification in surveillance video by combining MPEG-7 experts. *Signal Processing and Communications Applications Conference, 2005. Proceedings of the IEEE 13th*, pages 352–355, 2005.
- [82] J. Pavón, J. Gómez-Sanz, J.J. Valencia-Jiménez, and A. Fernández-Caballero. Desarrollo de un sistema inteligente de vigilancia multi-sensorial con agentes software. *CMPI-2006*, pages 337–338, 2006.
- [83] 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.
- [84] D. Rawlinson, P. Chakravarty, and R. Jarvis. Distributed Visual Servoing of a Mobile Robot for Surveillance Applications. *Australasian Conference on Robotics and Automation (ACRA)*, 2004.
- [85] P. Remagnino and GA Jones. Classifying Surveillance Events from Attributes and Behaviour. *The Proceeding of the British Machine Vision Conference*, pages 10–13.
- [86] 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.
- [87] K. Rohr. Towards model-based recognition of human movements in image sequences. *CVGIP: Image Understanding*, 59(1):94–115, 1994.
- [88] 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.
- [89] N. Rota and M. Thonnat. Video sequence interpretation for visual surveillance. *Proc. of Third IEEE Int. Workshop on Visual Surveillance*, 2000:59–68, 2000.
- [90] S. Russell and P.Ñorving. *Inteligencia Artificial. Un enfoque moderno*. Pretince Hall, Colección de inteligencia artificial, 1996.

- [91] K. Sato, A. Tsukada, F. Matsuda, K. Kawasaki, and M. Ozaki. Multimedia systems for industrial surveillance. *Proceedings of SPIE*, 3020:182, 1997.
- [92] L. Snidaro, G.L. Foresti, R. Niu, and P.K. Varshney. Sensor Fusion for Video Surveillance. *7th Int. Conf. on Information Fusion*, pages 2049–2074, 2004.
- [93] C. Stauffer and W.E.L. 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.
- [94] J.G. Stell. Part and complement: Fundamental concepts in spatial relations. *Anal. of Mathematics and Artificial Intelligence*, 41(1):1–17, 2004.
- [95] E. Stringa and CS Regazzoni. Content-based retrieval and real time detection from videosequences acquired by surveillance systems. *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, pages 138–142, 1998.
- [96] N. Sumpster and A. Bulpitt. Learning spatio-temporal patterns for predicting object behaviour. *Image and Vision Computing*, 18(9):697–704, 2000.
- [97] K. Takahashi, S. Seki, E. Kojima, and R. Oka. Recognition of dexterous manipulations from time-varying images. *Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on*, pages 23–28, 1994.
- [98] D. Thirde, M. Borg, J. Ferryman, F. Fusier, V. Valentin, F. Bremond, and M. Thonnat. A Real-Time Scene Understanding System for Airport Apron Monitoring. *Proceedings of the Fourth IEEE International Conference on Computer Vision Systems*, 2006.
- [99] H. Tsutsui, J. Miura, and Y. Shirai. Optical flow-based person tracking by multiple cameras. *Multisensor Fusion and Integration for Intelligent Systems, 2001. MFI 2001. International Conference on*, pages 91–96, 2001.
- [100] A. Utsumi, H. Mori, J. Ohya, and M. Yachida. Multiple-view-based tracking of multiple humans. *14th International Conference on Pattern Recognition*, 1:597–609.
- [101] M. Valera and SA Velastin. Intelligent distributed surveillance systems: a review. *Vision, Image and Signal Processing, IEE Proceedings-*, 152(2):192–204, 2005.
- [102] W.H.T.T.L. Wang and S. Maybank. A survey on visual surveillance of object motion and behaviors. *Systems, Man and Cybernetics, Part C, IEEE Transactions on*, 34(3):334–352, 2004.
- [103] G. Welch and G. Bishop. An Introduction to the Kalman Filter. *ACM SIGGRAPH 2001 Course Notes*, 2001.
- [104] A.D. Wilson, A.F. Bobick, and J. Cassell. Temporal classification of natural gesture and application to video coding. *Proc. Comp. Vis. and Pattern Rec.*, pages 948–954, 1997.
- [105] G. Wu, Y. Wu, L. Jiao, Y.F. Wang, and E.Y. Chang. Multi-camera spatio-temporal fusion and biased sequence-data learning for security surveillance. *Proceedings of the eleventh ACM international conference on Multimedia*, pages 528–538, 2003.
- [106] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

- 
- [107] LA 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, IEEE Transactions on [see also Circuits and Systems I: Regular Papers, IEEE Transactions on]*, 46(1):105–119, 1999.
- [108] F.B Zhan. Approximate analysis of binary topological relations between geographics regions with indeterminate boundaries. *Softcomputing*, 2:28–34, 1998.
- [109] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2(27):406–413, 2004.
- [110] L. Zhen. Simulation of pedestrian in computer animation. *Proceedings of the First International Conference on Innovative Computing, Information and Control, ICICIC'06*, 2:229–232, 2006.



# Resumen de las asignaturas cursadas en el Máster

---

## A.1. Metodologías y técnicas de investigación en informática

**Título:** Metodologías y técnicas de investigación en informática.

**Profesorado:** Dra. Marcela Genero Bocco y Dr. Mario Piattini Velthuis.

**Contenido:**

El objetivo principal de la asignatura es enseñar al alumno cómo realizar una tesis doctoral. Este objetivo, a su vez, se divide en cuatro subobjetivos: (1) dotar al alumno de los conocimientos necesarios para estructurar el contenido de la tesis doctoral, (2) ofrecer guías sobre cómo escribir una comunicación científica, (3) cómo realizar revisiones sistemáticas, y finalmente, (4) presentar los métodos de investigación más adecuados para la validación y contrastación de las hipótesis de investigación.

Los métodos de enseñanza aplicados en la asignatura son básicamente tres: clases magistrales impartidas por los profesores citados anteriormente, lectura de artículos científicos y debate entre los alumnos sobre dichos artículos. En cuanto al método de evaluación se tiene en cuenta la asistencia y la participación en clase, así como la entrega de un trabajo, el cual debe ser expuesto por el alumno en un tiempo máximo de doce minutos y tres para preguntas. Dicho trabajo consiste en la realización de una revisión sistemática sobre un tema seleccionado por el alumno, haciendo especial hincapié en los diferentes estudios empíricos.

**Resultados:**

Revisión sistemática sobre algoritmos de aprendizaje inductivos para la generación de

reglas que simulen el comportamiento de un sistema. Dicha revisión sistemática sirvió como punto de partida para la realización de un segundo trabajo en la asignatura *Técnicas de softcomputing*, donde se implementó mediante un sistema web el algoritmo de aprendizaje inductivo publicado en "Learning maximal structure rule in fuzzy logic for knowledge acquisition in expert systems", diseñado por J.L Castro, J.J Castro-Schez y J.M Zurita. El trabajo no sólo se limitaba a la implementación del algoritmo sino que también, se exigía la propuesta de posibles mejoras. Para ello, el estudio de algoritmos similares en la revisión sistemática ha servido de gran ayuda para realizar las propuestas.

## A.2. Sistemas heterogéneos en red

**Título:** Sistemas heterogéneos en red.

**Profesorado:** Dr. Franciso Moya Fernández y Dr. Fernando Rincón Calle.

**Contenido:**

La asignatura se basa, principalmente, en el estudio de los sistemas complejos interconectados, los cuales integran tanto componentes software como hardware. Además, se analizan diferentes niveles de abstracción de las metodologías de diseño de los sistemas en red, considerando aspectos de software, hardware y de comunicación entre componentes.

El método de evaluación empleado en la asignatura valora la asistencia, la participación del alumno en las clases y la realización de un trabajo relacionado con alguno de los temas principales de la asignatura: Interacción en sistemas heterogéneos, Aplicaciones distribuidas en Internet y Sistemas distribuidos empotrados.

**Resultados:**

Implementación de un prototipo de sistema de seguridad multi-agente y distribuido para la interpretación de comportamientos en escenas de tráfico. El prototipo está construido sobre una arquitectura multi-agente orientada a servicios, donde cada tipo de agente perteneciente a la arquitectura tiene una función bien diferenciada. Además, utiliza el middleware ZeroC ICE para ocultar la heterogeneidad de la red y, también, como núcleo de comunicación entre los agentes.

El principal objetivo del sistema es interpretar los comportamientos que vehículos y peatones tienen en un entorno virtual y simulado. Dicho objetivo es alcanzado con la colaboración de cuatro tipos de agentes inteligentes: (a) SimulationProxy, agente encargado de generar una nueva simulación en el escenario. Este agente sustituye o simula a la parte del sistema encargada de capturar el vídeo, audio y cualquier otro tipo de señal. (b) Formalizer, es un tipo de agente que se encarga de identificar los distintos elementos que conforman la escena a partir de los datos generados por un agente de tipo SimulationProxy. Para ello, el agente hace uso del conocimiento especificado en una ontología de dominio. (c) Analyzer, este tipo de agentes tiene como misión principal razonar sobre la normalidad o anormalidad de una situación, es decir, es el de encargado de determinar si lo que está ocurriendo en el momento actual es normal o, por el contrario, es anormal y debe ser castigado. (d) DecisionManager, el último tipo de agente, se encarga de tomar las decisiones pertinentes a partir de los resultados obtenidos tras el proceso de interpretación y razonamiento.

### A.3. Sistemas distribuidos avanzados: grid e inteligencia ambiental

**Título:** Sistemas distribuidos avanzados: grid e inteligencia ambiental.

**Profesorado:** Dra. Camelia Muñoz Caro, Dr. Alfonso Niño Ramos y Dr. José Bravo Rodríguez.

**Contenido:** La asignatura se divide en dos partes bien diferenciadas. Una primera parte titulada Sistemas Grid, impartida por los profesores Camelia Muñoz y Alfonso Niño. Y una segunda parte titulada inteligencia ambiental impartida por el profesor José Bravo. La primera de ellas tiene como gran objetivo dotar al alumno de los conocimientos necesarios para configurar y gestionar infraestructuras grid, así como diseñar sistemas grid a nivel de servicios básicos y servicios web. Respecto a la segunda parte, se estudian nuevas formas de interacción en donde se manifiesta la proactividad de los sistemas. Además, se pretende capacitar al alumno para el modelado de acciones de usuario en sistemas Aml, realizándose estudios reales sobre instalaciones en diferentes contextos, entre los que cabe destacar los centros de enseñanza y hospitalarios.

El método de evaluación de la asignatura se basa en la valoración de la asistencia a clase, participación en las actividades propuestas y, finalmente, en la entrega y defensa de dos trabajos teóricos.

**Resultados:**

Para superar la asignatura se realizaron y defendieron dos trabajos teóricos:

- *Sistemas grid y gestión del conocimiento.* El trabajo está enfocado en la búsqueda de información relacionada con los sistemas grid encargados de la gestión y extracción de conocimiento a partir de los datos almacenados en repositorios distribuidos geográficamente.
- *Programación en sistemas grid: metodologías, paradigmas, herramientas y entornos.* En este trabajo, como su propio nombre indica, se estudia en profundidad algunos aspectos claves en la programación grid como, por ejemplo, la gestión de procesos, gestión de recursos, sincronización y gestión de dependencias, etc. El trabajo finaliza con un análisis de las herramientas, lenguajes y entornos de programación que existen actualmente y están destinados para dicho fin.

### A.4. Cognición y colaboración

**Título:** Cognición y Colaboración.

**Profesorado:** Dr. José Ángel Olivas Varela, Dr. Manuel Prieto Méndez y Dra. Carmen Lacave Rodero.

**Contenido:** La asignatura está estructurada en tres bloques temáticos bien diferenciados, cada uno de ellos impartido por uno de los profesores de la asignatura.

- Bloque I: Gestión del conocimiento y recuperación eficiente de la información en la Web mediante técnicas de Soft Computing (impartido por el Dr. José Ángel Olivas).

El principal objetivo de este primer bloque es enseñar al alumno las principales técnicas basadas en Soft-Computing (tolerantes a la imprecisión e incertidumbre) para manipular información y extraer conocimiento de diversas fuentes.

- Bloque II: Gestión del conocimiento y computación gráfica (impartido por la Dra. Carmen Lacave Rodero). En este bloque se realiza una introducción al razonamiento probabilístico, haciendo mayor hincapié en el razonamiento con Redes Bayesianas.
- Bloque III: Gestión del conocimiento en grupo y aprendizaje comparado (impartido por el Dr. Manuel Prieto Méndez). El principal objetivo de este bloque es conocer las taxonomías y principios de funcionamiento de los sistemas de trabajo en grupo: análisis de los modelos CSCW, patrones para trabajo en grupo, aprendizaje automático y aprendizaje humano, etc.

Para superar el curso, el alumno debe realizar un trabajo encuadrado en alguno de los tres bloques anteriores.

**Resultados:** Se realizó un trabajo teórico-práctico encuadrado en el Bloque II: Gestión del conocimiento y computación gráfico, el cual fue supervisado y valorado por la Dra. Carmen Lacave Rodero. Este trabajo se divide en dos partes:

- Un trabajo teórico titulado “Aspectos generales del aprendizaje automático de modelos gráficos y aplicaciones en la clasificación supervisada”. El principal objetivo del trabajo es la búsqueda bibliográfica, el estudio y análisis de las diferentes alternativas que existen en el campo del aprendizaje automático de los modelos gráficos y sus aplicaciones en la clasificación supervisada.
- Implementación de una aplicación web, cuya función principal es la representación gráfica de Redes Bayesianas diseñadas con el entorno “Elvira”, además de servir como repositorio distribuido de las redes creadas por distintos usuarios.

## A.5. Procesos software y gestión del conocimiento

**Título:** Procesos software y gestión del conocimiento.

**Profesorado:** Dr. Francisco Ruíz, Dr. Félix Oscar García y Dra. Aurora Vizcaíno.

**Contenido:** El programa docente de la asignatura está dividido en cinco temas: Metamodelos y ontologías. Gestión del conocimiento y mejora de procesos, procesos software, MDE Ingeniería dirigida por modelos y diferentes casos de estudio: procesos de medición, mantenimiento del software, procesos de negocio y reingeniería.

Los métodos de enseñanza aplicados en la asignatura son bien numerosos: clases magistrales, clases de debate y dudas, tutorías, seminarios y conferencias, examen y estudio personal de los alumnos. Cabe destacar que, especialmente en esta asignatura, se fomenta y valora en gran medida la participación del alumno en clase. En cuanto a la evaluación, en la página oficial y el programa de la asignatura se muestra una tabla donde se detallan las diferentes actividades y puntuaciones correspondiente.

**Resultados:**

Además de la lectura de artículos científicos y la participación en los debates de clase, se realizaron y defendieron los siguientes trabajos.

- Estudio detallado de la metodología REFSENO. Representación formal de ontologías en ingeniería del software mediante tablas, texto y opcionalmente con diagramas.
- Fábricas de experiencia y su uso en la Ingeniería del Software. Transformación de la experiencia obtenida en conocimiento útil para cualquier organización.
- Principales funcionalidades de los sistemas para la gestión de procesos de negocio (BMPS). La gestión de los procesos de negocio permite a cualquier organización conocer qué es lo que hace la empresa y cómo lo hace. Además, permite gestionar de forma eficiente los distintos flujos de información, los materiales de los que dispone la empresa y sus recursos humanos.

## **A.6. Técnicas de softcomputing**

**Título:** Técnicas de softcomputing.

**Profesorado:** Dr. Luis Jiménez Linares, Dr. José Jesús Castro Sánchez y Dr. Juan Moreno García.

**Contenido:** En esta asignatura se presenta al alumno una nueva perspectiva para la definición de sistemas complejos mediante sistemas difusos, diferente a los métodos clásicos aprendidos durante la ingeniería en informática. Además, se pretende que el alumno obtenga cierta destreza en el uso y manejo de sistemas de reglas difusas, así como en el manejo de distintas metodologías para la definición de sistemas difusos. Para superar la asignatura se valora positivamente la asistencia a clase, así como la realización y la defensa de dos trabajos teórico-prácticos encuadrados en alguno de los temas principales enumerados anteriormente.

**Resultados:**

Se realizaron dos trabajos:

- Un trabajo teórico titulado “Tratamiento de la incertidumbre: MYCIN y el modelo de los factores de certeza”. El principal objetivo de este trabajo es la búsqueda de información y el estudio detallado del modelo de los factores de certeza para el tratamiento de la incertidumbre.
- Trabajo teórico-práctico: estudio detallado del algoritmo de aprendizaje inductivo presentado en el artículo “Learning Maximal Structure Rules in Fuzzy Logic for Knowledge Acquisition in Expert Systems”. Además del estudio de dicho algoritmo se propusieron diferentes modificaciones para mejorar los resultados que ofrecía el algoritmo original. Tanto las modificaciones como el algoritmo original fueron implementados mediante una aplicación web. A raíz de este trabajo se han realizado dos publicaciones: “Learning maximal structure rules with pruning based on distances between fuzzy sets”(IPMU’08) y “Aprendizaje de reglas difusas para la clasificación de comportamientos en un sistema de videovigilancia cognitiva” (ESTYL’08).





# **Currículum Vitae**

---

## **DATOS PERSONALES**

- APELLIDOS: Albusac Jiménez
- NOMBRE: Javier Alonso
- DNI: 5687900-T
- FECHA DE NACIMIENTO: 20/06/80
- DIRECCIÓN PARTICULAR: c/Libertad 59A, 1ºB, Miguelturra (Ciudad Real).
- TELÉFONO: 617342824

## **SITUACIÓN PROFESIONAL ACTUAL**

- Contratado por la Universidad de Castilla-La Mancha en el proyecto de investigación “HESPERIA: Homeland Security: tecnologías para la seguridad integral en espacios públicos e infraestructuras”, Proyecto CENIT (CDTI, mityc).

## **FORMACIÓN ACADÉMICA**

- Ingeniero en Informática por la Universidad de Castilla-La Mancha desde Diciembre de 2005.
- Actualmente cursando el segundo año del Máster en Tecnologías Informáticas Avanzadas de la UCLM.

## **BECAS DISFRUTADAS**

- Beca para la colaboración y desarrollo del sistema web de las Escuela Superior de Informática (Ciudad Real), desde 01/07/2004 hasta 20/12/2004
- Prácticas de empresa en VARNET (Ciudad Real) desde 1/11/2004 hasta 30/07/2005.
- Beca de Investigación del Vicerrectorado de la UCLM Ref. 2005-BCL-661-001 para el diseño y desarrollo de una visita virtual al Hospital General de Ciudad Real. Desde 01/10/2005 hasta 30/02/2006.

### **CURSOS DE FORMACIÓN DOCENTE Y OTROS CURSOS RECIBIDOS**

- Sistemas de Radiocomunicaciones (Junio 2000, UCLM).
- Tecnología de bases de datos post-relacionales (Diciembre 2002, UCLM).
- Introducción al diseño de aplicaciones informáticas interactivas con Macromedia Director y Macromedia Flash (Febrero 2003, UCLM).
- Desarrollo de componentes Enterprise JavaBeans (Marzo 2003, UCLM).
- Programación de aplicaciones web con Python Zope (Enero 2004, UCLM).
- Programación con Visual Basic .NET (Noviembre 2004, UCLM).

### **DOCENCIA EN TITULACIONES UNIVERSITARIAS DE 1ER, 2º Y 3ER CICLO**

- Profesor Asociado de la asignatura Fundamentos de Informática en la Escuela Universitaria Politécnica de Almadén, desde Noviembre hasta Febrero del año académico 2007/2008.
- Profesor del Curso de Enseñanzas Propias “Diseño gráfico con software libre”, en la Escuela Superior de Informática de Ciudad Real (Noviembre 2007).

### **PUBLICACIONES**

#### **LIBROS Y REVISTAS**

- Visión Estereoscópica. Javier Albusac Jiménez, Miguel García Corchero. Capítulo del libro Tecnologías libres para la síntesis de imagen digital con ISBN 84-689-9280-1.
- 3D Distributed Rendering and Optimization using Free Software. Carlos González Morcillo , Gerhard Weiss, David Vallejo Fernández, Luis Jiménez Linares, Javier Albusac Jiménez. Upgrade, Vol. VIII, Issue 6, Diciembre 2007.
- Optimización del proceso de render 3D distribuido con software libre. Carlos González Morcillo , Gerhard Weiss, David Vallejo Fernández, Luis Jiménez Linares, Javier Albusac Jiménez. Novática, N° 190, Noviembre-Diciembre 2007.
- A MultiAgent System for Physically based Rendering Optimization. Glez-Morcillo C., Weiss G., Jiménez L., Vallejo D., Albusac J.A. 4676/2007: 149-163. L.N. in Computer Science

## ARTÍCULOS EN CONGRESOS Y PONENCIAS

- Babel: A collaborative language learning system. Javier Albusac, Carlos González Morcillo y Luis Jiménez Linares. Floss International Conference, Jerez de la Frontera. pp 27-37, Marzo 2007. (PONENTE)
- An approximate reasoning technique for segmentation on compressed mpeg video. L. Rodríguez-Benítez, Juan Moreno-García y Javier Albusac. The International Joint Conference on Computer Vision and Computer Graphics, Theory and Applications. pp 184-191. Barcelona, Marzo 2007.
- Babel: Plataforma e-learning para el aprendizaje colaborativo de idiomas. Javier Albusac, Carlos González Morcillo y Luis Jiménez Linares. XVI Congreso Universitario de Innovación Educativa en Enseñanzas Técnicas (CUIEET'07). Valladolid, Julio 2007. (PONENTE)
- A MultiAgent System for Physically based Rendering Optimization. Carlos González Morcillo, Luis Jiménez Linares, David Vallejo, Javier Albusac. Cooperative Information Agents (CIA 2007). Holanda, Septiembre 2007.
- Seguimiento de objetos representados lingüísticamente utilizando técnicas de razonamiento aproximado. L. Rodríguez-Benítez, Juan Moreno-García, Javier Albusac, J.J Castro-Schez. CEDI 2007. Zaragoza, Septiembre 2007. (PONENTE)
- Learning Maximal Structure Rule with pruning based on distances between fuzzy sets. Javier Albusac, J.J Castro-Schez, David Vallejo. IPMU '08. Málaga, Junio 2008. (PONENTE)
- Aprendizaje de reglas difusas para la clasificación de comportamientos en un sistemas de videovigilancia cognitiva. Javier Albusac, J.J Castro-Schez, David Vallejo, L. Jiménez-Linares. Estylf'08. Mieres, Septiembre 2008. (PONENTE)
- Integrating a standar communication protocol into an e-commerce environment based on intelligent agents. D. Vallejo, J.J Castro-Schez, J. Albusac, C. Glez-Morcillo. International Conference on Enterprise Information Systems, Junio 2008, Barcelona (España).

**EXPERIENCIA LABORAL / MÉRITOS ASISTENCIALES**

- Programador Junior en la empresa VARNET de Ciudad Real. Desde 01/11/04 hasta 30/07/05.
- Ingeniero contratado en el proyecto de Investigación TIC2003-08807-C02-02 (UCLM), Dimoclust: métodos de softcomputing para el análisis y categorización de series de datos. Desde 01/07/06- 30/11/06
- Profesor asociado de la asignatura Fundamentos de Informática en la Escuela Politécnica de Almadén (UCLM), desde Noviembre hasta Febrero del año académico 2007/2008.
- Ingeniero contratado en el Proyecto de Investigación Hesperia: Homeland Security: tecnologías para la seguridad integral en espacios públicos e infraestructuras (UCLM). Desde 01/12/2006 hasta la fecha actual.