Universidad de Valladolid

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

DEPARTAMENTO DE TEORÍA DE LA SEÑAL Y COMUNICACIONES E INGENIERÍA TELEMÁTICA

TESIS DOCTORAL:

CARACTERIZACIÓN SEMÁNTICA DE ESPACIOS. SISTEMA DE VIDEOVIGILANCIA INTELIGENTE EN SMART CITIES

Presentada por Lorena Calavia Domínguez para optar al grado de doctora por la Universidad de Valladolid

Dirigida por:
Dra. Belén Carro Martínez
Dr. Antonio Javier Sánchez Esguevillas
Dr. Javier Manuel Aguiar Pérez

"The touch of your hand says
you'll catch me whenever I fall".

(Paul Overstreet & Don Schlitz)

A todos los que creyeron en mí y
fomentaron mi deseo de superación.
Gracias por vuestro apoyo,
comprensión y consejos.

“Logic will get you from A to B.
Imagination will take you everywhere.”

(Albert Einstein)

RESUMEN

El objetivo fundamental de las técnicas de visión artificial o por ordenador es que de forma autónoma se pueda caracterizar una escena identificando los objetos que hay en ella y sus comportamientos. Algunas de las principales líneas de interés de esta tecnología son el seguimiento de caras, control de procesos industriales, robótica o sistemas de videovigilancia. En estos últimos, la visión artificial se está afianzando como una metodología imprescindible.

En los sistemas tradicionales de videovigilancia un operador humano es el encargado de la interpretación de la escena y de realizar las acciones necesarias cuando se identifica una alarma. Pero con el paso del tiempo, se va reduciendo la capacidad de observación del personal, pudiendo pasar inadvertidas situaciones potenciales de riesgo.

En esta Tesis Doctoral, realizado dentro del proyecto europeo CELTIC HuSIMS (*Human Situation Monitoring System*), se presenta una nueva metodología inteligente para la caracterización de escenarios, aplicable a videovigilancia, capaz de detectar e identificar, de forma automática, situaciones anómalas analizando el movimiento de los objetos. El sistema está diseñado para reducir al mínimo el procesamiento y la transmisión de vídeo, lo que permite el despliegue de un gran número de cámaras (pequeñas y baratas) y sensores, y por lo tanto adecuada para *Smart Cities*.

El enfoque seguido se basa en un esquema de procesamiento de tres etapas. Primero, la detección de objetos en movimiento en las propias cámaras, utilizando algorítmica sencilla, para evitar el envío de datos de vídeo. En segundo lugar, la construcción, de forma automática, de un modelo de las diferentes zonas de las escenas captadas utilizando los parámetros de movimiento identificados por las cámaras. Y tercero, la realización de razonado semántico sobre el modelo de rutas y los parámetros de movimiento de los objetos de la escena actual para identificar

las alarmas a nivel conceptual, es decir, no sólo la detección de que un evento inusual está ocurriendo, sino también, la identificación de la naturaleza de ese evento.

Para verificar la validez de la metodología presentada se ha realizado una implementación de la misma y se ha aplicado a diferentes escenarios, tanto sintéticos como reales, demostrando su viabilidad.

ABSTRACT

The main objective of Computer Vision technologies is to allow the autonomous characterization of scenes by identification of objects, actions, events and behaviors from the video signal. Some of the achievements and research lines in this field include face recognition, industrial process control, robot sensing/guidance and automatic videosurveillance. Specifically, in this latter field of videosurveillance, computer vision has become a cornerstone in order to push the implantation of new paradigms.

In traditional videosurveillance systems, a human operator is in charge of the interpretation of the scene, and carrying out the necessary response actions when an alarm or specific situation is identified. But this approach presents several problems, including the inherent limits of the human mind regarding the number of simultaneous scenes analyzed or the fatigue that easily appears in the operator after some time has passed.

This Doctoral Dissertation, which has been carried out inside the European EUREKA-CELTIC Project HuSIMS, presents a new methodology for intelligent scene characterization to be applied in videosurveillance systems. It allows autonomous detection and identification of anomalous situations by analyzing motion parameters of the objects in the video signal only, and not the whole video signal itself. This innovative approach operates over a much more reduced amount of information, allowing great savings in memory, bandwidth and processing requirements, therefore becoming especially suitable for deployments of dense videosurveillance systems, with many small and cheap cameras which do not implement complex video processing algorithms but only moving object identification and tracking. It is worth mentioning that this kind of system is envisioned for integral videosurveillance of smart cities, which will gain a lot of importance in the next few years.

The proposed system employs a three-stage architecture. The first one is the detection of moving objects by the terminal sensor cameras, employing simple algorithms, which allows transmission of a reduced set of parameters instead of a heavy video signal while at the same time keeps under control the complexity of the video analysis algorithms in the camera by not performing any difficult and resource-greedy task like object identification. The second stage automatically learns and builds a route model of the scenes under surveillance by accumulating and processing the history of motion parameters received from the cameras. And the third stage applies real time semantic reasoning over the route model and the motion parameters in order to identify alarms at a conceptual level, that is, rich information about the nature of the situation causing the alarm is provided automatically together with the alarm itself.

In order to validate the proposed methodology, it has been implemented and applied to different operation scenarios, both synthetic environments in the lab and real deployments in the real world.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1

INTRODUCCIÓN

Los sistemas de vigilancia actuales no han ido evolucionando a la par que los nuevos avances tecnológicos, basándose, en su mayoría, en la instalación de dispositivos de alto coste que realizan grabación de imágenes y video. Esto hace que, a pesar de ser un mercado en crecimiento, no se satisfagan las exigencias que los consumidores demandan. La mayoría de ellos son circuitos cerrados de televisión (CCTV) en los que el tratamiento de los datos captados se realiza por personal de seguridad. Este planteamiento provoca rechazo social y que sea necesario adaptar los sistemas para el cumplimiento de las leyes de protección de datos y de privacidad para proteger los derechos de los ciudadanos.

Por otra parte, el precio de los aparatos de adquisición de imagen limita el alcance de las zonas a cubrir (la vigilancia, incluso de una pequeña zona, requiere una instalación de varias cámaras para monitorizarla desde distintos ángulos y evitar zonas muertas, y por tanto un presupuesto elevado, no asumible en muchos casos por los interesados). Además es necesario contratar personal humano que supervise en tiempo real las escenas procedentes de las cámaras. Pero existe una limitación en el número de cámaras a las que un operador humano es capaz de prestar

atención simultáneamente y el cansancio, tras un tiempo de trabajo, afecta a la eficacia de las apreciaciones [1]. Esto hace que un aumento del número de cámaras requiera un incremento proporcional del personal contratado, con el consiguiente aumento en el presupuesto, lo que hace inviable estas soluciones.

El propósito de la visión artificial es "entender" qué está pasando en la escena, tanto actividades como comportamientos. Una aplicación directa de esta caracterización de escenas es la videovigilancia. Si se consigue conocer de forma automatizada los diferentes escenarios y observar las actividades típicas se pueden determinar comportamientos anómalos sin ser necesaria la grabación u observación de imágenes por personal autorizado. Estos mecanismos determinan de forma autónoma que existe alguna anomalía en la zona vigilada pero su valor añadido es que son capaces de concretar qué está pasando. Esta identificación de acontecimientos hace que se decida, en cada momento, si es necesario avisar a los servicios de emergencia. En caso de precisarse actuaciones, los sistemas de seguridad pueden planificarla de antemano mientras se desplazan hacia el lugar del incidente ya que conocen a priori la situación que se van a encontrar. La utilización de estos nuevos mecanismos en videovigilancia constituye una alternativa viable y eficaz.

En las *Smart Cities*, también conocidas como Ciudades Inteligentes[1], se dispone de gran cantidad de información del entorno, por lo que es muy interesante la aplicación de técnicas de caracterización de escenarios a los diferentes dominios y la detección de eventos y determinadas situaciones. No obstante, la identificación de escenas requiere un análisis e interpretación de gran cantidad de datos en tiempo real. Parte de estos procederá de imágenes de cámaras de seguridad. Se tratarán imágenes de baja resolución para no coartar la privacidad de los usuarios. Por tanto, se reducirá la cantidad de datos a enviar y el precio de los dispositivos. Sin embargo, para este tipo de imágenes, los algoritmos de visión artificial están muy limitados. Esta información se complementará en algunos casos con la procedente de otros sensores para conseguir caracterizaciones más precisas. En este contexto, se hace necesario el desarrollo de innovadoras metodologías de procesado de datos y aprendizaje que se adapten a las necesidades de los usuarios actuales.

[1] Podemos considerar una ciudad como "inteligente" cuando las inversiones en capital humano y social, y en infraestructuras de comunicación tradicionales (transporte) y modernas (ICT), fomentan un desarrollo económico sostenible y una elevada calidad de vida, con una sabia gestión de los recursos naturales, a través de un gobierno participativo (A. Caragliu).

Este Capítulo se encuentra estructurado de la siguiente manera. Las motivaciones que han llevado a la realización de la Tesis Doctoral se introducen en la Sección 1.1. La Sección 1.2 incluye los objetivos fundamentales del trabajo realizado. Las principales contribuciones originales aportadas por la Tesis se presentan en la Sección 1.3. Finalmente, en la Sección 1.4 se detalla la organización de esta memoria.

1.1 Motivación

Las principales motivaciones que han llevado al desarrollo de esta Tesis Doctoral son las siguientes:

- Uso de los patrones de movimiento de los objetos para su identificación como alternativa a algoritmos de visión artificial.

  Para la identificación de objetos en imágenes se utilizan habitualmente técnicas y algoritmos de visión artificial. La metodología que siguen es la siguiente:

  - Realización de un preprocesado de la imagen.
  - Determinación de las diferentes entidades que aparecen en el fotograma.
  - Parametrización de los objetos para la extracción de los rasgos básicos de los mismos.
  - Asignación de una etiqueta al objeto en función de los descriptores y clasificación del mismo dentro de uno de los grupos de entidades predefinidos.

  Para realizar todo este procesado y evitar falsos positivos es necesario disponer de imágenes de alta calidad. Factores como la iluminación y el entorno influyen en el proceso de detección de este tipo de sistemas. Las cámaras deben contar con visores de alta resolución y capacidad de procesado, o bien, se transmiten las señales de video a un centro de control. En este último caso se utiliza un enorme ancho de banda, pero es el centro de control el que cuenta con la potencia necesaria para llevar a cabo el análisis.

Es crítico el uso de alternativas que realicen reconocimiento e identificación de objetos en imágenes de baja resolución dadas las limitaciones de los algoritmos de visión artificial. Estas soluciones deberán apartarse de la utilización de algoritmos de reconocimiento de objetos para basarse en una caracterización de los mismos en función de otros parámetros, como pueden ser sus patrones de movimiento. Propiedades como la velocidad y las dimensiones, fácilmente extraíbles incluso en este tipo de imágenes, servirán como complemento del movimiento para facilitar las identificaciones.

- Caracterización semántica de situaciones en diferentes dominios de las *Smart Cities.*

  Tan importante como determinar que se está produciendo una situación extraña es concretar qué es lo que está sucediendo y, si además se puede detallar utilizando el lenguaje formal, se facilita la comprensión para los operadores humanos. El sistema hace posible que los usuarios conozcan las alarmas generadas para poder gestionar más fácilmente las situaciones de crisis.

  Es necesario que las escenas y situaciones se expresen en términos semánticos para que el sistema sea capaz de identificar contextos con precisión. El uso de ontologías, combinado con el procesamiento en un motor de inferencia semántico, permite caracterizar de forma precisa los diferentes escenarios. Esta definición de la escena se basa en los datos que llegan de la red de sensores de la *Smart City*, combinados con reglas de comportamiento. Aplicando un razonador se consigue extraer información implícita en los datos originales de la que no se disponía a priori. Durante un periodo de aprendizaje se determinan las conductas usuales de los diferentes objetos en las diferentes zonas. Conociendo los comportamientos habituales dentro de cada escenario la detección de situaciones anómalas es inmediata. Esto hace que una de las aplicaciones inmediatas de la caracterización de escenarios sea la videovigilancia.

  Es preciso el diseño y utilización de ontologías específicas y adaptadas a cada dominio de aplicación. Además, en ciertos casos es imprescindible incorporar reglas adicionales que permiten una inferencia más completa de los datos. Como ventaja indicar que la combinación de ontologías es muy sencilla. Esta propiedad hace que se puedan determinar de forma semántica situaciones complejas empleando la combinación de varias ontologías

simples. Esta particularidad permite, además, la aplicación del sistema a diferentes campos dentro de una misma escena. Por ejemplo, se puede combinar un modelo que defina escenas en el dominio de la gestión del tráfico urbano con una ontología para el control de incendios. Esta Tesis Doctoral se valida en diferentes entornos para corroborar su versatilidad.

- Aumento de la privacidad de los individuos protegidos y disminución de los costes de videovigilancia de zonas extensas mediante la gestión automática de los eventos.

  Hay cada vez mayor tendencia a la instalación de sistemas de videovigilancia en los más diversos ámbitos (edificios públicos, estadios o instalaciones deportivas, transporte, gestión del tráfico, etc.) y con diferentes fines (la protección de la propiedad, detección de delitos, etc.).

  Hoy en día existen ya sistemas de videovigilancia desplegados en muchas ciudades del mundo. La mayoría de ellos se basan en el envío de imágenes o video a un centro de control donde son observadas por personal humano que se encarga de detectar las anomalías. La instalación de este tipo de sistemas de control hace que sea necesaria la protección de los derechos individuales de las personas observadas. La información procedente de las cámaras de seguridad, como puede aportar detalles sobre las personas y hacerla identificable, se trata como datos personales, y está regulada de forma general por diversos instrumentos jurídicos internacionales. Además, para videovigilancia existe una normativa específica, tanto a nivel nacional (Ley Orgánica 4/1997, de 4 de agosto de Protección de Datos Personales) como comunitario (la Directiva 95/46/CE, del Parlamento Europeo y del Consejo), ya que se realiza un uso y tratamiento de los datos personales obtenidos a través de las videocámaras.

  Aumentar la privacidad de la gente protegida (y no observada) es una de las principales motivaciones. Es imprescindible la implementación de una solución que gestione automáticamente los eventos que se generan sin ser necesario examinar la información gráfica. Las imágenes a tratar son de baja resolución para evitar la identificación de los individuos, lo que hace que no se vulneren los derechos y libertades de las personas. Sin embargo, esta disminución en la calidad no reduce la seguridad tradicionalmente conseguida con la instalación de las videocámaras de alta resolución.

Por otro lado, los sistemas tradicionales, como se ha comentado anteriormente, requieren personal humano, que constantemente y de manera manual, deben visualizar monitores identificando situaciones de riesgo. Ésto hace que si se quiere aumentar la zona a vigilar, además de ser necesaria la instalación de nuevas cámaras de alta resolución, también es imprescindible contratar personal que supervise las imágenes procedentes de dichas cámaras con el consecuente incremento del coste.

Por tanto, es necesario que el sistema permita, a partir de una imagen de baja resolución, analizar continuamente una escena compleja de la que extraer información relevante. En un segundo análisis se aplican mecanismos que permiten generar alarmas en caso de detección de situaciones de riesgo. Además, maneja gran cantidad de datos, estudiando un alto número de posibles alarmas y generando información en tiempo real entendible por los usuarios finales. Se consigue que el sistema funcione de un modo simple y con un bajo coste ya que la aplicación es capaz de identificar de forma autónoma situaciones peligrosas o irregulares, abriendo la puerta al despliegue de mecanismos de vigilancia altamente eficientes a gran escala, cubriendo, por ejemplo, ciudades enteras.

1.2 Objetivos

El principal objetivo de esta Tesis Doctoral es el estudio y aplicación de innovadoras metodologías basadas en semántica, para la identificación de las diferentes entidades y regiones de una escena basándose en el movimiento de los objetos que aparecen en una imagen, determinando, además, comportamientos anómalos dentro de diferentes dominios.

Con este fin se pueden enumerar los siguientes objetivos específicos:

- Análisis y adaptación de las metodologías para la determinación de patrones de desplazamiento en objetos en movimiento en función de la evolución de sus propiedades en el tiempo.
- Desarrollo de ontologías que permitan, en función de los datos de los objetos de las escenas actuales y anteriores, describir la escena.
- Diseño de ontologías innovadoras que determinen si se está produciendo en ese momento una situación especial e identificarla en lenguaje entendible por un operador humano.

- Propuesta e implementación de una arquitectura que incorpore mecanismos de detección y caracterización de objetos en movimiento y sus rutas, y a partir de estos datos, sea capaz de definir qué está sucediendo. La arquitectura debe ser fácilmente escalable y adaptable a nuevas situaciones y ámbitos de aplicación.
- Validación del sistema implementado, efectuando pruebas de la arquitectura sobre diferentes videos, para determinar si se han cumplido los objetivos planteados.

1.3 Contribuciones de la Tesis

Esta Tesis Doctoral incluirá las siguientes aportaciones originales:

- **Desarrollo de una metodología para la caracterización automatizada de escenarios, puramente semántica, utilizando las ontologías y reglas diseñadas.**

  Para su elaboración se van a considerar los patrones de movimiento y las características de los objetos (complementada, si se dispone, con la procedente de otros sensores). Para ello se procesará semánticamente su comportamiento tanto del momento actual como en momentos previos. De este proceso de inferencia se obtendrá la información necesaria para poder conseguir la definición de escenas. Se conocerán los objetos concretos que en un momento determinado aparecen en la imagen y su localización dentro de la misma, pero no sólo espacial, sino dentro del propio contexto. En el caso del control de tráfico, por ejemplo, una ontología podrá inferir que en la imagen existe un objeto peatón que se encuentra en una zona clasificada como un paso de peatones.

  Tras la traducción de la información de movimiento de los objetos al dominio semántico, el razonamiento permite la identificación de los mismos y la caracterización de la escena por lo que todo el procedimiento es puramente semántico. En la literatura los autores realizan la identificación de objetos utilizando otras metodologías, y posteriormente usan las tecnologías semánticas sólo para representar la información.

- Utilización de ontologías persistentes para el modelado de escenarios.

  Las metodologías semánticas existentes basan el conocimiento de la escena en las características de los objetos en un momento específico. Esto es debido a que utilizan ontologías que modelan instancias concretas de objetos para momentos puntuales y el proceso de inferencia proporciona en ocasiones resultados no esperados. Sin embargo, la utilización de persistencia aporta mejores resultados en el modelado de la escena ya que los procesos de inferencia incluyen, además de la información actual de la escena, históricos con las características previas de los objetos y conclusiones obtenidas tras razonados anteriores.

- Diseño e implementación de un sistema integrado que a partir de imágenes, incluso de baja resolución, indique, en un lenguaje entendible por un operador humano, la situación de alarma.

  Diseño y desarrollo de un sistema integral que sea capaz de:

  - Clasificar, utilizando semántica, los objetos en movimiento que en cada momento aparecen en la escena. Para la realización de esta identificación se dispondrá de los parámetros y características de los mismos que se han obtenido mediante un procesado a nivel de píxel que permite el tratamiento de imágenes incluso de baja resolución, que son las que imponen mayores limitaciones. En este sentido indicar, que los sistemas actuales reconocen los objetos sobre todo basándose en su forma y comparándola con modelos previamente establecidos, mecanismo que no es válido para imágenes con baja resolución, entornos con poca luz, etc.

  - Caracterizar situaciones y así conocer en todo momento, utilizando la semántica, qué es lo que está sucediendo en la escena. Además se conseguirá, en una situación anómala, concretar, en lenguaje formal, qué es lo que está pasando de manera automática y sin necesidad de intervención de operadores humanos.

  - Adaptarse, de forma sencilla, a diferentes dominios. Simplemente combinando ontologías o cambiándolas por las que definen otros ámbitos de conocimiento, se caracterizarán escenarios e identificarán anomalías diferentes, convirtiendo así la solución en versátil y escalable.

1.4 Estructura de la Tesis

La memoria de la Tesis Doctoral se estructura en ocho Capítulos tal y como se detalla a continuación:

- El Capítulo 1 se corresponde con la introducción e incluye la motivación, objetivos y principales contribuciones de la Tesis, así como su estructura.
- El Capítulo 2 analiza los antecedentes de mecanismos de caracterización de imágenes de video y de sistemas de videovigilancia prestando especial atención en aquellos basados en el movimiento.
- El Capítulo 3 estudia y define los requisitos del sistema que se va a implementar para realizar un diseño a alto nivel de la arquitectura.
- El Capítulo 4 describe la evolución de los sistemas de videovigilancia para determinar la importancia del procesado de imagen consiguiendo detecciones autónomas, exponiendo los sensores visuales inteligentes y las tecnologías de comunicación utilizadas dentro del proyecto en el que se enmarca el desarrollo de esta Tesis Doctoral.
- El Capítulo 5 detalla el sistema de detección de rutas de los objetos en función de los parámetros de movimiento obtenidos por los sensores visuales inteligentes.
- El Capítulo 6 analiza y expone las metodologías innovadoras de caracterización semántica de espacios implementadas, considerando que se van a utilizar para ello los patrones de movimiento y características de los objetos existentes la escena actual. Además, mediante ontologías y reglas, se identifican conductas anómalas para informar a un operador humano utilizando lenguaje natural.
- El Capítulo 7 describe los casos de uso y las pruebas realizadas al sistema para finalmente evaluar los resultados obtenidos.
- El Capítulo 8 extrae las conclusiones de la Tesis y futuras líneas de investigación que se derivan del trabajo realizado.

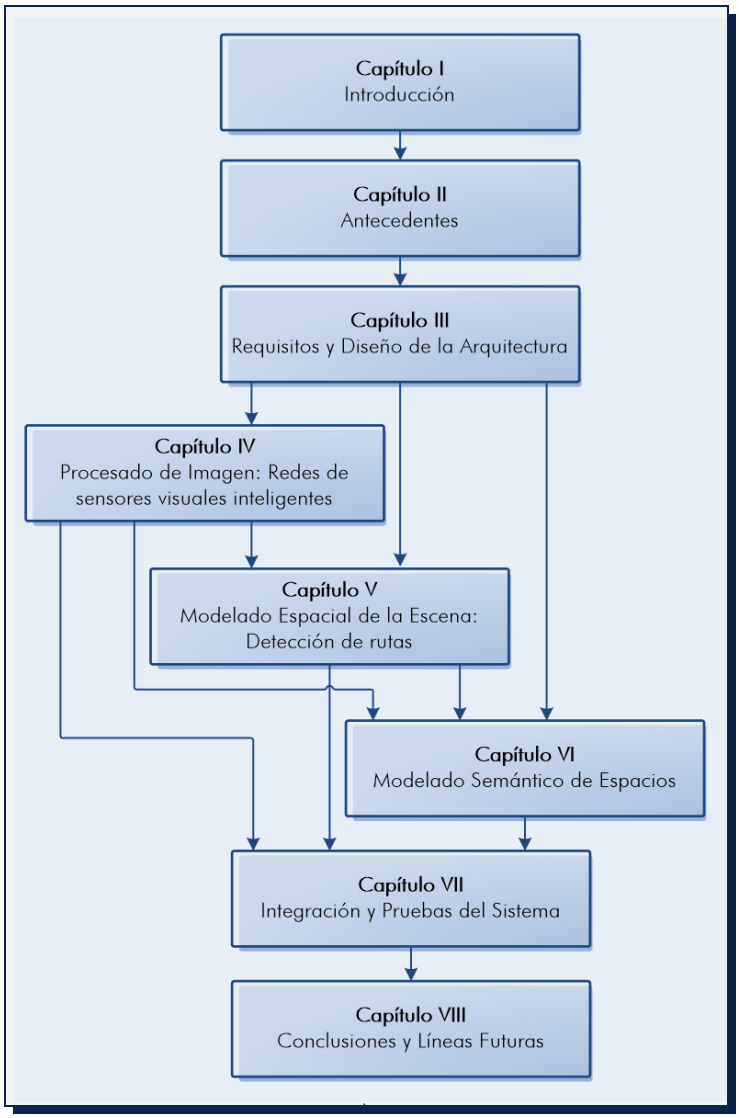


Figura 1.1. Estructura de la Tesis.

2

ANTECEDENTES

Una de las principales líneas de investigación en visión por ordenador es el estudio de las actividades que se están realizando en diferentes secuencias de video. Cada vez hay mayor interés en que este procedimiento se realice de manera automática. Para ello se realiza el procesado de las imágenes de alta calidad y se comparan los resultados con prototipos seleccionadas previamente. Este paradigma se está aplicando sobre todo en videovigilancia [2] y la monitorización del tráfico [3]-[5] para caracterizar los comportamientos y detectar comportamientos anómalos para poder actuar con rapidez.

Es un hecho que la presencia de los sistemas de videovigilancia aumenta progresivamente, debido entre otros a la disminución de los precios del hardware y a la mejora de los sistemas informáticos y las redes de comunicaciones para la gestión y transmisión de la información y las señales de video. Comenzando por los usos tradicionales en edificios con contenidos altamente valiosos o sensibles como instalaciones militares y bancos, estos sistemas se han extendido a lo largo de una gran variedad de otros emplazamientos, incluidos edificios y zonas públicas, redes de carreteras, pequeños comercios, propiedades particulares y residencias, y en

general cualquier lugar en el que la seguridad resulte un elemento a tener en cuenta.

Es por ello que la aplicación de la Inteligencia Artificial (AI - *Artificial Intelligence*), en cualquiera de sus variadas formas, al análisis de señales de videovigilancia resulta una solución muy atractiva y un campo en el que se está innovando y haciendo progresos rápidamente. La literatura muestra variadas soluciones [6]-[8] basadas en la aplicación de diferentes algoritmos al tratamiento de imágenes y su posterior interpretación para concretar el problema.

La automatización lograda por medio de estos sistemas permite la utilización de grandes números de cámaras sin necesidad de disponer de operadores humanos para controlar todas ellas. Esto abre la puerta al despliegue de mecanismos de vigilancia altamente eficientes a gran escala, cubriendo por ejemplo ciudades enteras.

En este Capítulo se estudian las metodologías existentes para la interpretación de imágenes, tanto para determinar los diferentes objetos en movimiento que en ellas aparecen, como aquellos específicamente diseñados para identificar las actividades y comportamiento de los mismos. Dentro de la gran variedad, el análisis se centrará especialmente en los mecanismos aplicados sobre todo a videovigilancia.

Este Capítulo se encuentra dividido en dos secciones. La Sección 2.1 recoge la evolución y tendencias de los mecanismos para el procesado y análisis de imágenes con el fin de identificar los comportamientos de los objetos, especialmente aquellos aplicados a videovigilancia. En la Sección 2.2 se exponen las principales conclusiones del Capítulo.

2.1 Mecanismos de caracterización de imágenes aplicados a la videovigilancia

En los últimos años diferentes organismos han financiado varias iniciativas centradas en el análisis de los contenidos de video de manera automática para determinar comportamientos y actividades:

- ADVISOR - *Annotated Digital Video for Surveillance and Optimised Retrieval* (FP5-IST; 2000-2002; http://www-sop.inria.fr/orion/ADVISOR/)

  El sistema ADVISOR trata de fomentar el uso de transporte público, en concreto del metro mediante la aplicación de tecnologías de visión por

ordenador para un uso efectivo de CCTV. Con ello pretende conseguir una detección automática de incidentes, anotación de contenido basado en las grabaciones de video, análisis de patrones de comportamiento de las personas, etc.

- ASSAVID - *Automatic Segmentation and Semantic Annotation of Sports Videos* (FP5-IST; 2000-2002; [9])

  El objetivo del proyecto es el desarrollo de técnicas para la segmentación automática y la anotación semántica de los videos de deportes. El nivel de anotación debe ser suficiente para permitir consultas simples basadas en texto. Para ello segmenta el material en tomas y agrupa y clasifica las fotos por categorías semánticas en función del tipo de deporte. Para ello, el sistema extrae información de cada toma identificando los aspectos más destacados de la pista de audio y de las reacciones visuales del público.

- AVITRACK - *Aircrafts surroundings, categorised Vehicles & Individuals Tracking for apRon's Activity model interpretation & Check* (FP6-AEROSPACE; 2004-2006; [10])

  AVITRACK utiliza sistemas de video y algoritmos inteligentes para rastrear objetos y personas e interpretar las operaciones normales de mantenimiento de aeronaves en la pista. Un conjunto de cámaras captan imágenes de la zona de estacionamiento y un programa de ordenador interpreta la representación en tres dimensiones en tiempo real de las actividades y movimientos de personas, objetos y vehículos.

- VIDI-VIDEO - *Interactive semantic video search with a large thesaurus of machine-learned audio-visual concepts* (FP6-IST; 2007-2010; http://www.vidivideo.info/)

  El proyecto VIDI-Video tiene como objetivo integrar y desarrollar, en un motor de búsqueda audio-visual semántico, componentes de última generación de aprendizaje automático, detección de eventos de audio y procesamiento de video mediante el uso de la información de diferentes fuentes: metadatos, palabras clave, información audio-visual, del habla, etc.

- SEARISE - *Smart Eyes: Attending and recognizing instances of salient events* (FP7-ICT; 2008-2011; http://www.fit.fraunhofer.de/en/fb/life/projects/searise.html)

El proyecto SEARISE desarrolla un sistema para la detección, seguimiento y clasificación de los eventos y comportamientos más destacados de las escenas que monitoriza. Tiene la capacidad de aprender de la información visual, auto-adaptarse a los cambios en el entorno, fijar los acontecimientos más destacados y seguir su movimiento.

- SAMURAI - *Suspicious and abnormal behaviour monitoring using a network of cameras & sensors for situation awareness enhancement* (FP7-SECURITY; 2008-2011; http://cordis.europa.eu/projects/rcn/89343_fr.html)

  El objetivo del proyecto es desarrollar e integrar un sistema de vigilancia inteligente que emplea sensores heterogéneos conectados en red para crear una visualización más completa de un espacio público concurrido. El objeto es detectar y predecir en tiempo real anomalías.

- 4DVideo - *4D spatio-temporal modelling of real-world events from video streams* (FP7-IDEAS-ERC; 2008-2013; https://www.rdb.ethz.ch/projects/project.php?type=&proj_id=19549&z_detailed=1)

  El objetivo de este proyecto es el desarrollo de algoritmos que permitan capturar y analizar los eventos dinámicos que tienen lugar en el mundo real. Para ello, despliegan redes de cámaras inteligentes para realizar las tareas de observación para después realizar reconstrucciones tridimensionales de hechos reales dinámicos tanto en interiores como al aire libre.

- SAVASA - *Standards Based Approach to Video Archive Search and Analysis* (FP7-SECURITY; 2011-2014; http://www.savasa.eu/)

  El proyecto SAVASA propone la creación de una plataforma de búsqueda de archivos de video. Para ello los usuarios autorizados realizan consultas semánticas sobre diferentes archivos de video remotos.

- ADDPRIV - *Automatic Data relevancy Discrimination for a PRIVacy-sensitive video surveillance* (FP7-SECURITY; 2011-2014; http://www.addpriv.eu/)

  El proyecto trata de determinar de forma automática qué información de la obtenida por un sistema de cámaras de vigilancia distribuido es relevante desde el punto de vista de la seguridad para eliminar la información que no es de interés. Así se minimiza el almacenamiento de datos y se protege la privacidad de los ciudadanos.

- ADVISE - *Advanced Video Surveillance archives search Engine for security applications* (FP7-SECURITY; 2012-2015; http://www.advise-project.eu/)

  El objetivo del proyecto ADVISE es diseñar y desarrollar un marco para la unificación de los sistemas de archivos de videovigilancia para hacer frente a la necesidad de soluciones automatizadas e inteligentes de vigilancia. El sistema se centra en el uso de semántica enriquecida y el análisis de video basado en eventos para una búsqueda eficiente en los archivos y hará cumplir las restricciones legales, éticas y de privacidad que se aplican al intercambio y procesamiento de los datos de vigilancia.

- ACTIVIA - *Visual Recognition of Function and Intention* (FP7-IDEAS-ERC; 2013-2017; http://cordis.europa.eu/search/index.cfm?fuseaction=proj.document&PJ_RCN=13205230)

  El objetivo del proyecto es aprender de forma automática el uso, el propósito y la función de los objetos y las escenas a partir de datos visuales. Para ello utiliza reconocimiento visual de objetos, escenas y acciones humanas y las imágenes y videos públicos disponibles para entrenar modelos visuales.

Además, existen en la literatura soluciones complejas que tratan de modelar la escena en tres dimensiones, uno de los objetivos principales en robótica y visión por ordenador [11]. El propósito de realizar una caracterización 3D de la escena en robótica es, utilizando mapas y dispositivos GPS (*Global Positioning System*), conseguir determinar el espacio por el que se pueden realizar los desplazamientos [12]. En visión por ordenador, los esfuerzos se han centrado en crear mapas en 3D y tratar de etiquetar los diferentes elementos de los mismos, detectando los objetos de interés y sus actividades y comportamientos [13].

En los últimos años la exactitud de la detección de objetos se ha incrementado considerablemente. Si bien existen muchos detectores de objetos genéricos [14], sólo unos pocos se centran en escenarios urbanos. Ess *et al.* [15] y Gavrila *et al.* [16] en sus estudios recuadran peatones en 3D en escenas complejas y realizan el seguimiento de los mismos en los diferentes fotogramas.

Ess *et al.* [15] abordan el problema de seguimiento de varias personas en zonas peatonales concurridas utilizando un equipo montado en una plataforma móvil. Específicamente, se centran en la aplicación del sistema como apoyo a algoritmos de planificación de ruta para evitar obstáculos dinámicos. La complejidad del

problema requiere la extracción de la mayor cantidad de información visual posible y lo combina utilizando retroalimentación cognitiva. Se estima la posición de la cámara y la profundidad y se realiza la detección de objetos y trayectorias basándose únicamente en la información visual. Para ello primero estima la superficie del suelo y detecta los objetos. A continuación, aborda las interacciones de los objetos y estima las trayectorias. Por último, predice el movimiento de los objetos dinámicos.

Por su parte Gavrila *et al.* [16] presentan un sistema de visión multi-señal para la detección en tiempo real y el seguimiento de los peatones desde un vehículo en movimiento. El componente de detección implica una serie de módulos en cascada donde cada uno utiliza criterios visuales complementarios para procesar las escenas. La novedad del sistema es la estrecha integración de los módulos consecutivos: generación de las regiones de interés utilizando visión estéreo, detección basada en la forma, la clasificación usando la textura y verificación apoyada en estéreo. Por ejemplo, la detección fundamentada en la forma, activa una combinación ponderada de los clasificadores basados en la textura, cada uno sintonizado de forma particular. El rendimiento de módulos individuales y su interacción se analiza por medio de la Característica Operativa del Receptor (ROC - *Receiver Operating Characteristic*). Una técnica de optimización secuencial permite la combinación sucesiva de las ROC individuales, proporcionando los parámetros de ajuste del sistema optimizándolo de una manera sistemática.

Wojek *et al.* [17] proponen un modelo para realizar el seguimiento de objetos por un observador móvil con una sola cámara. Para ello presenta un modelo de escena 3D probabilístico que abarca la detección de múltiples clases de objetos, seguimiento, etiquetado de la escena, y relaciones geométricas en 3D. El modelo 3D es capaz de representar interacciones complejas como la oclusión entre objetos, la exclusión física y el contexto geométrico. La inferencia permite recuperar el contexto de la escena y realizar el seguimiento multi-objeto 3D de un observador móvil, para objetos de varias categorías, utilizando un solo video.

Algunos estudios se centran en la división de la escena en distintas regiones de interés. En la literatura aparecen múltiples opciones para la creación de segmentaciones de la escena y su posterior etiquetado en categorías semánticas. Wojek *et al.* [18] realizan la detección de objetos conjunta y segmentación de imágenes en entornos urbanos. En su trabajo proponen un nuevo enfoque basado en campos aleatorios condicionales (CRF - *Conditional Random Field*), modelos estocásticos utilizados habitualmente para etiquetar y segmentar secuencias de

datos o extraer información de documentos. Esta metodología extiende el trabajo existente mediante la formulación de la integración como un problema de etiquetado común de las clases de entidades y la escena y, usando la integración sistemática de información dinámica para la detección de objetos. Como resultado, el enfoque es aplicable a escenas altamente dinámicas, que incluyen movimientos rápidos tanto de la cámara como de los objetos. Por otro lado, Sturgess *et al.* [19] combinan la apariencia y las características del movimiento para la segmentación de imágenes. Está diseñado para manejar imágenes a nivel de calle como las proporcionadas por *Google Street View* y mapas de Microsoft Bing. Se formula el problema como un CRF con el fin de modelar las probabilidades de la etiqueta y el conocimiento previo. Para ello utiliza un conjunto de características basadas en la apariencia como el color, la ubicación y los descriptores HOG (*Histogram of Oriented Gradients*).

Sin embargo, el nivel de comprensión generada por la detección de objetos [15]-[17] y la segmentación de imágenes [18][19], sin razonamiento de alto nivel dice relativamente poco sobre la estructura de la escena subyacente.

Uno de los principales dominios de aplicación de los sistemas de interpretación de video es la videovigilancia. La preocupación para la seguridad personal y la seguridad en sitios públicos es creciente. Se espera que las ventas de sistemas de videovigilancia alcancen una tasa de crecimiento (CAGR - *Compound Annual Growth Rate*) del 14.33 % durante el período 2011-2015 [20]. En los entornos en los que la seguridad es crítica, estos sistemas de vigilancia a través de video son monitorizados por operadores humanos que se encargan de revisar las imágenes en tiempo real. Si el número de cámaras a monitorizar es grande, algún elemento adicional puede ser utilizado para hacer un primer filtrado de la señal de video, tales como detectores de movimiento u otras alarmas, que ayudan a determinar de forma automática qué imágenes son aquellas en las que potencialmente está ocurriendo algo interesante. Sin embargo, hay un gran número de escenarios en los que no es posible ayudarse de este tipo de elementos adicionales (lugares públicos muy transitados) y no es eficiente o rentable contar con un operador humano para llevar a cabo la vigilancia en tiempo real. En estas situaciones el uso de los dispositivos adquiere una dimensión más bien disuasiva (como elemento de obtención de imágenes que puedan servir como pruebas) o para facilitar las actuaciones de contingencia (siendo capaz de generar alarmas en tiempo real) que como elemento de vigilancia.

Actualmente existe una amplia variedad de sistemas de videovigilancia utilizados en diferentes campos [21] como la detección de intrusos [22], vigilancia del tráfico [23], espacios públicos o el hogar [24], detección de incendios [25] o aplicada al deporte [26]. El principal problema de estos sistemas es que la mayoría están diseñados para dominios de aplicación específicos o situaciones determinadas. Habitualmente el modelado del conocimiento usado se diseña a priori y el sistema sólo determina los eventos de interés para esos campos de aplicación. Una de las líneas de investigación abiertas es conseguir sistemas multi-dominio o fácilmente escalables y adaptables que soporten la inclusión de nuevos módulos para conseguir razonamientos más completos.

Nuevas cámaras inteligentes, sensores, entornos multi-cámara, etc., requieren el desarrollo de nuevas tecnologías para aprovechar los datos obtenidos por estos componentes y transformarlos en información útil y de alto nivel para los operadores [27]. Es por eso que hoy en día, la vigilancia inteligente es un campo de investigación que está en constante crecimiento. Se están desarrollando varias alternativas en el procesado para la comprensión de la escena, reconocimiento facial / matrículas / objetos o la detección de alarmas.

Algunas empresas están ofreciendo nuevas plataformas de videovigilancia en el mercado. IBM, por ejemplo, ha presentado un sistema de vigilancia inteligente que, no sólo proporciona la capacidad de monitorizar la escena de forma automática, sino también la capacidad de gestionar los datos de vigilancia, realizar recuperación basada en eventos, recibir alertas en tiempo real y extraer patrones estadísticos de largo plazo de la actividad [28].

Están apareciendo además herramientas como ETISEO (*Evaluation for video understanding; French Ministry of Defence & French Ministry of Research*; 2005-2006) [29] que testea de forma automática sistemas de videovigilancia. Muchos otros proyectos ya han evaluado la eficacia de los sistemas de vigilancia por video, pero más en el punto de vista del usuario final. ETISEO tiene como objetivo estudiar la dependencia entre los algoritmos y las características de video.

El proyecto SMART (*Scalable Measures for Automated Recognition Technologies;* FP7-SECURITY; 2011-2014; http://www.smartsurveillance.eu/) aborda las cuestiones a tomar con respecto a las tecnologías automatizadas de "vigilancia inteligente" en una sociedad donde la privacidad y protección de datos es un derecho fundamental. Los riesgos y oportunidades inherentes de la utilización de la vigilancia inteligente hacen que sea necesario cumplir una normativa que proteja al

ciudadano. Este proyecto pretende crear un conjunto de herramientas con un código de buenas prácticas para informar a los diseñadores de sistemas, diseñadores de políticas y órganos legislativos. Por otro lado existen conjuntos de datos de video realistas diseñados para evaluar los diversos algoritmos de reconocimiento de eventos.

En [30] Oh *et al.* introducen un nuevo conjunto de datos de video diseñado para evaluar los diversos algoritmos de reconocimiento de eventos visuales existentes, especialmente aquellos para el reconocimiento visual de eventos continuos (CVER - *Continuous Visual Event Recognition*) en zonas al aire libre de amplia cobertura. Proponen estos conjuntos de datos como alternativa a los existentes para el reconocimiento de acciones ya que consideran los anteriores no realistas para la vigilancia en el mundo real debido a que suelen ser videos cortos que muestran la acción de un único individuo [31]. Además, la mayoría de los conjuntos de datos existentes se han desarrollado específicamente para películas [32] y el deporte [33], pero estas acciones y las condiciones ambientales no son las mismas que en los entornos de videovigilancia. La base de datos se compone de diversas escenas al aire libre con las acciones que se producen de forma natural por actores no profesionales en videos capturados del mundo real.

Estas iniciativas estimulan la investigación de metodologías de visión por ordenador haciendo que el reconocimiento automático de objetos sea una línea de investigación en auge [6]-[8][34][35]. Existen multitud de mecanismos que llevan a cabo el procesamiento de video para la comprensión de las escenas pero la mayoría siguen la misma metodología. El procesado no se realiza directamente sobre el video en sí sino que se le aplica segmentación para ir procesando fotograma a fotograma las distintas imágenes que se van obteniendo. El siguiente paso es determinar los objetos que aparecen en la escena. A continuación se realiza un seguimiento de los mismos para su identificación en los distintos fotogramas en los que aparecen. Este proceso se denomina *tracking*. El siguiente paso es la identificación de los objetos para poder etiquetarlos en función de sus propiedades. Como último paso, y una vez reconocidos los objetos, se analizan para determinar su comportamiento y la actividad que están realizando dentro de las imágenes. En videovigilancia en este último paso de determinan las anomalías o situaciones de riesgo.

En esta fase se identifican los distintos objetos que aparecen en la escena y se determinan sus parámetros básicos (posición, velocidad, dimensiones, etc.). Para

realizar este procesado se utilizan distintos mecanismos de procesado de imagen [7]:

- La extracción del fondo: Esta técnica consiste en comparar píxel a píxel con el fondo las imágenes que van apareciendo [36]. Como resultado se van identificando regiones donde se encuentran los objetos en movimiento que son lo que en un principio no aparecían. Para que esta metodología funcione adecuadamente es necesario disponer de una buena imagen de fondo con la que comparar y así evitar los problemas derivados de cambios de iluminación, objetos extraños, etc. También se va actualizando la imagen de fondo cada cierto tiempo para así adaptarla a las nuevas condiciones y evitar falsas detecciones.

- La diferencia entre fotogramas consecutivos: Este mecanismo compara píxel a píxel el fotograma actual con los fotogramas anteriores de manera que esta diferencia indica las zonas en las que se ha producido un cambio y por tanto movimiento [37]. En ocasiones, para obtener resultados más precisos, se compara la diferencia con un umbral, de manera que si el cambio es mínimo no se considere [38]. Para hacer esta metodología más robusta se suele realizar la comparación con varios fotogramas previos.

- La utilización de flujo óptico: Esta técnica utiliza los vectores que definen el movimiento relativo de los objetos con respecto al observador [39]. Son técnicas muy complejas computacionalmente y requieren hardware especial por lo que no se suelen utilizar en videos reales. Como ventaja, estos mecanismos permiten detectar los objetos en movimiento incluso si la imagen procede de una cámara móvil.

En la segunda fase se realiza un seguimiento de la evolución de los objetos en los diferentes fotogramas utilizando sus localizaciones, tamaños, formas, etc. Este procedimiento se complica bastante sobre todo en videos con gran cantidad de objetos. En ocasiones los objetos desaparecen temporalmente durante varios fotogramas porque se encuentran ocultos detrás de otros y es imposible realizar el proceso de seguimiento. A este problema se le denomina oclusión. Por otra parte, si los objetos están muy juntos, en ocasiones se detectarán como un único objeto. Existen varias herramientas matemáticas para realizar este proceso una vez determinados los objetos que aparecen en la escena como el filtro de Kalman [40], el algoritmo de condensación [41] o redes bayesianas dinámicas [42]. De forma

general las técnicas existentes se pueden clasificar en: basadas en regiones, en contornos activos o utilizando modelos [7].

- Los algoritmos de seguimiento basado en regiones procesan las zonas donde existe movimiento. Para ello utilizan el mecanismo de comparación con un fondo dinámico.

  Siguiendo esta aproximación, algunos autores descomponen los objetos en zonas diferentes y van realizando el seguimiento de manera separada.

  Para el seguimiento de personas, por ejemplo, brazos, piernas, torso y cabeza se consideran independientes y el seguimiento de cada una de esas partes permite el seguimiento de la persona. Este es el mecanismo utilizado por los algoritmos *Pfinder* y W^4.

  Pfinder [43] modela de forma independiente el color de cada píxel utilizando un modelo gaussiano. Del análisis de *frames* consecutivos se extrae la información del fondo de manera que se pueden extraer los píxeles pertenecientes al cuerpo humano y se asignan a las diferentes partes del cuerpo.

  W^4 [44] es un algoritmo para la detección y seguimiento de personas en exteriores. Opera con imágenes en escala de grises procedentes de una cámara de infrarrojos, por lo que no hace uso del color para el seguimiento. Emplea una combinación de análisis de la forma y de seguimiento para localizar a las personas y sus partes (cabeza, manos, pies, torso) creando modelos que se puedan seguir incluso en situaciones de oclusión.

  McKenna *et al.* [45] proponen un sistema de visión artificial para el seguimiento de varias personas en distintos ambientes de interior y al aire libre. Para ello utilizan un método dinámico de extracción de fondo que utiliza el color y el gradiente y cromaticidad. El uso del gradiente trata de eliminar los cambios de iluminación en la escena ya que los autores consideran que no son modificaciones lentas sino que las sombras de las personas que se mueven producen cambios bruscos. El seguimiento se realiza en tres niveles de abstracción: regiones, personas y los grupos. Cada región es un conjunto de píxeles que permanecen conectados durante un periodo de tiempo. Una persona se compone de una o más regiones agrupadas bajo una serie de condiciones geométricas para formar la

estructura del cuerpo humano, y un grupo se compone de una o más personas que se mueven juntas.

Para el seguimiento de vehículos, hay algunos sistemas de uso extendido como el sistema desarrollado en el proyecto PATH por la Universidad de California [46]. Los autores proponen un sistema de seguimiento basado en correlación, un modelo físico de movimiento y la utilización de un filtro de Kalman para extraer las trayectorias de vehículos en una secuencia de imágenes de escenarios de tráfico. Un módulo de razonado obtiene la información inferida para determinar los distintos eventos de tráfico que se producen.

Estos algoritmos funcionan bien en escenas que contienen pocos objetos pero no son capaces de manejar las situaciones en las que se produce oclusión entre objetos.

- Las técnicas de seguimiento basado en contornos activos utilizan las siluetas de los objetos y las van actualizando en los distintos fotogramas. Estás técnicas extraen la forma del objeto para conseguir descripciones más simples y precisas que las obtenidas por el método de seguimiento basado en regiones.

  Freedman y Zhang [47] utilizan la distribución de probabilidad de variables como la intensidad, el color o la textura para caracterizar los objetos. Para realizar el seguimiento tratan de encontrar la región, dentro de la imagen actual, en la que la distribución de muestra en la región coincide mejor con la distribución del modelo.

  En [48] Yilmaz *et al.* proponen un mecanismo de seguimiento basando en la evolución del contorno fotograma a fotograma. Propone que el contorno de energía está formado por dos términos: la energía de la imagen y de la forma. La energía de la imagen se basa en el color y la textura de las observaciones y se evalúa en una banda alrededor del contorno. Por otro lado la energía de la forma utiliza las últimas observaciones del contorno y conserva la forma del objeto durante oclusiones parciales y completas. En este caso, se propone un modelo actualizado en tiempo real, que determina las deformaciones del contorno durante el proceso de seguimiento. El rastreo se consigue mediante el análisis de la evolución del contorno, que se representa usando conjuntos de nivel, reduciendo al mínimo la energía en la dirección de descenso del gradiente.

En [49] se presenta un mecanismo para la detección y seguimiento de múltiples objetos en movimiento en secuencias de imágenes basado en contornos activos geodésicos.

- Las técnicas de seguimiento basadas en modelos utilizan prototipos de objetos diseñados previamente con los que comparar. Estas técnicas se aplican de manera diferente a objetos rígidos, que mantienen su silueta más o menos constante durante todo el recorrido, que a objetos cuyo contorno varía.

  La utilización de este enfoque para el seguimiento de objetos no rígidos como personas utiliza básicamente una metodología de tres pasos: predecir, encontrar y actualizar. En primer lugar, se predice la pose en el siguiente fotograma utilizando los conocimientos previos. Esta pose se proyecta sobre la imagen y se va comparando buscándola dentro de la misma. Una vez localizada se actualiza el modelo con la nueva pose. Para la realización de este proceso se necesitan crear modelos, tanto humanos como del movimiento, y la implementación de estrategias de predicción y búsqueda. Un ejemplo de uso de esta metodología es el presentado por Aggarwal *et al.* en [50].

  Si lo que se desea rastrear es un objeto rígido como un vehículo, el contorno del objeto no varía tan rápidamente con el tiempo. En [51] se modelan los vehículos como un conjunto de rectángulos teniendo en cuenta las dimensiones, el ancho y largo del capó y el tamaño de las ruedas del mismo. Este modelo va rotando y cambiando en función de la variación del modelo que simula la gravedad en la imagen.

  Como ventajas, este método es más robusto que los anteriores ya que conoce a priori los contornos de los objetos y se han obtenido buenos resultados incluso en situaciones de oclusión. Como contraposición, estos algoritmos requieren del diseño de modelos previos y tienen una carga computacional alta.

Una región en movimiento puede ser un objeto u otro en función del escenario. En el control de tráfico, por ejemplo, los objetos podrán ser vehículos, peatones, bicicletas, un árbol, etc. En paralelo al seguimiento del objeto se realiza la clasificación de los objetos. Para la identificación y etiquetado de los objetos se utilizan principalmente dos mecanismos: el etiquetado basado en la forma de los mismos y utilizando sus patrones de movimiento [7].

- La clasificación de objetos basada en la forma utiliza las siluetas de los objetos para compararla con patrones predeterminados de los que se conoce la clase a la que pertenecen. De cada clase se dispone de uno o varios patrones representativos. La silueta del objeto se compara con todos los patrones definidos calculando la similitud con cada uno de ellos. El objeto pertenecerá a la clase a la que representa el patrón con el que más se parece. Dado que las siluetas varían con el tiempo y esto puede afectar al proceso de clasificación, en ocasiones autores como Lipton *et al.* [38] y Kuno *et al.* [52] contemplan la evolución de las siluetas durante un periodo de tiempo, realizando este proceso fotograma a fotograma para asegurar que el etiquetado es el adecuado.

  En el proyecto VSAM (*Video Surveillance and Monitoring*) [53] utilizan redes neuronales con el fin de, utilizando la forma de los objetos, identificar dentro del entorno personas independientes, en grupos y vehículos.

- Con respecto a la clasificación basada en los patrones de movimiento de los objetos, se utilizan las variaciones de la silueta con el tiempo y la periodicidad de los movimientos. En [54] se utiliza la similitud de los movimientos en su evolución con el tiempo para identificar los distintos objetos, en el caso del estudio, personas, vehículos y perros corriendo. El estudio presentado en [55] permite diferenciar vehículos y personas por la rigidez de las siluetas y su evolución con el tiempo. Mientras que las siluetas de las personas varían con el movimiento de manera más o menos periódica, la de los vehículos se mantiene más o menos inalterable durante todo su recorrido.

Por supuesto, ambas metodologías pueden utilizarse de manera conjunta o complementarse con parámetros como la velocidad de los objetos para conseguir mejores resultados.

Cuando el sistema es capaz de identificar objetos, la AI y algoritmos de interpretación de video son capaces de detectar comportamientos anormales de los objetos. La aplicación de complejos algoritmos de visión artificial es una de las principales tendencias para videovigilancia [56]-[59]. Sin embargo, normalmente requieren potencia de cálculo, por lo que o bien la propia cámara incluye una unidad de procesamiento de gran alcance, por lo que se incrementa el coste, o la señal de video de alta definición se envía a una unidad central de proceso, solución con alto consumo de ancho de banda (especialmente en un escenario con un gran

número de cámaras). Estas limitaciones se resuelven empleando paradigmas más ligeros para procesar la imagen, lo que implica normalmente la reducción del video a un conjunto de parámetros de segundo nivel (por ejemplo, el movimiento del objeto [60]). Estos enfoques presentan la ventaja de ser más ligeros, pero por lo general pierden mucha información en la imagen (como el color o la forma de los objetos) al reducir el flujo de video a los parámetros. Por lo tanto, se requieren herramientas de análisis avanzadas para maximizar la información de alto nivel que se puede extraer y su posterior interpretación.

Las soluciones disponibles documentadas en la literatura, como por ejemplo [61], emplean básicamente dos estrategias diferentes para el proceso de interpretación del video y la posterior identificación de alarmas: o bien el análisis estadístico (matemáticamente se caracterizan las variaciones normales de los parámetros de la imagen para identificar las situaciones fuera de estos valores) o análisis "semántico" (el significado de los objetos de la imagen se utiliza para entender cuándo se está produciendo una situación de alarma), aunque algunos autores como SanMiguel y Martínez [62] proponen soluciones híbridas.

- En [56] y [63] se trata la utilización del análisis estadístico para procesar la información visual centrado en la videovigilancia y el uso de "*Latent Semantic Analysis*", presentan modelos probabilísticos donde se aplica clasificación estadística y aprendizaje relacional para identificar rutinas recurrentes.

  Sistemas como los presentados en [64][65] usan soluciones basadas en redes bayesianas. El teorema de Bayes es muy útil para determinar las probabilidades de una alarma mediante el uso de las relaciones entre todas las variables en una situación específica. Otros autores [66] prefieren enfoques probabilísticos más complejos como modelos ocultos de Markov (HMM) (típicamente empleados en muchos dominios para el reconocimiento de patrones), para extraer parámetros desconocidos pero significativos a partir de los datos en bruto proporcionados por las cámaras. Otras técnicas, utilizadas en el reconocimiento de patrones también, son la *Dynamic Time Warping* (DTW) [67] y la *Longest Common Subsequence* (LCSS) [68]. Esos mecanismos comparan grupos de variables para encontrar similitudes entre ellos, y se emplean con éxito, por ejemplo, para agrupar trayectorias similares en las escenas de vigilancia [69][70].

La red bayesiana y los métodos basados en similitud utilizan la información explícita proporcionada por el sistema para detectar alarmas. Los sistemas basados en HMM van más allá. Extraen nueva información implícita, oculta en los datos proporcionados por las cámaras y sensores. Algunas técnicas deductivas utilizan redes neuronales [71][72] o algoritmos de agrupación [3] para clasificar los comportamientos y contextos. Normalmente estas metodologías requieren gran cantidad de recursos y el procesamiento de datos es lento.

Las distintas soluciones están enfocadas a realizar el procesamiento estadístico de imágenes con el fin de reconocer y localizar diferentes objetos como en [8][57] o dirigidas a detectar el comportamiento estadístico y la asignación de funciones a los objetos como en [63][73][74]. En ambos casos están implementadas para su funcionamiento en ubicaciones predefinidas y escenarios controlados. Su traslado a entornos reales es difícil debido a su baja flexibilidad: el sistema tiene que ser rediseñado por completo y adaptado para cada dominio.

- En la literatura, la palabra "semántica" [2][75]-[78] a veces se usa para especificar que la detección de alarma no se hace sólo estadísticamente (como en [79]), pero está fuertemente unido al dominio de la escena visualizada, por lo que la detección de las alarmas se realiza sin tener en cuenta el sentido del resto de los comportamientos que aparecen en la imagen, codificada dentro de los algoritmos de detección de alarma. Por ejemplo, un sistema para la detección de exceso de velocidad en una carretera podría llevar a cabo la identificación de objetos para detectar automóviles y escanear su velocidad, pero que el algoritmo no puede ser modificado para detectar alarmas de incendio o personas sospechosas, anomalías que suceden dentro de un edificio. Este trabajo va un paso más allá de este enfoque, ya que se basa en la representación formal del significado de la imagen, utilizando las tecnologías de la *Web* Semántica, que permiten un diseño ontológico fácil y rápido. Por lo tanto, sólo con la modificación de la ontología, el propietario puede cambiar las condiciones de alarma, los parámetros empleados e incluso todo el dominio de funcionamiento. El otro uso de la palabra "semántica" en la literatura es en documentos que en realidad emplean tecnologías de *Web* Semántica formales.

La utilización de semántica para la representación y procesado del conocimiento es una disciplina que empezó a introducirse en el mundo de las tecnologías de la información hace aproximadamente 10 años [80]. Estas tecnologías semánticas se han desarrollado para superar las limitaciones de los sistemas tradicionales de gestión y representación de datos sintácticos/estadísticos, y se están aplicando en la nueva generación *Web*, etiquetada en ocasiones como *Web* 3.0 o *Web* Semántica [81]. Sin embargo, la semántica también se está aplicando a nuevos dominios que pueden beneficiarse de esta representación estructurada del conocimiento y la capacidad de razonamiento (que proporciona ventajas como la interoperabilidad entre sistemas heterogéneos, la posibilidad para inferir relaciones que no se almacenan de forma explícita en las bases de datos, etc.) [82][83].

Algunos autores [84][85] presentan el concepto de la "*Semantic Information Fusion*", donde los datos en bruto procedentes del sensor se convierten en información semántica que puede interpretarse para conseguir una comprensión de la misma. Algunos sistemas clasifican las diferentes regiones de la imagen [2][69][80] o plantillas visuales [82], mientras que otros [84][85] se centran en el procesado a bajo nivel de la imagen utilizando métodos bayesianos. Sin embargo para la identificación de objetos y alarmas utilizan otras metodologías, y posteriormente usan las tecnologías semánticas sólo para representar la información. Zhang *et al.* [79] y Marraud *et al.* [86] en sus estudios, usan la semántica para permitir búsquedas avanzadas de las características de videos en bases de datos; François *et al.* [87] proponen un nuevo lenguaje para la representación de los acontecimientos en un video para facilitar la búsqueda y Poppe *et al.* [88] especifican un sistema para unificar información semántica en videovigilancia.

Este procesamiento semántico de datos por lo general se realiza en dos etapas: en primer lugar, la especificación del modelo de conocimiento, y en segundo lugar, el reconocimiento de patrones. La primera de estas fases se lleva a cabo *off-line*, en fase de diseño, e implica la creación de una ontología que describe el dominio de conocimiento específico en el que el sistema funciona identificando las entidades implicadas y las relaciones entre ellas [63][73][74][81][89]-[91]. Este modelo de conocimiento se emplea en la segunda fase que lleva a cabo la interpretación semántica de los datos de

entrada de acuerdo con el modelo de conocimiento especificado en la primera etapa.

Sin embargo, todos los casos anteriores implican una alta carga computacional, ya que los algoritmos operan directamente sobre las imágenes. Esto significa que, o bien todas las cámaras tienen que incluir procesadores de alto rendimiento o la señal de video tiene que ser enviada al centro de control, donde se ejecutan los algoritmos inteligentes. Para redes de vigilancia densas (que normalmente es el caso de los entornos inteligentes), esto es muy ineficiente, ya que las cámaras inteligentes son caras o se requiere un enorme ancho de banda para el envío del video.

Típicamente, las señales de video procedentes de estos sensores son tratadas de forma independiente. Sin embargo, existen muchos casos en los que sus salidas se pueden combinar con el fin de obtener una mejor comprensión de lo que está sucediendo. Los entornos multi-camara son muy útiles ya que permiten ampliar las zonas de cobertura y la visualización de las escenas desde diferentes localizaciones con lo que se evitan fenómenos como la oclusión. Sin embargo para que estos sistemas sean eficientes se necesita planificar bien la localización de las cámaras para utilizar el menor número posible, calibrarlas adecuadamente y aplicar metodologías que permitan el procesado conjunto de la información procedente de todas ellas de manera coordinada.

Con respecto a la instalación, un despliegue adecuado minimiza costes y además permite un mejor funcionamiento del mismo. El uso de cámaras redundantes que cubren la misma zona aumentará, no sólo el coste de instalación, sino también el tiempo de procesado y la complejidad de los algoritmos de fusión [7]. Si por el contrario se utilizan menos cámaras de las estrictamente necesarias aparecerán zonas muertas que no se están monitorizando lo que reduce la eficiencia del sistema. Autores como Pavlidis *et al.* [92] tratan de buscar el algoritmo adecuado para resolver este problema.

La calibración de las cámaras se complica en sistemas muti-camara. Ya no son adecuados los métodos de calibración tradicionales para una única cámara (basados en coordenadas 3D y las coordenadas de la imagen de algunos puntos conocidos para calcular los parámetros de la misma). Se necesitan nuevos métodos que contemplan la evolución temporal de la información [7].

Con respecto a la utilización de manera coordinada de la información procedente de varias cámaras hay que tener en cuenta varios aspectos.

Por una parte es imprescindible identificar el mismo objeto dentro de todas las imágenes que visualizan la misma escena. Para ello existen distintos métodos [7]. Hay autores como Cai *et al.* [93] que utilizan la localización del objeto, la intensidad o la geometría, otros como Krumm *et al.* [94] basan el reconocimiento en el color o autores como Javed *et al.* [95] establecen relaciones espaciales entre los FOV para determinar las correspondencias entre las imágenes.

Es importante también realizar un seguimiento coordinado de los objetos de manera que cuando desaparezcan del FOV de una cámara sea otra la que lo identifique y lo siga. Esto permite también el seguimiento de los objetos cuando se producen oclusiones. Para evitar este fenómeno los autores proponen diferentes métodos. Seleccionar la mejor vista, aplicar seguimiento basado en flujo óptico, predicción de la oposición y la velocidad del objeto utilizando información de otras cámaras son algunos de los utilizados [7]. Además, un tratamiento adecuado de la información procedente de las distintas cámaras permite la representación de escenas completas en 3D y la obtención de trayectorias integradas como composición de todas las imágenes de las múltiples cámaras.

Habitualmente este tipo de sistemas se centran únicamente en el procesado de imagen, sin embargo, los despliegues incluyen sensores heterogéneos que pueden aportar información adicional optimizando la detección y permitiendo identificaciones en casos en los que únicamente con el procesado visual no sería posible.

Por otro lado los avances en las Tecnologías de la Información y las Comunicaciones (ICT - *Information and Communication Technologies*) están provocando una transformación de los ambientes en los que vivimos para convertirlos en entidades inteligentes conocidas de forma global como *Smart Spaces* (*Smart Homes, Smart Buildings, Smart Cities*, etc.). Estos sistemas se nutren de la información proveniente de grandes redes de sensores distribuidos por todo su dominio (una casa, un edificio, una ciudad entera, etc.) y la utilizan para adaptar de forma inteligente su comportamiento a las necesidades del usuario [96].

Cuando se integran en los *Smart Spaces*, estas redes de sensores aportan a los sistemas inteligentes la posibilidad de visualización y, combinado con su inteligencia, permiten detectar e identificar las diferentes situaciones anormales que puedan surgir. El máximo problema es que para cubrir áreas extensas se necesita el despliegue de gran cantidad de cámaras, con lo que un operador humano no puede analizarlas individualmente y requieren de algoritmos de procesado

autónomo. Sensores de movimiento, detectores de humo, sensores de localización GPS son algunas de las posibilidades.

Además, en los últimos años están apareciendo nuevas propuestas que contemplan como complemento a la información visual el procesado de los sonidos del entorno. En el proyecto PRISMATICA (*PRo-active Integrated Systems for Security Management by Technological, Institutional and Communication Assistance,* FP5; 2000-2005) [97][98], por ejemplo, se utilizan algoritmos de procesado de video y audio para mejorar la seguridad de los pasajeros en el transporte público.

El principal problema a afrontar para utilizar esta información es cómo procesar o integrar toda esta información heterogénea. Con el fin de tomar ventaja de la creciente conciencia de la situación que surge de la interpretación combinada de varias salidas de sensores, aparecen técnicas de fusión de datos. En [84][85], por ejemplo, se presentan una recopilación de las técnicas existentes de la fusión de la información de las redes de sensores inalámbricos mediante distintos métodos, algoritmos, arquitecturas y modelos.

2.2 Conclusiones

En los últimos años el reconocimiento de escenas dinámicas se está convirtiendo en una de las principales áreas de investigación. Es por ello que se han financiado diferentes iniciativas de este campo invirtiendo muchos esfuerzos sobre todo en videovigilancia. Existen además multitud de publicaciones que analizan distintas secuencias de video pero la mayoría de las aplicaciones están destinadas también a este fin.

Por otro lado los sistemas de videovigilancia modernos cada vez tienen mayores demandas. Cobertura de espacios amplios, escenarios complejos, redes de sensores heterogéneas o funcionamiento en tiempo real son algunos de los requisitos que se imponen a los nuevos sistemas de videovigilancia autónomos.

La mayoría de los autores utilizan una metodología en cuatro fases para la implementación de estos sistemas: detección de objetos en movimiento, seguimiento de los mismos en los distintos fotogramas del mismo, clasificación de los objetos para poder etiquetarlos y definir las actividades y comportamientos de los objetos. La literatura en este campo es bastante extensa ya que para cada una de estas fases los autores proponen distintos mecanismos.

En la mayoría de los casos se proponen técnicas de visión por ordenador para realizar el procesado de las imágenes de video segmentadas. Sin embargo, la carga computacional de dicho procesado y la necesidad de imágenes de alta resolución encarece los sistemas con lo que se están buscando alternativas. Algunas soluciones no utilizan la forma, color, textura, etc., de los objetos sino que basan sus razonamientos en sus parámetros de movimiento, por ejemplo. Velocidad, posición, patrones de movimiento, son algunos de los parámetros que surgen como alternativa a un procesamiento puro de la imagen.

Por otro lado, el despliegue de sensores visuales para cubrir áreas extensas consigue mejoras con respecto a los sistemas monocámara ya que se consiguen eliminar problemas de apreciación como la oclusión. La visualización de la escena desde distintas localizaciones proporciona diferentes perspectivas de los objetos y por lo tanto mejores resultados. Es muy importante en estos casos colocar las cámaras de manera adecuada para evitar zonas muertas que no se visualicen o información redundante que aumenta la complejidad del sistema y el coste.

Detectores de humo, sensores de localización GPS, etc., son algunas de las fuentes de información alternativas a la visual que pueden aportar información adicional sobre los entornos y que, contemplándolos en los sistemas de videovigilancia pueden mejorar los resultados. Sin embargo, el procesado de fuentes heterogéneas supone un reto que la mayoría no es capaz de afrontar.

La realidad es que la mayoría de los sistemas existentes funcionan para escenarios específicos y con sensores predeterminados siendo complicado la adaptación a nuevas situaciones y el procesado de información procedente de distintas fuentes a las previstas a priori.

El objetivo de esta Tesis Doctoral es proporcionar una metodología que incorpore las bondades de los sistemas estudiados y además, solucione algunos de sus inconvenientes. Para ello el sistema utiliza, como alternativa al procesado de imagen, los parámetros de movimiento de los objetos mejorando la privacidad de los usuarios. Como añadido, es importante tener en cuenta la información que, procedente de otros sensores, se puede tener del entorno. Esta característica no es usual en los sistemas estudiados ya que se limitan al estudio de las propiedades de los objetos, y sin embargo, puede aportar información crítica para la detección de ciertas alertas. Una característica adicional es la identificación de situaciones anómalas en lenguaje formal, entendible por un operador humano. Esta característica podría no considerarse especial, sin embargo, un modelado

puramente semántico de la escena permite, no sólo una jerarquización de la información, sino también que el dominio de conocimiento sea independiente del diseño, de manera que, con la modificación de la ontología, se permite la adaptación a nuevas áreas de aplicación. Es decir, las tecnologías de la *Web* Semántica no sólo se utilizan como una herramienta de modelado, sino como un mecanismo de abstracción, no habiéndose utilizado hasta ahora para este fin.

3

REQUISITOS Y DISEÑO DE LA ARQUITECTURA

HuSIMS (*Human Situation Monitoring System*) es un proyecto Eureka-Celtic, iniciativa impulsada por la industria europea para definir, ejecutar y financiar proyectos públicos y privados de investigación en el área de las telecomunicaciones e Internet del futuro y aplicaciones y servicios centrados en un nuevo "Mundo Inteligente Conectado" (*Smart Connected World*). El proyecto, con un presupuesto de 6 millones de euros y una duración de tres años, está financiado por el Ministerio de Industria, Turismo y Comercio dentro del Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica 2008-2011 y el Fondo de Desarrollo Regional (FEDER).

El proyecto HuSIMS pretende diseñar una red de sensores de vigilancia capaz de identificar, de forma autónoma, cuándo en la imagen se está dando una situación de emergencia, desde un accidente de tráfico a un incendio. Para implementar este comportamiento inteligente, se desarrolla dentro del mismo un módulo denominado

Motor Semántico. Esta Tesis Doctoral se centra sobre todo en el diseño e implementación de este mecanismo de caracterización de imágenes y detección de anomalías, basado en la detección de rutas y el modelado ontológico de espacios.

En esta Tesis Doctoral se va a realizar el diseño de la arquitectura y la implementación de este módulo utilizado únicamente como aportación de los miembros del proyecto, el sensor visual comercial proporcionado por uno de las compañías participantes.

El Motor Semántico es un sistema automatizado de caracterización totalmente semántica de escenas de diferentes dominios. Por su flexibilidad, puede aplicarse a situaciones de muy diversa índole (gestión y control del tráfico en *Smart Cities*, medio ambiente, vandalismo, etc.) permitiendo además, detectar alarmas y situaciones anómalas y/o potencialmente peligrosas.

Como se estudió en el Capítulo 1, los actuales sistemas se basan en su mayoría en el análisis estadístico de las características de la imagen para identificar los objetos que aparecen en ellas. Este análisis sólo permite la detección de comportamientos anómalos, entendiendo por anómalos aquellos que no suceden frecuentemente, basándose en criterios puramente matemáticos. Este tipo de desarrollos son muy limitados y no permiten identificar en lenguaje formal el carácter de la alarma.

Por otro lado existen desarrollos sencillos basadas en ciertos algoritmos implementados en el propio código. Este tipo de sistemas se diseñan con un objetivo definido y para un entorno cerrado por lo que, en caso de variar las condiciones o querer adaptarlos a otros dominios, es necesario modificar de forma manual los algoritmos con lo que su flexibilidad es muy limitada.

Para evitar que una vez desarrollado el sistema aparezcan estas limitaciones, antes de comenzar a seleccionar las tecnologías y la implementación, es necesario definir de forma clara los requisitos del mismo. Estas exigencias se toman como punto de partida para diseñar, a alto nivel, la arquitectura que incluye todas las características. Estos estudios y diseños previos a la implementación del sistema se llevarán a cabo en este Capítulo.

Este Capítulo se encuentra dividido en tres secciones. Los requisitos del sistema son analizados en la Sección 3.1. En la Sección 3.2 se diseña la arquitectura del sistema especificando los principales componentes de la infraestructura y su funcionalidad. Finalmente, en la Sección 3.3 se exponen las principales conclusiones del Capítulo.

3.1 Requisitos

Como aparece recogido en la introducción, el objetivo fundamental de esta Tesis Doctoral es el diseño y desarrollo de un sistema inteligente y autónomo de interpretación de escenarios de diferentes dominios.

Para ello, los principales requisitos del sistema, algunos de ellos impuestos por el propio proyecto HuSIMS, son:

1. Operar con un gran número de cámaras de forma económica.

   Las cámaras utilizadas deben ser unidades sencillas, con poca resolución y capacidad de proceso limitada, que hace imposible la utilización de técnicas avanzadas como reconocimiento de caras u objetos. Para la identificación de los distintos elementos se emplea la información de la que se dispone, básicamente tamaños, posiciones y velocidades de los objetos en movimiento ya que van a ser éstas las únicas regiones que se van a poder procesar.

   Tan importante como la obtención de las características de los objetos en movimiento va a ser el seguimiento de los mismos. De esta manera se puede determinar cómo van variando sus parámetros y así obtener información adicional de los mismos.

   Además, las cámaras no envían al centro de control el video para un análisis en tiempo real sino sólo la información básica obtenida del mismo, con lo que se simplifican las transmisiones y el ancho de banda necesario.

   Aunque de forma genérica se hable de información procedente de cámaras, el sistema funcionará igualmente con grabaciones de video incluso de situaciones reales.

2. Incorporar un algoritmo de interpretación de escenas de video.

   A partir de información básica de los objetos en movimiento, el sistema debe incorporar un algoritmo que sea capaz de identificarlos y con ello las diferentes zonas de la escena. Volviendo sobre el ejemplo del control y gestión del tráfico, un objeto en movimiento se etiqueta como coche cuando tiene una velocidad y dimensiones dentro de unos umbrales. A partir de ahí, se determina que es una carretera aquella zona de la imagen por la que habitualmente circulan vehículos. Además, se conocen los patrones de

movimiento habituales de esa región como pueden ser la dirección y la velocidad de los objetos que se desplazan por ella.

3. Debe ser capaz de identificar situaciones anómalas y dar información específica de la alarma en lenguaje natural.

   El sistema debe completar la caracterización de escenas de video con un mecanismo de identificación de las situaciones anómalas. Como se estudia en el estado del arte, la mayoría de estos sistemas existentes utilizan métodos puramente matemáticos y estadísticos para determinar que una actuación poco usual es una situación de alarma. Sin embargo, el sistema a implementar debe ser capaz de identificar, de forma concreta qué está sucediendo, como por ejemplo, indicar que hay un coche que ha sufrido un accidente.

4. Podrá funcionar con información no sólo procedente de cámaras, sino también con la que pueden aportar otro tipo de sensores.

   En algunas ocasiones puede ser interesante o se podrá disponer de información adicional a la aportada por las cámaras. Sensores de temperatura, humedad, vibración, fuerza, etc., pueden facilitar la caracterización de las escenas y la identificación de nuevas situaciones de emergencia como terremotos, incendios, etc. El sistema debe poder incluir la información de sensores adicionales en su procesamiento para obtener resultados más precisos.

5. Será fácilmente adaptable para su funcionamiento en diferentes dominios dentro de las *Smart Cities*.

   Una de las mayores limitaciones de los sistemas estudiados en el estado del arte es que restringen su funcionamiento a un dominio específico y bajo una serie de condiciones. Un cambio de escenario hace necesario el desarrollo de un nuevo algoritmo o incluso una nueva implementación completa del sistema. Para evitar este problema el sistema deberá diseñarse y desarrollarse para que sea flexible y se pueda adaptar de forma fácil a cualquier dominio, es decir, que sea capaz tanto de realizar el control y gestión del tráfico como, por ejemplo, la detección de incendios.

3.2 Arquitectura Propuesta

El sistema a diseñar e implementar debe ser capaz de cumplir los requisitos especificados en la Sección 3.1.

Una posible solución a este problema es utilizar algoritmos de visión artificial para detectar la presencia de objetos conocidos ante la cámara. La bibliografía muestra avances sensibles en este campo, tales como [56]-[59].

Una vez detectada la presencia de estos objetos, ésta se podría incorporar a un modelo de conocimiento semántico que permitiría caracterizar los comportamientos normales y anómalos.

Sin embargo, esta utilización de algoritmos de visión artificial para la detección de los objetos, al tratarse de una tarea con una alta carga computacional, entraría en conflicto con el requisito número 1.

Otra solución es emplear análisis semántico de los datos, es decir, dar una interpretación, con significado acorde a la percepción humana del mundo real, de lo que está pasando en la escena. En lugar de, por ejemplo, utilizar parámetros matemáticos y/o estadísticos para determinar sucesos que están fuera de lo normal, el análisis semántico está enfocado en caracterizar la información de la señal de video según un modelo de conocimiento del mundo real, es decir, identificando objetos con significado, tales como peatones, calzadas, incendios, coches, etc.

La realización de este tipo de razonamiento semántico directamente sobre la señal de video requiere una gran potencia de cálculo, principalmente en la identificación de todos los objetos en la imagen, y que entraría en conflicto con el requisito número 1.

Por lo tanto, en el sistema propuesto en este trabajo no se realiza la identificación de objetos directamente sobre el flujo de video. En su lugar, se describe un método basado en el trabajo de Dee *et al.* [99] en el que se propone la construcción de "mapas visuales" de los videos utilizando los parámetros de movimiento de los objetos. Esto permite la clasificación de las distintas regiones de la escena en función de los patrones de movimiento observados dentro de ellas.

Para ello, el video es preprocesado en una primera etapa consiguiendo extraer información sobre objetos en movimiento, su tamaño y trayectorias. Con esa información se crea, para cada cámara (modo de aprendizaje), un modelo de las trayectorias de la escena y los objetos son identificados sobre la base de los

parámetros de su movimiento (modo de operación). Por último, el razonamiento semántico se realiza sobre esta interpretación permitiendo así detallar la escena completa, requerimiento número 2.

El razonamiento semántico se lleva a cabo con un alto nivel de abstracción utilizando conceptos humanos (como "un coche debe moverse a lo largo de un carril" o "hay una alarma si un coche se encuentra en una acera"), verificando así el requisito número 3.

Además, un sistema basado en un razonador semántico es capaz de realizar la interpretación de las imágenes, siendo posible de forma fácil un cambio de dominio de aplicación (requisito número 5) o la inclusión de información procedente de sensores u otras fuentes (requisito número 4) simplemente especificándolo en la ontología apropiada.

Por eso, la estrategia que se va a utilizar se aparta de la utilización de estos algoritmos de reconocimiento de objetos para basarse en una caracterización semántica de los mismos en función de sus patrones de movimiento.

La idea principal es que es posible extraer el significado semántico de un objeto y el comportamiento empleando los parámetros identificados por los sensores y las cámaras. Por lo tanto, es posible implementar una ontología que contenga todo el conocimiento de un dominio (tráfico, detección de incendios, etc.) y utilizar la caracterización semántica de la escena para determinar situaciones normales y anormales y comportamientos.

Para llevar a cabo estas tareas el sistema realiza un proceso en dos etapas: aprendizaje y operación.

Durante la primera etapa, de aprendizaje, se genera automáticamente un modelo de las trayectorias de los objetos que se mueven en la escena etiquetando cada una de ellas (carretera o acera, por ejemplo, serían algunas de las etiquetas que se podrían utilizar en el dominio del control de tráfico). Para ello se utiliza un algoritmo de detección de rutas y una ontología específica para ese ámbito de aplicación (programado manualmente por un operador humano). En esta etapa, tomando como entrada las rutas que se van descubriendo, en función del movimiento y el tamaño relativo de los objetos detectados por la cámara, se identifican las diferentes zonas de la escena.

En la segunda etapa, de operación, se utiliza también un modelo de conocimiento del dominio del problema para etiquetar los objetos en movimiento que aparecen

en la escena actual y tratar para discernir las situaciones normales de las alarmas. Se aplica sobre la caracterización de zonas de la escena obtenida en la etapa anterior y la información en tiempo real de los objetos en movimiento, que de nuevo, a través de su tamaño relativo, se asignan a los diferentes actores contemplados.

En ambos casos, el modelo ontológico especifica conceptos que definen, por ejemplo, el comportamiento normal de los coches (como "*los coches deben estar en la carretera*") son siempre los mismos, independientemente de la carretera concreta que la cámara supervisa. Esto significa que hay una sola ontología para cada dominio de vigilancia (control de tráfico, detección de incendios, vigilancia perimetral, etc.) compartida por todas las cámaras que miran una escena relacionada con ese campo de aplicación.

Simplemente cambiando este núcleo de conocimiento, totalmente separado del sistema desarrollado, se consigue el funcionamiento en diferentes dominios. Las ontologías especifican las diferentes entidades involucradas en cada ámbito de conocimiento concreto, sus propiedades y las relaciones entre las mismas de una manera altamente estructurada y lógica, agrupándolas en clases. Gracias a esta estructura, un razonador es capaz de aplicar normas lógicas para inferir conocimiento que se encuentra implícito en el sistema, pero que no ha sido incluido de forma explícita. Una ontología que puede utilizarse como ejemplo sencillo es la que modela las relaciones familiares: una entidad de la clase persona puede relacionarse con otra entidad diferente por medio de las relaciones "*ser hijo de*", "*ser hermano de*" o "*ser tío de*". En el caso en el que la persona A sea hijo de B, y la persona B sea hermano de C, el razonador puede inferir que la persona C es tío de A si la ontología está correctamente definida.

3.2.1 Visión global

El sistema propuesto se muestra en la Figura 3.1. Se basa en una arquitectura de tres etapas:

- Un módulo de sensorización: formado por redes de cámaras de vigilancia inteligentes y/o sensores (incendio y detectores de movimiento, por ejemplo) que obtienen la información de la escena.
- Un módulo de detección de rutas: capaz de determinar la disposición de la escena en función del movimiento y dimensiones de los objetos.

- Un módulo de modelado ontológico e inferencia: encargado de traducir el modelo generado anteriormente y las lecturas de las cámaras y sensores en tiempo real a los términos semánticos para introducirlos en la ontología, razonarla y modelar la escena y poder identificar una situación de alarma cuando se está produciendo.

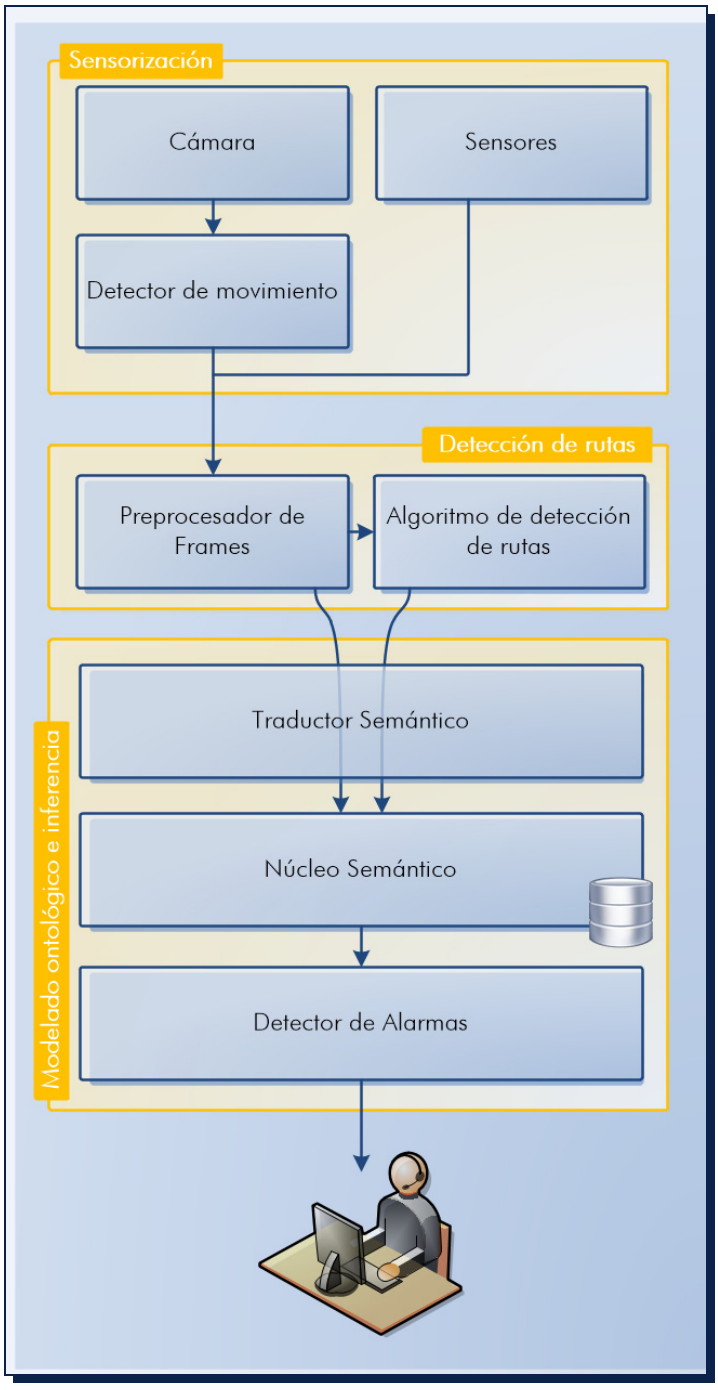


Figura 3.1. Arquitectura a alto nivel del sistema.

En los siguientes subapartados se presentan los diferentes módulos para en los Capítulos 4, 5 y 6, realizar una descripción detallada.

Como se ha introducido, en la Sección 3.2, el funcionamiento se realiza en dos etapas: aprendizaje y operación.

En la primera de ellas el módulo que realiza la detección de rutas, haciendo uso de los parámetros de movimiento de los objetos proporcionados por el módulo de sensorización. Con estos datos va construyendo un modelo de regiones. Estos patrones y las características de los objetos en movimiento se envían al módulo de modelado ontológico para que vaya etiquetándolos. En el ejemplo del control y gestión de tráfico, los objetos en movimiento que aparecen en la imagen, en función de sus parámetros, serán clasificados por el modelo ontológico como peatones, vehículos, etc., y las zonas por las que transcurren como aceras o carreteras en función de quien transite por ellas.

Una vez etiquetadas las distintas zonas descubiertas se pasa a modo funcionamiento (el módulo de detección de rutas deja de estar activo). En esta etapa el módulo de modelado ontológico toma como entrada las regiones ya clasificadas y los parámetros de los objetos que se van descubriendo (cada nueva detección de objetos se produce una proceso de razonado), clasificando los nuevos. Una vez conocidos los objetos el mismo proceso de inferencia determina si se han producido situaciones de alarma ya que conoce los comportamientos habituales de los distintos elementos incluidos en la ontología.

En esta Tesis Doctoral, para las tareas realizadas por el módulo de sensorización, se van a utilizar sensores comerciales proporcionados por uno de las empresas participantes en el proyecto HuSIMS. Por tanto, el trabajo realizado se centra en el diseño e implementación del proceso de detección de rutas y el modelado ontológico e inferencia.

3.2.2 Principales Entidades

3.2.2.1 Sensorización

Durante esta etapa una red de sensores formada por cámaras de vigilancia inteligentes y/o sensores (de temperatura y detectores de movimiento, por ejemplo) obtiene la información de la escena. Las propias cámaras ejecutan algoritmos de detección de movimiento para transformar el flujo de video en paquetes de datos que contienen parámetros de los diferentes objetos móviles (velocidad, posición,

tamaño, etc.). Es muy importante tener en cuenta que las cámaras que se van a utilizar captan imágenes de baja resolución.

El procedimiento comienza determinando los objetos en movimiento. Una vez identificados los objetos mediante algorítmica sencilla se obtienen sus parámetros principales. Además de las características básicas, cada objeto posee un identificador para poder realizar el seguimiento del mismo en los distintos fotogramas en los que aparece.

El conjunto de datos, que en un momento determinado se obtiene de los objetos en movimiento de un fotograma, se empaqueta para su posterior envío.

Las imágenes monitorizadas por las cámaras son procesadas en ellas mismas por lo que, salvo petición expresa, nunca se envía el video capturado. Con ello se consigue que no sean visualizadas o procesadas por ningún operador humano permitiendo mantener la privacidad de las personas captadas.

3.2.2.2 Detección de Rutas

En esta etapa se realiza la caracterización de la escena vista por cada cámara en términos de rutas (zonas de la imagen a través de la cual los objetos generalmente transitan), y fuentes/sumideros de objeto (zonas de la imagen en la que los objetos generalmente aparecen/desaparecen, respectivamente). Por lo tanto, después del periodo de aprendizaje, cada cámara ha construido un modelo de ruta/fuente/sumidero de la escena que observa. Cada uno de los elementos tiene un significado diferente dependiendo del dominio específico de vigilancia. En el control y gestión de tráfico, los objetos se desplazan por carreteras y aceras (rutas) y un semáforo o la entrada a un garaje, por ejemplo, son zonas de aparición/desaparición de objetos (fuentes y sumideros).

Para identificar las rutas, el módulo contiene información acerca de cada una de las trayectorias de los objetos, es decir, la historia de puntos a través de la que se ha movido. Cuando el móvil desaparece de la imagen, esa trayectoria se agrupa con la ruta más similar de las identificadas con anterioridad o genera una nueva ruta si no hay correspondencia con ningún grupo previo.

El mismo criterio se sigue para identificar fuentes y sumideros. El módulo de aprendizaje construye una historia de los puntos en que los objetos han aparecido y desaparecido. Estos conjuntos de puntos se agrupan juntos utilizando un algoritmo de clustering. Cuando un grupo determinado contiene más de un número

configurable de puntos iniciales o finales, se considera una fuente o sumidero, respectivamente.

Para la implementación de este proceso se utilizan dos submódulos internos. En primer lugar, el Preprocesador de *Frames* (fotogramas) recibe de la cámara un archivo con los parámetros de movimiento de los objetos detectados por la cámara, separa los datos de cada fotograma (un solo archivo puede incluir la información de varios *frames* para optimizar las comunicaciones), corrige la distorsión de perspectiva usando altura y los valores de ángulo de inclinación de la cámara de origen mediante la aplicación de una asignación de perspectiva inversa descrita por Mallot *et al.* en [100], y vuelve a formatear la información para almacenarla en una matriz de datos sin procesar. De esta matriz de datos, el Algoritmo de Detección de Ruta, utilizando un conjunto de rutinas, determina las rutas/fuentes/sumideros de la escena.

3.2.2.3 Modelado Ontológico e Inferencia

El objetivo de esta etapa es traducir los parámetros sintácticos de los objetos, las rutas, los sumideros y fuentes obtenidas por las cámaras y módulo de Detección de Rutas en clases semánticas ("coche" en lugar de "objeto", "carretera" en lugar de "ruta"), realizar un modelo de la escena e identificar cualquier situación de alerta (un "coche está en la acera") de acuerdo con la ontología y reglas semánticas (un modelo de conocimiento formal especificado por un humano).

El Traductor Semántico convierte la información sintáctica en datos semánticos formales (de acuerdo a los formatos estándar de la *Web* Semántica) y rellena la ontología con ellos, es decir, es el encargado de interpretar los parámetros de movimiento, rutas y fuentes/sumideros y darles un significado de acuerdo al modelo desarrollado para cada dominio.

El módulo está activo tanto durante el tiempo de aprendizaje (cuando el motor semántico está construyendo el modelo de ruta de la escena) como el de operación. Pasado el tiempo de aprendizaje, el traductor se emplea para generar el modelo semántico de la escena y poblar la ontología con él, y al tiempo de la operación, la traducción se utiliza para transformar los parámetros de movimiento de los objetos en los datos semánticos (identificar, por ejemplo, si un objeto es un vehículo o un peatón, y en qué carril se encuentra), y también poblar la ontología.

Después de la traducción, el Núcleo Semántico procesa la ontología (recientemente populada con nuevos individuales) gracias a un razonador semántico para inferir

propiedades nuevas acerca de los objetos, identificarlos y reconocer si hay una situación de alarma. Para ello hace uso de ontologías y un conjunto de reglas.

La ontología es un modelo de conocimiento que especifica formalmente (utilizando OWL-*Web Ontology Language*) y en detalle el comportamiento esperado en condiciones normales de los elementos reconocibles de la vigilancia de dominio, por lo que es posible inferir que un elemento no funciona con normalidad. En la ontología se detallan las condiciones de una situación normal en términos de clases y las relaciones entre ellos, por ejemplo, en el ámbito de control de tráfico, tales como:

- Los Coches en movimiento están en la Carretera.
- Los Coches no se detienen por más de 2 minutos en una Carretera.
- Las Personas no se detienen en Carreteras.

Las reglas complementan la ontología relacionando los parámetros de los objetos con el modelo para identificar a los individuos dentro del dominio. Por ejemplo, en el dominio de control de tráfico, para distinguir si una ruta es carretera o acera, las reglas pueden determinar que si el tamaño medio y la velocidad de los objetos que van a través de una ruta específica son más altos que la media de los tamaños y las velocidades de todos los de las rutas de la imagen, la ruta se considera una carretera, y si no, una acera.

Después del razonamiento semántico de la ontología poblada con individuales y complementada con las reglas se dispone de un modelo de la escena. Siguiendo con el dominio del control de tráfico, este modelo podría ser:

- O1, O3 y O4 son Coches
- O1, O3 y O4 están en R1
- R1 es una Carretera
- O2 es una Persona
- O2 está detenida en R1

A continuación, el razonador identifica una alarma al detectar que O2, al ser una Persona está detenida en un lugar donde no debe estar.

Una vez detectadas todas las alarmas de la escena actual el Detector de Alarmas las procesa y muestra por pantalla.

3.3 Conclusiones

Los sistemas de caracterización de situaciones y comportamientos actuales presentan ciertas restricciones en su funcionamiento. La arquitectura presentada parte de la que quizá es la mayor limitación de los sistemas existentes hasta ahora, la falta de flexibilidad para adaptarse a diferentes dominios. No es tarea sencilla diseñar un mecanismo que pueda adaptarse fácilmente a las distintas situaciones y comportamientos que puedan producirse dentro de las *Smart Cities*.

Con objeto de conseguir que la adaptación sea lo más sencilla y automática posible ésta se realiza utilizando semántica. Mediante un modelo del dominio de conocimiento denominado ontología y un conjunto de reglas que lo complementan se consigue la independencia del sistema del dominio de funcionamiento.

Respecto a las restricciones, debido al uso de cámaras sencillas que captan imágenes de baja resolución y con poca capacidad de procesamiento, la solución planteada propone la utilización de los parámetros de movimiento de los objetos de la imagen reduciendo el volumen de las transmisiones. Utilizando esta información como entrada del algoritmo de detección de rutas propuesto se identifican las diferentes zonas de la imagen.

Esta información y la procedente de la cámara se introducen en el núcleo de conocimiento que define el dominio en forma de objetos de una determinada clase. Si el modelo semántico está preparado para tratar información de objetos de otras clases, es muy fácil incluir la información complementaria procedente de otro tipo de sensores. La ontología es capaz de diferenciar las clases y procesarlas adecuadamente.

Como resultado del proceso de inferencia, la arquitectura proporciona una escena modelada en la que cada zona y objeto está correctamente identificado. Una vez se ha caracterizado la escena, es fácil encontrar situaciones extrañas ya que se conoce cuál debe ser el comportamiento de los diferentes objetos en cada una de las regiones. Además, como este procesado se realiza también semánticamente, los resultados obtenidos son en lenguaje natural, entendibles por un operador humano.

La arquitectura expuesta cumple todas las exigencias. Un módulo de modelado ontológico e inferencia es el punto clave y da al sistema toda la flexibilidad que requiere. Este núcleo puramente semántico permite, no sólo el modelado de la escena y la detección de alarmas, sino también el reconocimiento de los diferentes objetos dentro de la misma, cualidad todavía no explotada y que normalmente se

realizaba utilizando visión artificial. El resto de la arquitectura gira en torno a este elemento principal amoldando de forma genérica la información para que el sistema sea totalmente adaptable.

4

PROCESADO DE IMAGEN: REDES DE SENSORES VISUALES INTELIGENTES

Para la monitorización de las escenas se necesita el despliegue de un gran número de sensores visuales inteligentes pequeños, baratos y no intrusivos (baja resolución), capaces de comunicarse de forma inalámbrica con un centro de control. Se habla de sensor visual y no de cámara por sus características de análisis de imagen embebidas, aunque al tratarse de una distinción sutil, a lo largo de esta Tesis Doctoral se utiliza sensor visual o cámara de manera indistinta para referirse a los sensores visuales utilizados.

Dentro de HuSIMS se utilizan sensores visuales con una sensibilidad óptima. Estos sensores pueden funcionar incluso en condiciones visuales extremas, tanto en interiores como al aire libre, independientemente de la iluminación y de la

meteorología, son de bajo coste y tienen requerimientos energéticos bajos. Para el proyecto HuSIMS, Emza, uno de los socios participantes en el mismo, ha realizado modificaciones sobre su sensor visual comercial *WiseEye*, para conseguir una plataforma capaz de cubrir todos los requisitos antes mencionados.

En este Capítulo se realiza una descripción general de las características y funcionamiento de los sensores visuales utilizados y las tecnologías utilizadas para su comunicación con el centro de control.

Este Capítulo se encuentra dividido en cuatro secciones. La Sección 4.1 hace una breve introducción de la evolución de los sistemas de visualización utilizados en videovigilancia para resaltar la importancia del procesado de imagen con el fin de conseguir una detección automática. En la Sección 4.2 se describe el funcionamiento de los sensores inteligentes empleados en el proyecto HuSIMS. En la Sección 4.3 se definen las tecnologías utilizadas para la transmisión de la información procedente de las redes de sensores al centro de control. Finalmente, en la Sección 4.4 se exponen las principales conclusiones del Capítulo.

4.1 Las cámaras y el procesado de imagen aplicado a videovigilancia

Una forma natural muy importante de obtener información sobre el mundo que nos rodea es a través de la vista. Incluso los pequeños animales, aves e insectos pueden interpretar fácilmente el mundo visual con menor potencia de cálculo que la que posee un ordenador normal. Por desgracia, los sistemas de visión artificial actuales son generalmente costosos, y en lugar de tener capacidades cognitivas, a menudo se limitan a la grabación de imagen.

El concepto de la utilización de cámaras en seguridad se remonta a la década de los 40, aunque no es hasta los años 70 cuando aparecen los primeros sistemas de videovigilancia analógicos comerciales [101]. Inicialmente, una persona tenía que vigilar continuamente el flujo de video. Con la introducción de los grabadores de video y más tarde de video multiplexado se permite una mayor flexibilidad en la visualización, almacenamiento, y recuperación del video (analógica). Las cámaras de video digitales se introdujeron en la década de los 90 y en los últimos años sus ventas han superado las cámaras analógicas en el mundo de la vigilancia. En [102] se hace una revisión exhaustiva de la evolución histórica de los sistemas de videovigilancia.

Se determinó que la observación humana de los videos de vigilancia es ineficaz [103] y el análisis automatizado de video se convirtió en un tema de interés [104]. Los algoritmos utilizados a menudo se basaban en una sencilla diferenciación de fotogramas que debía superar un umbral para considerar la zona como objeto en movimiento, llamados habitualmente de detección de movimiento en video (VMD - *Video Motion Detection*). Las primeras aplicaciones estuvieron centradas en la monitorización de tráfico [105] y detección de intrusos [106].

Estos sistemas de análisis automatizado de secuencias de video por lo general utilizaban una arquitectura separada, donde las cámaras se limitaban a la transmisión de video en vivo a una instalación central con un servidor. En sistemas más avanzados, estos servidores, aplicando complejos algoritmos de procesado de imagen pretenden detectar bordes, seguimiento y reconocimiento de objetos e incluso de gestos, utilizando para ello operaciones de cálculo y matemáticas complejas [6][107]. Hay muy pocos productos en los que el análisis se realiza en la propia cámara que visualiza la imagen.

Otro enfoque destinado a reducir la monitorización de los entornos son los sensores de vigilancia híbridos. Estos sensores permanecen en suspensión hasta que se activan cuando detectan movimiento. Una vez más, estos sensores sólo son un instrumento para la grabación de imagen o la transmisión de la misma ya que no disponen de capacidad de análisis.

Aunque hay sistemas que afirman que realizan un análisis automatizado de video para la identificación de personas sospechosas o la detección de objetos perdidos en lugares concurridos, estos productos todavía tienen que penetrar en el mercado y ganar fuerza comercial.

En la comunidad investigadora, el tema de las "cámaras inteligentes" y "cámaras inteligentes integradas" ha ganado un gran interés [102][108][109]. Estudios recientes se centran en el seguimiento realizado por una red de cámaras inteligentes integradas [110][111] y arquitecturas de hardware específicas [112]. Algunos de los diseños más actuales para los sistemas de vigilancia de bajo consumo combinan un sensor estéreo de baja resolución con una cámara a color de alta resolución [113], el desarrollo de algoritmos ligeros para cámaras inteligentes integradas [109][114], o emplean las redes de cámaras inteligentes para la aplicación en agricultura [115].

4.2 Sensores visuales inteligentes

El sensor visual inteligente desarrollado para HuSIMS es novedoso en varios aspectos. En primer lugar, normalmente (salvo por solicitud expresa de los sistemas de emergencia si lo consideran necesario) no produce salida de video, sólo un fichero XML (*eXtensible Markup Language*- http://www.w3.org/XML/) con la descripción de la actividad observada en la escena que supervisa. En segundo lugar, se propone reducir drásticamente el tamaño, precio, ancho de banda requerido, el consumo de energía, y la complejidad de la instalación, manteniendo el rendimiento de dispositivos de gama alta en diferentes condiciones meteorológicas y de iluminación.

Este sensor visual se basa en los componentes y la arquitectura estándar, con un sensor CMOS (*Complementary metal-oxide-semiconductor*) CIF (*Common Intermediate Format*) o VGA (*Video Graphic Array*) (similar al de los teléfonos móviles) y el procesador ARM9 que ejecuta algoritmos de cálculo capaces de procesar 10-15 fotogramas por segundo (fps – *frames per second*) con bajos consumos de potencia. Todo esto permite que los sensores sean desplegados en áreas extensas cubriendo cada uno un área limitada de la ciudad (por ejemplo, un cruce de carreteras, la entrada de un bar, o una parada de autobús).

Cada sensor visual controla el movimiento de los objetos en una región de aproximadamente 30x30 metros con formato de video VGA. Estos sensores visuales al trabajar con baja resolución, permiten preservar la intimidad de las personas, ya que no pueden reconocer las caras (una característica muy importante para una implementación real de la ciudad inteligente). Aun así, el video es procesado en la propia cámara usando algoritmos de detección de movimiento capaces de identificar objetos en movimiento y determinar sus parámetros (dirección, tamaño, velocidad, etc.). El resultado de este proceso es una secuencia de ficheros XML que contienen la información de los diferentes objetos en movimiento en cada fotograma.

La arquitectura está pensada para maximizar los datos obtenidos a partir de cada píxel, permitiendo la reducción de la resolución del sensor visual y la potencia del procesador. El algoritmo utilizado en este proyecto se basa en un procesamiento en tres capas: análisis a nivel de píxel, fase de segmentación y capa de análisis de objetos y movimiento.

Una innovación que introduce este diseño es el procesado a bajo nivel o píxel. Esta parte es la que demanda mayor carga computacional por lo que es necesario que

sea lo más eficiente posible. Este enfoque requiere de una simplificación de los bloques de cálculo, en contraste con los mecanismos tradicionales de procesamiento de imágenes que tienden a considerar el dispositivo de adquisición de imágenes como una herramienta de medición y el procesamiento de sus datos como una aplicación de herramientas matemáticas rigurosas, tales como detectores de bordes gaussianos, análisis de Fourier, etc. Estas operaciones necesitan aritmética de punto flotante y complejas arquitecturas o hardware específico, tales como DSPs (*Digital Signal Processor*) o FPGAs (*Field Programmable Gate Array*) con consumos elevados de potencia.

En HuSIMS el procesado a nivel de píxel se inspira en la arquitectura de procesamiento visual de la naturaleza, que utiliza un gran número de receptores analógicos simples, cada uno sensible a un aspecto particular de la visión: color, contraste, movimiento, dirección, resolución espacial, etc. Los sensores utilizados son estáticos y monitorizan en una escena fija. Para cada píxel se debe determinar si para un fotograma específico, la intensidad de la luz que incide sobre él es regular o irregular, con respecto valores históricos de la misma en un intervalo previo (varios segundos a unos pocos minutos). En las aproximaciones tradicionales se necesita almacenar los históricos de niveles de intensidad y calcular las estadísticas pertinentes. Sin embargo, esto implica la necesidad de recursos de almacenamiento significativos y alta carga computacional. El enfoque utilizado en HuSIMS capta el comportamiento histórico de píxeles utilizando umbrales que definen un límite inferior y un límite superior. Estos topes definen la envolvente de las intensidades esperadas de una señal de píxel de entrada "regular". Este algoritmo sólo emplea la aritmética de enteros, incrementos/decrementos, comparaciones y búsquedas en tablas, por lo que el sensor es capaz de analizar el video en tiempo real con un procesador de bajo rendimiento ARM9.

La Figura 4.1 muestra un ejemplo de comportamiento de los umbrales como una función del perfil de intensidad de píxel. Cuando la intensidad de píxel excede el umbral superior (como en el fotograma 1500 en la Figura 4.1 donde se sobrepasan las 100 A.U. (*Arbitrary Units*) siendo el límite superior aproximadamente 30 A.U.) o cae por debajo del umbral inferior, se convierte en un “píxel caliente”.

La siguiente fase, la segmentación, identifica conjuntos de “píxeles calientes” conectados. Es muy importante determinar en cualquier sistema de vigilancia el tamaño mínimo de los objetos detectables en píxeles. Por ejemplo, para un tamaño de objeto mínimo de 8 píxeles, la probabilidad de que en un suceso aleatorio ocho píxeles sean calientes es p^8 donde p es la probabilidad aleatoria de que un píxel sea

caliente. Estableciendo p = 1/100 y suponiendo formato VGA (640 x 480 píxeles) y 30 fps, un solo caso de 2 × 4 píxeles aleatorios (en cualquier parte del cuadro) se producirá una vez en $100^8 = 10^{16}$ eventos. Esto es equivalente a 30 fps × 31536000 segundos / año × 307.200 píxeles VGA × 34,5 años.

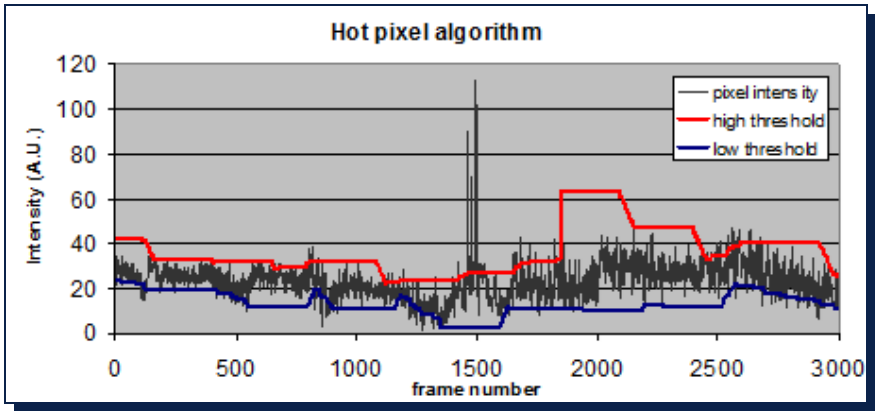


Figura 4.1. Algoritmo de "píxel caliente": muestra los umbrales superior e inferior y su comportamiento adaptativo como una función del perfil de intensidad de píxel [116].

Así, con píxeles sintonizados a la sensibilidad de 1:100 fotogramas, se puede lograr una buena detección con bajas falsas alarmas inherentes, maximizando el uso de cada píxel y evitando la necesidad de un gran número de píxeles, lo que reduce aún más el consumo de potencia.

El análisis y la experiencia muestran que el factor óptimo de p es 0,01. Podría considerarse un valor muy general, permitiendo a los sensores operar bien en gran variedad de aplicaciones, incluyendo la detección de intrusos o la seguridad en el hogar. Pero se puede comparar con el proceso natural de la sensibilidad retiniana adaptativa, que es un proceso universal que funciona de manera similar tanto en interiores como al aire libre, noche o día, en zonas urbanas, regiones con vegetación o desérticas. La reducción de este valor (por ejemplo a 0,001) disminuiría drásticamente la sensibilidad del sistema, mientras que el aumento (por ejemplo a 0,1) da lugar a numerosas falsas alarmas.

Por último, la fase de detección de objetos y movimiento se ejecuta cuando se ha detectado al menos un grupo de píxeles con tamaño superior al mínimo en la fase anterior. En ese caso se le aplican a la imagen una serie de filtros para eliminar los efectos de las sombras, brillos, etc., y conseguir una detección y seguimiento de objetos adecuada.

Por otro lado, es posible completar la información de las cámaras con otros tipos de sensores, tales como detectores de humo, humedad y/o acelerómetros, para

permitir una detección más precisa de las situaciones peligrosas y la detección de alarmas que no son fácilmente identificables únicamente con las cámaras.

4.3 Red de comunicaciones

Los datos recogidos por los distintos sensores se envían a un centro de control donde son procesados. En casas y edificios cuya infraestructura incluye cables coaxiales, cables eléctricos y cables de la línea telefónica se puede reutilizar infraestructura para un despliegue masivo de los sensores más rápido.

En escenarios al aire libre, las redes inalámbricas son la mejor solución. Las tecnologías inalámbricas como Wi-Fi (802.11) y WiMAX (802.16) permiten agregar y colocar cámaras y sensores en lugares que antes eran inaccesibles, y ofrecen una calidad de servicio (QoS – *Quality of Service*), de alta capacidad y disponibilidad, mecanismos de cifrado de datos y conectividad de baja latencia esencial para transmisiones en tiempo real.

Uno de los principales objetivos de HuSIMS es que el sistema sea rentable y pueda desplegarse en zonas amplias y heterogéneas. El despliegue de soluciones híbridas que incluyen tramos inalámbricas combinados con espacios cableados (por ejemplo sensores visuales con Wi-Fi vinculados a la red de la línea eléctrica tecnologías PLC (*Power Line Communications*)) permite un despliegue inalámbrico de bajo coste y rápido reutilizando el cableado existente.

Por otro lado, para el proyecto se han desarrollado redes con características SON (*Self-Organizing Networks*) como la auto-configuración para conseguir despliegues más rápidos o la posibilidad de reducir el impacto ante fallos buscando de forma automática caminos alternativos para el envío de la información. En lugar de utilizar redes de malla cuyo rendimiento se degrada rápidamente en escenarios multi-punto, el sistema emplea nodos de acceso inalámbrico punto a multipunto basados en el protocolo 802.11n. En cuanto a la planificación de la red, los sensores siempre son capaces de llegar a más de un nodo de acceso con el fin de proporcionar rutas redundantes de acceso al centro de control. Por otra parte, en interiores, las tecnologías de comunicaciones que se utilizan son Wi-Fi y PLC una como la infraestructura de seguridad de la otra con el fin de disponer siempre de conectividad.

4.4 Conclusiones

En este Capítulo se han introducido los sensores visuales inteligentes, la alternativa actual a las cámaras de videovigilancia tradicionales. Estos sensores tienen la característica especial de realizar el procesado de la imagen que captan ellos mismos, sin necesidad de transmitir la imagen a un centro de control. Esta ventaja permite mantener la privacidad personal y requiere anchos de banda de transmisión bajos, ya que no es necesaria una transmisión de la imagen en alta resolución en tiempo real sino sólo los resultados del procesado realizado en formato texto.

Por otro lado, el procesamiento que realizan es sencillo, no utilizan algoritmos complicados de procesado de imagen con alta carga computacional lo que hace que la potencia consumida se reduzca considerablemente y el precio del dispositivo se minimice.

Para la transmisión de la señal se recurre a un enfoque híbrido, inalámbrico-cableado con un mecanismo de auto-configuración y auto-adaptación para conseguir mejores tiempos de instalación y robustez ante posibles fallos.

Este Capítulo es meramente descriptivo y se ha incluido para entender de forma general el funcionamiento del sistema completo. Por otro lado, los desarrollos e implementaciones de este módulo dentro del proyecto son confidenciales y por tanto no se ha podido hacer una reseña más exhaustiva del hardware y los componentes de red.

5

MODELADO ESPACIAL DE LA ESCENA: DETECCIÓN DE RUTAS

La capacidad de identificar y razonar los sucesos de las secuencias de video está condicionada por la validez del modelo del escenario. A su vez, para un correcto modelado de la escena es necesario descomponerla en las diferentes regiones espaciales de interés. Estas regiones se van a poder caracterizar por su apariencia pero también por el comportamiento de los objetos que hay en ellas.

El enfoque de la detección de las diferentes zonas de la arquitectura propuesta se basa en la caracterización de la escena vista por cada cámara en términos de rutas (zonas de la imagen a través de la cual los objetos se desplazan), y fuentes/sumideros (zonas de la imagen en la que los objetos generalmente aparecen/desaparecen, respectivamente). Por lo tanto, después del periodo de

aprendizaje, cada cámara ha construido un modelo de ruta/fuente/sumidero de la escena que observa. Dependiendo del dominio específico, los elementos del modelo tienen significados diferentes. En el control del tráfico, las rutas serán carreteras y aceras, y los sumideros/fuentes representan (además de los bordes de la imagen), por ejemplo, la entrada de un aparcamiento, una puerta en un edificio, o un semáforo (dado que la cámara sólo detecta objetos en movimiento, cuando un coche se detiene en un semáforo, desaparece de la imagen a efectos de dicha detección de objetos).

Para construir un modelo de la escena se necesita reconocer las diferentes zonas mediante el seguimiento de los objetos individuales, estableciendo fotograma a fotograma una correspondencia. A partir de las posiciones en las que han estado los diferentes objetos, el módulo es capaz de construir trayectorias de los mismos y descubrir un patrón de movimiento. Después, acumulando trayectorias durante un periodo de tiempo se establecerá una norma de los movimientos típicos y esto permitirá el reconocimiento de movimientos atípicos y patrones de comportamiento.

El objetivo de este Capítulo es estudiar un método para utilizar el movimiento de los diferentes objetos dentro de las escenas con el fin de reconocer las regiones significativas en las mismas, ya que un modelado automático de las regiones en el escenario facilitará las interpretaciones al sistema de razonamiento posterior.

Este Capítulo se encuentra dividido en seis secciones. En la Sección 5.1 se recogen los mecanismos existentes para la detección de las diferentes zonas de la imagen. La Sección 5.2 describe cómo se realiza el modelado de las escenas en términos de rutas y fuentes/sumideros. La detección de las rutas y la identificación de las mismas en función del movimiento de los objetos se recogen en la Sección 5.3 y para las fuentes/sumideros en la Sección 5.4. En la Sección 5.5 se analizan los resultados de la validación del algoritmo propuesto. Finalmente, en la Sección 5.6 se exponen las principales conclusiones del Capítulo.

5.1 Mecanismos de detección de zonas

Para la identificación de las diferentes zonas que componen la escena existen diferentes técnicas, muchas de ellas basadas en el procesamiento de la imagen mediante algoritmos de visión artificial y/o comparación con modelos predefinidos [117]-[119]. Sin embargo para esta Tesis estas aproximaciones no son viables ya que con el objetivo de mantener la privacidad de los individuos y realizar un procesado de escenas con baja carga computacional no se dispone de imágenes en

alta resolución. En este caso los mecanismos a analizar están centrados en determinar los patrones de movimiento de los objetos para identificar zonas diferenciadas que posteriormente se etiquetan de forma automática utilizando una aproximación semántica.

En la literatura existen diferentes metodologías a la hora de afrontar el problema de la detección de las rutas o trayectorias habituales de los objetos en movimiento, sobre todo para su aplicación en videovigilancia.

Una de las primeras propuestas es la presentada por Fernyhough, Cohn y Hogg [120]. En ella se presenta un sistema que, partiendo de la creación de una base de datos de caminos, pretende detectar regiones para conseguir una representación espacial, modelado propuesto por Howarth y Buxton en [121]. El proceso se divide en tres fases: seguimiento de objetos, generación de caminos y generación de regiones.

En la primera etapa se detectan las formas de los distintos objetos en movimiento. Para realizar el seguimiento de los objetos, se asigna una etiqueta a cada objeto y en el fotograma siguiente se asignará la misma al objeto cuyo tamaño sea similar al anterior y se encuentre en una posición próxima.

En la segunda etapa los caminos se generan como la representación de los píxeles en los que un objeto ha estado posicionado a lo largo de la imagen. Mediante comparación, los caminos se actualizan, incrementando además el número que indica los caminos equivalentes, o en el caso en el que no haya similitudes con otros se crea uno nuevo. Al final se dispone de una base de datos que contiene la distribución más frecuente de los caminos permitiendo detectar ruido y determinar los más comunes.

Por último, se realiza un modelado espacial de la escena mediante la identificación de regiones. Se analiza la base de datos de caminos eliminando los poco frecuentes (los que tienen menor número de caminos equivalentes) e intentando fusionar caminos equivalentes que no se hayan detectado en el proceso de generación. Finalizado este proceso se dispone de áreas donde los objetos tienen comportamientos similares.

Con el tiempo, la base metodológica utilizada para el modelado de escenarios dinámicos (en los que los objetos aparecen, se mueven y desaparecen generando continuamente nuevas rutas) se va aproximando a la introducida por Makris y Ellis en [2]. Según esta aproximación, las escenas se podrían descomponer en zonas de

entrada y salida, zonas de parada y rutas o caminos que siguen los objetos en su movimiento.

Como zonas de entrada y de salida se señalan aquellas franjas por las cuales los objetos normalmente aparecen y desaparecen de la escena. Normalmente son zonas coincidentes (salvo las encontradas en los bordes del campo de visión de la cámara) seleccionando para su representación modelos GMMs (*Gaussian Mixture Models*) en 2D. Como mecanismo de aprendizaje, en [122] Makris y Ellis realizan la comparación entre los algoritmos de clustering K-means [123] y EM (*Expectation–maximization*) [124] decantándose al final de su estudio por el algoritmo EM por su comportamiento frente al ruido.

En cuanto a las zonas de parada, éstas se entienden como localizaciones en las que las velocidades son bajas o muy bajas respecto a unas predefinidas [2] o aquellas en las que, dentro de un área, el objeto permanece en ellas más de unos segundos [125]. En algunos casos las zonas de parada pueden necesitar para su caracterización una propiedad adicional que incluya la duración de la detección en cada una de las zonas, que puede ser aproximada por una función exponencial [2]. Estas zonas son utilizadas para filtrar el ruido y falsos positivos en las rutas ya que sólo consideran válidas aquellas que empiezan y acaban en una región considerada como entrada/salida.

Una vez definidos los puntos de interés, se pasa a la definición de los caminos. Para la determinación de las rutas existentes se utilizan los parámetros de movimiento de los objetos que realmente es una secuencia de medias que variará con el tiempo, típicamente la posición del objeto y su velocidad. A partir de estos parámetros, Morris y Trivedi en [126] proponen un modelo de tres pasos, que habitualmente es el utilizado por los diferentes autores para realizar este procedimiento. Preprocesado de las trayectorias, agrupamiento de los caminos similares y modelado de las rutas descubiertas son las fases a seguir. Los autores en su estudio hacen una recopilación de las técnicas que la comunidad científica utiliza para completar con éxito las tres fases. A continuación se recopilan algunas de las que proponen.

Las distintas velocidades de los objetos y la variación de las mismas con el tiempo (para una frecuencia de muestreo del video predefinida), hacen que se obtengan trayectorias de longitudes desiguales dentro de la misma ruta. Ante este problema, la mayoría de las investigaciones combinan la normalización de trayectorias con una reducción dimensional (habitualmente reducir las trayectorias a curvas en 2D),

para manejar trayectorias aleatorias de modo que se permita el uso de técnicas de agrupamiento estándar.

El proceso de normalización asegura que todas las trayectorias tengan la misma longitud. Para ello se realiza relleno con ceros [127] (incluir ceros al final de la ruta hasta conseguir la longitud deseada), extensión de rutas en función de los valores que ha tomado la misma hasta su fin [7][128], remuestreo (interpolación de la trayectoria original) [2][129]-[131] o técnicas de suavizado (eliminación de ruido y aplicación de técnicas de interpolación) [4][132].

En cuanto a la reducción dimensional, el mapeado de trayectorias permite lograr un procesado más manejable del espacio. Entre las técnicas de reducción dimensional se encuentran [126]:

- La cuantificación vectorial o *Vector quantization* (VQ): ignorando que las trayectorias son dinámicas y utilizando sólo coordenadas espaciales, se consigue la reducción limitando el número de trayectorias únicas utilizando una cantidad finita de vectores prototípico que las simbolizan [36][133].
- La transformada *Wavelet*: aplicando esta herramienta matemática se consigue suavizar las trayectorias manteniendo la forma y la estructura [4][134].
- *Hidden Markov Model* (HMM): asume que las trayectorias se producen por procesos estocásticos pudiendo utilizar este modelado para caracterizar las dependencias temporales entre los distintos puntos [135].
- El Análisis de Componentes Principales (PCA-*Principal Components Analysis*): estudia los parámetros que influyen en la variación de los datos ordenándolos en función de su importancia. Aplicando está técnica se proyectan las trayectorias en un subespacio que incluye la mayor parte de la señal eliminando direcciones y pequeñas variaciones de las mismas [136][137].
- Métodos espectrales: a partir de la matriz que recoge la similitud entre trayectorias y aplicando una serie de transformaciones se consigue una nueva matriz que incluye las nuevas trayectorias en el espacio espectral [138].

Como alternativa a la normalización, existen técnicas de métrica que permiten determinar la similitud entre trayectorias que no tienen por qué tener la misma

longitud. Ésto hace que no sea necesario el preprocesado previo para realizar la agrupación de trayectorias para formar rutas. Entre las técnicas utilizadas a la hora de determinar la distancia mínima existente entre trayectorias destacan la Distancia Euclidea [139], *Dynamic Time Warping* (DTW) [140][141], *Longest Common Subsequence* (LCSS) [142][143] y la Distancia de Hausdorff [132][144]. En [69][70] se realiza una comparativa de las diferentes técnicas para tratar de determinar, a través de la experimentación, cuál es la más robusta frente al ruido y eficiente para escenarios de videovigilancia en exteriores, llegando a la conclusión de que la mejor técnica la determina el escenario concreto.

Alternativas al cálculo de la similitud entre rutas son las técnicas de clustering que, en este caso, sí que requieren preprocesado. Hay distintas técnicas empleadas para la realización de este proceso recopiladas por Jain [145] y Berkhin [146] como indican Morris y Trivedi en [126]. Entre ellas se encuentran:

- Mejora iterativa: realiza una agrupación inicial que se va refinando con cada iteración. Típico de este grupo es el algoritmo de clustering K-Means [137][138] (o su variante FCM-*Fuzzy c-Means* [131]). Este algoritmo, conocido el número de grupos, selecciona de manera aleatoria el centro de los mismos y realiza grupos de muestras según la cercanía de las mismas a esos centros. En nuevas iteraciones, con los grupos establecidos se recalculan nuevos centros y se reasignan las muestras a este nuevo centro. Este proceso se repite hasta lograr la convergencia.

- Técnicas de adaptación en tiempo real: Como las trayectorias a agrupar se actualizan durante todo el proceso de aprendizaje, las uniones se modifican y actualizan en función de estas nuevas trayectorias que aparecen. Además, a priori se desconoce el número de grupos en los que se clasificarán las trayectorias. Dentro de este grupo se encuentran mecanismos como la utilización de umbrales [2][3] o I-kMeans [147].

- Clustering jerárquico: Hay dos aproximaciones, la acumulativa [136] y la divisiva [4][132]. En ambas se definen estructuras de árbol que establecen las relaciones de similitud siguiendo un procedimiento ascendente para el caso acumulativo o descendente para el divisivo. En la aproximación divisiva, por ejemplo, el nodo raíz incluye todas las trayectorias y los niveles inferiores van incrementando el número de agrupaciones (el siguiente nivel distribuiría las trayectorias en dos grupos, el siguiente en cuatro, y así sucesivamente). Esto

permite que se pueda "cortar el árbol" por donde se desee para establecer el número de agrupaciones que interese.

- Redes neuronales: Utilizan mapas SOM (*Self-Organizing Map*) [148] para realizar las agrupaciones [149][150]. Cada nodo de salida de la red neuronal corresponde a una ruta y nodos vecinos corresponden a las rutas más similares. Estas redes pueden ser entrenadas de manera secuencial y fácilmente actualizadas con nuevas trayectorias. Sin embargo, requieren de mucho tiempo para alcanzar la convergencia debido a la complejidad de las mismas y a las cantidades grandes de datos que necesitan.

Una vez realizado el agrupamiento de las trayectorias en una ruta se lleva a cabo la validación del mismo. Este paso es muy importante ya que se debe verificar la calidad del camino aprendido ya que a priori se desconoce el número real de rutas en la escena. Para ello existen diferentes técnicas, entre las que destacan los procedimientos de combinación por acumulación para agrupar clusters similares. Otras técnicas buscan el número correcto de clusters mediante la minimización o maximización de un criterio óptimo (partiendo de un número inicial de clusters, lo van variando hasta encontrar el que mejor ajusta ese criterio). En este grupo se encuentra el TSC (*Tightness and Separation Criterion*) [131] o el *Bayesian Information Criterion* o Criterio de Información Bayesiano (BIC) [151].

Cuando las trayectorias se encuentran agrupadas, las rutas o caminos resultantes son modelados. Para ello se siguen dos aproximaciones. En la primera de ellas se considera la ruta completa, desde el punto de inicio hasta el fin. Una ruta se modela a partir de un conjunto de puntos central complementado con dos envolventes que se extienden a lo largo del camino y que representan las variaciones de anchura de éste [2][128][130]-[132][134]. En la segunda aproximación se descompone el camino en partes más pequeñas que se denominan subpaths, bien sean diferentes trayectorias o subzonas delimitadas por intersecciones, por ejemplo [3][137]. El proceso de aprendizaje de las rutas es útil también como realimentación para funciones de bajo nivel que permitan eliminar sombras o determinar de una manera más fiable y robusta los caminos.

Una vez modelada la escena como composición de diferentes zonas se puede pasar a analizar los comportamientos y actividades de los objetos. Mediante la observación de la escena, un sistema puede ser capaz de determinar cercados virtuales, perfiles de velocidad, clasificar rutas, detecciones falsas, análisis de actividades y caracterizar la interacción entre objetos.

Basharat, Gritai y Shah [152] presentan sistemas donde, aplicando similares fundamentos, se consideran propiedades adicionales de los objetos en movimiento que intervienen en la escena. En él cada conjunto de observaciones contiene el tiempo, la localización (x, y), la anchura y la altura del objeto. Por medio del conjunto de observaciones, se modelan los patrones de movimiento en la escena gracias al movimiento y a los tamaños de los objetos. Dentro de las características de los objetos, se utilizan los datos de tamaño (anchura y altura) como fuente para determinar comportamientos anómalos y detectar objetos.

Anjum y Cavallaro [153] presentan un algoritmo de clusterización de trayectorias, que calcula patrones comunes en los comportamientos. El algoritmo sigue cuatro pasos principales: la extracción de un conjunto de características representativas de la trayectoria, la agrupación no paramétrica, unión de clusters y la fusión de información para la identificación de los patrones de movimiento considerados normales y especiales. Primero se transforman las trayectorias en un conjunto de espacios en los que la técnica Mean-shift identifica los grupos correspondientes. Por otra parte, se ideó un procedimiento de fusión para refinar estos resultados mediante la combinación de agrupaciones adyacentes similares. Los patrones comunes finales se estiman mediante la unión de los resultados de la agrupación a través de todos los espacios.

Cambiando la metodología, Johnson y Hogg [149] proponen la utilización de redes neuronales para modelar la distribución de las trayectorias. En este caso se define la trayectoria de los objetos como un conjunto de vectores de flujo que representan la posición y velocidad instantáneas calculadas a partir de la evolución de las coordenadas del centro del objeto. Un objeto que ha existido en n fotogramas es representado por un conjunto de n vectores de flujo 4D (coordenada x, coordenada y, velocidad x, velocidad y).

A continuación se modela la función de densidad probabilística (pdf) de los n vectores. Una alternativa es dividir el espacio en una red o cuadrícula n dimensional con un contador que se incrementa cada vez que un vector caiga en esa celda. Esta alternativa proporcionaría un modelo poco conciso por lo que se propone utilizar cuantificación vectorial (VQ-*Vector quantization*), método clásico para realizar esta tarea. La cuantificación vectorial se implementa utilizando dos redes de aprendizaje, la primera modela la distribución de los vectores de flujo y la segunda la distribución de las trayectorias. Para conectar estas dos redes y dotar de memoria a la arquitectura se introduce una capa de neuronas con fugas [154]. Utilizando esta

aproximación se detectan movimientos instantáneos atípicos en los objetos lo que permite identificar posibles incidentes.

Sumpter y Bulpitt [150] utilizan esta metodología pero van un paso más allá. Utilizan la salida a modo de realimentación para la capa de neuronas de fuga para aprender los patrones de activación.

Además, en este caso el sistema de vectores tendrá 10 a 20 dimensiones dependiendo del número de objetos a modelar y del número de parámetros usados para describir las variaciones de forma. Con el objetivo de reducir las dimensiones de los datos para el modelado de las formas se utiliza PDM (*Point-Distribution Model*). Cada objeto se representa como la composición de una forma previamente conocida y una serie de parámetros que definen las variaciones con respecto a ese modelo predefinido.

Este mecanismo permite predecir la trayectoria y las variaciones en la forma que experimentarán los objetos durante su transcurso a lo largo de la escena.

Una alternativa a lo planteado anteriormente es lo propuesto por Boyd, Meloche y Vardi [155]. En primer lugar se propone la utilización del método de Sudderth *et al.* [156] para conseguir una estimación de los píxeles del fondo e identificar los que se encuentran en primer plano. A continuación el sistema establece una conexión entre los grupos de píxeles que entren dentro de un cierto umbral. Con esto se obtiene una lista de centros de objetos situados en primer plano para cada secuencia de video. Además, a cada píxel se le asigna un parámetro intensidad que se corresponde con el número de veces que un centro es detectado en esa localización.

A continuación, con el objetivo de mapear la red dentro de la escena, se crea una red arbitraria de celdas hexagonales que se superponen a la imagen. A continuación se agrupan las celdas adyacentes con intensidades similares asignando a cada conjunto un número de nodo. Para establecer las relaciones entre los distintos nodos, se realiza un seguimiento de los centros de los objetos. Dos nodos están relacionados cuando para dos fotogramas consecutivos el centro del objeto pasa de uno a otro nodo. La escena es modelada como una red de regiones interconectadas. Adicionalmente se determina la media de la intensidad de tráfico basándose en estadísticas acumuladas en un periodo de tiempo.

5.2 Modelo espacial de la escena

El objetivo principal de este Capítulo es identificar las diferentes zonas de la escena para realizar un modelado espacial. Para ello se implementa un mecanismo similar al diseñado por Makris y Ellis en [2][130] basado en rutas y fuentes/sumideros:

- Rutas: zonas de la escena por las que pasan los objetos. Cada ruta puede considerarse como un grupo de trayectorias similares.
- Fuentes y sumideros: zonas por donde habitualmente aparecen (fuentes) o desaparecen (sumideros) los objetos.

De la metodología propuesta por estos autores de referencia se toma la idea de la detección de trayectorias, su agrupamiento para la formación de rutas y el concepto de fuentes y sumideros. Sin embargo, para el modelo propuesto se ha realizado una elección tecnológica y las modificaciones necesarias para su adecuación a los objetivos a conseguir. Estas decisiones se especifican en las Secciones 5.3 y 5.4 que detallan el diseño presentado.

Por otro lado, dependiendo del dominio del espacio que se esté caracterizando, estos conceptos tienen significados diferentes. Si, por ejemplo, se está visualizando una calle, las Rutas serán las carreteras y las aceras mientras que las fuentes y sumideros se encontraran en los bordes de las imágenes, semáforos, paradas de autobús, etc.

5.3 Trayectorias y rutas

5.3.1 Modelado de rutas

Cada ruta está caracterizada con:

- Una secuencia de puntos centrada.
- Dos envolventes que representan el tamaño medio de la ruta.

En la Figura 5.1 se puede ver la secuencia centrada de puntos en dos dimensiones, representada en color verde y las envolventes en amarillo. También aparecen en amarillo los vectores normales a la dirección de la ruta.

La dirección de la ruta está marcada en la Figura 5.1 con una punta de flecha al final. Así, un camino en el que los objetos viajan en ambas direcciones esta representado por dos rutas, una para cada dirección. Las envolventes de las rutas se

obtienen utilizando los valores de anchura y altura dados por la cámara de los objetos que se desplazan por ellas.

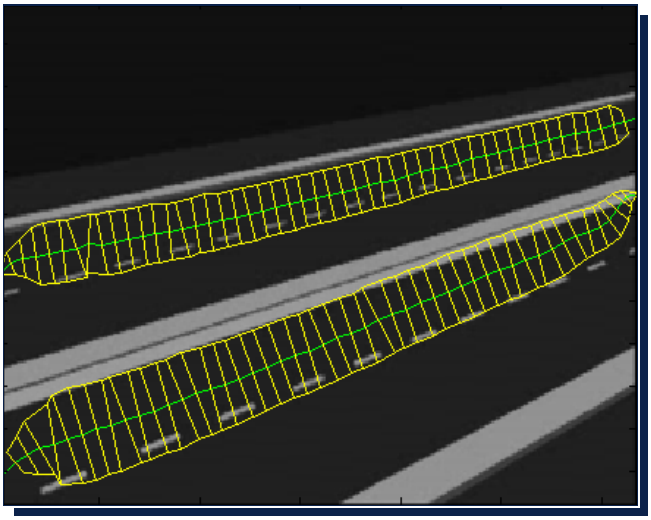


Figura 5.1. Modelado de rutas.

5.3.2 Identificación de rutas

En primer lugar, y antes de comenzar a explicar el procedimiento seguido para identificar de forma automática las diferentes rutas, se definen unos conceptos previos.

Según la Real Academia de la Lengua una trayectoria es *una "línea descrita en el espacio por un cuerpo que se mueve".* Siguiendo esa definición, de forma general se toma como trayectoria el conjunto de puntos por los que pasa un objeto (centro del objeto) en su movimiento a lo largo de la escena. De esta forma, para un objeto O_i, su trayectoria T_i es un conjunto de vectores t_{in} que representan la posición espacial de O_i en coordenadas *x* e *y* a lo largo del tiempo *k*:

$$T_i = \{t_{i1} \dots t_{in}\} \text{ donde } t_{ik} = \{x_{ik}, y_{ik}\} \qquad (5.1)$$

Considerando que el objeto aparece en la escena en el instante 1 y en el instante *n* desaparece.

La trayectoria de un objeto se "construye" según dicho objeto se desplaza por la escena. Se inicia con la aparición del objeto y concluye con su desaparición. Durante este proceso, la trayectoria del objeto es una "trayectoria abierta":

$$T_i = \{t_{i1} \dots t_{ia}\} \text{ con } a < n \qquad (5.2)$$

y pasa a ser una "trayectoria cerrada" ($T_i = \{t_{i1} \dots t_{in}\}$) cuando concluye el movimiento y desaparece.

Además, el sensor visual proporciona otros parámetros del objeto como sus dimensiones o velocidad en cada momento. De esta forma, para O_i, la evolución de su anchura (*width*) y altura (*height*), por ejemplo, están recogidos en los vectores:

$$W_i = \{w_{i1} \dots w_{in}\} \tag{5.3}$$

$$H_i = \{h_{i1} \dots h_{in}\} \tag{5.4}$$

Por otra parte aparece el concepto de "ruta". Una ruta es un conjunto de una o varias trayectorias cerradas. Las rutas incluyen un parámetro denominado "*strength*" (S) o fuerza que indica el número de trayectorias agrupadas en ella. Además, como se comentó en la Sección 5.3.1, las rutas están caracterizadas por una secuencia de puntos centrada (X, Y) y dos envolventes (E1, E2). Igualmente, se caracteriza por un vector de direcciones (D). Así pues, la ruta R_i posee los siguientes parámetros:

$$X_i = \{x_1 \dots x_N\} \tag{5.5}$$

$$Y_i = \{y_1 \dots y_N\} \tag{5.6}$$

$$E1_i = \{e1_{i1} \dots e1_{iN}\} \text{ donde } e1_{ik} = \{x_{ik}, y_{ik}\} \tag{5.7}$$

$$E2_i = \{e2_{i1} \dots e2_{iN}\} \text{ donde } e2_{ik} = \{x_{ik}, y_{ik}\} \tag{5.8}$$

$$D_i = \{d_{i1} \dots d_{iN}\} \text{ donde } d_{ik} = \{dx_{ik}, dy_{ik}\} \tag{5.9}$$

$$S_i \in \mathbb{N}\text{-}\{0\} \tag{5.10}$$

Donde N es el número de muestras de la ruta y es el mismo para todas con el objetivo de facilitar los cálculos y agilizar el procesamiento.

Además, las rutas también contienen el conjunto de parámetros de los objetos que se agrupan dentro de las trayectorias (tamaño medio, velocidad media máxima y velocidad media promedio, en el caso de control de tráfico u otras para otros dominios). Esto ayuda al Traductor Semántico a identificar el tipo de objetos que utilizan esa ruta, y así asignarle un significado propio. Por ejemplo, en un escenario de control de tráfico, rutas con tamaño y la velocidad superior que la media de

todas las rutas, pueden ser consideradas candidatas a ser carreteras y las demás, aceras.

5.3.2.1 Agrupamiento de trayectorias cerradas en rutas

Para la detección de las rutas y fuentes/sumideros, se sigue una estrategia análoga a la descrita en [2][3][69][122][126][130][157].

Normalmente se encuentran inconsistencias en los movimientos debidas a las diferentes interacciones: espacios concurridos, desviaciones de las trayectorias habituales, etc. Para evitar estas inconsistencias se eliminan las trayectorias cortas y aquellas de objetos cuya dirección cambia frecuentemente en pequeños periodos de tiempo. Para ello se desechan las trayectorias cerradas con pocos puntos o muy cortas (definido por umbrales) evitado que se tengan en cuenta objetos ruidosos.

Por otro lado, la distancia entre puntos consecutivos de una trayectoria varía considerablemente con la velocidad de los objetos. Por ello se interpolan linealmente las trayectorias para que todas tengan N muestras y así normalizarlas y hacerlas independientes de la velocidad. Es decir, cuando un objeto desaparece de la imagen, el vector con los puntos por los que ha pasado su centro se considera su trayectoria cerrada. Esta trayectoria se vuelve a muestrear para que contenga sólo un número fijo de puntos N (configurable).

Una vez determinadas y normalizadas las trayectorias válidas se agrupan las similares para formar Rutas.

Para medir esta semejanza entre trayectorias se pueden utilizar varios métodos que abarcan desde los más simples como la distancia Euclidea hasta otros más complejos como DTW (*Dynamic Time Warping*) y LCSS (*Longest Common Subsequence*). En la literatura hay autores como Zhang *et al.* en [70] y Morris y Trivedi [69] que estudian esta problemática y tratan de determinar cuál es el sistema de clustering de trayectorias más adecuado. Sin embargo, el método a utilizar depende de la escena a analizar consiguiéndose diferentes rendimientos para cada método en función de la situación y no pudiéndose seleccionar un mecanismo óptimo.

En este caso se ha utilizado la distancia de Hausdorff [158] propuesta por [132][144] para la medida de similitud espacial entre trayectorias y se incorpora la medida del ángulo que ayuda a diferenciar dos trayectorias afines en las que los objetos se mueven en sentidos contrarios (problema tratado ya en [70]).

La distancia de Hausdorff tradicional es una herramienta matemática capaz de medir la similitud de dos conjuntos de puntos. Formalmente, la distancia definida entre los conjuntos A y B, donde A y B son conjuntos de puntos se define como:

$$H(A, B) = \max\{h(A, B), h(B, A)\} \tag{5.11}$$

Donde:

$$h(A,B)=\max_{a\in A}\{\min_{b\in B}\{\|a - b\|\}\} \tag{5.12}$$

La función h(A, B) (ecuación (5.12)) se conoce como la distancia directa de Hausdorff desde A a B, y ordena cada punto de A basado en su distancia al punto más cercano de B. La distancia de Hausdorff mide la diferencia entre conjuntos fijos de puntos. En la Figura 5.2 se representa de forma gráfica como se realiza el proceso para calcular la distancia de Hausdorff entre dos trayectorias $A=\{a_1 \dots a_N\}$ y $B=\{b_1 \dots b_N\}$.

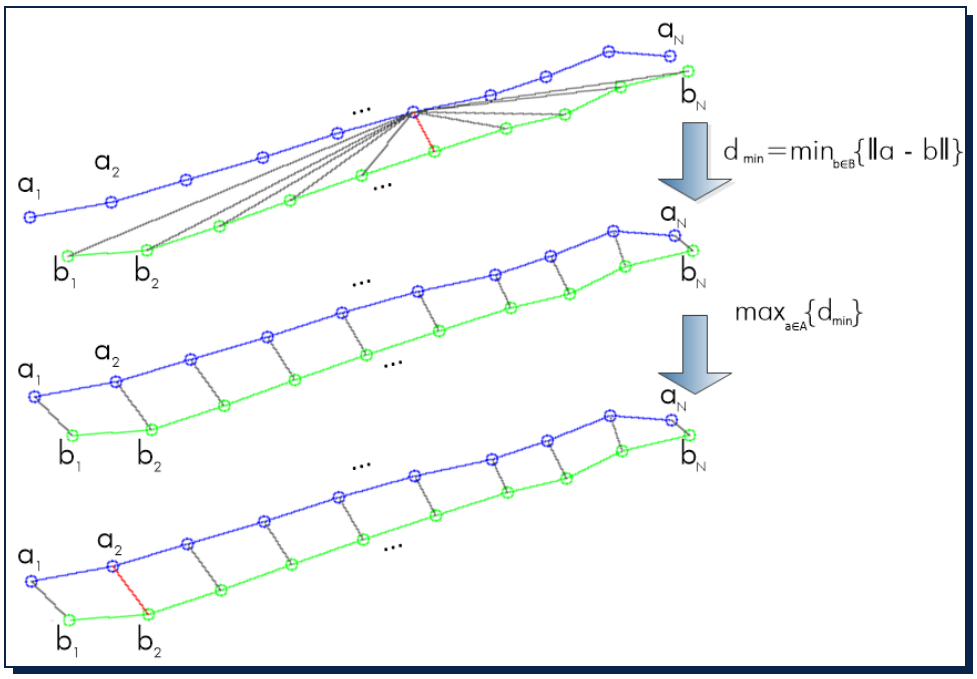


Figura 5.2. Cálculo de la distancia de Hausdorff.

La media de la diferencia angular punto a punto se utiliza como complemento para distinguir trayectorias semejantes (con distancia de Hausdorff pequeña) pero en las que los objetos se desplazan en sentidos contrarios. Para obtener este valor se calculan los ángulos de todas las direcciones de las dos trayectorias. Haciendo la resta punto a punto se obtiene el ángulo que forman ambas en cada punto, se

calcula la media y es ese el valor que se compara con el umbral establecido para determinar la semejanza de las direcciones de las trayectorias.

La Figura 5.3 muestra el ángulo de las direcciones de las trayectorias A y B en cada punto:

$$\theta_A=\{\theta_1 \dots \theta_N\} \quad (5.13)$$

$$\phi_B=\{\phi_1 \dots \phi_N\} \quad (5.14)$$

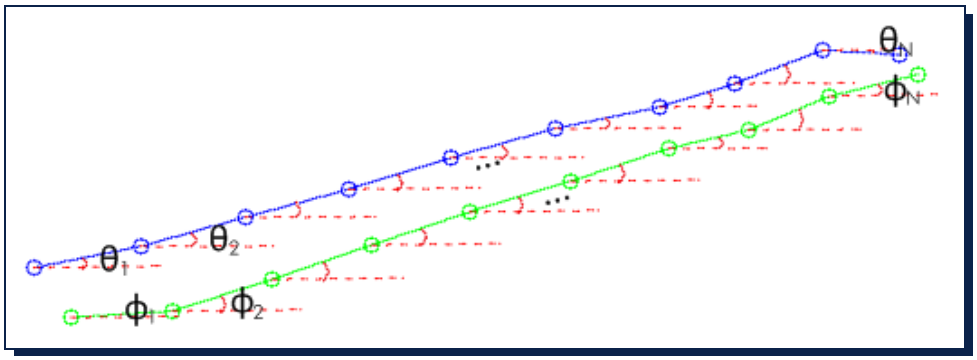


Figura 5.3. Cálculo del ángulo de las direcciones de las trayectorias.

La media de la diferencia angular punto a punto se calcula a partir de estos ángulos como:

$$d_{ang}=(\Sigma^{N}_{i=1}|\theta_i-\phi_i|)/N \quad (5.15)$$

Dos trayectorias se agrupan en el mismo cluster cuando los valores de la distancia de Hausdorff y la diferencia angular están por debajo de dos umbrales diferentes. Las comparaciones punto a punto son posibles porque todas las trayectorias son muestreadas con la misma tasa. Esto tiene también la ventaja de reducir el número de puntos en una trayectoria, optimizando así la cantidad de cálculo necesaria para la obtención de la distancia Hausdorff (que es, por definición, una operación costosa).

A modo de resumen, el algoritmo para generar el modelo de rutas es el siguiente:

1. Cuando un objeto desaparece de la imagen, se verifica si el número de puntos de la trayectoria cerrada y la longitud de la misma superan los umbrales preestablecidos para determinar que es una trayectoria válida. En caso contrario se elimina y no se tiene en cuenta para la identificación de rutas.

2. Esta trayectoria cerrada se vuelve a muestrear para que contenga sólo un número fijo de puntos (configurable).

3. Cada vez que se completa una nueva trayectoria válida:

   a. Si no hay todavía ninguna ruta en el sistema, esta trayectoria se considera una ruta de fuerza uno. Es decir, si la trayectoria cerrada T_i tiene como parámetros:

$$T_i = \{t_{i1} \dots t_{iN}\} \text{ donde } t_{ik} = \{x_{ik}, y_{ik}\} \quad (5.16)$$

$$W_i = \{w_{i1} \dots w_{iN}\} \quad (5.17)$$

$$H_i = \{h_{i1} \dots h_{iN}\} \quad (5.18)$$

   Se crea la ruta R_1 caracterizada por:

$$X_1 = \{x_{i1} \dots x_{iN}\} \quad (5.19)$$

$$Y_1 = \{y_{i1} \dots y_{iN}\} \quad (5.20)$$

$$S_1 = 1 \quad (5.21)$$

   y $E1_1$ y $E2_1$ (envolventes 1 y 2) calculados utilizando el ángulo de inclinación y la altura a la que está colocada la cámara y los parámetros W_i y H_i de la trayectoria cerrada mediante la aplicación de perspectiva inversa descrita en [100] y D_1 como los vectores diferencia de dos centros consecutivos de la ruta.

   b. Si ya existe alguna ruta, se calculan los dos valores que miden la similitud con una de las rutas existentes: la distancia de Hausdorff y la diferencia angular entre las direcciones de la trayectoria y la ruta en cada punto.

      i. Si ambos valores están por debajo de un umbral previamente configurado sólo para una de las rutas, esta trayectoria se agrupa con la ruta (los puntos centrales, direcciones y envolventes se promedian con los de la nueva trayectoria añadida) y la fuerza se incrementa en uno. La metodología para realizar esta actualización de la ruta se especifica en la Sección 5.3.2.2 Actualización de rutas.

ii. Si los dos valores están por debajo del umbral para varias de las rutas, la trayectoria se agrupa con aquella más cercana del modo que se describirá en la Sección 5.3.2.2.

iii. Si las medidas de similitud superan los umbrales, se crea una nueva ruta de fuerza uno sólo con esta trayectoria del modo especificado en el apartado a de este mismo punto.

En la Figura 5.4 se puede observar el funcionamiento del algoritmo propuesto en un video real de una intersección procedente del proyecto ITEA CANDELA - *Content Analysis and Network DELivery Architectures* (*ITEA Programme*; 2003-2005; http://www.hitech-projects.com/euprojects/candela/). En rojo aparecen las envolventes de las trayectorias cerradas que se convierten en rutas con peso 1 y en amarillo las envolventes de las rutas formadas como agrupación de varias trayectorias cerradas.

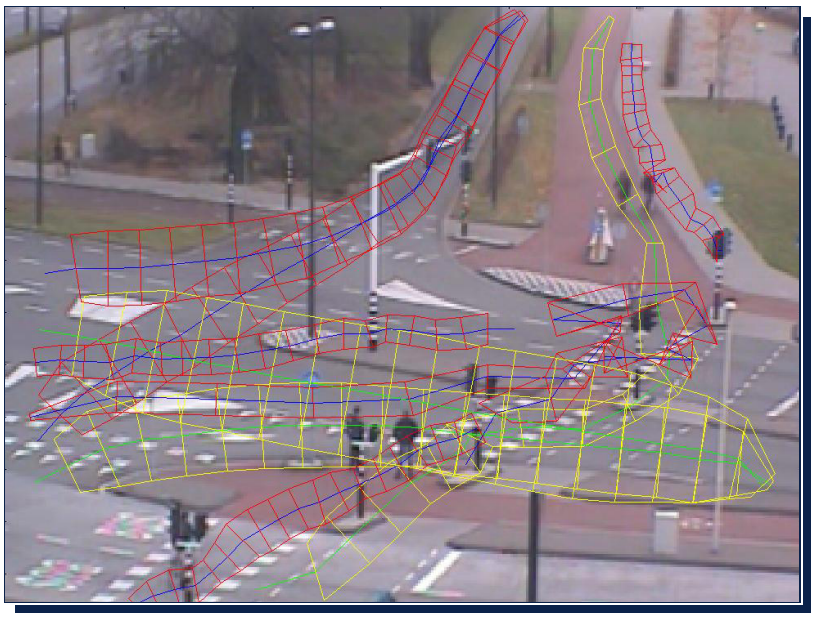


Figura 5.4. Ejemplo de detección de trayectorias y rutas en un video real.

5.3.2.2 Actualización de rutas

Cuando una nueva trayectoria coincide con una ruta, los puntos de la ruta actualizada se calculan como la media de los de la trayectoria y los de la ruta antigua ponderándola con el parámetro "*strength*". De esta manera las rutas más establecidas varían menos que las que contienen pocas trayectorias y todas las trayectorias que forman la misma ruta tienen el mismo peso.

Si una trayectoria T_i con:

$$T_i = \{t_{i1} \dots t_{iN}\} \text{ donde } t_{ik} = \{x_{ik}, y_{ik}\} \quad (5.22)$$

$$W_i = \{w_{i1} \dots w_{iN}\} \quad (5.23)$$

$$H_i = \{h_{i1} \dots h_{iN}\} \quad (5.24)$$

es similar a una ruta R_i con parámetros:

$$X_i = \{x_1 \dots x_N\} \quad (5.25)$$

$$Y_i = \{y_1 \dots y_N\} \quad (5.26)$$

$$E1_i = \{e1_{i1} \dots e1_{iN}\} \text{ donde } e1_{ik} = \{x_{ik}, y_{ik}\} \quad (5.27)$$

$$E2_i = \{e2_{i1} \dots e2_{iN}\} \text{ donde } e2_{ik} = \{x_{ik}, y_{ik}\} \quad (5.28)$$

$$D_i = \{d_{i1} \dots d_{iN}\} \text{ donde } d_{ik} = \{dx_{ik}, dy_{ik}\} \quad (5.29)$$

$$S_i \quad (5.30)$$

Se crea la ruta temporal con los parámetros de la trayectoria:

$$X_{temp} = \{x_{temp1} \dots x_{tempN}\} \quad (5.31)$$

$$Y_{temp} = \{y_{temp1} \dots y_{tempN}\} \quad (5.32)$$

y $E1_{temp}$ y $E2_{temp}$ (envolventes) calculadas, utilizando perspectiva inversa [99] a partir del ángulo de inclinación y la altura de la cámara y los parámetros W_i y H_i de la trayectoria cerrada y D_{temp} como los vectores diferencia de dos centros consecutivos de la ruta.

A continuación se actualiza R_i, a modo de ejemplo:

$$X_i = \{(x_1 * S_i + x_{1temp})/(S_i + 1) \dots (x_N * S_i + x_{Ntemp})/(S_i + 1)\} \quad (5.33)$$

$$Y_i = \{(y_1 * S_i + y_{1temp})/(S_i + 1) \dots (y_N * S_i + y_{Ntemp})/(S_i + 1)\} \quad (5.34)$$

$$S_i = S_i + 1 \quad (5.35)$$

Y los valores de $E1_i$, $E2_i$ y D_i de forma similar a X_i e Y_i.

5.3.2.3 Fusión de rutas

Como se indica en la Sección 5.3.2.1, cuando aparece una nueva trayectoria cerrada válida se calcula la distancia de ésta con todas las rutas detectadas previamente. Esta distancia mide la similitud entre ambas. En situaciones como la mostrada en la Figura 5.5, en las que una trayectoria está incluida dentro de una ruta (y además tienen la misma dirección), los valores de distancia entre esa ruta y la trayectoria superan los umbrales ya que no son similares.

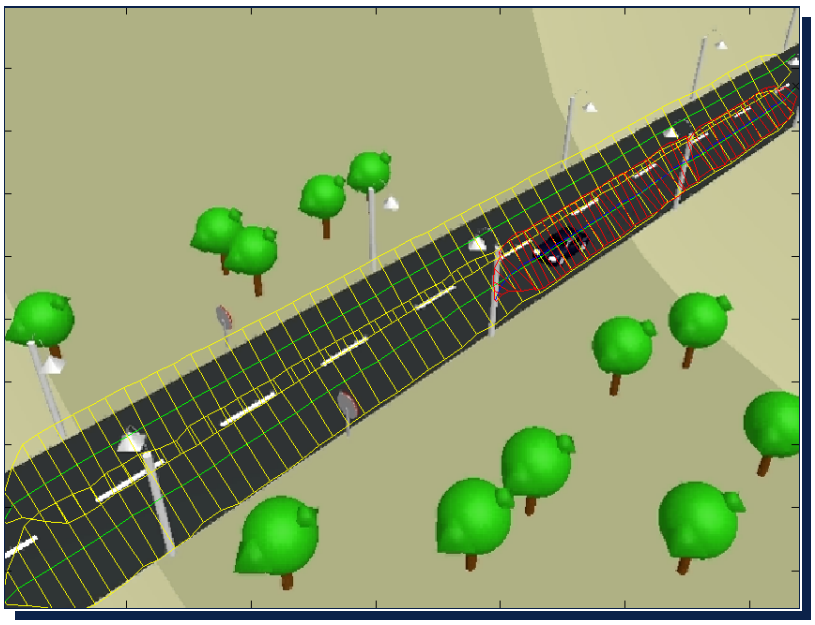


Figura 5.5. Ejemplo de fusión de rutas.

Se realiza una búsqueda, dentro de todas las rutas, de trayectorias (en este caso rutas con peso uno) más pequeñas que las incluyen eliminando estos "trozos" cuando haya coincidencia. Para ello cada ruta se divide en sub-rutas y, utilizando la distancia de Hausdorff y la diferencia angular, se comparan con las otras rutas. Si en algún caso estos valores son inferiores a los umbrales establecidos se elimina la ruta corta (la sub-ruta con la que se compara pertenece a la ruta más larga).

5.4 Fuentes y sumideros

5.4.1 Modelado de fuentes/sumideros

Las fuentes y sumideros se representan como agrupaciones de los puntos donde ha aparecido/desparecido un objeto. En la Figura 5.6 se pueden ver en cian (marcados con una x) los puntos que han sido fuentes y en magenta (también

marcados con x) los sumideros. Además, aprecian las zonas de fuentes de objetos señaladas con un rectángulo blanco (que agrupa los puntos cian).

Figura 5.6. Ejemplo de detección de rutas y fuentes en un video sintético.

Las fuentes y sumideros representan zonas de la escena donde generalmente los objetos "aparecen y desaparecen". Normalmente van a encontrarse en los bordes del campo de visión de la cámara (FOV - *Field Of View*). También se identifican como fuentes aquellas zonas en las que los objetos habitualmente se paran (no son detectados por el sistema de procesado de imagen al no estar en movimiento) y como sumideros a las zonas donde comienzan otra vez a moverse. Puede producirse además una identificación errónea de fuentes/sumideros en las zonas donde se produce habitualmente ocultación de los objetos y no hay una detección precisa del sistema de procesamiento de imagen en movimiento produciéndose discontinuidades, problema denominado oclusión en procesamiento de imagen.

5.4.2 Localización de fuentes/sumideros

Para la localización de fuentes y sumideros, los puntos inicial y final del vector de la trayectoria cerrada se añaden a las matrices de fuentes y sumideros respectivamente, creadas para el almacenamiento de los mismos. Los puntos de estas matrices se agrupan usando un algoritmo de clustering. El clustering permite concentrar datos en clases de tal forma que los objetos de un grupo tienen una similitud alta entre ellos, y baja (sean muy diferentes) con objetos de otros clusters. Esta medida de semejanza, en este caso, está basada en la situación de los objetos

y las zonas de la escena definidas por estos grupos se consideran fuentes y sumideros.

Existen diversos algoritmos de clustering que se pueden aplicar para definir las agrupaciones en este tipo de sistemas. Herramientas como MATLAB incluyen, de forma nativa, un *toolbox* para realizar este tipo de operaciones. En concreto MATLAB, en el apartado de estadística, dispone de una función que realiza el algoritmo Kmeans. Sin embargo, para la utilización de otros algoritmos es necesario recurrir a implementaciones adicionales. *Fuzzy Clustering and Data Analysis Toolbox* es un conjunto de funciones de MATLAB desarrolladas por Balazs Balasko, Janos Abonyi and Balazs Fiel pertenecientes al *Department of Process Engineering* de la Universidad de Veszprem en Hungría.

La documentación de la herramienta proporcionada [159] incluye:

- Algoritmos de clustering: son los encargados de dividir los datos en grupos o clusters según diferentes criterios. Están implementados los algoritmos típicos como Kmeans [123] y Kmedoid [160], así como FCM (*Fuzzy c-Means*) [161], GK (*Gustafson-Kessel*) [162] y GG (Gath-Geva)[163] que son algoritmos basados en densidad. Además se dispone de diferentes medidas de distancia para utilizar.
- Validación: se trata de funciones que proporcionan medidas de validez para la partición. Son muy apropiadas cuando el número de clusters se desconoce a priori. De esta manera, el número óptimo de clusters se determina por el valor extremo (alto o bajo) del índice de validación, es decir, se calcula el índice para distinto número de agrupaciones, seleccionando como válida aquella con la que le marcador alcanza el valor máximo o mínimo, según esté definido para la metodología concreta. Los índices que incorpora esta herramienta son:
    - *Partition Coefficient* (PC) [164].
    - *Clasification Entropy* (CE) [165].
    - *Partition Index* (SC) [166].
    - *Separation Index* (S) [166].
    - *Xie and Beni´s Index* (XB) [167].
    - *Dunn´s Index* (DI) [168].

- *Alternative Dunn´s Index* (ADI) [169].

Analizando cada uno de ellos, se puede ver que la única diferencia entre S, SC y XB es la forma de medir la separación de los grupos. Entre estos tres índices, S y SC son los más usados. Además, en el caso de tener clusters solapados, los índices DI y ADI no son fiables ya que tienen que ser recalculados con métodos *hard clustering*. Hay que mencionar que ningún índice es concluyente por sí mismo, sino que hay que comparar los resultados para todos los posibles números de clusters de todos y cada uno de ellos para llegar a encontrar el k óptimo, con lo que no es una solución eficiente.

En el caso que se está tratando se desconoce a priori del número de clusters, que resulta crítico ya que el cálculo del mismo lleva asociado un elevado coste computacional.

Como alternativa se utiliza el algoritmo DBSCAN (*Density-based spatial clustering of applications with noise*) [170][171] implementado por Michal Daszykowski del *Department of Chemometrics*, Institute of Chemistry de University of Silesia. Este mecanismo, utiliza el radio de proximidad máximo entre puntos pertenecientes a la misma agrupación, en lugar del número de clusters, dato más fácil de estimar por trabajar en escenas de tamaño limitado. Además DBSCAN permite descubrir grupos de formas arbitrarias, incluso puede encontrar un cluster completamente rodeado por un grupo diferente (y no conectado a él). DBSCAN tiene la posibilidad de tratar elementos ruidosos.

DBSCAN es un algoritmo basado en densidad que necesita dos parámetros de entrada: Epsilon (ε) y minPoints. Comienza seleccionando un punto de partida arbitrario P. Si el número de puntos alcanzados al trazar una circunferencia de centro en P y radio Epsilon es superior o igual a minPoints, se comienza a formar un cluster. P, denominado punto central, y todos los alcanzados forman parte de esa agrupación. A continuación se repite el proceso de forma recursiva con cada uno de los puntos del grupo. Los puntos que se encuentren en las cercanías de los puntos alcanzables por P se denominarán denso-alcanzables desde P. Si un punto no es central se visita otro del conjunto de datos. El proceso continúa hasta que se procesan todos los puntos de la partición. Si un grupo está totalmente expandido (todos los puntos están al alcance de la visita), el algoritmo procede a iterar a través de los puntos restantes no visitados hasta que se agote. Los puntos que quedan fuera de los grupos formados se llaman puntos ruido, los puntos que no son ni ruido ni centrales se llaman puntos borde. De esta forma DBSCAN construye grupos en

los que sus puntos son o puntos centrales (un grupo puede tener más de un punto central) o puntos borde y el conjunto de todos los puntos marcados como ruido se consideran valores atípicos. DBSCAN es además insensible al orden de los puntos.

En la Figura 5.7 se puede ver el funcionamiento del algoritmo. Partiendo del punto P_1, se encuentra que para ε hay dos puntos vecinos (P_2 y P_3) con lo cual P_1 es un punto central y se comienza a generar el cluster C_1 con estos tres valores. Haciendo el mismo proceso con los dos puntos vecinos, P_2 es un punto borde (sólo tiene P_1 como vecino) y P_3 es punto central también (tiene P_1 y P_4 como vecinos). Después, analizando P_4 se determina que es un punto de borde. Además, P_4 es densamente alcanzable por P_1 y P_2 es densamente alcanzable por P_3. El cluster C_1 queda formado por los puntos P_1, P_2, P_3 y P_4. Los puntos centrales aparecen representados en color verde y los puntos de borde en amarillo. De igual modo, siguiendo con el resto de puntos, se determina que hay otro cluster C_2 y dos puntos marcados como ruido (representados en rojo).

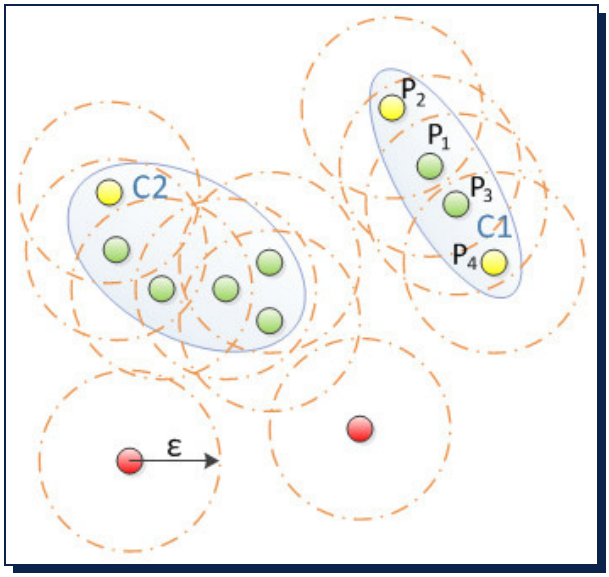


Figura 5.7. Algoritmo DBSCAN para minPoints 2.

En las Figura 5.8 y Figura 5.9, se observan ejemplos de la utilización del algoritmo DBSCAN con parámetros minPoints=2 y ε=25 para dos escenarios diferentes, uno sintético y un video real del MIT - *Massachusetts Institute of Technology*. Los puntos de las fuentes y sumideros se representan con cruces cian y magenta respectivamente y los clusters se representan con cuadrados.

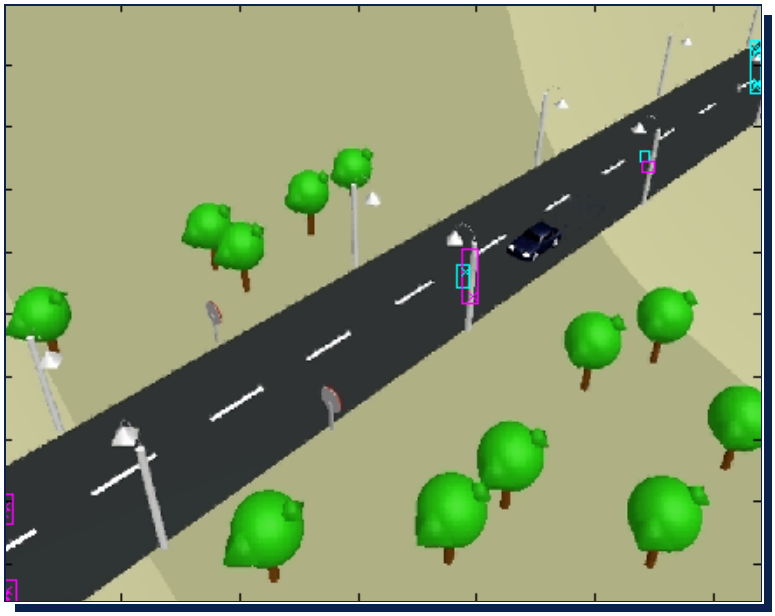


Figura 5.8. Ejemplo de detección de fuentes y sumideros con el algoritmo DBSCAN en un video sintético.

Como se comenta en la Sección 5.4.1 Modelado de fuentes/sumideros puede producirse una detección errónea de fuentes/sumideros debido a la oclusión. En el caso de la Figura 5.8 una farola interrumpe la continuidad en el movimiento de los objetos de movimiento y el sistema de procesado de imagen no es capaz de evitar este problema identificando esa falsa fuente. Puede verse que al llegar a la farola se detecta un sumidero y el objeto "desaparece" de la imagen y al volver a aparecer, una vez pasada la farola, se descubre una nueva fuente.

Por otro lado, observando la imagen, a la derecha aparece un cluster de fuentes y a la derecha se encuentran dos sumideros, por lo que se puede determinar que la carretera es de dos carriles con el mismo sentido.

En la Figura 5.9 se puede ver como, no sólo se identifican fuentes y sumideros en los bordes de las escenas, sino también en aquellas zonas en las que los objetos habitualmente se detienen (no son detectados por el sistema de procesado de imagen al no estar en movimiento) y como sumideros a las zonas donde comienzan otra vez a moverse. Así, se observa como aparecen combinaciones de fuente y sumideros en zonas con semáforos, tramos de acera con pasos de peatones, cruces, etc.

También se muestra en la Figura 5.10 como hay puntos atípicos que no pertenecen a ningún cluster. En algunas ocasiones simplemente son espurios mal detectados por el sistema de procesamiento de imagen o fenómenos puntuales, pero en otros

casos no se ha ejecutado el algoritmo suficiente tiempo y por lo tanto ha habido muy pocos objetos que se detuvieran o arrancasen en esa zona. Se puede observar, una fuente en la puerta de un garaje por lo que ha salido un solo coche y otro garaje en el que aparece una fuente y un sumidero pero al tratarse sólo de un coche entrante y uno saliente no se han formado todavía los clusters oportunos en esa zona.

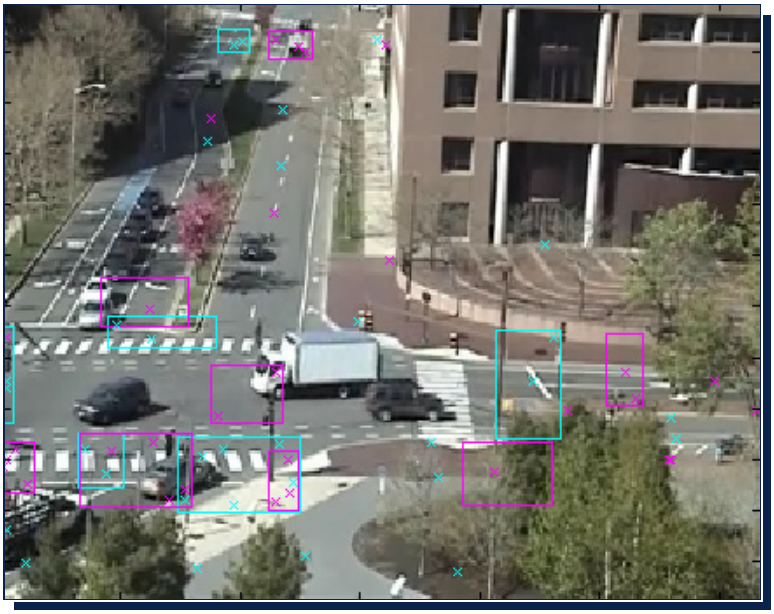


Figura 5.9. Ejemplo de detección de fuentes y sumideros con el algoritmo DBSCAN en un video real.

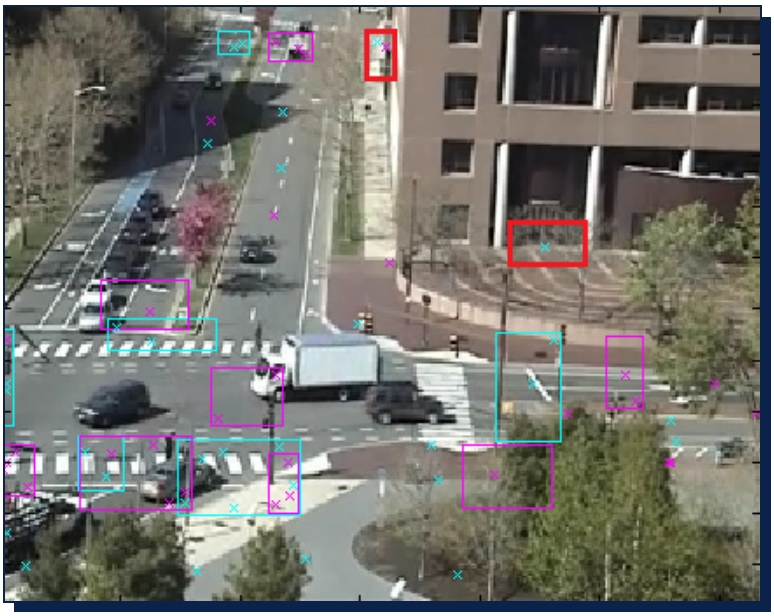


Figura 5.10. Ejemplo de detección de ruido con el algoritmo DBSCAN en un video real.

Descripción del video	N	Número de objetos	Número de rutas	Número de rutas identificadas	Número de rutas identificados erróneamente	% rutas identificadas	% rutas erróneas
Video sintético de una carretera de dos carriles de una dirección por carril.	5	359	2	2	2	100%	100%
	10	359	2	2	0	100%	0%
	20	359	2	2	0	100%	0%
Video sintético de una carretera de cuatro carriles, dos carriles para cada sentido.	5	58	4	3	1	75%	25%
	10	58	4	4	0	100%	0%
	20	58	4	4	0	100%	0%
Un video real de una intersección compleja, con tres vías para vehículos con dos direcciones y varias aceras.	10	37	5(coches) 6(personas)	4(coches) 5(personas)	0(coches) 0(personas)	80%(coches) 83%(personas)	0%(coches) 0%(personas)
	20	37	5(coches) 6(personas)	4(coches) 5(personas)	0(coches) 0(personas)	80%(coches) 83%(personas)	0%(coches) 0%(personas)
Parte de un video del MIT que muestra una compleja intersección con carreteras y aceras.	10	151	9(coches) 7(personas)	8(coches) 7(personas)	0(coches) 1(personas)	89%(coches) 100%(personas)	0%(coches) 14%(personas)

Tabla 5.1. Comportamiento del detector de rutas.

5.5 Validación

El comportamiento del algoritmo se evalúa en términos de precisión en la identificación de ruta. Para ello, se analizan cuatro videos diferentes con el detector de rutas (dos videos sintéticos y dos reales, uno del proyecto ITEA CANDELA (disponible en http://www.multitel.be/image/research-development/research-projects/candela.php) y otro del MIT - *Massachusetts Institute of Technology* (disponible en http://www.ee.cuhk.edu.hk/ ~ xgwang / MITtraffic.html)), comparando, mediante la observación, las trayectorias identificadas con las trayectorias reales que deben ser detectadas. Para ello se superponen las zonas descubiertas con la imagen real, facilitando así la determinación de los errores en el proceso de descubrimiento de rutas y especificando el porcentaje de rutas correctamente identificados y el porcentaje de caminos erróneos (para un funcionamiento ideal, estos datos es 100% y 0% respectivamente).

La Tabla 5.1 recoge los resultados del análisis. También se ha incluido el número de objetos que aparecen en cada uno de los videos y el número de puntos N de cada trayectoria/ruta considerado para cada ejecución.

Evaluando la precisión, los números obtenidos para escenas sencillas, como las que se esperan para carreteras con carriles bien definidos, son muy buenos. Es fácil identificar todos los carriles sin errores al ejecutar el algoritmo con 10 o más puntos por trayectoria. En las escenas más complicadas, las tasas siguen siendo buenas pero, a medida que aumenta la complejidad del escenario son necesarios mayor número de objetos para conseguir las mismas precisiones.

5.6 Conclusiones

Los sensores inteligentes que se utilizan proporcionan parámetros de movimiento de los objetos que captan, no imágenes. Esto hace que se descarten mecanismos basados en procesamiento de la imagen mediante algoritmos de visión artificial y/o comparación con modelos predefinidos para determinar las zonas de la imagen.

Existen distintos métodos para la identificación de regiones basadas en la detección de patrones de movimiento de los objetos que por ellas se desplazan. El sistema descrito en esta Tesis Doctoral sigue la tendencia de la mayoría de los estudios. En ellos, partiendo del movimiento de los objetos que transcurren por la escena, se diferencian dos tipos de regiones fundamentales. Por una parte se determinan los caminos propiamente dichos que corresponden a las trayectorias habituales de los

objetos y que se han denominado rutas. Como complemento se reconocen las zonas de entrada y salida de los objetos que se etiquetan como fuentes y sumideros.

Para la identificación de las distintas regiones se utilizan filtros que eliminan ruido en las rutas (desechando trayectorias de objetos con pocos puntos o de dimensiones reducidas), interpolación para facilitar la comparación de trayectorias, una modificación de la distancia de Hausdorff como técnica para medir la similitud entre trayectorias, y el algoritmo de clustering DBSCAN para agrupar en regiones los puntos de entrada y salida de objetos.

La aplicación de estos mecanismos no es crítica en cuanto a tiempo de ejecución ya que sólo se ejecutan durante el tiempo de aprendizaje del sistema pero sí que deben proporcionar un resultado coherente y completo de la escena ya que sirve de base para el modelado semántico de los espacios. Un buen modelado espacial proporciona una base robusta para el reconocimiento de las regiones y es muy importante para reducir los falsos positivos en la determinación de situaciones anómalas. En este sentido, el sistema descrito cumple las expectativas esperadas y proporciona resultados satisfactorios.

6

MODELADO SEMÁNTICO DE ESPACIOS

Para la correcta caracterización de escenarios es necesario identificar los distintos elementos de la representación que tienen significado de acuerdo con el dominio y el conocimiento que se tenga del mismo. En este sentido, es muy importante el modelado y la consistencia del mismo para que los resultados de la inferencia sean válidos. Por ello, existen diferentes mecanismos de representación del conocimiento, cada uno de los cuales proporciona diferentes niveles de complejidad en la aplicación de sus propias estrategias de inferencia. Marcos [172], redes semánticas [173], reglas de producción [174] o lógica de predicados [175] son algunas de las posibles soluciones.

En los últimos años el modelado semántico como mecanismo para la representación del conocimiento se está imponiendo. Las ontologías proporcionan un mecanismo de relación conceptual que hace que sea fácilmente adaptado para su utilización en muy diversas áreas. Los conceptos se convierten en elementos con

significado pudiendo determinar a qué se refiere un término cuando está siendo utilizado en un determinado contexto. Sin embargo, para poder establecer relaciones entre ellos, a priori independientes se requiere el uso de herramientas de razonamiento adicionales que lleven a cabo el proceso de inferencia. Esta complementación de tecnologías se convierte en una herramienta muy potente permitiendo identificar los elementos que aparecen en la escena y establecer relaciones entre ellos. Además, si en un contexto genérico se utilizan las ontologías para definir los distintos conceptos, un cambio de la misma permite de forma fácil la adaptación a un nuevo dominio. Es decir, un objeto del escenario se identifica con una etiqueta u otra en función del modelo ontológico que se haya definido.

De esta forma se establece una clara separación entre los datos de bajo nivel y el modelo de la base de conocimientos. Este enfoque presenta varias ventajas. Por un lado, el modelo se hace independiente de las técnicas y herramientas utilizadas para extraer la información de bajo nivel, por lo que puede ser actualizado y mejorado de forma separada. Por otro lado, es posible modificar también los algoritmos y métodos para la extracción de datos de manera independiente, y fuentes de datos adicionales pueden ser incluidas con pocos cambios.

Por otro lado, una vez caracterizada la escena, etiquetadas las distintas zonas de la misma y los objetos que en ella aparecen, es fácil ir un poco más allá y utilizar toda esta información para informar a un posible usuario de situaciones de su interés o directamente, aplicándolo a videovigilancia, notificar alertas.

Así pues, el objetivo de este Capítulo es determinar en lenguaje formal (entendible por un operador humano) la clase de regiones (ruta, fuente o sumidero) que aparecen en la escena clasificándolos convenientemente de manera que se conozca la disposición de los mismos y las características que habitualmente tienen los elementos que se mueven por ellos. En este sentido, también se clasifican los objetos en movimiento dentro de clases con comportamientos definidos. Si un móvil experimenta un comportamiento diferente se lanza una alerta concretando este evento. Además, si dentro de una región algún elemento sigue un movimiento determinado atípico o especial se informa especificando de tal situación.

Este Capítulo se encuentra dividido en ocho secciones. En la Sección 6.1 se realiza una introducción a la Semántica y las posibilidades que ofrece. La Sección 6.2 describe las ontologías prestando especial atención en las ontologías persistentes por sus ventajas adicionales y en la Sección 6.3 los lenguajes que las definen. En la Sección 6.4, como complemento de las ontologías, se detallan las reglas

ontológicas que posibilitan razonamientos más completos. En la Sección 6.5 se analizan los tipos de razonadores que se pueden aplicar a los modelos ontológicos para realizar la inferencia. En la Sección 6.6 se describe la arquitectura propuesta para la caracterización semántica de las regiones y los objetos, validando la correcta definición ontológica en la Sección 6.7. Finalmente, en la Sección 6.8 se exponen las principales conclusiones del Capítulo.

6.1 Semántica

La representación y tratamiento de datos mediante el uso de la semántica es una disciplina de relativa novedad [176]. Esta metodología se introdujo inicialmente mediante su aplicación en la *Web* ya que un modelado uniforme del conocimiento disponible permitía establecer relaciones entre conceptos para realizar búsquedas o ejecutar servicios *Web* mejorando lo resultados. Así nació la denominada *Web* Semántica [81].

La aparición de nuevas herramientas y metodologías ha permitido utilizar dicha tecnología en diversos campos de la ingeniería, posibilitando su aplicación a nuevos escenarios donde puedan aprovecharse sus ventajas. Entre ellas destacan los siguientes:

- Facultad de facilitar la interoperabilidad entre los sistemas heterogéneos gracias a la posibilidad de utilizar un lenguaje común para definir las señales de entrada y salida.
- Capacidad de mejorar el acceso basado en contenidos ya que, al estar la información descrita mediante el lenguaje semántico, se posibilita la realización de búsquedas avanzadas basadas en conceptos en lugar de en palabras clave como se ha realizado hasta ahora. Como ventaja se pueden obtener resultados que no estén almacenados con esos términos pero que se refieran al concepto deseado o realizar filtrados en función del contexto de la búsqueda cuando un mismo término se usa para referirse a varios conceptos.
- Posibilidad de reutilizar el conocimiento al poder combinar diferentes ontologías en una sola o usar una creada anteriormente para un fin al que no estaba originalmente dirigida. Esto dota al sistema de una mayor flexibilidad y eficiencia a la hora de enfrentarse a situaciones nuevas.
- Realización de un procesamiento automático de la información, evitando así la intervención humana para tareas tediosas o complejas. Al definir los

conceptos previamente y gracias a la capacidad de la semántica de establecer nuevas relaciones entre los miembros de la ontología, se permite a las máquinas realizar tareas de manera más automatizada. Además, como la definición del conocimiento sigue un modelo, se simplifica la comunicación entre dispositivos facilitando también esta tarea.

La semántica trata de expresar formalmente un dominio de conocimiento representándolo en función de las características de los diferentes objetos o individuos del entorno y sus relaciones. Para ello se realiza un procesado automático de la información conocida de una cierta entidad, se le incluye dentro de una clase o varias en función de la misma y se le asignan ciertas propiedades que la vinculan con otra u otras instancias.

Las tecnologías semánticas basan su funcionamiento en cuatro pilares fundamentales: una o varias ontologías, el lenguaje semántico para definirlas, en algunos casos unas reglas de inferencia que sirven como complemento y un razonador.

Las ontologías son estructuras que representan de manera formal las relaciones existentes entre los miembros definidos en ella. Por una parte se divide en clases o categorías que agrupan a los individuos y por otra define las propiedades utilizadas para describirlos y relacionarlos. Hay que diferenciar muy bien entre clases e individuales o instancias de objetos. Una clase incluye un esquema con las características de los individuos que contiene y un individual o instancia es un objeto concreto que pertenece a una o varias clases y por tanto tiene las propiedades que se asignan a los objetos de esa clase.

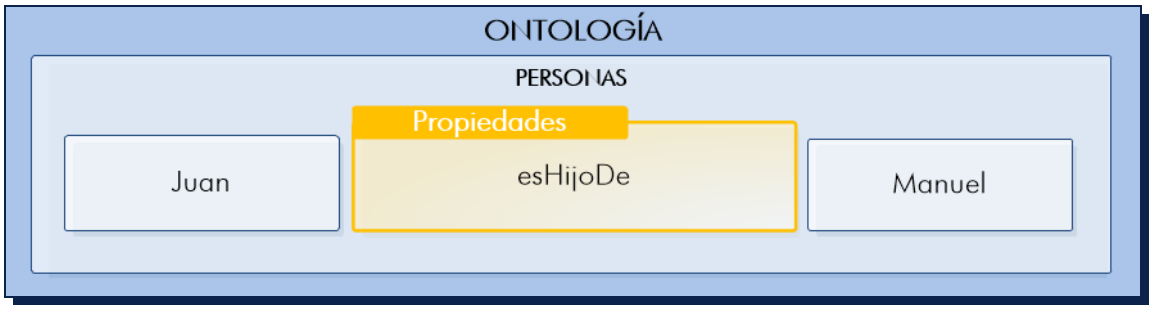


Figura 6.1. Diferenciación en la ontología entre clases, individuales y propiedades.

Esto puede verse en la Figura 6.1. Por una parte aparecen dos individuales "Juan" y "Manuel", individuos concretos, que pertenecen a la clase "Personas". Y por otra, dentro de esta clase se ha definido una propiedad de sus miembros que es "esHijoDe" que, en este caso establece una relación entre ambos elementos.

El lenguaje semántico permite la definición de la ontología para la representación del conocimiento. Este lenguaje utiliza una estructura denominada tripleta formada por tres componentes: el sujeto, el predicado y el objeto. El sujeto identifica a una clase o individual que posee la propiedad definida en el predicado y que lo relaciona con otra clase o individual en el objeto de la tripleta. Cada una de estas tres partes se define mediante una URI (*Uniform Resource Identifier*). En la Figura 6.2 se muestra un ejemplo de este lenguaje y de su estructura.

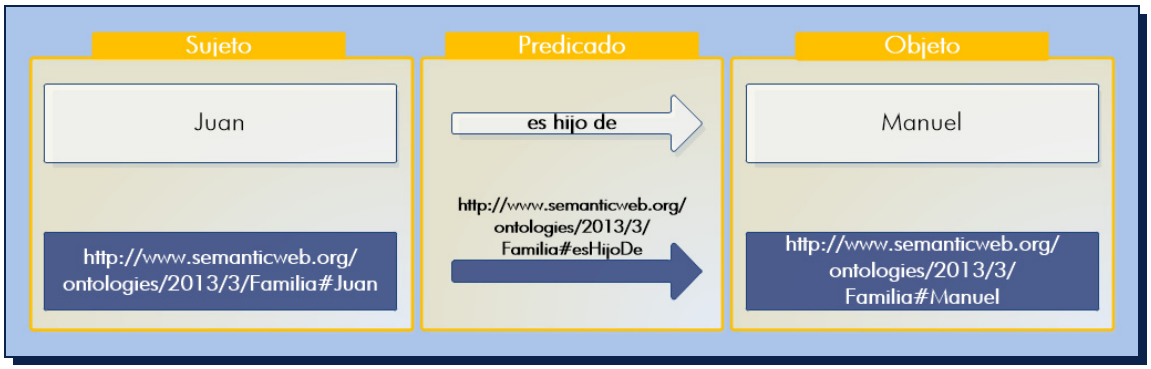


Figura 6.2. Ejemplo del lenguaje semántico: la tripleta.

En ocasiones se necesita dotar al sistema de información adicional para que durante el proceso de inferencia se puedan crear nuevas relaciones entre los elementos que forman la ontología. Para ello se utilizan las reglas de inferencia.

Éstas proporcionan nuevos datos al razonador, relaciones, límites, etc., que o bien no ha sido posible definir en la propia ontología o se han incluido posteriormente como complemento a las definiciones ontológicas iniciales. Gracias a ellas se dota al razonador de una mejor capacidad de asimilación de los datos y por lo tanto de creación de conocimiento.

Por último el razonador, componente que se encarga de, a partir de las relaciones existentes en la ontología y las instrucciones indicadas en las reglas, obtener nuevas relaciones y categorías de los contenidos incluidos en dicho modelo de conocimiento.

Un ejemplo de lo que implica su uso es lo mostrado en la Figura 6.3. La ontología incluye los individuales "Juan" y "Manuel" relacionados por la propiedad "esHijoDe". La ontología o las reglas definen que la propiedad "esHijoDe" tiene como inversa "esPadreDe", por lo tanto, en el proceso de razonado se determina que "Manuel esPadreDe Juan".

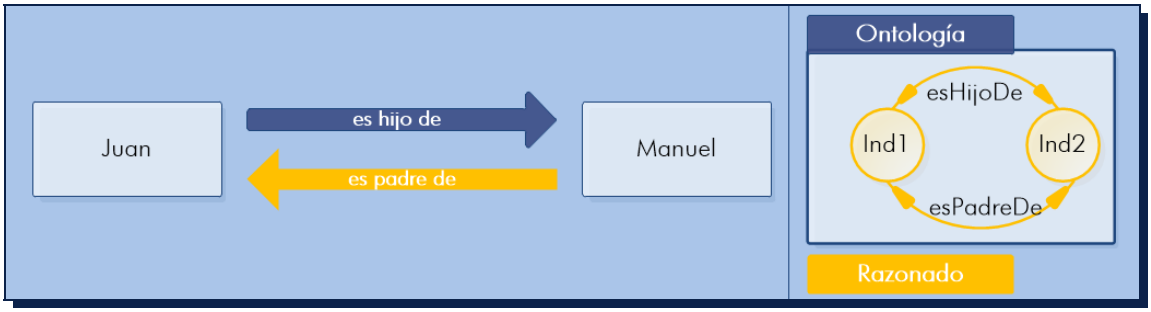


Figura 6.3. Resultado de un razonamiento en base a propiedades entre objetos.

La semántica puede ser entendida como un modelo, en el que mediante el razonado semántico de una o varias ontologías que esquematizan una escena, los conceptos (atributos, propiedades, situación, etc.) y relaciones conocidas entre los mismos, descritas en un lenguaje semántico y complementadas en algunos casos con reglas, permite determinar nuevas relaciones entre los individuos que se desconocían a priori.

6.2 Ontologías

El término ontología, proveniente de los términos griegos "ontos" (existencia) y "logos" (estudio), ha ido ganando importancia impulsado por el aumento de la utilización de la semántica en diversos ámbitos de la ciencia. Comenzó a utilizarse en filosofía, pero poco a poco fue adquiriendo significado propio en las ciencias de la computación, para denominar a la representación conceptual que permite la comunicación entre sistemas y con el usuario.

Existen definiciones de ontología distintas dentro de la comunidad investigadora adaptadas a los intereses o necesidades del campo en el que se use. Entre ellas destaca la proporcionada por Weigand en 1997 [177] en la que se especifica que una ontología es "una base de datos que describe conceptos generales o sobre un dominio, algunas de sus propiedades y cómo se relacionan unos con otros".

Otra de las interpretaciones a tener en cuenta a la hora de entender lo que es una ontología es la proporcionada por Gruber en [178], para el que es "una especificación formal y explícita de una conceptualización compartida". Los términos "formal" y "explícita" indican que todo lo incluido en ella está organizado y relacionado entre sí mediante conceptos y restricciones y dentro de una estructura cerrada o dominio. Con "conceptualización" se refiere a la generación de un modelo abstracto, definido por unas reglas, y que expresa, mediante relaciones y definiciones, los conceptos incluidos en él. Y con el término "compartida" se indica

que la comunidad que vaya a utilizarla debe ponerse de acuerdo en la selección de los términos incluidos en este modelo puesto que todos los miembros deberán hacer uso de las expresiones establecidas. Por lo que, de forma general, una ontología sería un modelo regido por unas especificaciones consensuadas por una cierta comunidad, que indican relaciones entre los diferentes conceptos de un dominio concreto.

En [179] Hendler detalla que la ontología es "un conjunto de términos de conocimiento donde se incluyen un vocabulario, unas relaciones y un conjunto de reglas para realizar una inferencia sobre un dominio particular". Esta nueva definición utiliza ya el concepto "reglas" para dotar al proceso de razonado de la capacidad de añadir al modelo de relaciones nuevas restricciones.

Independientemente de la interpretación, las ontologías tienen una serie de características que las describen. En [178][180]-[182] se incluyen algunas de ellas. Las más relevantes son las siguientes:

- Utilizan un vocabulario común con lo que se pretende proporcionar términos concretos, evitando ambigüedades a la hora de describir ciertos términos y con miras a compartir o combinar diversas ontologías entre diferentes ámbitos de aplicación.
- Incluyen una taxonomía para clasificar en grupos y subgrupos los individuales definidos en ella.
- Determinan las relaciones entre clases e individuos dentro de las mismas con el objetivo de poder crear nuevas relaciones durante el proceso de inferencia.
- Para su definición se utilizan diferentes lenguajes que determinarán la especificación de la ontología.

El objetivo final del diseño ontológico es el modelado del conocimiento. Esquematizar los conceptos y relaciones de un determinado dominio para poder realizar operaciones e inferencia sobre ellos. Sin embargo, en ocasiones el modelado de la escena actual se puede completar con el conocimiento obtenido en procesos de inferencia previos, proporcionando resultados más adecuados. Con ese fin surge la persistencia aplicada al modelado ontológico. En las siguientes subsecciones se introducen ambos conceptos para tener una visión más precisa de ellos.

6.2.1 Modelado del conocimiento

Existen diferentes técnicas para realizar el modelado del conocimiento cuyo último fin será el diseño e implementación de una ontología en un lenguaje concreto.

Las primeras técnicas utilizadas para este propósito se basaban en marcos y lógica de primer orden, mecanismos considerados de AI [183]. A pesar de ser técnicas de modelado antiguas, sus fundamentos siguen siendo utilizados o han sido adoptados como base de otros métodos que se aplican en la actualidad. En concreto, todavía se emplean las definiciones de las cinco categorías en las cuales se dividen los datos dentro de la ontología [178]. Aunque ya se habían introducido algunos de estos conceptos básicos previamente, hay que tener muy claro la diferencia entre ellos para usarlos adecuadamente y entender correctamente el modelo ontológico:

- Clase o concepto: es la base del conocimiento. Definen grupos de objetos con propiedades comunes. Se utilizan para estructurar el conocimiento dividiéndolo de forma jerárquica y permitiendo que puedan ser usados mecanismos de clasificación o razonado.
- Relación o propiedad: son utilizadas para establecer las relaciones entre clases o entre clases y miembros de las mismas. Se estructuran de modo que hay un inicio o dominio (concepto o elemento de la ontología) y un fin o rango (elemento al que apunta la relación o propiedad) que está relacionado de alguna manera con el dominio. Suelen estar definidas en la propia taxonomía.
- Función: determina un nuevo elemento de la ontología a través del cálculo de una expresión donde intervienen otros miembros conocidos.
- Instancia o individual: cada uno de los elementos de una clase. Representan a un objeto, situación, escena, etc., que forman parte de una o varias de las clases.
- Axiomas o reglas: definen relaciones entre elementos de la ontología fuera de la taxonomía permitiendo la inferencia o razonado de nuevas características o relaciones entre los objetos de la ontología.

En los últimos años el modelado ontológico ha evolucionado hacia métodos basados en lógica descriptiva, lenguaje UML (*Unified Modelling Language*) o diagramas Entidad/Relación o E/R.

El fundamento de la lógica descriptiva es la división de la ontología en dos partes complementarias diferenciadas por su fundamentación [183], el TBox y el ABox.

- En el TBox (*Terminological Box*) se definen los conceptos correspondientes a los términos de un determinado ámbito y que son declarados mediante propiedades generales. Representa por tanto el esquema o taxonomía del dominio, es decir, detalla la estructura y las relaciones conceptuales.
- El ABox (*Assertional Box*) contiene el conocimiento extendido, donde se declaran los individuales de la clase, roles entre instancias y otras afirmaciones (de ahí la A).

Se podría definir por tanto el TBox como la parte de la lógica que incluye las propiedades o relaciones entre clases de la ontología, mientras que el ABox contiene los individuales o conceptos de la escena y sus relaciones.

Las operaciones del TBox se basan en la inferencia y el rastreo o la verificación de miembros de la clase en la jerarquía (es decir, la colocación o la relación estructural de los objetos en el esquema) mientras que las operaciones del ABox se centran en normas para la comprobación de hechos, consistencia, etc.

Por otro lado, una de las principales razones por la que se utiliza la técnica UML para el modelado del conocimiento es que, debido a sus orígenes y a su facilidad de comprensión, su uso se ha extendido dentro de la comunidad de desarrolladores de software. Esto proporciona a este tipo de métodos una gran cantidad de herramientas para realizar el modelado. Este tipo de técnica es utilizada conjuntamente con OCL (*Object Constraint Language*) para representar los conceptos, propiedades y jerarquización de los individuales.

La última de las técnicas se basa en el uso de diagramas E/R. Esta metodología se centra en un modelado semántico en el que las clases de la ontología y las propiedades que las unen se representan como entidades y relaciones respectivamente en el diagrama E/R. Para el uso de axiomas se recurre a notación adicional que complemente el modelo.

6.2.2 Ontologías persistentes

En sus orígenes el uso de la semántica se centró en el procesado de la información para la realización de búsquedas contextuales y tratamiento de *Webs* mediante la creación de modelos de conocimiento representados por ontologías. Sin embargo, el almacenamiento de ese conocimiento inferido no era necesario, ya que con cada

acceso se generaba un nuevo entendimiento que no dependía de lo razonado anteriormente.

En la actualidad existen sistemas que han de tratar grandes cantidades de información, diferentes modelos o datos que puedan necesitar una clasificación o razonado previo.

Por ello es necesario aplicar mecanismos para almacenar el conocimiento generado con anterioridad y que, bajo nuevas condiciones, se pueda utilizar tanto el modelo existente como los históricos con los datos inferidos y así generar un conocimiento más completo. Es ahí donde entra en escena la persistencia.

Las ontologías persistentes se basan en tres elementos fundamentales: una ontología, una base de datos y una base de programación que proporcione soporte para el acceso a la base de datos y realización de consultas. Un ejemplo de este tipo de mecanismos son los descritos en [184]-[188].

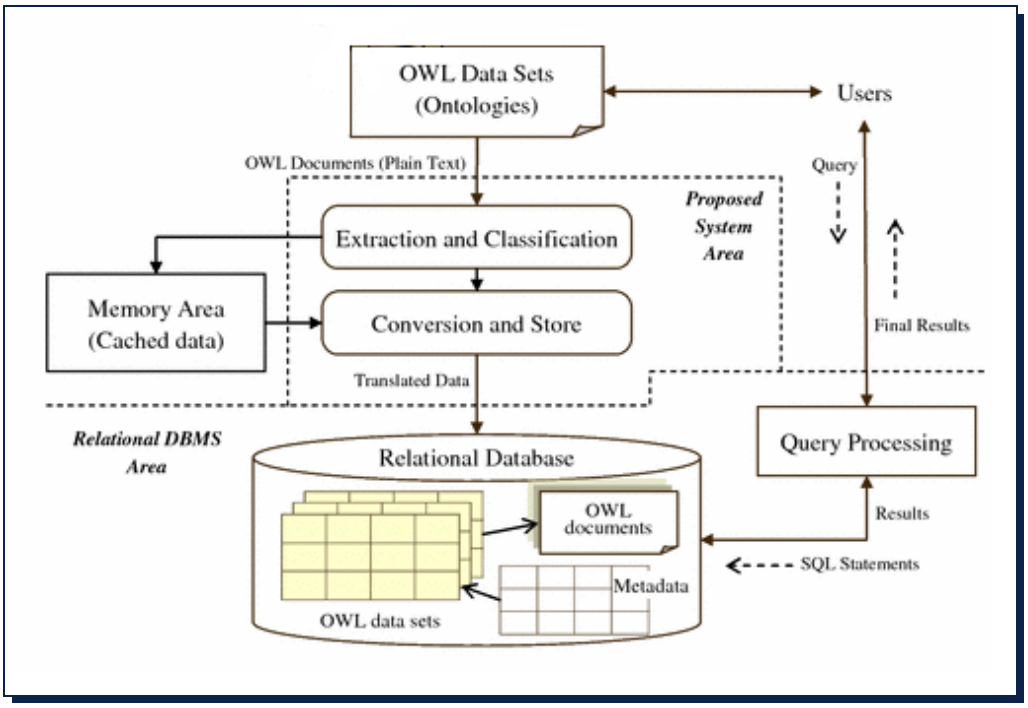


Figura 6.4. Esquema de un sistema configurado para realizar persistencia semántica [188].

En la Figura 6.4 puede verse una propuesta de arquitectura básica. El objetivo fundamental es que, para el usuario, una consulta o un proceso de inferencia sobre una ontología persistente sea exactamente igual a la que haría en el caso de una ontología que no posee esta característica. Por ello es necesario implementar una API de comunicaciones entre la interfaz de la base de datos, donde está

almacenado el conocimiento y el usuario que trata la ontología de la manera habitual.

La novedad en este tipo de propuestas es el uso de bases de datos. Permiten almacenar el conocimiento generado mediante el razonado, en disco, al contrario que los casos no persistentes que lo hacen en memoria volátil y en cuanto se apaga el sistema todo lo inferido desaparece.

Habitualmente se emplea una base de datos relacional o RDBMS (*Relational DataBase Management System*) ya que consiguen mantener las uniones generadas y de la ontología debido a su estructura basada en tablas y relaciones que no pueden estar duplicadas. Además, tienen una gran compatibilidad con el soporte de programación que debe permitir la conexión entre los datos generados mediante la inferencia y la propia base de datos y permitir las consultas para extraer la información incluida en ella.

Uno de los *frameworks* más usados para el almacenamiento y tratamiento de ontologías persistentes es Jena [189]. Esta API basada en Java posibilita, tanto la introducción de datos dentro de la ontología como su extracción.

6.3 Lenguajes

Una de las funciones principales de los lenguajes semánticos es la de codificar el modelo de conocimiento contenido en una ontología. Existe una gran variedad de lenguajes para realizar esta tarea pero los más utilizados son OIL (*Ontology Inference Layer*) [190], la combinación de DAML (*DARPA Agent Markup Language* - http://www.daml.org/) + OIL [182], RDF (*Resource Description Framework* - http://www.w3.org/RDF/) y OWL (*Web Ontology Language* - http://www.w3.org/TR/owl-features/).

El consorcio W3C (*World Wide Web Consortium*) que realiza recomendaciones de lenguaje basadas en la decisión conjunta del personal propio y del público en general para la elección de uno u otro, centradas en el ámbito de la *Web*, aconseja RDF y OWL.

Para la selección del lenguaje más adecuado para cada ontología hay que tener en cuenta dos aspectos fundamentales:

- Debe especificar de forma correcta los términos de la ontología, ya que puede que ésta sea utilizada posteriormente en otro sistema que utilice un lenguaje distinto.
- Tiene que realizar de forma correcta y lo más eficientemente posible los razonamientos.

6.3.1 Lenguaje RDF

Este lenguaje está basado en pares de objetos o recursos relacionados mediante propiedades [191]. Los objetos simbolizan a cualquier individuo o clase de la ontología. Por su parte, los atributos o propiedades se emplean para relacionar los recursos.

La principal característica de este lenguaje es su capacidad para dotar de semántica a un documento sin necesidad de ahondar en su estructura. Mediante este tipo de lenguaje se forma una red de conocimiento semántico basándose en un grafo dirigido. En él, el conocimiento es representado mediante nodos que son conectados con otros mediante las propiedades o atributos.

La unidad básica representativa de conocimiento es la tripleta, que como se indicó en la Sección 6.1 Semántica, está formada por un sujeto (clase o individuo), un predicado (propiedad) y un objeto (clase, individuo o valor) (Ver Figura 6.2).

6.3.2 Lenguaje OWL

Este lenguaje está diseñado para aplicaciones o ámbitos de la ciencia que necesitan realizar un procesado de la información [192]. Más enfocado a la publicación y compartición de ontologías, OWL se considera una variación de DAML+OIL y a su vez una extensión de RDF (Figura 6.5).

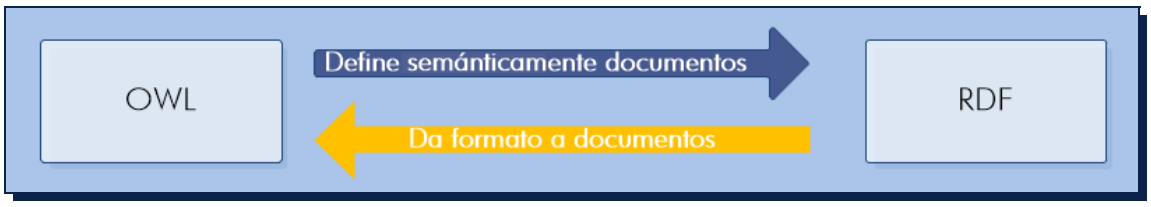


Figura 6.5. Relación entre el lenguaje OWL y RDF.

RDF es un lenguaje que especifica un formato. Cualquier tripleta es válida en RDF, sin embargo, si el contenido no tiene significado coherente por lo que no es

semánticamente correcta, no es apropiada en OWL. Por ejemplo, una tripleta que podría incluir un documento RDF bien formado sería:

Juan esHijoDe Manzana

porque incluye sujeto predicado y objeto, sin embargo no se podría expresar en lenguaje OWL porque semánticamente no tiene sentido.

A su vez, este lenguaje tiene tres variaciones en función de la las capacidades de expresión y complejidad que se quieran dar a la ontología. Estos son, clasificados por complejidad, de menor a mayor: OWL Lite, OWL DL (*Description Logic*) y OWL Full.

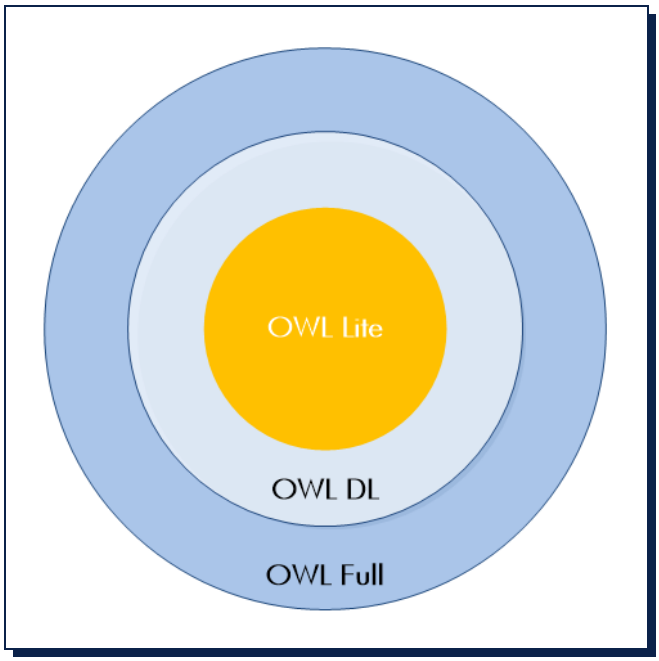


Figura 6.6. Relación entre los diferentes niveles del lenguaje OWL.

En el primero de los casos, el denominado OWL Lite, su uso se centra en crear o definir un axioma donde se especifiquen, de forma clara y simple, la relación jerárquica entre las diferentes entidades de la ontología, añadiendo por supuesto, un conjunto sencillo de restricciones entre las mismas.

Por su parte OWL DL aumenta la complejidad de la ontología, pero a su vez dota a este modelo de la capacidad de realizar cálculos, aumentado la descripción de los miembros de la misma pero manteniendo su eficiencia a la hora de razonar y generar nuevas relaciones.

Por último, OWL Full es más expresivo que el anterior y su enfoque se centra en esa característica siendo menos importante el procesamiento de datos. La diferencia con OWL DL, sobre todo, es la posibilidad de este lenguaje de definir varios niveles jerárquicos dentro de las clases, consiguiendo que la ontología pueda diferenciar mejor entre los distintos niveles de la misma.

La relación entre los tres niveles del lenguaje puede verse en la Figura 6.6.

6.4 Reglas

En la literatura existen varios trabajos como [193]-[196], que presentan un estudio de nuevas propuestas para el uso de reglas de inferencia o de cómo influye la aplicación de las mismas a la hora de realizar consultas.

Las reglas son expresiones en lenguaje semántico que permiten crear relaciones entre entidades de la ontología. Se utilizan como apoyo o conocimiento extra del razonador. Esta información le deja inferir relaciones que no habían podido ser expresadas en la propia ontología.

Las reglas de inferencia están formadas por dos partes diferenciadas. En primer lugar establecen unas relaciones entre las clases u objetos, llamadas premisas, que cuando se cumplen hacen que se llegue a un resultado o conclusión.

Al igual que en el caso de los lenguajes, el W3C tiene un protocolo de definición de reglas recomendado basado en OWL DL y OWL Lite, SWRL (*Semantic Web Rule Language* - http://www.w3.org/Submission/SWRL/). Un ejemplo de su sintaxis es el siguiente:

$$padre(?a,?b) \wedge hermano(?b,?c) \Rightarrow tío(?a,?c)$$

que se podría resumir como:

si tenemos dos individuales "a" y "b" donde "a" tiene de padre a "b" y a la vez la entidad "b" tiene de hermano a "c", una consecuencia de ello es que el sujeto "a" tiene de tío al objeto "c".

La regla presentada establece unas relaciones genéricas dentro de la clase "Personas". En función de los individuales instanciados para esta clase en un momento determinado, la regla se ejecutará o no considerándose como válida la conclusión. Por ejemplo, si y sólo si "Manuel esPadreDe Juan" y "Manuel esHermanoDe Carlos" se puede determinar que "Carlos esTíoDe Juan".

En este caso Jena [189][197] proporciona la capacidad de trabajar con RDF, OWL y realizar consultas SPARQL (*SPARQL Protocol and RDF Query Language*). La sintaxis de las reglas generadas para su aplicación en Jena, denominadas *Jena Rules* [198][199], es específica para esta aplicación. La estructura se basa en la de RDF fundamentada en sujeto-predicado-objeto. Un ejemplo de la misma regla pero escrita para Jena sería:

(?a owl:tienePadre ?b) (?b owl:tieneHermano ?c) ⇒ (?a owl:tieneTío ?c)

6.5 Razonadores

La ontología, gracias al lenguaje semántico, modela conocimiento a través de las relaciones entre diferentes entidades. Sin embargo, la generación de nuevas conexiones y por lo tanto conocimiento es una de las características principales del pensamiento humano. Es aquí donde entran en escena los razonadores.

Existen diversidad de técnicas a la hora de afrontar la inferencia de nuevos datos en función de la representación del conocimiento. En este caso el estudio se centra en los mecanismos de razonado basados en reglas.

Los primeros sistemas que utilizan reglas como complemento son los descritos en [200]. Éstos cuentan con una arquitectura basada en tres puntos: una base de conocimiento, una de reglas y, por supuesto, el razonador.

La base de conocimiento corresponde a la ontología que, con la aplicación de reglas, se va razonando hasta llegar a un estado final que contempla el modelado completo y que incluye la base de conocimiento inicial y el inferido.

Sin embargo, la parte más importante de este razonamiento es la base de reglas. Estas reglas, descritas basándose en una teoría conductivista (si A entonces B), se utilizan para apoyar y modificar el comportamiento del razonador permitiendo crear relaciones de forma más precisa o que antes no hubiesen sido capaces de descubrirse.

Puede ocurrir que ciertas reglas, aunque bien definidas, lleven a generar una situación no deseable como un conflicto o inconsistencia. Para ello existen diferentes medidas o protocolos que intentan evitarlo determinando la prioridad entre las reglas. Entre ellos destacan los siguientes:

- Regla más general: toda regla contiene un número determinado de declaraciones o elementos que indican su complejidad. En este caso se da prioridad a aquella que incluye menos condiciones en su declaración.
- Regla más concreta: a diferencia del caso anterior, en esta ocasión se prioriza lo determinado por la regla que contenga más condiciones.
- Regla más novedosa: se aplica primero la que contenga las condiciones que impliquen el uso de datos más nuevos.
- Regla más antigua: contraria a la anterior, se elige la regla con los datos más viejos.
- Orden: las reglas se definen acompañadas por un número. Este número puede ser utilizado para indicar el orden de ejecución.
- Principio de refracción: se emplea para evitar que se produzca un bucle infinito en la inferencia. Una regla no puede volver a aplicarse salvo que se incluyan nuevos individuales en el proceso de inferencia.

Por su parte, el razonador, utilizando la base de conocimiento y en función de las reglas, genera la inferencia, deduciendo las nuevas relaciones.

Ésta se puede realizar de dos formas distintas: hacia adelante o hacia atrás.

En el primero de los casos este proceso implica que el razonador crea unas relaciones inferidas coherentes con los datos iniciales. Es el proceso más lógico. Se selecciona una condición establecida y se realiza el razonado.

Por su parte, el razonamiento hacia atrás ejecuta el proceso inverso, tratando de demostrar una premisa partiendo de un hecho ya razonado. Si esto no se demuestra se siguen buscando reglas que consigan demostrarlo.

Existen distintas herramientas que incluyen la posibilidad de usar esta técnica en el proceso de razonado, entre las que se encuentra Jena. Esta herramienta hace posible crear un modelo inicial a partir del modelo existente, las reglas de inferencia incluidas y los individuales instanciados. Este modelo, antes del proceso de razonado, puede ser validado para verificar que no existen inconsistencias. Jena dispone de distintos razonadores para seleccionar el más adecuado en función de la situación específica:

- Razonador transitivo: básico, permite únicamente almacenar y consultar jerarquías de clases y sus propiedades (únicamente las propiedades transitiva y simétrica de rdfs:subPropertyOf y rdfs:subClassOf).
- Razonador RDFS: implementa un subconjunto configurable de la funcionalidad ofrecida por RDFS (*RDF Schema*).
- Razonador para OWL específico para el lenguaje OWL Lite.
- Razonador DAML: soporte mínimo para la inferencia mediante DAML.
- Razonador general de reglas: consigue inferir hechos incluyendo la resolución de reglas.

Además, Jena permite la incorporación de nuevos razonadores externos como Pellet (http://clarkparsia.com/pellet). Pellet es un razonador independiente por lo que se puede utilizar de manera individual. Es de código abierto, está implementado en Java y efectúa un razonado incremental, es decir, para cada pequeño cambio en el ABox realiza una nueva inferencia. Está especialmente diseñado para el procesamiento de ontologías escritas el lenguaje OWL DL. Se puede encontrar una amplia descripción de este razonador en [199].

6.6 Caracterización de escenas: el modelado semántico

De forma general, el modelado semántico de las escenas se lleva a cabo en dos subetapas diferentes, la traducción semántica y el proceso de inferencia en el núcleo semántico:

- En el proceso de traducción semántica los datos obtenidos por las cámaras son procesados, interpretando los distintos parámetros de los objetos en movimiento. Esta información, convertida en datos semánticos, se introduce en la ontología para cada objeto, previamente definido, como una nueva instancia. Además, si como complemento se dispone de datos procedentes de sensores distribuidos por la escena, se incluyen los valores detectados como nuevos individuales dentro de la ontología. A su vez, la información de las zonas detectadas en el proceso de modelado espacial de la escena es tratada y cada ruta detectada se integra en el modelo ontológico como un nuevo objeto. Indicar que, los datos insertados en la ontología se procesan previamente y se les aplican las correcciones necesarias en función de la colocación de la cámara, tanto en altura como en ángulo.

- En la segunda subetapa, el nuevo modelo populado con los individuales de la escena actual es razonado junto con las reglas específicas diseñadas para ese dominio concreto. Como resultado de este proceso, cada objeto es identificado como miembro de una clase concreta dependiendo de los parámetros de la misma. En el dominio del tráfico, por ejemplo, un objeto es identificado como peatón o vehículo en función de sus dimensiones, velocidad, etc. Además, cada ruta se etiqueta como acera o carretera en función del tipo de objetos que circulen por ella, las velocidades de los mismos, etc.

  Resaltar que el núcleo semántico es un módulo fácilmente configurable que permite el cambio o adaptación a diferentes dominios, una de las principales ventajas del sistema propuesto. Sencillamente un cambio de ontología y reglas consigue el cambio de dominio.

Por otra parte, como complemento del proceso de modelado semántico, es muy sencillo incluir reglas adicionales en el modelo para conocer si el comportamiento de un objeto es normal o se encuentra en una situación excepcional. Por ejemplo, una regla puede especificar que, si se detecta un objeto clasificado como *Vehículo* circulando por una ruta de la clase *Acera,* se lanza una alarma que especifica esta incidencia.

Para ilustrar el funcionamiento se presentan ejemplos de su aplicación en el dominio del tráfico.

En este caso particular, la cámara recibe la señal de video procedente de las calles en las que vehículos y peatones se mueven en carreteras, aceras o pasos de peatones. El objetivo en este caso es construir un modelo de la escena indicando la clase a la que pertenece cada objeto en función de sus propiedades de movimiento, obtenidas aplicando algoritmos de visión artificial de bajo nivel. En este ámbito de conocimiento se identifican como vehículos los objetos de mayor velocidad y tamaño (superando un determinado umbral) y como peatones los de menor, por ejemplo.

Una vez identificados los objetos, se etiquetan las diferentes zonas detectadas en la etapa de modelado espacial. En el dominio usado a modo de ejemplo, las reglas clasifican como calzadas las zonas por las que circulan objetos previamente identificados como vehículos, aceras como las zonas en las que habitualmente se encuentran peatones y pasos de peatón si se identifican ambos tipos de objetos.

Así, pasado un tiempo de aprendizaje se dispone, para cada fotograma de la colocación del objeto dentro de la imagen. En este caso, puede ser algo como: *Objeto1 que es un Vehículo está Localizado_en la Ruta2 que es una Calzada.*

A partir de este momento, determinar situaciones anómalas es muy sencillo ya que la ontología identifica las diferentes clases de localizaciones en las que se pueden encontrar los distintos objetos. En el dominio del tráfico, una situación anómala será un vehículo circulando por un acera, un exceso de velocidad, etc.

A continuación se va a describir el diseño del modelo ontológico por su importancia en la caracterización de las escenas de un determinado dominio y el funcionamiento del sistema a bajo nivel para entender la solución propuesta y verificar su viabilidad.

6.6.1 Diseño del modelo ontológico y las reglas de inferencia

El núcleo semántico utiliza un modelo de conocimiento basado en ontologías y un conjunto de reglas pensadas específicamente para los objetos, rutas y situaciones de alarma que pueden aparecer en la escena de un dominio concreto.

La ontología es la representación abstracta de los diferentes actores del ámbito de conocimiento. De forma general se pueden identificar:

- Clases: grupos de actores con similares características.
- Individuales o instancias: actores particulares que pertenecen a una determinada clase.
- Propiedades: definen las características de cada actor y relacionan las distintas clases.

En el dominio del tráfico, por ejemplo, la ontología incluye la clase *Vehículo* con propiedades como *Velocidad* o *Tamaño.* En un escenario específico aparece, por ejemplo, un objeto *Objeto1* que pertenece a la clase *Vehículo* que tiene una *Velocidad* de 50 km/h y un individual *Objeto2* identificado como perteneciente a la clase *Peatón* con una *Velocidad* de 4km/h.

Por otro lado, las reglas de inferencia son una serie de condiciones y operaciones lógicas que se ejecutan sobre la ontología populada durante el proceso de razonado, y que, en función de los individuales y las propiedades de los mismos, se activan o no generando una salida y otra. Por ejemplo, la ontología posiciona los objetos de la clase *Vehículo* dentro de los de la clase *Localización*, es decir, tienen la propiedad *Localizado_en.* Si un *Objeto1* de la clase *Vehículo* está *Localizado_en*

un individual *Ruta1* de la clase *Localización* y hay una regla que indica que si un *Vehículo* se encuentra en una *Acera*, la situación es de alarma y hay dos posibilidades:

- Si la *Ruta1* es una *Carretera* esta regla no se activa.
- Si la *Ruta1* es una *Acera*, *Objeto1* pasará a tener la propiedad *Situación_de_alarma* con valor *Fuera_de_Carretera*.

6.6.1.1 El modelo ontológico

En esta Tesis, el razonado semántico se realiza tomando como base tres ontologías, que se dividen en dos categorías. Dos ontologías son genéricas e independientes del dominio de funcionamiento y la otra es específica del ámbito de estudio, en este caso de ejemplo, una ontología concreta para el control de tráfico.

Las ontologías generales sirven como base para la inclusión en el modelo semántico de los distintos individuales detectados en movimiento en la escena y las diferentes rutas o zonas determinadas en la fase de caracterización espacial, por tanto, son independientes del campo de aplicación. Estos individuales o zonas pertenecen a una u otra clase en función de la ontología específica y las reglas adicionales que se apliquen. Es decir, un objeto en movimiento tiene una serie de características como son la posición dentro de la escena, la dirección de movimiento, una velocidad, etc., independientes del dominio de funcionamiento del sistema. Es la ontología específica y las reglas las que determinan que este objeto es un vehículo, si se trabaja en el dominio particular del control del tráfico o un lobo si el sistema monitoriza un espacio nacional protegido.

Pero, ¿por qué esta diferenciación? Una limitación de las ontologías y las reglas, es que el razonador sólo se basa en la información de la que dispone en un momento concreto para realizar el proceso de inferencia. Esto hace que en muchas ocasiones tras este proceso se llegue a conclusiones erróneas. Por ejemplo, si en un momento determinado un peatón está localizado en una ruta (donde, en el momento actual, no existen otros objetos), esta ruta en este proceso de inferencia se considera una acera sin tener en cuenta si previamente han pasado por ella multitud de vehículos. Para evitar estos problemas es necesario utilizar ontologías persistentes que posibilitan el almacenamiento de históricos y proporcionan mejores resultados.

Sin embargo, ¿es necesario almacenar todas las propiedades de todos los objetos que van apareciendo en la escena? Cuando un objeto aparece en la escena se

determinan sus parámetros de movimiento. Posición, velocidad, dirección de movimiento, dimensiones son algunas de estas características. Sin embargo, cuando el objeto desaparece de la escena, muchos de estos parámetros no se van a utilizar en procesos de razonamiento posteriores y lo único que va a interesar son las conclusiones extraídas con el procesamiento de los mismos. Por ejemplo, mediante un proceso de análisis de los datos extraídos y su evolución con el tiempo para un objeto concreto, se determina la clase a la que pertenece (en el dominio del tráfico se clasifica como peatón o vehículo). Es esta información la que va interesar mantener y no los datos concretos que dan como resultado esa conclusión. Ahí es donde entra en juego la necesidad de utilizar diferentes ontologías.

Por una parte, de forma general e independiente del dominio, todos los objetos en movimiento y las localizaciones disponen de los mismos parámetros, que sirven como base en el proceso de razonamiento y, por otro lado, las deducciones a las que se llegue después del procesado de los mismos, va a clasificarlos dentro de una u otra clase del dominio de la escena. Es esta información inferida la que es almacenada para que sirva de entrada en nuevos procesos de razonado y por tanto requiere la utilización de una ontología persistente. Sin embargo, las características de un objeto o ruta en un fotograma concreto es información "volátil" y por tanto las ontologías que las definan no necesitan esa propiedad. Por otro lado, objetos y regiones son modelados utilizando dos ontologías diferentes ya que son conceptos diferentes aunque entre ellos existan relaciones.

A modo de resumen, el modelado del conocimiento se realiza utilizando tres ontologías diferentes como puede verse en la Figura 6.7. Por un lado una ontología define los objetos en movimiento como el conjunto de una serie de propiedades. De igual forma, diferentes rutas obtenidas en el proceso de modelado espacial son individuales de la clase *Localización* y están definidos mediante su propia ontología. Cada dominio se modela con una ontología persistente que almacena la información inferida a lo largo del tiempo para servir de realimentación al sistema y mejorar los procesos de razonamiento que se realicen con posterioridad. Son esta ontología y las reglas que la complementan los únicos elementos que hay que rediseñar para que el sistema funcione en otros dominios. Esta ontología, por una parte, define los objetos que se pueden encontrar en ese campo de aplicación y por otra parte se identifica las posibles localizaciones.

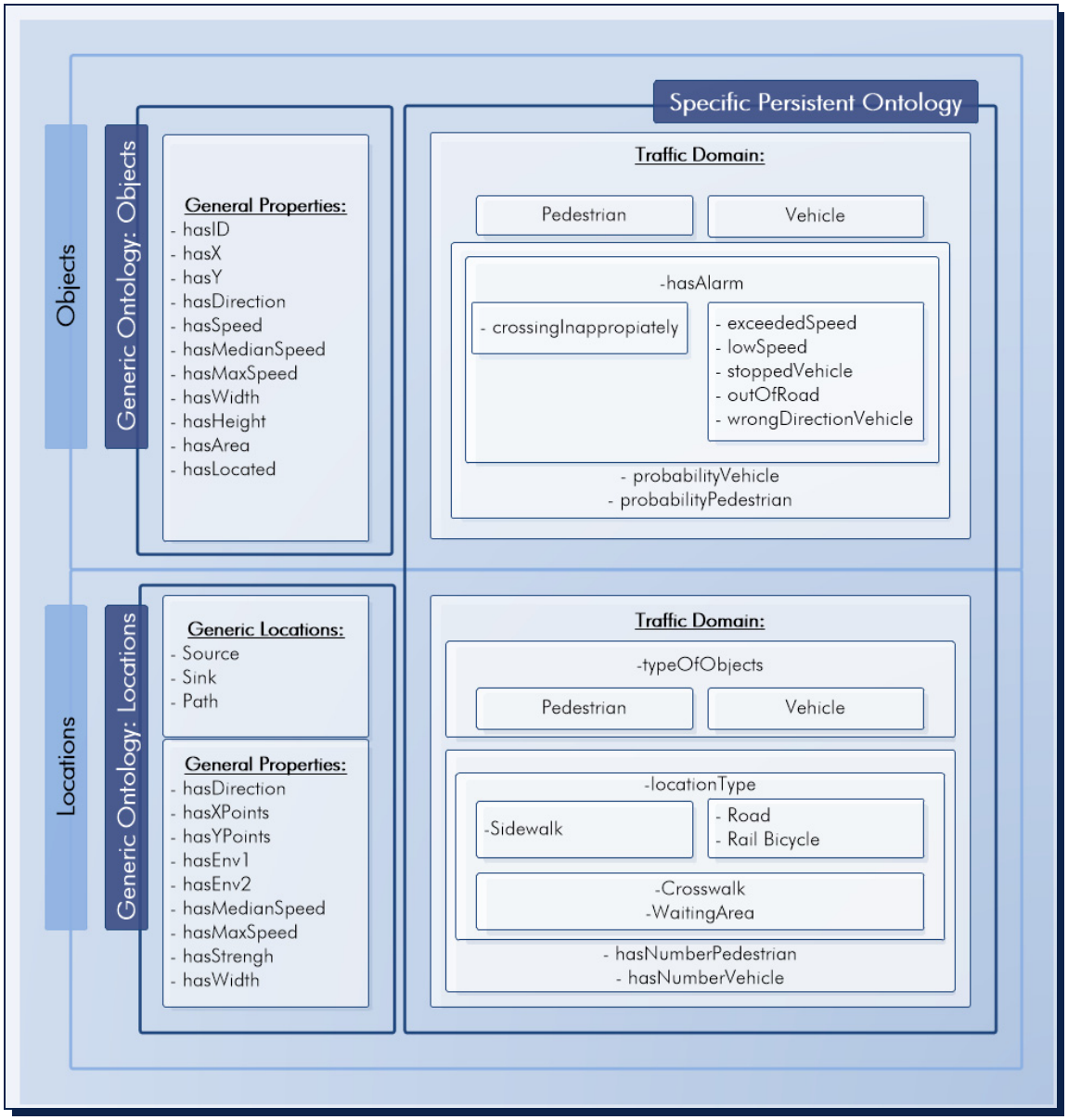


Figura 6.7. Diseño del modelo ontológico.

6.6.1.1.1 Modelado de los objetos en movimiento

La primera ontología que se va a describir es la que se utiliza para definir, de forma general, las características que tienen los objetos en movimiento de la escena. Para ello se crea una clase *OBJECT* (Figura 6.8). Esta clase incluye todos los individuales que se detectan en el fotograma mediante el análisis de la escena. A su vez, sirve de base para la categorización de estos individuales dentro de subclases creadas en la ontología específica.

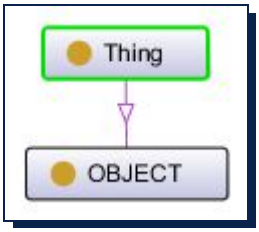


Figura 6.8. Clase "Objeto".

Esta ontología contiene las siguientes propiedades que son asignadas a los individuales de la clase *OBJECT*:

- *Data Property* (Las siguientes propiedades serán asignadas a todo individual de la clase *OBJECT* introducido en la ontología):
    - *hasID:* Asocia al individual un valor numérico correspondiente a su identificador. Este identificador se asigna con la aparición del individual en la escena y es único para cada objeto.
    - *hasX:* Asigna al individual un valor numérico correspondiente a su posición (coordenada en el eje X) dentro de la escena.
    - *hasY:* Fija al individual un valor numérico correspondiente a su posición (coordenada en el eje Y) dentro de la escena.
    - *hasDirection:* Asocia al individual un valor numérico correspondiente a la dirección de su movimiento. Este valor toma como referencia las agujas del reloj, es decir, si un objeto parte de la posición (0,0) y en el fotograma siguiente se encuentra en la (1,0), la dirección asignada será 3. Del mismo modo, si desde la posición (1,0) se desplaza a la (1,-1), la dirección será 6.
    - *hasSpeed:* Asigna al individual un valor numérico correspondiente a su velocidad en el fotograma (se calcula como la variación de su posición con respecto a la localización en el fotograma previo entre el tiempo entre fotogramas).
    - *hasMedianSpeed:* Asocia al individual un valor numérico correspondiente a la velocidad media del individual a lo largo de su recorrido en la escena.

- *hasMaxSpeed*: Establece al individual un valor numérico correspondiente a la máxima velocidad del individual a lo largo de su recorrido.
- *hasWidth*: Asocia al individual un valor numérico correspondiente a su anchura en la imagen en un fotograma determinado.
- *hasHeight:* Asigna al individual un valor numérico correspondiente a su altura en la imagen en un fotograma determinado.
- *hasArea:* Fija al individual un valor numérico correspondiente al valor de su área en el fotograma concreto.

- *Object Property*
  - *hasLocated:* Esta propiedad se utiliza para asociar un individual de la clase OBJECT con un individual de la clase LOCATION, indicando que un objeto se encuentra dentro de una ruta.
    - *Object1 hasLocated Ruta1*

6.6.1.1.2 Modelado de las localizaciones

La segunda de las ontologías genéricas es la que modela las localizaciones. En ella se define la clase general *LOCATION* que sirve de base para la introducción de los individuales de localización determinados durante el proceso de detección de rutas de la escena a estudio. Dentro de esta clase está definida la subclase *GenericLocations* que contiene tres nuevas clases que etiquetan los distintos tipos de localizaciones (Figura 6.9):

- *Path*: Pertenecen a esta clase todas las zonas de la escena por las que se mueven los objetos. Es decir, son individuales de la clase *Path* las rutas determinadas en el proceso de caracterización de espacios.
- *Source*: Engloba las zonas por las que aparecen los objetos en la escena. En este caso, son individuales de esta clase las fuentes detectadas.
- *Sink*: Engloba las zonas por las que habitualmente desaparecen los objetos de la escena. Son individuales de esta clase los sumideros determinados durante el proceso de detección de rutas.

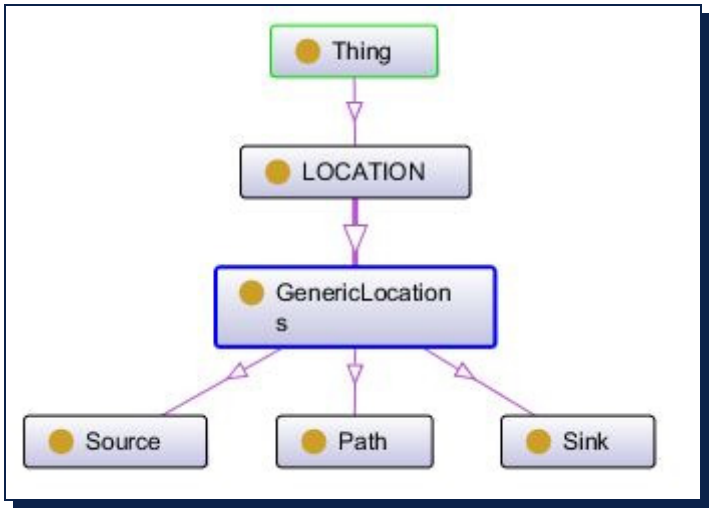


Figura 6.9. Clase "Localización".

En este caso las propiedades, definidas todas de tipo *Data Property,* son las siguientes:

- *hasDirection*: Asocia al individual de tipo *Path* un valor numérico indicando la dirección habitual de movimiento de los objetos que circulan por esa ruta. Al igual que en el caso de la ontología que define los objetos en movimiento toma como referencia las agujas del reloj.

Tal y como se define en la Sección 5.3.1, cada ruta está caracterizada por una secuencia de puntos centrada y dos envolventes que representan el tamaño medio de la ruta. Así pues:

- *hasXPoints:* Asocia al individual de tipo *Path* las coordenadas X de la posición de su secuencia de puntos centrada.
- *hasYPoints:* Asocia al individual de tipo *Path* las coordenadas Y de la posición de su secuencia de puntos centrada.
- *hasEnv1X:* Asocia al individual de tipo *Path* las coordenadas X de la posición de una de sus envolventes, denominada en este caso envolvente 1.
- *hasEnv1Y:* Asocia al individual de tipo *Path* las coordenadas Y de la posición de su envolvente 1.
- *hasEnv2X:* Asocia al individual de tipo *Path* las coordenadas X de la posición de la otra de sus envolventes, denominada en este caso envolvente 2.

- *hasEnv2Y:* Asocia al individual de tipo *Path* las coordenadas Y de la posición de su envolvente 2.
- *hasMaxSpeed:* Asigna al individual de tipo *Path* la velocidad máxima de los individuales del tipo OBJECT que han pasado por él.
- *hasMedianSpeed:* Asigna al individual de tipo *Path* la velocidad media de los individuales del tipo OBJECT que han pasado por él.
- *hasStrengh:* Asigna al individual de tipo *Path* un valor numérico con el número de puntos que caracterizan la ruta (número de puntos de la secuencia de puntos centrada).
- *hasWidth:* Asigna al individual de tipo *Path* un valor numérico con la anchura de la ruta en cada punto (distancia entre envolventes).

6.6.1.1.3 Modelado semántico específico: Aplicación al control de tráfico

Por otro lado se encuentra la ontología específica del dominio de aplicación. Este tipo de ontologías se crea como complemento de las anteriores y su objetivo es dotar a éstas de unas etiquetas específicas que describan la escena en lenguaje formal. Esta ontología es persistente, característica que no influye en el diseño de la misma y que permite el almacenamiento de forma transparente en una base de datos relacional, aportando funcionalidades adicionales al sistema y posibilitando una caracterización de escenas más fiable.

A modo de ejemplo, se diseña esta ontología para el modelado del dominio del control de tráfico. Hay que tener en cuenta que se utiliza como complemento de las dos anteriores con lo que, en el caso de tener los modelos generales clases ya definidas, éstas han de especificarse con el mismo nombre.

En este sentido se definen dentro de la ontología tres clases principales (Figura 6.10):

1. La primera de las clases principales corresponde a la clase *OBJECT* y contiene dos subclases correspondientes a los dos tipos de sujetos a estudio:
    - *Pedestrian:* Engloba las instancias de tipo *OBJECT* que el proceso de razonado determina que por sus características son de tipo peatón.
    - *Vehicle:* Abarca los individuales de tipo *OBJECT* que, en este caso, sus características hacen que se clasifiquen como de tipo vehículo.

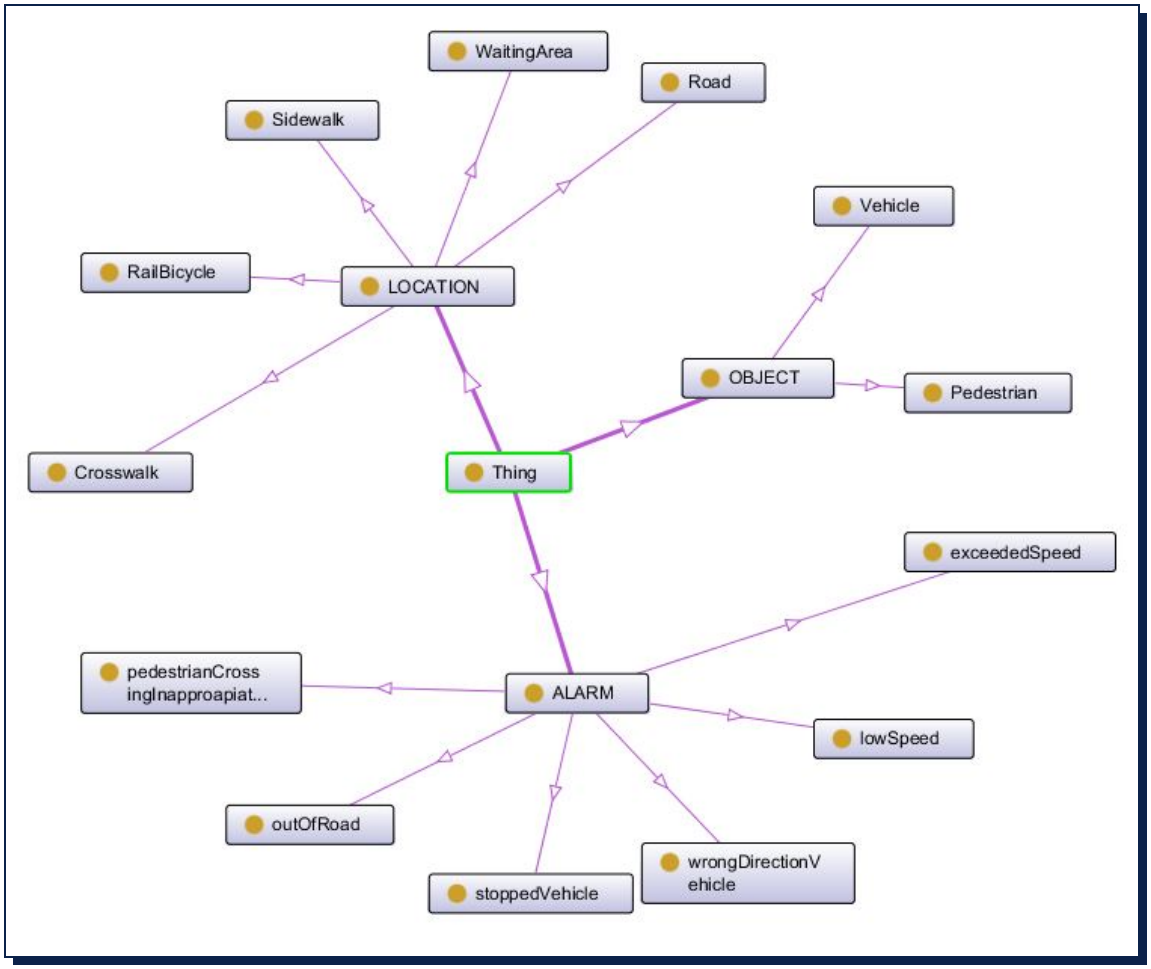


Figura 6.10. Ontología específica para el control de tráfico.

2. Por otro lado se encuentra la clase *LOCATION*. En ella están definidas las subclases correspondiente al tipo de rutas que se pueden encontrar en este campo de aplicación:

   - *Crosswalk:* Incluye todos los individuales de la clase *Path* que el proceso de inferencia ha determinado que son paso de peatones.
   - *RailBicycle:* Engloba a todos los individuales de la clase *Path* que tras el razonado se etiquetan como carril bici.
   - *Road:* Contiene todos los individuales de la clase *Path* que son clasificados como carretera.
   - *Sidewalk:* Abarca a todos los individuales de la clase *Path* que son catalogados como acera.

- *WaitingArea:* Engloba a todos los individuales de la clase *LOCATION* que representan una zona que se considera zona de espera o transición.

3. Por último se define la clase *ALARM.* Esta clase está diseñada para dotar al sistema de la capacidad de determinar e identificar, de manera semántica, las diversas situaciones anómalas que puedan darse en una escena. En la Figura 6.7 se incluyen dentro de *OBJECT* porque están relacionadas con el comportamiento de un objeto concreto. Dicha clase contiene varias subclases:

    - *exceededSpeed:* Engloba a todos los objetos de la clase *Vehicle* que, encontrándose en una *LOCATION* del tipo *Road*, poseen una velocidad muy superior a la asignada como máxima para ese tipo de localización.

    - *lowSpeed:* Etiqueta a todos los objetos de la clase *Vehicle* que, dentro de una determinada *LOCATION* del tipo *Road*, poseen una velocidad inferior a la esperada para ese tipo de localización.

    - *outOfRoad:* Contiene los objetos de la clase *Vehicle* que se encuentran en una *LOCATION* distinta del tipo *Road*.

    - *stoppedVehicle:* Incluye los objetos de la clase *Vehicle* que encontrándose en un *Path* del tipo *Road* poseen velocidad cero y a su vez no están localizados en una zona de espera (*WaitingArea*).

    - *wrongDirectionVehicle:* Engloba a todos los objetos de la clase *Vehicle* que, encontrándose en una *LOCATION* del tipo *Road*, poseen una dirección distinta a la asignada como habitual para esa localización.

    - *pedestrianCrossingInapproapiately:* Agrupa a los objetos de la clase *Pedestrian* que se encuentran posicionados en un *Path* de tipo *Road.*

Para esta ontología, al igual que en el caso de la primera ontología genérica, se definen dos tipos de propiedades. Éstas pretenden complementar las incluidas en las dos ontologías genéricas:

- *Object Property:*

    - *hasAlarm:* Asocia a un individual de la clase *OBJECT* a una de las subclases de la clase *ALARM.*

- *Data Property:*

  Ante la aparición de un nuevo objeto en movimiento en la escena mediante el proceso de inferencia hay que determinar la clase a la que pertenece. Dentro del dominio del control de tráfico se han definido dos, *Pedestrian* y *Vehicle*. Para conocer si este objeto pertenece a una clase u otra se necesitan saber las características del mismo durante un tramo de su desplazamiento en la escena, ya que los valores que presenta en un momento puntual no son concluyentes. Por ejemplo, si en un fotograma determinado aparece en la escena un vehículo, que por perspectiva tiene unas dimensiones reducidas y además circula muy despacio porque hay un semáforo en rojo, el sistema de razonamiento puede determinar que, en ese *frame* ese objeto es un peatón. Sin embargo, cuando el semáforo se ponga otra vez en verde y el objeto adquiera velocidad, al circular por una localización de tipo carretera el sistema determina que es un vehículo. Es decir, el razonador no va a clasificar un determinado objeto en movimiento dentro de una clase hasta que no dispone de suficiente información como para determinarlo de manera fiable. Para realizar este procesamiento se incluyen las propiedades *probabilityPedestrian y probabilityVehicle,* datos numéricos que se le asigna a un individual de la clase *OBJECT* y que ayuda a determinar si dicho elemento es de la clase *Pedestrian* (peatón) o *Vehicle* (vehículo). Estos valores se actualizan cada fotograma con las características concretas que posee el objeto en ese momento. Así, si en el primer fotograma el objeto por sus parámetros puede determinarse que es un peatón, la propiedad *probabilityPedestrian* tomar el valor 1 y *probabilityVehicle* se mantiene a 0, valor que toman por defecto ambas propiedades. Si los parámetros del siguiente fotograma indican que el objeto en ese fotograma es un peatón se incrementa en 1 el valor de *probabilityPedestrian* o, en caso contrario se suma 1 al valor de la propiedad *probabilityVehicle.* Con la evolución de estas probabilidades las reglas diseñadas son las encargadas de discernir realmente la clase a la que pertenece el objeto.

  Por otro lado se han definido propiedades que ayudan al razonador a determinar la subclase en la que está incluida una determinada localización:

  - *hasNumberPedestrian:* Asigna a un individual de la clase *Path* un valor numérico correspondiente al número de objetos de tipo peatón que han pasado por esa localización.

- *hasNumberVehicle:* Establece, para un individual de la clase *Path*, el valor numérico de los objetos identificados como vehículos que han pasado por esa ruta.

6.6.1.2 Las reglas de inferencia

En ciertas ocasiones los procesos de razonado de un modelo ontológico populado no proporcionan el conocimiento esperado por lo que se hace imprescindible incluir una serie de reglas de inferencia que complementen el modelo y aporten información adicional durante el proceso de razonado.

El proceso de inferencia permite obtener conclusiones en base a los datos de los que se dispone y las expresiones que marcan las relaciones entre los mismos. En lógica, una regla de inferencia es un esquema para construir inferencias válidas. Estos esquemas establecen relaciones sintácticas entre un conjunto de expresiones llamadas premisas, que, cuando se cumplen, hacen que se llegue a un resultado llamado conclusión formada también por un conjunto de fórmulas. Todas las expresiones son tripletas RDF donde sujeto, predicado u objeto son variables. Cuando, para un determinado valor, las premisas son válidas, la regla se ejecuta y las tripletas establecidas como conclusión son correctas.

En la Figura 6.11 se observa un ejemplo de regla de inferencia. En este caso, r1 identifica la regla y las dos líneas siguientes son las premisas. En ellas se puede observar que hay dos variables, ?o y ?value. Cuando estas variables toman ciertos valores las dos premisas se verifican con lo que la conclusión (precedida por ->) es válida.

```
[r1:
  (?o rdf:type wrongPlace:OBJECT)
  noValue(?o wrongPlace:probabilityVehicle ?value)
  ->(?o wrongPlace:probabilityVehicle 0)]
```

Figura 6.11. Regla de inferencia.

En el sistema diseñado se utilizan con diferentes fines:

- Inicializar tripletas RDF en el dominio específico que no se han contemplado en el diseño general.

  En el dominio del tráfico, las reglas de inferencia permiten incluir en el modelo, por ejemplo, una tripleta para el objeto que aparece en escena

indicando que la probabilidad de que sea vehículo o peatón es 0. Un ejemplo de este tipo de reglas es el mostrado en la Figura 6.11. En él, se señala que, si un objeto con identificador ?o, variable, de tipo *OBJECT* no posee todavía la propiedad *probabilityVehicle*, se incluye en la ontología una tripleta en la que se le asigna a este objeto la propiedad con valor 0.

Esta tripleta no se puede incluir en el modelo genérico ya que el número de clases y las etiquetas de las mismas, que identifican los objetos, no están definidas a priori (no se conocería el nombre de las propiedades). En el modelo específico tampoco tiene cabida ya que no se conocen previamente los identificadores de los objetos que van apareciendo en la escena.

- Actualizar valores de probabilidades cuando se verifiquen ciertas condiciones en los objetos para determinar la clase a la que pertenecen.

  No se conoce la clase de un nuevo objeto en movimiento que asoma en la escena. Para poder clasificarlo, a cada nuevo individuo se le asignan propiedades con probabilidades de que pertenezca a una clase u otra. En el dominio del tráfico, por ejemplo, como se indicó en la Sección 6.6.1.1.3 Modelado semántico específico: Aplicación al control de tráfico, a un *Objeto1* se le asocian como propiedades la probabilidad de que sea vehículo y la de que sea peatón.

  Para la actualización de estos valores es necesario incluir reglas que los modifiquen con cada fotograma. Las premisas jugarán con los valores de los parámetros de los objetos en movimiento, como la velocidad o dimensiones del objeto en el fotograma actual, para actualizar una u otra propiedad. En el dominio del tráfico, por ejemplo, si el *Objeto1* tiene en el fotograma actual velocidad 2 km/h y ancho 0,60 m, inferiores a los umbrales establecidos para vehículos (10 km/h y 1,50 m), la regla correspondiente se activa al cumplirse esas premisas y, como consecuencia, la propiedad *probabilityPedestrian* de *Objeto1* se incrementará en 1 (realmente se borra la tripleta previa y se agrega una nueva con el valor actualizado). No obstante, la regla que actualiza la propiedad *probabilityVehicle* no se ejecutará con lo que el valor de la misma se mantiene.

- Establecer la clase en la que incluir un objeto cuando una probabilidad supere un umbral.

  Como se ha indicado, la evolución de los parámetros de los objetos hace que las probabilidades de que pertenezcan a una clase u otra se vayan actualizando para que el objeto se pueda clasificar adecuadamente cuando, tras varios fotogramas, se determine que la posibilidad de que sea un Vehículo o Peatón, si se habla del dominio del control de tráfico, supera un umbral o es muy superior una con respecto a la otra.

  Por tanto, va a ser necesaria una regla de inferencia que evalúe las probabilidades para que, en caso de que se cumplan las premisas establecidas, se asigne el objeto a una determinada clase. Así pues, si la propiedad *probabilityPedestrian* de *Objeto1* es 21 y la *probabilityVehicle* es 5, y las premisas establecen que si la *probabilityPedestrian* es mayor de 17 y la diferencia de probabilidades es mayor de 15 se llega a la conclusión de que *Objeto1* es un peatón, incluyendo además en la ontología la tripleta *Objeto1 hasType Pedestrian.*

  Para evitar incongruencias en los datos, cuando un objeto se incluye dentro de una clase, este objeto no se vuelve a reclasificar. Es decir, si se determina que un individual es un vehículo esta asignación es definitiva y aunque sus propiedades cambien, no se va a poder a posteriori decir que es un peatón. Los umbrales establecidos en las reglas deben estar estudiados para que cuando se identifique un objeto como perteneciente a una determinada clase sea veraz.

- Recalcular valores de cada clase de objeto que han transitado las distintas localizaciones.

  Para determinar si una ruta detectada pertenece a una clase u otra y así etiquetarla va a ser necesario, además de conocer características de la misma como su anchura, saber la clase de objetos que por ella transitan. En el dominio del control de tráfico, una ruta concreta es identificada como acera, si por ella circulan sobre todo peatones, por ejemplo. A cada ruta descubierta durante el proceso de modelado espacial, se le asignan propiedades con el número de objetos de cada posible clase determinada por el dominio de aplicación. En el campo del control de tráfico, por una ruta pueden circular peatones o vehículos con lo que cada ruta dispone de las propiedades *hasNumberVehicle* y *hasNumberPedestrian* que inicialmente

tienen valor 0. Cuando un nuevo objeto aparece en la escena y se identifica la clase a la que pertenece, estos valores se actualizan para cada una de las rutas por las que pasa.

- Identificar la clase de ruta que es una zona pudiendo etiquetarla así para un determinado dominio.

  Al igual que sucede en el caso de los objetos, cuando por una ruta concreta han pasado un número de objetos de una clase o la diferencia entre los que han circulado de los distintos tipos supera un umbral, se determina la clase a la que pertenece la ruta.

  En el ámbito del control de tráfico, las localizaciones definidas en la ontología específica pueden pertenecer a las clases *Sidewalk*, *Crosswalk* o *Road* y son los valores de las propiedades *hasNumberVehicle* y *hasNumberPedestrian* de la ruta las que activan una regla u otra, asignando a la localización la etiqueta que corresponda.

- Borrar tripletas de las ontologías cuando sea necesario eliminarlas del modelo.

  Si bajo ciertas condiciones se decide que el valor asignado a una propiedad es necesario eliminarlo porque es una afirmación que ya no es válida, se pueden incluir reglas que indiquen al sistema la necesidad de eliminar del modelo persistente una determinada tripleta RDF. Este es el caso, por ejemplo, de la actualización de valores a determinadas propiedades durante el transcurso de los diferentes fotogramas.

  Para realizar este borrado se pueden definir esquemas de tripletas que digan al sistema que realice el borrado. Por ejemplo, si en un momento puntual el sistema de reglas genera a modo de conclusión las siguientes tripletas:

  Objeto1 removeValue 6

  Objeto1 probabilityPedestrian removeProperty

  El sistema determina que tiene que borrar del modelo persistente la tripleta:

  Objeto1 probabilityPedestrian 6

  Una única tripleta no es suficiente para que se entienda que hay que realizar el borrado ya que no incluye palabras clave que se lo indiquen. Si estas palabras clave sólo están presentes en el sujeto, predicado u objeto de una

única tripleta, la respuesta podría no ser la esperada. Por ejemplo, si sólo se incluye la tripleta *Objeto1 removeValue 6* entendiendo como clave en este caso el predicado de la tripleta, se pueden eliminar todas las tripletas cuyo objeto sea *6* para el sujeto *Objeto1* independiente del predicado de las mismas. La inclusión de una tripleta adicional que especifique el predicado que se desea borrar asegura que la respuesta sea la esperada.

- Especificar si en la escena se está produciendo una situación de alarma asignando al objeto concreto, en lenguaje natural, el problema detectado.

  Una vez caracterizada la escena, si se cumplen ciertas condiciones predefinidas como no habituales en el dominio de estudio, se genera una tripleta RDF que asigna a un objeto una alarma.

  Este tipo de regla es, en el campo del control de tráfico, la que concluye que *Objeto1* tiene la propiedad *hasAlarm* con valor *pedestrianCrossingInappropriately* cuando *Objeto1* de tipo peatón está localizado en una ruta *Ruta1* que no es de la clase *Sidewalk* ni *Crosswalk*. La regla de inferencia que se ejecuta en ese caso es similar a la incluida en la Figura 6.12.

```
[r16:
  (?o rdf:type wrongPlace:OBJECT)
  (?o wrongPlace:hasType wrongPlace:Pedestrian)
  (?r rdf:type wrongPlace:Path)
  (?o wrongPlace:hasLocated ?r)
  noValue(?r wrongPlace:hasType wrongPlace:Crosswalk)
  noValue(?r wrongPlace:hasType wrongPlace:Sidewalk)
  ->(?o wrongPlace:hasAlarm wrongPlace:pedestrianCrossingInappropriately)]
```

Figura 6.12. Ejemplo de regla de inferencia para la detección de situaciones anómalas.

Los propósitos indicados sólo son algunas de las posibilidades que ofrecen las reglas de inferencia, que pueden aumentar su complejidad o completarse con otras para conseguir resultados adicionales.

Por otro lado, en todos los casos las reglas de inferencia son específicas del dominio de conocimiento y, para mejorar la modularidad del mecanismo propuesto y así facilitar su adecuación a un nuevo campo de aplicación, se incluyen en un fichero de texto plano.

6.6.2 Funcionamiento del sistema

En la introducción de la Sección 6.6 se incluye de forma general el funcionamiento del mecanismo del modelado semántico de espacios. En este apartado se va a detallar a bajo nivel como, partiendo de la información detectada durante el proceso de modelado espacial y la información de los elementos en movimiento que se obtienen fotograma a fotograma en tiempo real, se pueden etiquetar semánticamente los diferentes objetos de la escena.

La explicación del proceso es genérica pero, para facilitar el seguimiento del proceso, se utiliza el ejemplo utilizado hasta ahora, la aplicación al dominio del control de tráfico.

En primer lugar se crea una base de datos relacional donde debe estar almacenada la ontología persistente. Para la generación de la misma se utiliza el modelo diseñado para el ámbito de aplicación y que es un esquema que poco a poco se va populando con los distintos resultados obtenidos de los procesos de inferencia.

En la Figura 6.13 se muestra el esquema de la ontología persistente utilizado para este dominio de aplicación. Para la creación de la misma se ha utilizado el software de código libre Protégé (http://protege.stanford.edu/), uno de los editores de ontologías más usados.

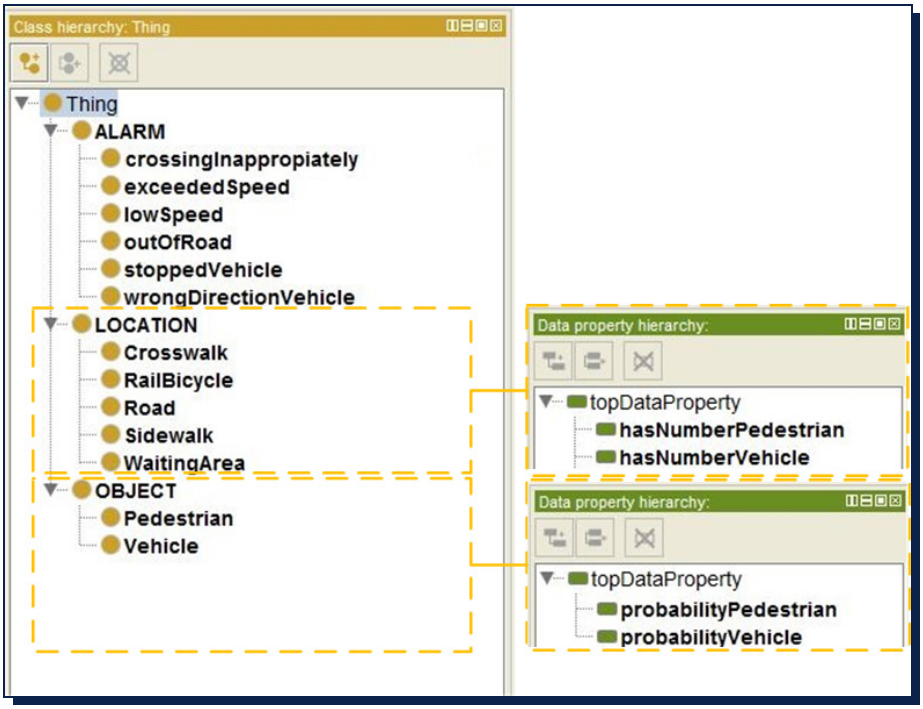


Figura 6.13. Esquema de la ontología persistente para el dominio del control de tráfico.

En ejecuciones posteriores, la ontología y los individuales almacenados en la misma se cargan en un modelo que sirve como base en el proceso de inferencia. Este modelo incluye la información necesaria para el modelado de la escena que se lleva a cabo durante el proceso de aprendizaje y la caracterización de la escena cuando comienza la fase de operación.

A continuación se carga como modelo la ontología que define los objetos en movimiento. En la Figura 6.14 se pueden observar las propiedades de los mismos.

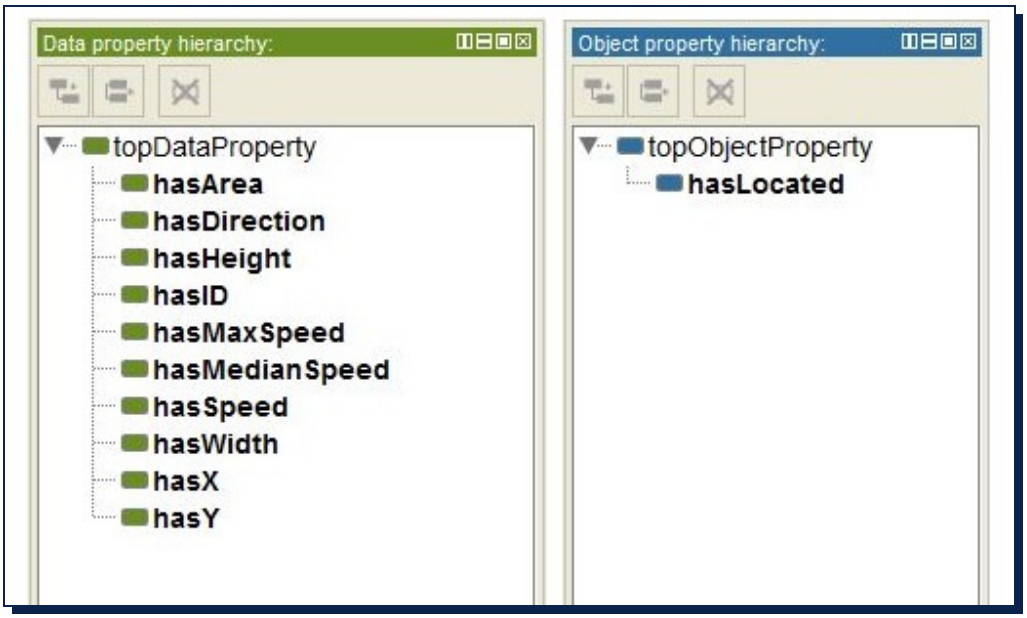


Figura 6.14. Propiedades de los objetos de la clase *OBJECT*.

En este esquema se crean y generan instancias con los individuales que se detectan en el fotograma actual, incluyendo las propiedades de los mismos. Esta ontología es volátil y cuando finaliza el proceso de inferencia del fotograma la información de todos los individuales del momento actual se pierde.

En la Figura 6.15 se muestra la definición de *Objeto1*, un ejemplo de individual de la clase *OBJECT*. En ella se ven los valores de las distintas propiedades incluidas en la ontología.

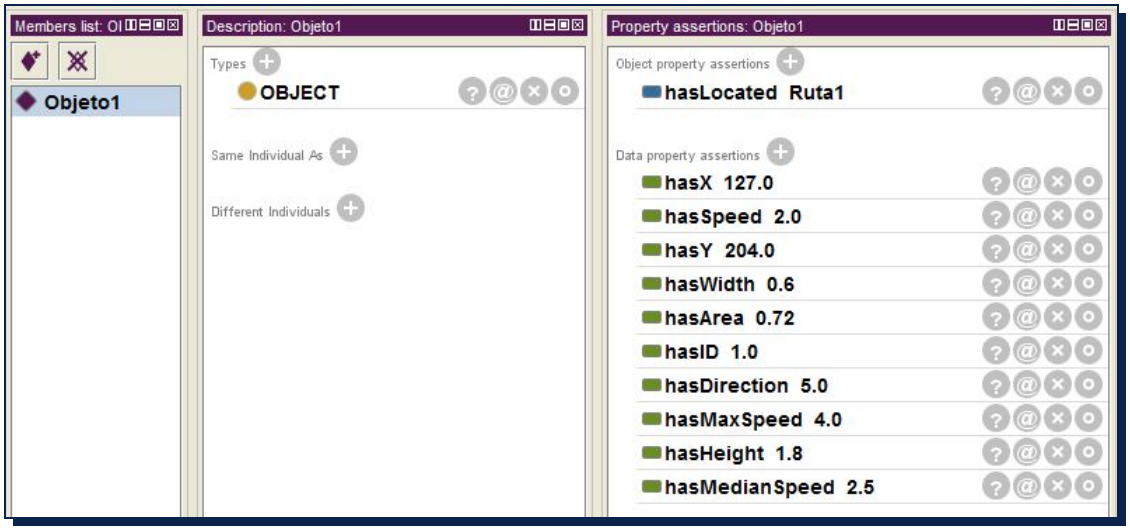


Figura 6.15. Instancia de la clase *OBJECT*.

En la Figura 6.16 y Figura 6.17 se muestran las propiedades de los objetos de la clase *LOCATION* y la definición de *Ruta1*, un ejemplo de individual de tipo *Path* perteneciente a esta clase. Se pueden ver además, en la Figura 6.17, los valores de las distintas propiedades incluidas en la ontología. En este caso *Ruta1* tiene una longitud (*hasStrengh*) de 10 puntos. Para cada uno de ellos, la ruta tiene un punto central (*hasXPoints* - *hasYPoints*), dos puntos definiendo la envolvente superior e inferior (*hasEnv1X* - *hasEnv1Y*, *hasEnv2X* - *hasEnv2Y*) y una dirección (*hasDirection*) siguiendo la definición que se presentó en la Sección 5.3.1 Modelado de rutas.

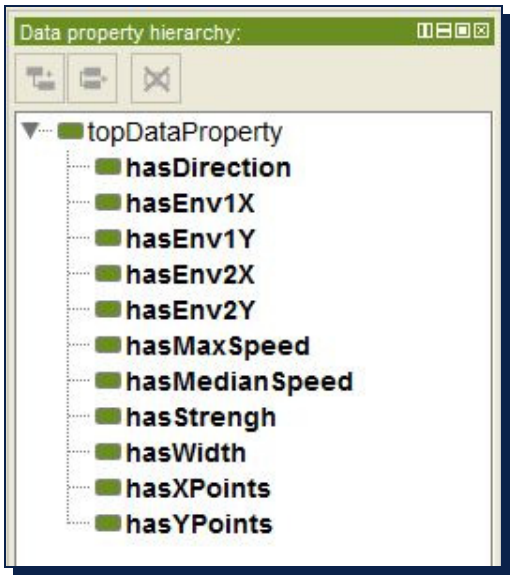


Figura 6.16. Propiedades de los objetos de la clase *LOCATION*.

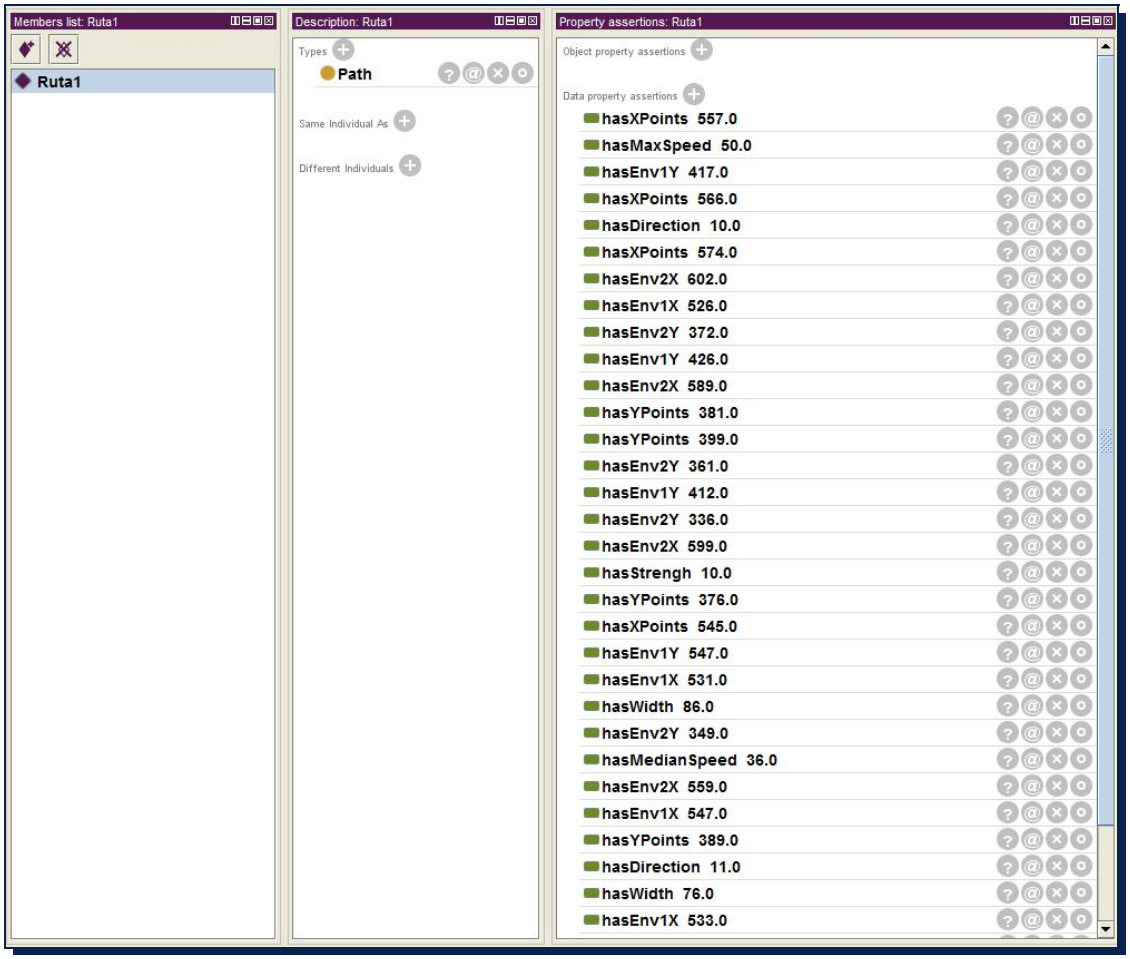


Figura 6.17. Individual de la clase *LOCATION*.

El siguiente paso es posicionar los objetos detectados para el fotograma actual dentro de las rutas identificadas hasta el momento. En la Figura 6.15 se observa como entre las propiedades se incluye *hasLocated* que posiciona el objeto dentro de las rutas descubiertas.

Los tres esquemas con el conocimiento se importan dentro de un único modelo que, en combinación con el conjunto de reglas de inferencia es procesado por un razonador genérico basado en reglas. La información adicional generada en este proceso sirve como entrada para la actualización del modelo persistente y permite la detección de situaciones anómalas. En la Figura 6.18 se muestra el procedimiento.

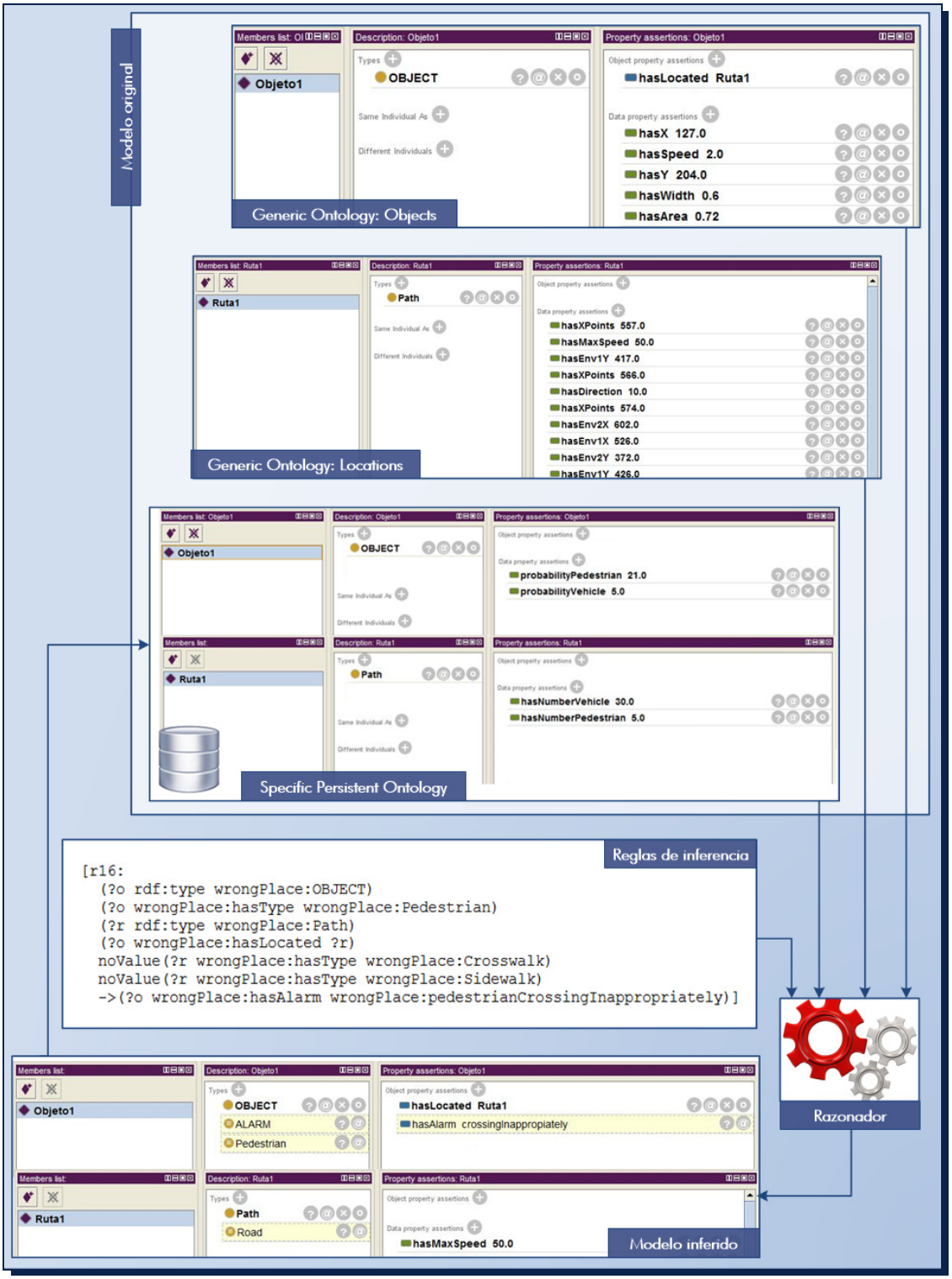


Figura 6.18. Esquema del proceso de integración de modelos, inferencia y actualización de la ontología persistente.

En la Figura 6.19 se indica observar como, a partir de la información del individual en el momento actual y las propiedades en fotogramas anteriores del *Objeto1*

(*probabilityPedestrian* y *probabilityVehicle*) inferidas previamente y almacenadas en la ontología persistente, el razonador etiqueta el objeto como *Pedestrian*.

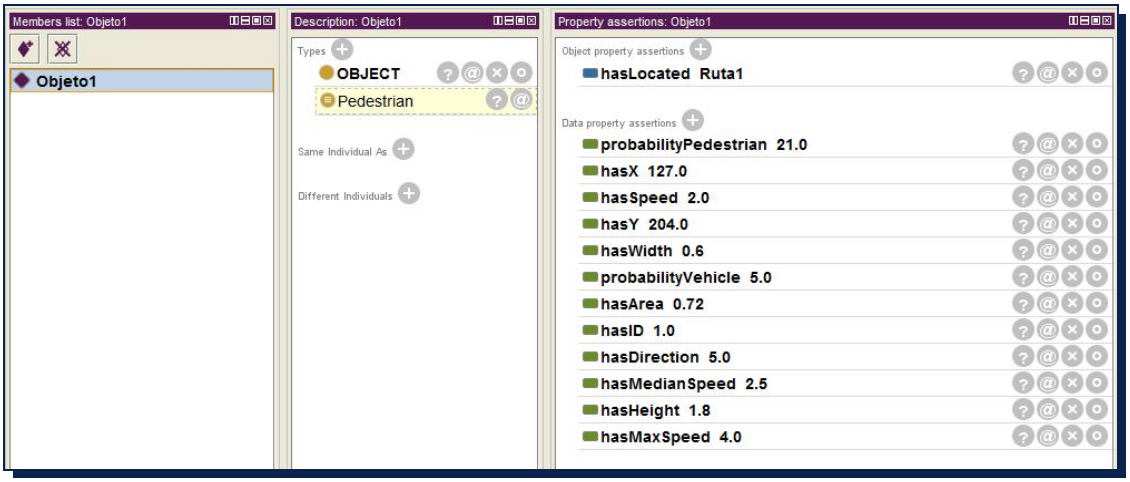


Figura 6.19. Identificación de un objeto perteneciente a la clase *Pedestrian*.

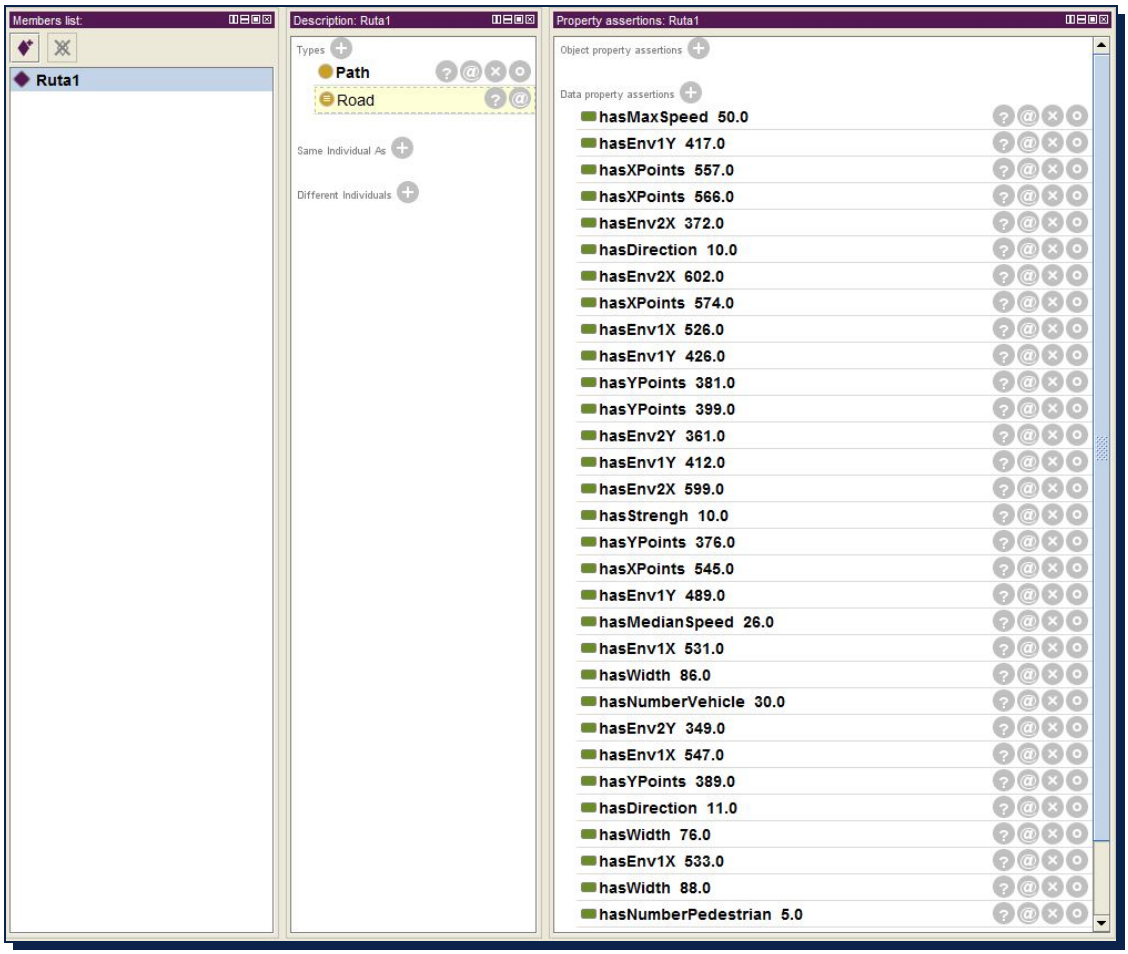


Figura 6.20. Identificación de una ruta perteneciente a la clase *Road*.

En la Figura 6.20 se puede ver como, con las propiedades actuales de la ruta *Ruta1* y las inferidas previamente (*hasNumberVehicle* y *hasNumberPedestrian,* el razonador etiqueta el objeto como *Road*.

Dado que el individual *Objeto1* es de tipo *Pedestrian* y está ubicado dentro de *Ruta1* de tipo *Road*, las reglas de inferencia diseñadas detectan una situación anómala y la definen haciendo que se incluya el objeto dentro de la clase *ALARM* y se le asigne la propiedad *hasAlarm* con valor *pedestrianCrossingInappropiately* como se muestra en la Figura 6.21.

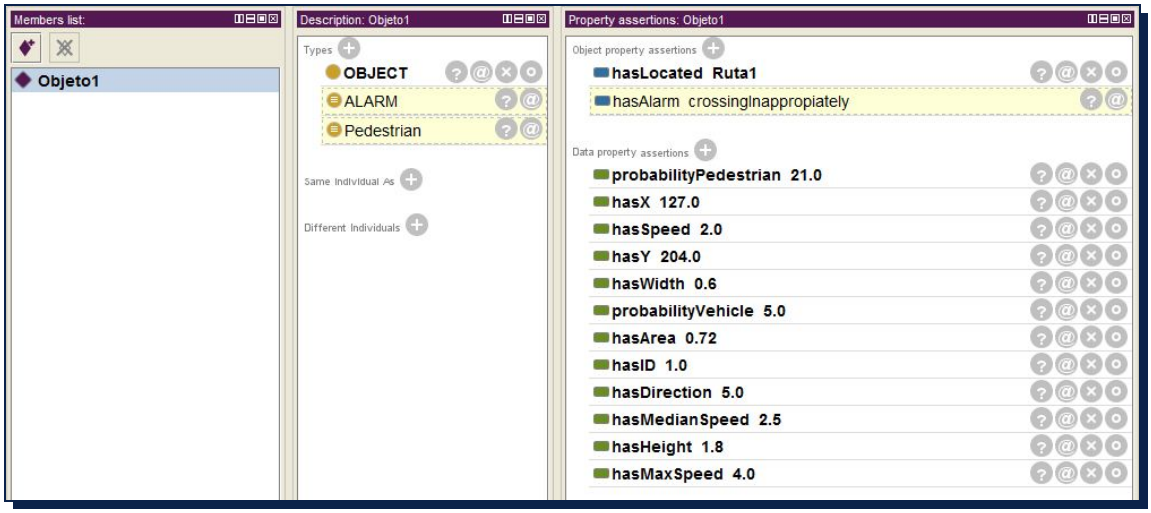


Figura 6.21. Alarma asignada a una instancia de objeto.

6.7 Validación

Protégé, la herramienta de edición de ontologías utilizada permite realizar la validación de la ontología utilizando los razonadores de los que dispone. La evaluación que realiza incluye tres aspectos:

- Detectar inconsistencias en la ontología para determinar si existen contradicciones dentro de la misma.
- Validar la taxonomía de las clases comprobando todas las relaciones entre las clases y determinando si el diseño jerárquico es coherente.
- Verificar los procesos de inferencia para encontrar la clase o clases a la que pertenece cada individual incluido en la ontología determinando si esta clasificación es la adecuada.

Utilizando el razonado, en caso de inconsistencias en la ontología se lanza un mensaje al usuario indicando la clase o clases en la que está el error marcándolas en rojo. En la Figura 6.22 se puede ver un ejemplo de funcionamiento.

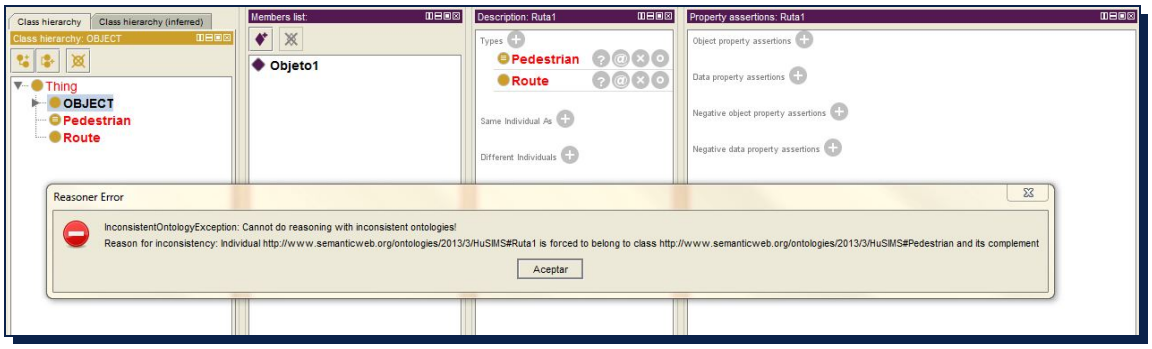


Figura 6.22. Ejemplo de alarma indicando inconsistencia en la ontología.

Para tener un mayor conocimiento de la inconsistencia, Protégé proporciona la posibilidad de acceder a su explicación (ver Figura 6.23).

Figura 6.23. Ejemplo de inconsistencia en la ontología.

En el caso de esta Tesis Doctoral, se han validado las tres ontologías por separado utilizando el razonador Pellet proporcionado por la herramienta ya que es el de uso más extendido.

Indicar que no ha detectado errores en ninguna de las ontologías evaluadas.

6.8 Conclusiones

El uso de ontologías para el modelado semántico y la caracterización de escenas proporciona una serie de ventajas que no se consiguen empleando otros mecanismos y tecnologías.

Por una parte, la facilidad de diseño de ontologías permite que la adaptación y cambio de dominio sea muy sencillo. La utilización de varias ontologías genéricas

combinadas con una ontología específica del campo de aplicación simplifica la tarea. El código que implementa el sistema permanece inalterable y la información del dominio se aporta de manera separada. Una modificación o ampliación de la ontología específica del ámbito de conocimiento, incluida en un fichero de texto, y el ajuste de las reglas de inferencia hacen que el sistema sea muy modular y fácilmente escalable.

La minuciosidad en la caracterización depende de la completitud y complejidad de la ontología diseñada. Indicar también que, varias ontologías se pueden fusionar fácilmente de manera que, se pueden contemplar varios dominios de aplicación si se dispone de ontologías particulares para cada uno de los campos específicos.

Además, la inclusión de nuevos datos de entrada, como la procedente de sensores, puede mejorar los resultados del proceso de inferencia y permite identificar nuevas situaciones que no podrían detectarse únicamente con la información visual de la escena. Para entender estas nuevas posibilidades sólo es necesario incluir en las reglas de inferencia los datos proporcionados por los mismos y las conclusiones a las que se llega cuando se alcanzan ciertos valores.

El uso de la semántica para la caracterización de escenas y su aplicación a la identificación de alarmas proporciona un modelo de conocimiento que utiliza conceptos del lenguaje formal. Precisamente debido a esto, cuando se dispara una alarma existe gran cantidad de información disponible relativa a la emergencia, en un formato de muy alto nivel directamente interpretable por una persona (colisión, atropello, disturbios, incendio, etc.), no un simple aviso de que ha ocurrido un suceso no habitual. Esta información adicional puede resultar muy ventajosa a la hora de ahorrar tiempo durante la gestión de la alerta, puesto que el operador humano sabe inicialmente a qué se enfrenta y se evita la inspección inicial de los archivos de video para identificar la situación. Incluso más aún, es posible automatizar en cierta medida las reacciones a diferentes tipos de emergencias, distribuyendo de forma inteligente las mismas a los operadores implicados.

7

INTEGRACIÓN Y PRUEBAS DEL SISTEMA

Con el objetivo de verificar la validez del sistema propuesto se implementa un prototipo integrado del mismo para su aplicación en el dominio del control de tráfico (que sirve de ejemplo a lo largo del desarrollo de los Capítulos precedentes de esta Tesis). Las pruebas para la evaluación del sistema se ejecutan sobre videos sintéticos y reales, que proporcionan los mismos resultados que la ejecución con una cámara situada en el escenario real.

Este Capítulo se encuentra estructurado de la siguiente manera. En la Sección 7.1 se recoge el proceso de integración del sistema completo incluyendo las tecnologías utilizadas para la implementación de los diferentes módulos y su funcionamiento de manera coordinada. La Sección 7.2 define una serie de casos de uso en los que se analiza la arquitectura global para determinar la validez de la propuesta, recogiendo los resultados de las pruebas de esfuerzo como de precisión en la caracterización de la escena en la Sección 7.3. La Sección 7.4 describe la

aplicación del sistema a otros dominios distintos al utilizado hasta ahora como ejemplo. Finalmente, en la Sección 7.5 se exponen las principales conclusiones.

7.1 Integración

En las secciones 4, 5 y 6 se describe el diseño propuesto para la caracterización semántica de escenarios. La arquitectura es modular, de manera que cada bloque tiene su función particular para así comprender mejor el cometido de los mismos, aportar escalabilidad al sistema y facilitar su adaptabilidad a diferentes campos de aplicación.

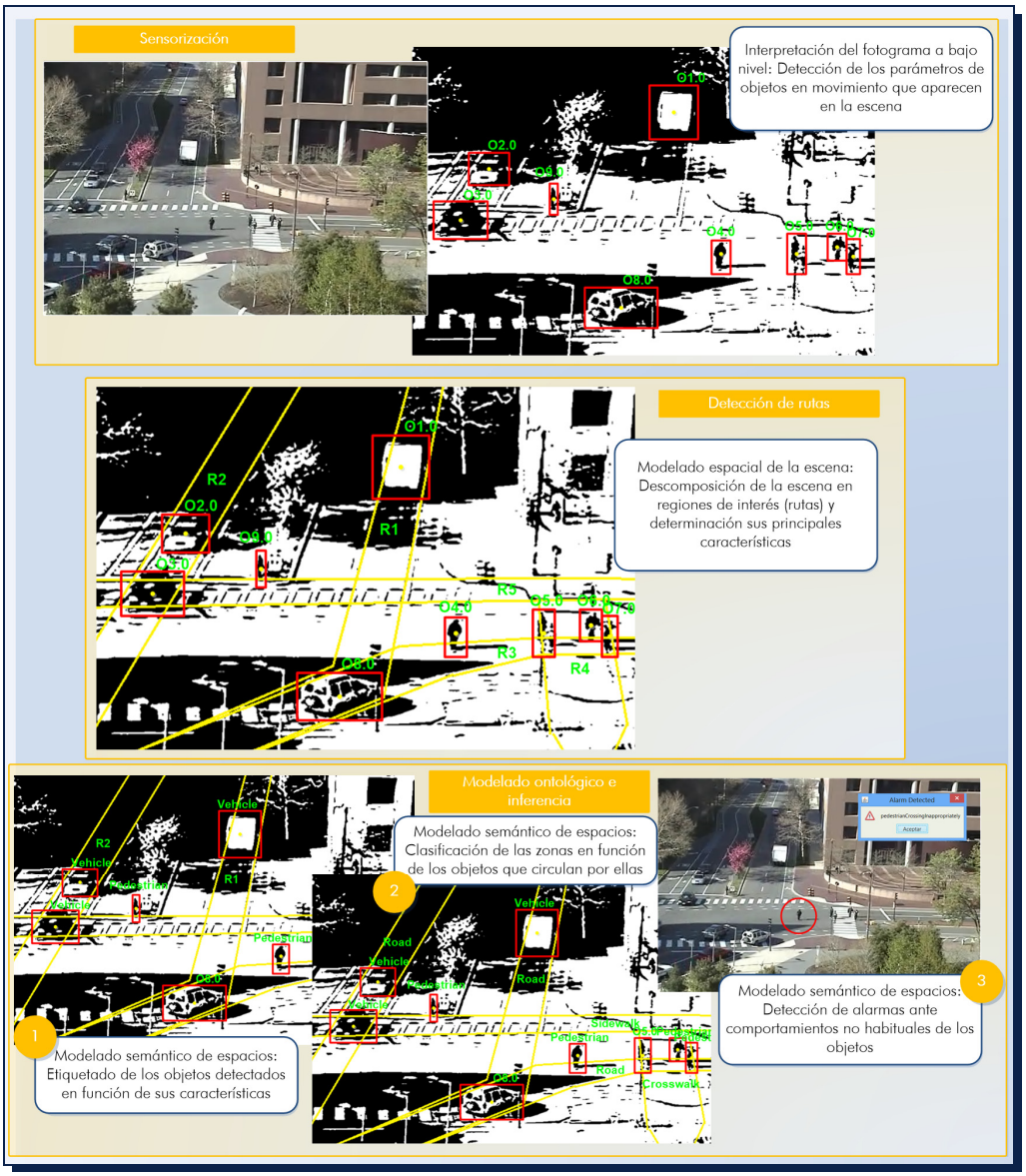


Figura 7.1. Funcionamiento del sistema integrado.

En esta sección se especifica cómo se ha llevado a cabo el proceso de integración completo justificando las tecnologías utilizas para la implementación de los diferentes bloques y la definición los mecanismos de comunicación entre los mismos. La Figura 7.1 resume el resultado de la ejecución de cada módulo individual para entender el funcionamiento del sistema integrado.

7.1.1 Sensorización

El primer módulo es el encargado de la Sensorización. El video captado por el propio sensor inteligente es procesado por él mismo, proporcionando información acerca de los objetos de movimiento que aparecen en la escena. En la Figura 7.2 se pueden ver los diferentes subbloques de los que está compuesto y la información obtenida en cada uno de ellos.

Para cada fotograma, se envía un archivo XML, a través de la red, con las características de todos los objetos detectados en él. De cada elemento se incluye su anchura, altura, posición (x, y), área, velocidad, dirección del movimiento, etc. Así mismo, cada objeto tiene un identificador único para permitir su seguimiento a lo largo de la escena.

Además de los parámetros de movimiento obtenidos por la cámara, la arquitectura admite información procedente de cualquier tipo de sensor distribuido por la escena que pueda complementar la información visual. El procesado de estos datos y la adaptación y transmisión con el formato definido para esta arquitectura requiere módulos hardware y software adicionales específicos para cada sensor y que escapan de los objetivos de esta Tesis Doctoral.

En la Figura 7.3 aparece un ejemplo sencillo de XML para una escena en la que solo hay un objeto en movimiento y un despliegue de dos sensores en la zona. Las etiquetas <Objects> y </Objects> delimitan la información aportada por el sensor visual y entre <Sensors> y </Sensors> la procedente de otros sensores. Dentro de cada par de etiquetas <Object> y </Object> se incluyen los parámetros del objeto detectado, de manera que, por cada elemento en movimiento descubierto, aparecerá un par de estas marcas con su propia información. En el caso del resto de sensores el proceso es similar salvo que, en este caso, la información proporcionada por cada sensor aparece entre las etiquetas <Sensor>y </Sensor>. Además, las etiquetas <ID></ID> y <SensorID></SensorID> delimitan los identificadores que distinguen cada objeto y sensor respectivamente de manera única.

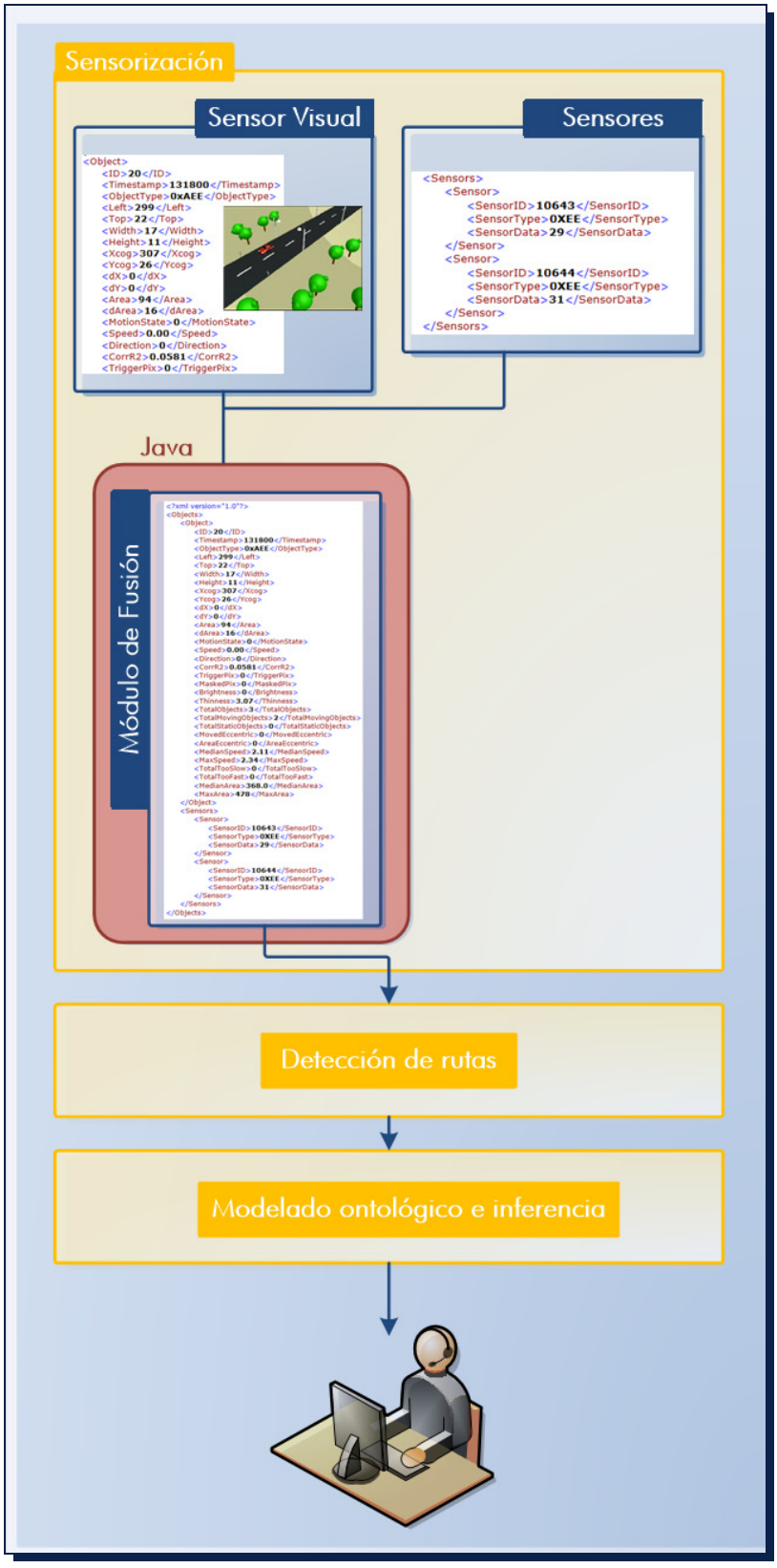
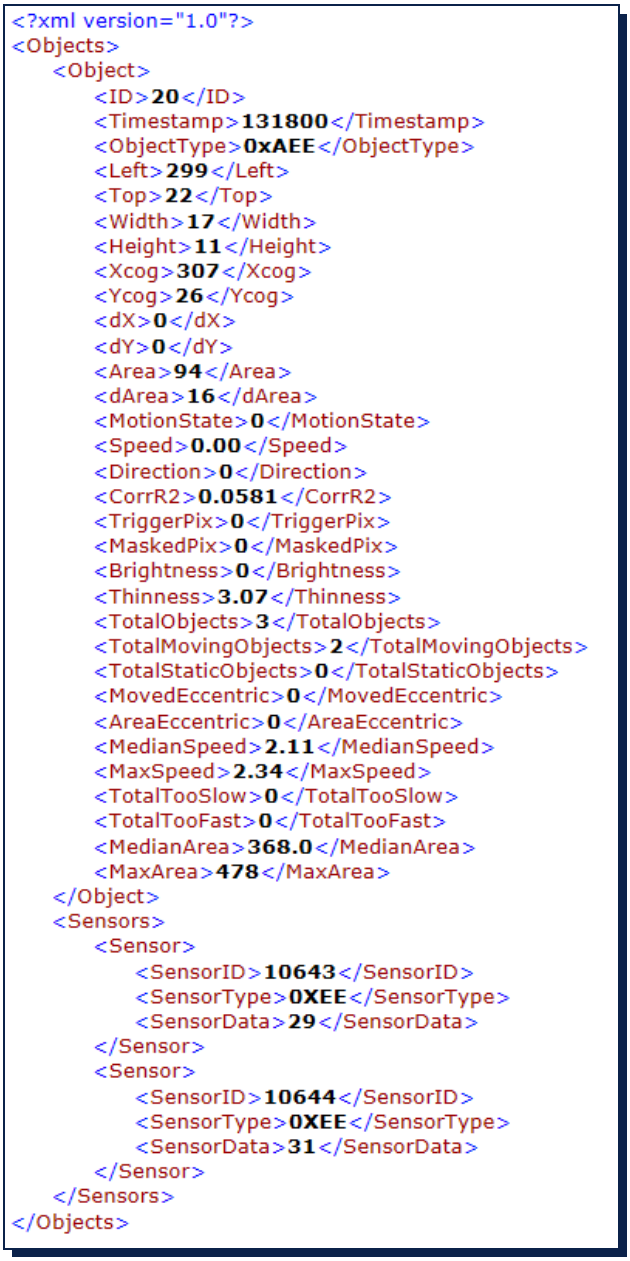


Figura 7.2. Implementación e integración del módulo de Sensorización.



```
<?xml version="1.0"?>
<Objects>
    <Object>
        <ID>20</ID>
        <Timestamp>131800</Timestamp>
        <ObjectType>0xAEE</ObjectType>
        <Left>299</Left>
        <Top>22</Top>
        <Width>17</Width>
        <Height>11</Height>
        <Xcog>307</Xcog>
        <Ycog>26</Ycog>
        <dX>0</dX>
        <dY>0</dY>
        <Area>94</Area>
        <dArea>16</dArea>
        <MotionState>0</MotionState>
        <Speed>0.00</Speed>
        <Direction>0</Direction>
        <CorrR2>0.0581</CorrR2>
        <TriggerPix>0</TriggerPix>
        <MaskedPix>0</MaskedPix>
        <Brightness>0</Brightness>
        <Thinness>3.07</Thinness>
        <TotalObjects>3</TotalObjects>
        <TotalMovingObjects>2</TotalMovingObjects>
        <TotalStaticObjects>0</TotalStaticObjects>
        <MovedEccentric>0</MovedEccentric>
        <AreaEccentric>0</AreaEccentric>
        <MedianSpeed>2.11</MedianSpeed>
        <MaxSpeed>2.34</MaxSpeed>
        <TotalTooSlow>0</TotalTooSlow>
        <TotalTooFast>0</TotalTooFast>
        <MedianArea>368.0</MedianArea>
        <MaxArea>478</MaxArea>
    </Object>
    <Sensors>
        <Sensor>
            <SensorID>10643</SensorID>
            <SensorType>0XEE</SensorType>
            <SensorData>29</SensorData>
        </Sensor>
        <Sensor>
            <SensorID>10644</SensorID>
            <SensorType>0XEE</SensorType>
            <SensorData>31</SensorData>
        </Sensor>
    </Sensors>
</Objects>
```

Figura 7.3. Ejemplo de fichero XML que el módulo de Sensorización envía al Detector de Rutas.

Una vez en el centro de control, se realiza el procesado de la información obtenida de todas las fuentes disponibles para ese fotograma y con ella se genera un fichero XML que unifica los formatos de las diferentes fuentes. Este procesado acontece en el módulo de fusionado, que incluye un pequeño programa Java, lenguaje utilizado para la implementación de los demás módulos.

Se ha seleccionado XML, como tecnología para la transferencia de datos, porque es un estándar que se ha adoptado para manejar la gran cantidad de información

disponible en la *Web* y es muy común su aplicación a distintos ámbitos ya que permite unificar formatos para fuentes diferentes [201].

Los informes no se envían de forma continua sino sólo cuando se dispone de información significativa. La tasa con la que se procesa la imagen es configurable aunque habitualmente se utilizan velocidades de fotogramas de 15 fps.

7.1.2 Detección de Rutas

El bloque que lleva a cabo la detección de rutas recibe del módulo de fusión el fichero XML con la información procedente del sensor visual inteligente y los sensores de la escena (si los hubiera) y genera un modelo espacial descubriendo las diferentes zonas existentes en la misma y los comportamientos de los objetos que por ella transcurren. En la Figura 7.4 se puede ver la implementación realizada para conseguir este objetivo.

Dentro de este módulo, el Preprocesador de *Frames* analiza cada fotograma que contiene el XML mediante un código implementado en Java, obtiene la información incluida en el fichero y genera clases Java con la misma. Se ha utilizado esta tecnología por su flexibilidad y su facilidad de uso para la realización es esta tarea. Java proporciona varias APIs (*Application Programming Interface*) para realizar este procedimiento:

- La API para DOM (*Document Object Model* - http://www.w3.org/DOM/): En DOM el acceso al documento XML se representa como un árbol de nodos jerárquico cargado en memoria. Se puede leer cualquier parte del mismo ya que, el procesando se realiza de arriba abajo, se puede volver hacia atrás. Además, soporta la realización de modificaciones sobre los nodos.
- SAX (*Simple API for XML* - http://www.saxproject.org/) es una API de Java que permite la lectura secuencial de documentos XML. En este caso la API funciona con eventos (SAX los va lanzando según recorre el documento y detecta las distintas etiqueta)s. El fichero se va leyendo de manera secuencial y no permite volver atrás.
- Stax (*Streaming API for XML* - http://stax.codehaus.org/) es también una API que admite tanto lectura como escritura de documentos XML. En este caso el acceso al documento es aleatorio y no necesita requisitos especiales de memoria. La principal diferencia con SAX es que, en vez de ir enviando eventos según encuentra etiquetas, en esta aproximación es el usuario el que

tiene el control de la aplicación seleccionando los datos de las mismas que le van interesando.

- JAXB (*Java Architecture for XML Binding* - http://www.oracle.com/technetwork/articles/javase/index-140168.html) es un estándar Java que define cómo convertir los objetos en ficheros XML y viceversa. JAXB define una API completa de lectura y escritura muy útil cuando el XML tiene una estructura fija.

En este caso se hace uso de Stax porque DOM y SAX son API's antiguas y para utilizar JAXB se necesita generar un XSD (*XML Schema Definition* - http://www.w3.org/TR/2012/REC-xmlschema11-1-20120405/), esquema del XML a procesar, con lo que, si se realiza una modificación del mismo agregando una nueva etiqueta, hay que realizar un nuevo XSD que la contemple, con el consiguiente tiempo de desarrollo y la posibilidad de cometer errores, algo habitual en esta tarea. Los objetos Java formados se envían al Algoritmo de Detección de Rutas donde se les aplican algoritmos para la detección de regiones, ampliamente descritos en la Sección 5 Modelado Espacial de la Escena: Detección de Rutas. Se utiliza MATLAB para el procesamiento, en vez de realizar la implementación en Java, por su potencia de cálculo para trabajar con gran cantidad de datos en tiempo real. Para poder llevar a cabo este intercambio de información entre Java y MATLAB se emplea *MATLAB Builder JA* (http://www.mathworks.es/products/javabuilder/), que permite crear clases Java desde MATLAB. Estas clases se integran en los programas Java y se pueden desplegar en cualquier ordenador sin necesidad de tener MATLAB instalado, simplemente utilizando el *MATLAB Compiler Runtime* (MCR). Sólo se ejecutan estos algoritmos cuando el sistema está en modo aprendizaje identificando las distintas zonas que aparecen en la imagen y determinando los valores correctos o normales de los objetos que transcurren por la escena.

7.1.3 Modelado ontológico e inferencia

El modelo que se va obteniendo, junto con los datos de los objetos detectados en cada momento por la cámara y la información procedente de los sensores, se integra en el módulo de modelado ontológico e inferencia. Este bloque funciona tanto en modo aprendizaje como en operación. En modo aprendizaje el sistema va a ir poco a poco etiquetando las diferentes zonas que se van descubriendo y los objetos que van apareciendo. En modo operación, cuando ya se han detectado y

clasificado las distintas zonas de la escena, se van identificando los nuevos objetos que aparecen y se señalan las situaciones de alerta.

En la Figura 7.5 se muestra la estructura interna del módulo de Modelado Ontológico e Inferencia.

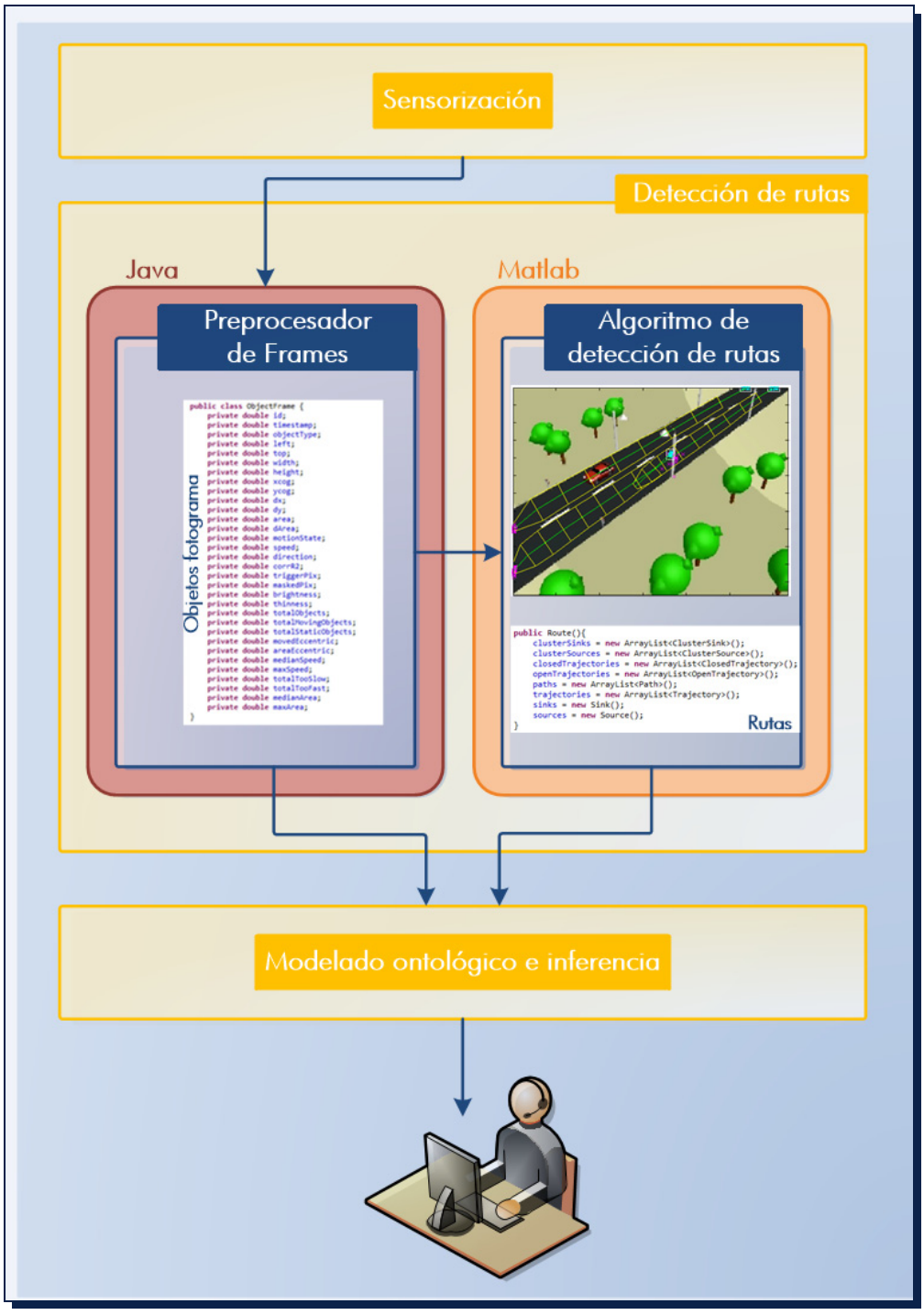


Figura 7.4. Implementación e integración del módulo de Detección de Rutas.

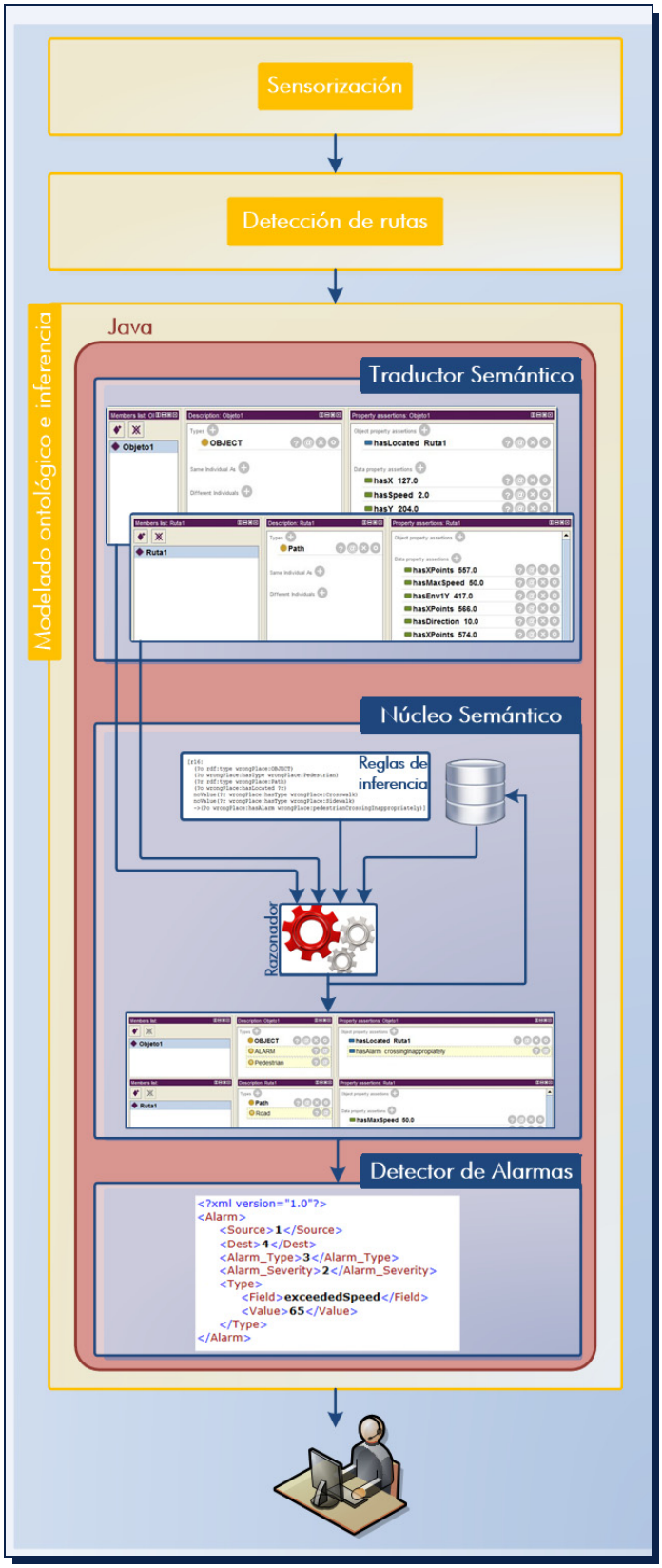


Figura 7.5. Implementación e integración del módulo que realiza el modelado ontológico.

Para realizar todo el proceso se utiliza el *framework* Jena que maneja todas las operaciones semánticas realizadas en Java.

Jena es un marco de código abierto para la *Web* Semántica escrito en Java basado en las recomendaciones de W3C para RDF y OWL. Proporciona distintas APIs que permiten el procesamiento de documentos escritos en ambos lenguajes Además admite el uso de distintos tipos de razonadores así como la realización del proceso de inferencia ontológico. Dentro de estas posibilidades se incluye un razonador genérico basado en reglas que posibilita el razonado cuando se incluyen reglas como complemento al modelo ontológico. Jena facilita también soporte para la consulta de modelos RDF mediante RDQL y SPARQL.

Así mismo, las ontologías diseñadas están escritas en lenguaje OWL, lenguaje estándar de marcado para publicar y compartir datos usando ontologías en la *Web* Semántica, utilizando la herramienta de código libre Protégé.

La información extraída del fotograma actual y las diferentes rutas identificadas hasta el momento se envían al bloque denominado Traductor Semántico en forma de objetos Java. Como su nombre indica, este bloque traduce la información sintáctica en semántica y popula las dos ontologías genéricas con los distintos individuales detectados. En el núcleo semántico, se importan dentro de un único modelo, las ontologías genéricas ya instanciadas y la ontología persistente, que incluye la información relevante obtenida en procesos de inferencia previos. Esta última se encuentra almacenada en una base de datos relacional, en este caso la seleccionada es MySQL, de forma totalmente transparente para el usuario. Sobre el nuevo modelo integrado y las reglas, diseñadas siguiendo las indicaciones de Jena, se realiza el proceso de inferencia utilizando el razonador genérico basado en reglas proporcionado por Jena.

En este proceso se realiza la clasificación de los distintos objetos y zonas. Este nuevo conocimiento servirá como realimentación en la ontología persistente para futuros razonamientos. Además, identifica si se está produciendo una alarma, concretando al bloque Detector de Alarmas la información que posee de la misma (el tipo de evento, localización de la incidencia, etc.), para generar un fichero XML (Figura 7.6) que se envía a los servicios de emergencia o al centro de control según se haya definido para esa alerta concreta.

Todo este proceso está ampliamente documentado en la Sección 6.6.2 Funcionamiento del sistema.

```
<?xml version="1.0"?>
<Alarm>
    <Source>1</Source>
    <Dest>4</Dest>
    <Alarm_Type>3</Alarm_Type>
    <Alarm_Severity>2</Alarm_Severity>
    <Type>
        <Field>exceededSpeed</Field>
        <Value>65</Value>
    </Type>
</Alarm>
```

Figura 7.6. Ejemplo de XML con la información relativa a la alarma.

7.1.4 Interfaz gráfica

Aunque la imagen de la escena monitorizada no es transmitida y los sensores visuales sólo envían la información de movimiento, para facilitar la comprensión del funcionamiento general del sistema integrado se ha incluido una interfaz gráfica. En ella se muestra la evolución de la fase de aprendizaje, el proceso de detección y la generación de alarmas.

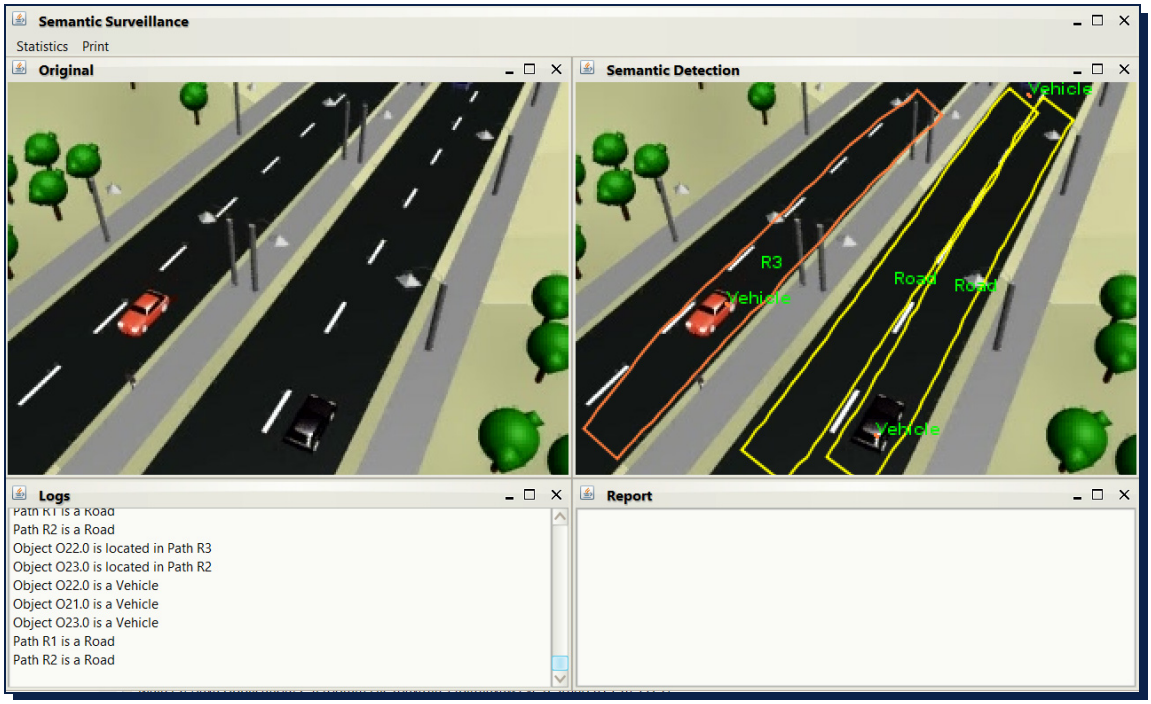


Figura 7.7. Interfaz gráfica del sistema: aprendizaje.

En la Figura 7.7 se observa un ejemplo de dicha interfaz. De forma general consta de:

- Menú de selección superior:

  El sistema está pensado para la identificación los objetos de los diferentes escenarios y su aplicación principal es determinar situaciones anómalas. Sin embargo, el conocimiento de la escena permite utilizar la identificación de los objetos de los videos para obtener estadísticas. Este menú de selección hace que el usuario pueda elegir los informes que desea, en el desarrollo actual, el número total de vehículos que han circulado por una vía y la velocidad media de los objetos que transitan por ella. Por una parte, aparece "S*tatistics"*, desplegable en el que se selecciona la información que se desea y que se mostrará en el área de texto "*Reports*" y por otra "*Print",* que imprime en PDF del informe solicitado.

- Pantalla con la imagen original (izquierda):

  Muestra la escena tal y como es captada por el sensor visual. En esta pantalla se rodean con una circunferencia roja los objetos que tienen un comportamiento anómalo. Además se incluye una ventana de alarma con la información de la misma.

- Pantalla con el estado del procesado (derecha):

  Expone la imagen y, superpuesto, de los objetos y las rutas que en ella aparecen. Los objetos, previo a su identificación incluyen una etiqueta con el texto "O+identificador del objeto". Cuando el proceso de inferencia determina el tipo de objeto, esta etiqueta se sustituye por la de la clase a la que pertenece. En el caso de la Figura 7.7 todos los individuales son vehículos y por tanto incluyen ese rotulo.

  Las rutas determinadas por el detector de rutas aparecen en naranja. Éstas incluyen inicialmente una etiqueta genérica, "R+identificador de ruta", en el caso de la Figura 7.7, R3, que las distingue. Cuando el modelado ontológico determina el tipo concreto de ruta (carretera, acera, etc.) cambia a color amarillo e incluye un rótulo que sustituye al primero. En el caso de la Figura 7.7, las rutas se establece que son carreteras y por tanto se etiquetan como *Road*.

- Área de Texto "Logs" (izquierda):

  Muestra la información de la escena. Ésta incluye:

  - Los objetos identificados y la clase a la que pertenecen. *Object O22.0 is a Vehicle*, por ejemplo.
  - Las rutas descubiertas y su etiqueta, como en *Path R1 is a Road.*
  - La localización de los objetos en las distintas rutas (*Object O22.0 is located in Path R3*).
  - Los mensajes de alarma especificando el momento y el tipo de situación (Ver "*Logs*" en Figura 7.8).

- Área de Texto "Reports" (derecha):

  Imprime las estadísticas que el usuario solicita en la pestaña "Statistics" del menú inicial.

La Figura 7.7 corresponde a la ejecución del sistema en modo de aprendizaje, ya que no se han determinado todas las rutas que aparecen en la imagen.

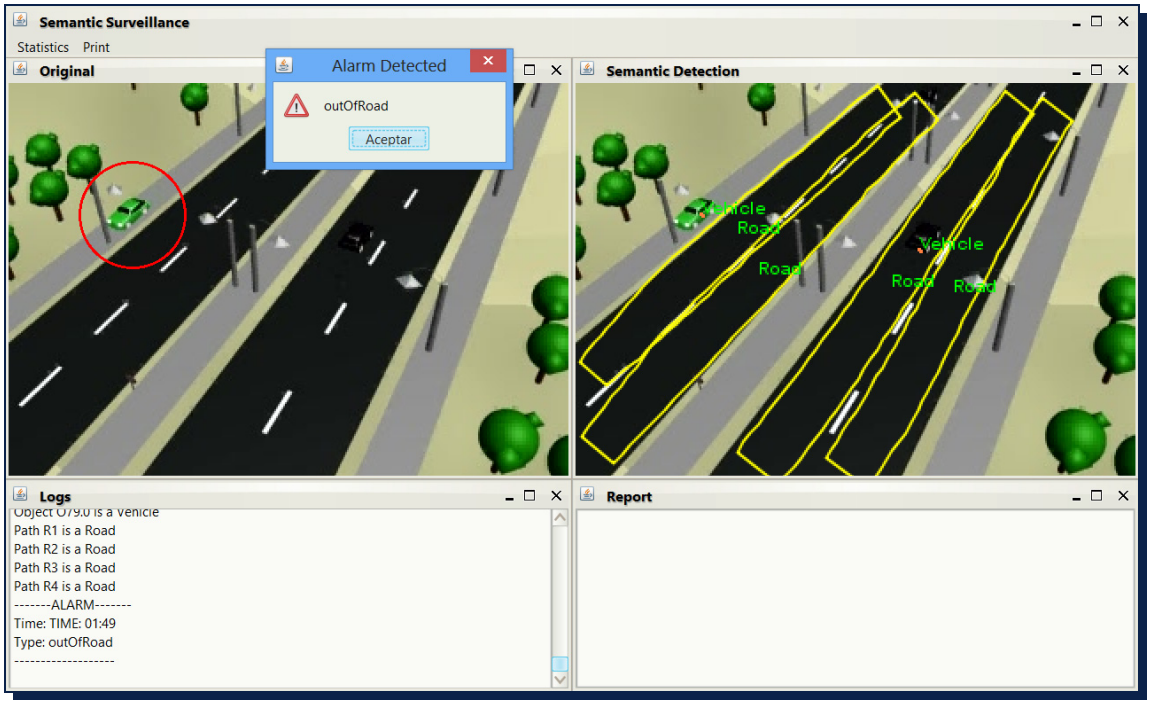


Figura 7.8. Interfaz gráfica del sistema: detección de alarmas.

En la Figura 7.8, tras completarse el aprendizaje, se han descubierto todas las posibles rutas identificándolas como carreteras (en amarillo). En un momento determinado un vehículo (en el área de texto "*Logs*" se puede ver que el objeto O79.0 es de esta clase) se sale fuera y aparece un mensaje de alarma, rodeando el objeto problemático y concretando el evento como "*outOfRoad*".

7.2 Definición de casos de uso

En esta sección se presentan los casos de uso donde se observa el procesado del sistema. Sobre ellos, además, se realizan las pruebas de esfuerzo y se obtiene la precisión del mecanismo en la caracterización de escenas y detección de anomalías. La algorítmica de los sensores visuales inteligentes y las tecnologías de comunicación son impuestas por el proyecto HuSIMS y están fuera de los objetivos de esta Tesis Doctoral.

7.2.1 Exposición de escenarios

El sistema se testea con videos sintéticos y reales para validar su funcionamiento. La complejidad de los escenarios se va incrementando partiendo de escenarios sintéticos sencillos a videos reales más complicados.

7.2.1.1 Escenario 1: Extrarradio de una ciudad

En este caso se utiliza un video sintético desarrollado especialmente dentro del proyecto HuSIMS para llevar a cabo las pruebas de evaluación del mismo.

Figura 7.9. Escenario 1: Extrarradio de una ciudad.

Este escenario muestra un área del extrarradio de una ciudad. Corresponde a una carretera de cuatro carriles, dos para cada sentido separados por una mediana. Además, la carretera dispone de arcenes en ambos lados. En este escenario sólo van a circular vehículos.

En la Figura 7.9 se puede ver una captura de la escena.

7.2.1.2 Escenario 2: Carretera urbana

Igual que en el caso anterior, para este escenario se utiliza un video sintético grabado dentro del proyecto HuSIMS.

Este caso de uso corresponde a un área urbana que incluye una carretera de dos carriles en el mismo sentido. A ambos lados la misma hay aceras por las que transitan peatones. En la Figura 7.10 puede verse la imagen captada por la cámara.

Figura 7.10. Escenario 2: Carretera urbana.

7.2.1.3 Escenario 3: Cruce en área urbana

En este caso se emplea un trozo del videojuego *Grand Theft Auto IV*, especialmente grabado para la validación de esta Tesis Doctoral. Es un juego de acción-aventura de mundo abierto desarrollado por *Rockstar North*.

Este video, aunque sintético, se contempla por su realismo en las escenas, utilizándose como punto intermedio, previo al procesado de escenas reales.

Este escenario corresponde a un cruce en área de urbana donde una carretera de dos carriles, uno en cada sentido, desemboca en otra que cruza con características similares. Un paso de cebra atraviesa la carretera y, a ambos lados de la misma, hay aceras por las que circulan peatones. El cruce y el paso de peatones están controlados por un semáforo. En la Figura 7.11 aparece una captura de la escena.

Figura 7.11. Escenario 3: Cruce en área urbana.

7.2.1.4 Escenario 4: Intersección en área urbana

Este caso de uso muestra en un video real una intersección urbana. Las imágenes proceden del proyecto ITEA CANDELA (disponible en http://www.multitel.be/image/research-development/research-projects/candela.php).

Este escenario corresponde a una intersección dentro de un área urbana. Se pueden observar varias carreteras por las que circulan distintos tipos de vehículos, aceras, carril bici, carril bus, etc. En la Figura 7.12 se visualiza este escenario.

7.2.1.5 Escenario 5: Área urbana muy transitada

Este escenario se recurre a un video real de vigilancia de tráfico del MIT (disponible en http://www.ee.cuhk.edu.hk/ ~ xgwang / MITtraffic.html).

Podría tratarse de un área urbana de cualquier ciudad. Una cámara monitoriza una escena compuesta por varias carreteras y aceras controladas por semáforos, un carril bici y los vehículos y las personas que se desplazan sobre ellos. La Figura 7.13 captura la imagen de este video.

Figura 7.12. Escenario 4: Intersección en área urbana.

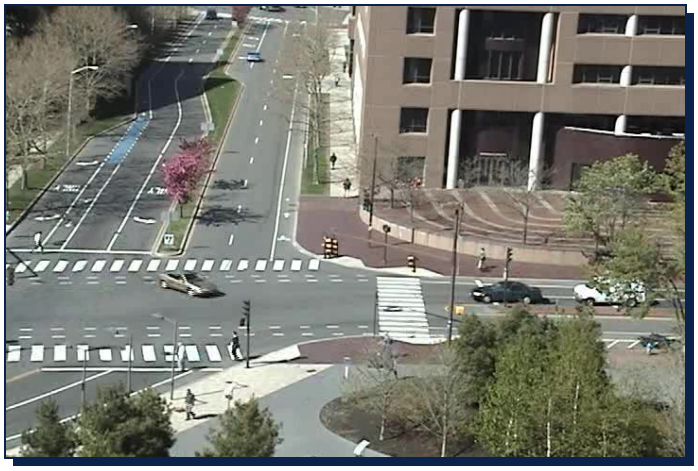


Figura 7.13. Escenario 5: Área urbana muy transitada.

7.2.2 Descripción y análisis de eventos

En esta sección se describen los eventos anómalos que tienen lugar en los diferentes escenarios y la manera en la que el diseño propuesto los procesa para llegar a identificar, en lenguaje formal, la situación concreta.

El detalle de como se produce la detección de los eventos es independiente del escenario, aunque con umbrales específicos para cada uno de los parámetros (adaptados a la escena concreta). El cálculo de los límites para determinar el tipo de objeto, tipo de ruta, etc., se realiza durante un proceso de adaptación del sistema

previo a la ejecución. En esa fase, mediante la observación y la experiencia se determinan los valores óptimos para el escenario y se incluyen en las reglas de inferencia adaptadas para ese caso.

7.2.2.1 Vehículo fuera de la vía

Se va a detectar un vehículo fuera de la vía en el Escenario 1: Extrarradio de una ciudad.

Durante el proceso de aprendizaje (ver Figura 7.7), las reglas definidas permiten identificar los nuevos individuales como vehículos utilizando para ellos los valores de altura, anchura y velocidad. Según se van detectando las rutas durante el proceso de aprendizaje, se van posicionando los objetos dentro de las mismas. En función de la clase a la que estos pertenecen, el proceso de inferencia determina que los distintos caminos son carreteras. Además, la dirección, velocidad, etc., de los objetos establece las características típicas de la vía.

Una vez completado el modo aprendizaje, el sistema entra en la fase de operación, descubriendo las anomalías que va detectando.

El O79.0 aparece en la escena. Sus dimensiones (incluidas en la ontología como *hasWidth* y *hasHeight*) y su velocidad (propiedad *hasSpeed* en la ontología) superan los umbrales establecidos para los vehículos. Esto hace fotograma a fotograma se incremente la propiedad *hasProbabilityVehicle*. Cuando, tras un tiempo de ejecución este valor supera el umbral considerado óptimo para asegurar que el objeto es un vehículo (en este caso se definió como límite 20) y, además, la diferencia con la propiedad *hasProbabilityPedestrian* es elevada (25, mayor que 24, valor establecido como tope), se etiqueta el objeto O79.0 como *Vehicle*.

Durante su desplazamiento por la escena el objeto se ha localizado dentro de la ruta R4, pero en un momento determinado, el vehículo abandona la ruta y no aparece en ninguna otra, con lo que ya no tiene asignada la propiedad *hasLocated*. Este evento hace que se detecte un abandono de la carretera, lanzando una alarma *outOfRoad* indicando esta incidencia.

La Figura 7.8 muestra el proceso completo para la identificación de esta alarma. En la parte de arriba de la figura aparecen las características de O79.0 en el momento t_1.

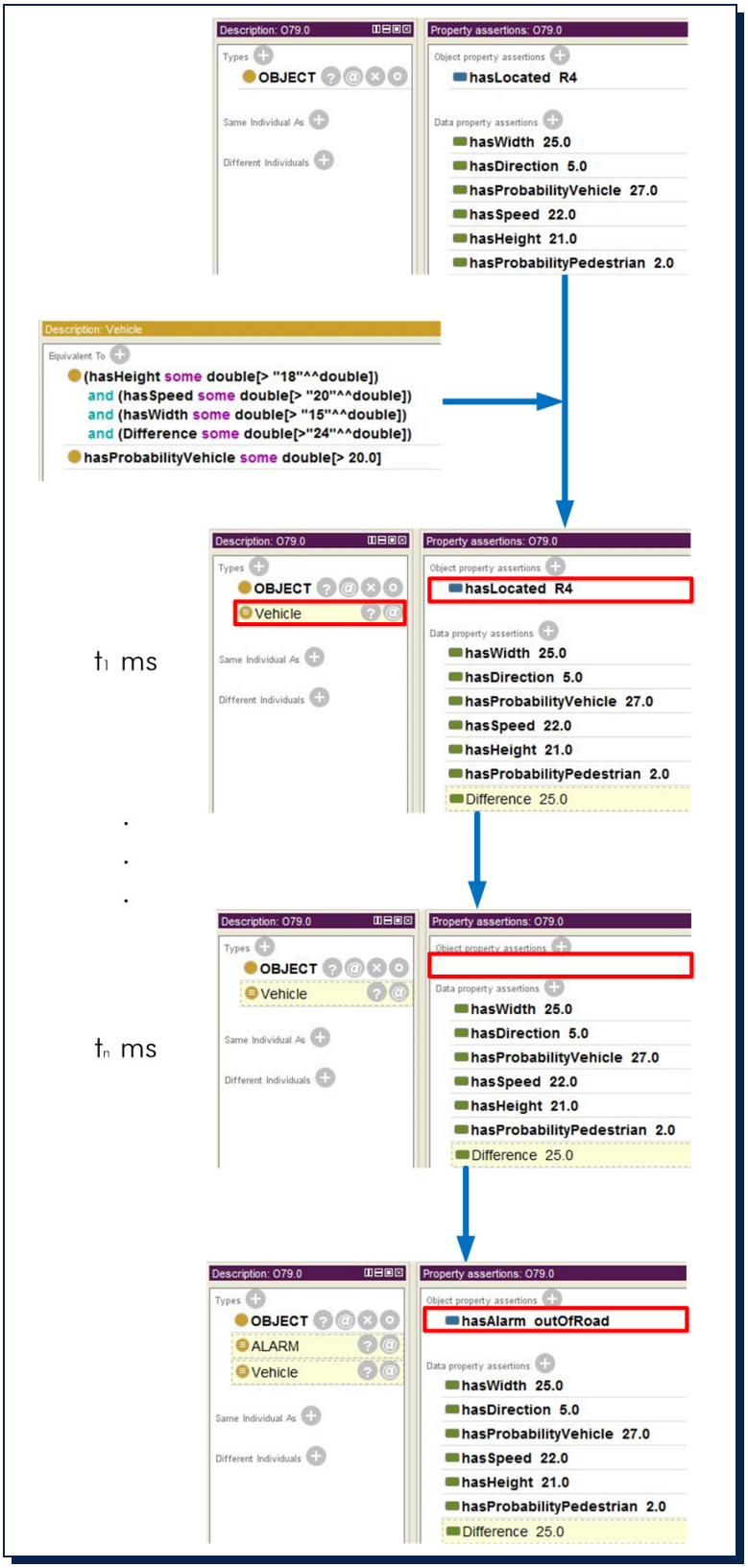


Figura 7.14. Ejemplo de razonamiento semántico para vehículo circulando fuera de vía.

El proceso de razonado determina en ese momento que es un vehículo ya que verifica las propiedades que los describen. Posteriormente, en t_n desaparece la propiedad *hasLocated* (señalado en rojo el lugar donde debería aparecer) y se lanza la alarma.

Para evitar errores, el sistema verifica que este evento sucede un período de tiempo mínimo antes de avisar al centro de control o los servicios de emergencia adecuados. Una de las ventajas del enfoque semántico es que la alarma incluye información detallada sobre el suceso que la causó. En algunos casos la situación podría no merecer la atención de un operador humano (porque sucede muy a menudo o no es extremadamente peligrosa), por lo que el sistema decide automáticamente que la alarma no debe avanzar a la consola de control, si así se ha definido en la ontología. En otros, el sistema de vigilancia podría accionar automáticamente actuadores automáticos, en función de la especificación de las condiciones semánticas para las respuestas autonomas a cada tipo de alarma.

7.2.2.2 Vehículo circulando en dirección inapropiada

Este evento aparece en el Escenario 1: Extrarradio de una ciudad. En este escenario un coche que circula en la dirección equivocada está causando un riesgo para el resto de los vehículos.

La Figura 7.15 muestra el proceso completo para la detección de esta alarma. Al igual que en el caso anterior, la ontología clasifica al individuo como un miembro de la clase de vehículo debido a los valores de las propiedades *hasWidth*, *hasHeight* y *hasSpeed* (26, 30 y 42, respectivamente, y por lo tanto dentro de los intervalos predefinidos para la clase *Vehicle*) durante un número de veces superior al umbral establecido (en este caso *hasProbabilityVehicle* toma el valor 39, mayor que el límite 20).

Además, se asigna este vehículo a la ruta R1, que se ha determinado que es de la clase *Road* ya que han circulado un número de vehículos superior a un tope fijado (*hasNumberVehicle* toma el valor 15, mayor que 10, concretado en este caso en las reglas de inferencia) y muy superior a la cantidad de peatones registrada. Durante el aprendizaje se establece también la dirección de la ruta en función de la de los objetos que habitualmente circulan por ella, tomando su propiedad *hasDirection* valor 1.0.

Recordar que las direcciones se definen tomando como referencia las agujas del reloj (6.6.1.1.1 Modelado de los objetos en movimiento).

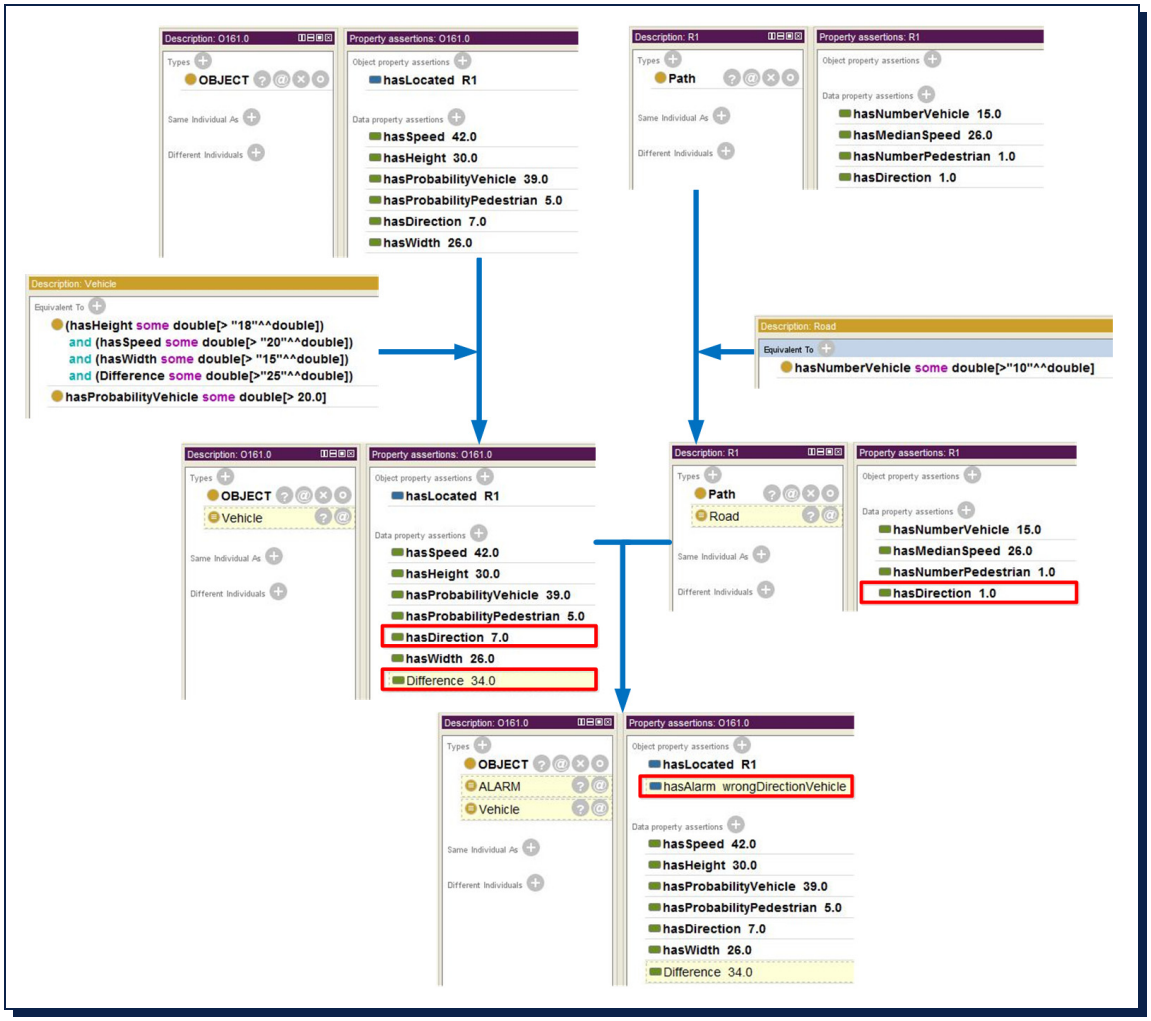


Figura 7.15. Ejemplo de razonamiento semántico para vehículo circulando en dirección errónea.

Para comprobar que el vehículo está en la dirección correcta para la vía en la que se ubica, el razonador analiza el valor de su propiedad *hasDirection* (en este caso el está circulando en dirección 7.0). Como la dirección del vehículo no coincide con la apropiada para la carretera, se produce una alarma de tipo *wrongDirectionVehicle*.

La Figura 7.16 muestra la interfaz gráfica de este evento para el Escenario 1: Extrarradio de una ciudad.

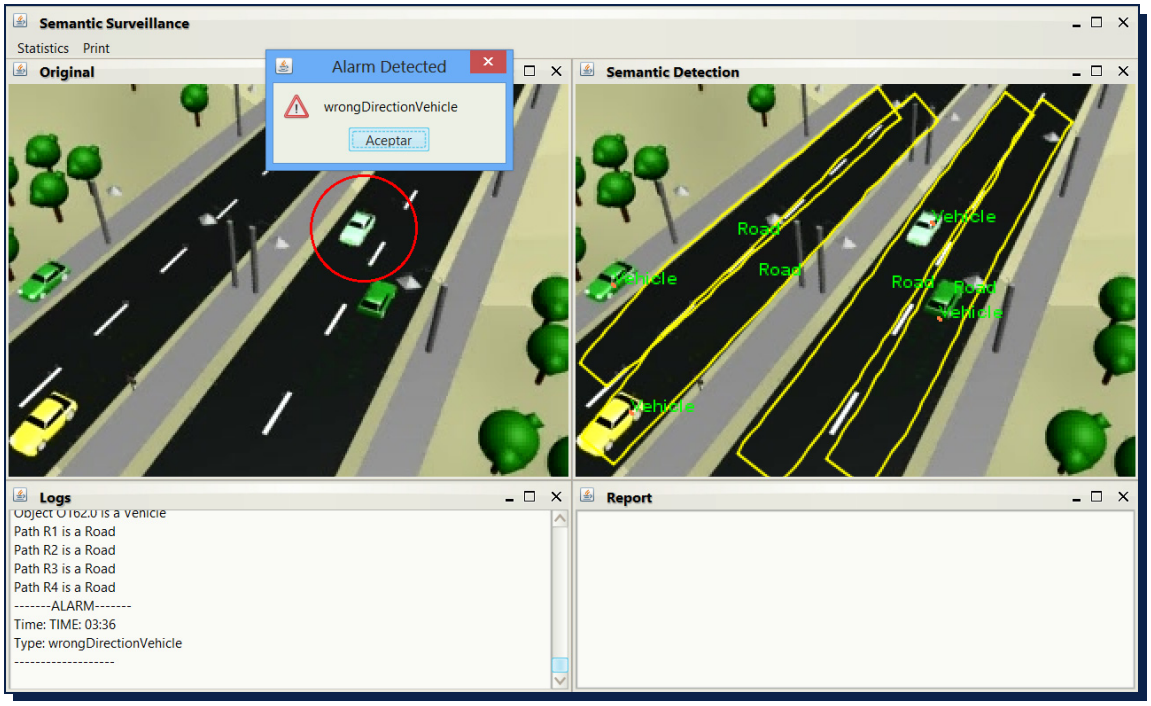


Figura 7.16. Alarma: vehículo circulando en dirección contraria para el escenario 1.

7.2.2.3 Peatón cruzando la calle de forma inapropiada

En este escenario un peatón cruza la carretera inadecuadamente al no utilizar el paso de peatones. Este evento aparece en el Escenario 2: Carretera urbana y Escenario 4: Intersección en área urbana.

La Figura 7.17 muestra el proceso completo para la detección de esta alarma. En este caso, utilizando el mismo procedimiento que en los casos anteriores, la ontología clasifica al individuo como miembro de la clase *Pedestrian*.

Durante su desplazamiento por la escena el objeto se ha localizado dentro de la ruta R4, que se ha determinado que es de la clase *Sidewalk*. En un momento, el peatón abandona esta ruta pasando a la R1, que previamente se ha identificado de tipo *Road*. Este evento hace que el sistema detecte que el patón no transita por un área adecuada, lanzando una alarma *pedestrianCrossingInapproapiately* indicando esta incidencia.

En ocasiones, un objeto puede pertenecer a varia rutas. Cuando un peatón cruza por un paso de peatones, por ejemplo, se localiza a la vez en una ruta tipo *Road* y la ruta por la que realmente está circulando, de tipo *Crosswalk*. Las reglas contemplan esta posibilidad para evitar el lanzamiento de falsas alarmas.

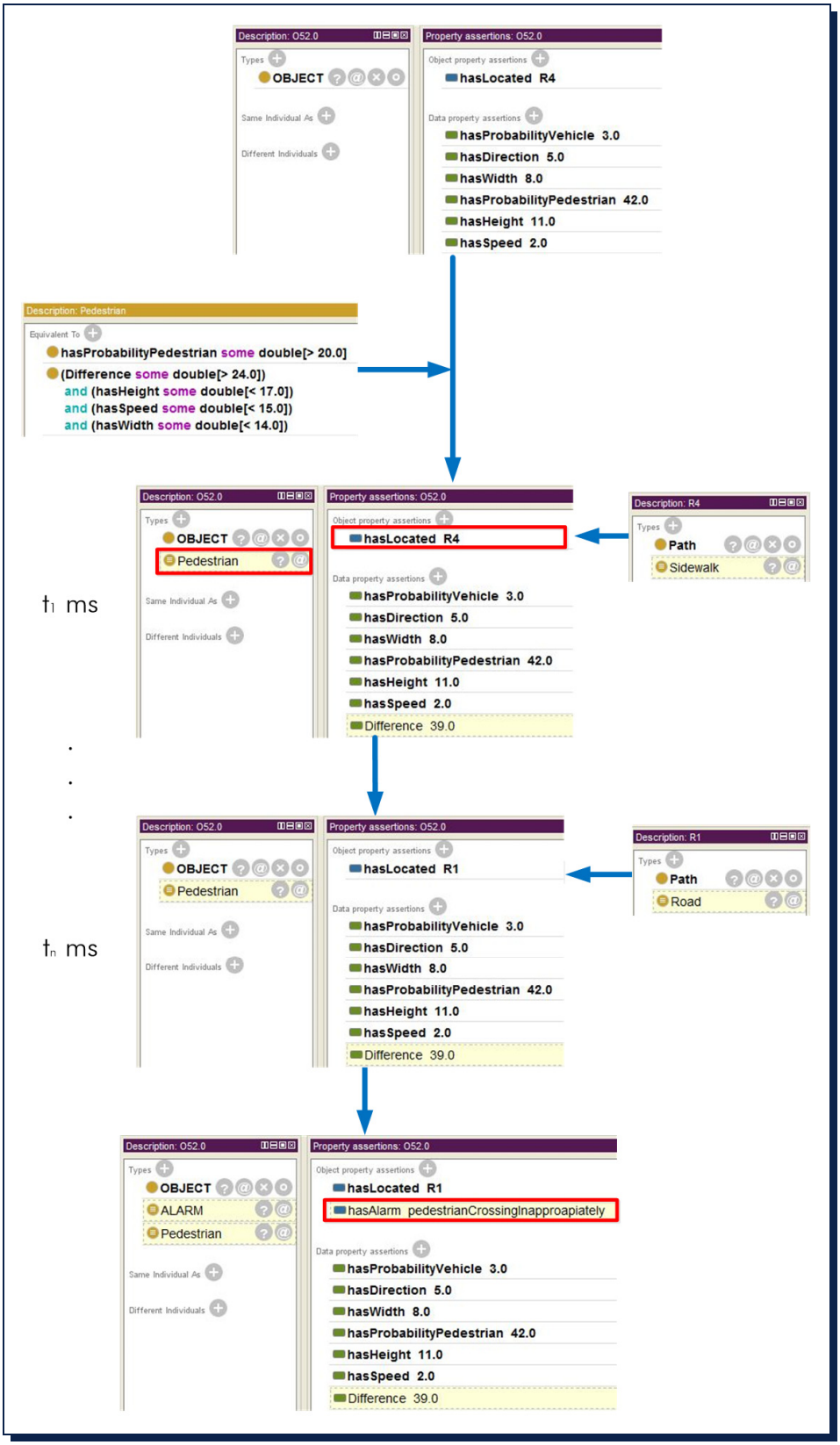


Figura 7.17. Ejemplo de razonamiento semántico para un peatón cruzando por la carretera.

La Figura 7.18 muestra la alarma de este evento para el Escenario 4: Intersección en área urbana.

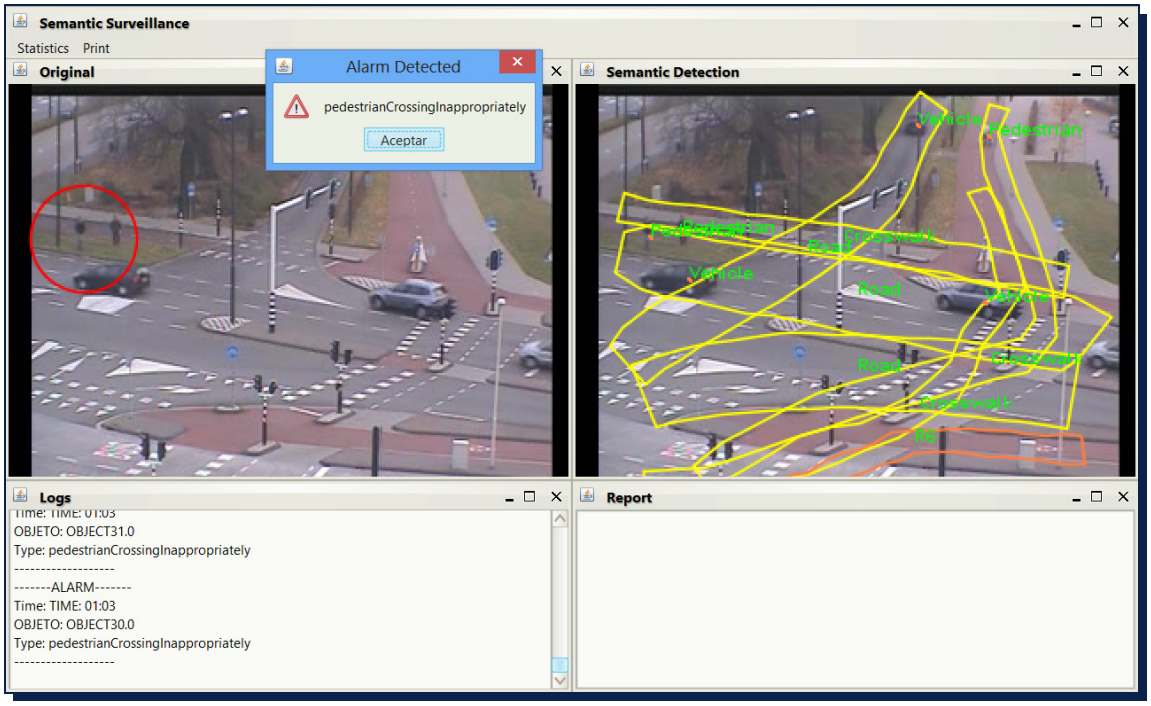


Figura 7.18. Alarma: peatón cruzando de forma inapropiada en el escenario 4.

7.3 Pruebas Experimentales y Resultados

Con el fin de evaluar el comportamiento del sistema propuesto, es importante estudiar tanto el consumo de recursos como la precisión del sistema en la caracterización de escenarios e identificación de las alarmas.

7.3.1 Pruebas de esfuerzo

Para la realización de todas las pruebas se utiliza un ordenador portátil con un procesador Intel Core i7 2,20 GHz con 8 GB de RAM.

En primer lugar se analiza, de forma general, la eficiencia del sistema, determinando el tiempo necesario para el procesado de un video completo, para saber si es viable su funcionamiento en tiempo real. Con este objetivo se prueba en los distintos escenarios, concretando qué módulos son los que consumen más recursos.

Núm. de escenario	Duración (s)	Tiempo de procesado (s)	Núm. de objetos total /Máx. objetos en fotograma	N	Núm. de rutas total	Tiempo medio de procesado (por fotograma) (ms)	Tiempo medio para detección de rutas (por fotograma) (ms)	Tiempo medio para modelado semántico (por fotograma) (ms)
1	219	102	164/4	40	4	43,2	54,28	18,8
2	155	89	58/14	20	10	55,0	46,40	17,8
3	82	38	25/13	10	15	38,7	22,91	15,3
4	85	41	37/11	20	8	38,9	40,48	14,5
5	334	192	153/15	10	29	56,5	43,30	27,3

Tabla 7.1. Pruebas de esfuerzo.

Durante el procesado de cada escenario se recogen los tiempos consumidos, para cada fotograma individual, por el módulo de detección de rutas y el de modelado semántico, así como el tiempo total invertido. Los valores medios por fotograma obtenidos aparecen reflejados en la Tabla 7.1. Ademas, se incluye la duración total del video y el tiempo que dura su procesado completo. Se observa que, independientemente de la complejidad del escenario, el tiempo que tarda en procesarse completamente es inferior a su duración.

Por otro lado se especifican el número de objetos del escenario completo, el número máximo de objetos encontrados en un fotograma, el número de rutas y N (número de puntos de cada trayectoria o ruta seleccionado para ese escenario en concreto). El objetivo de incluir estos valores es determinar su influencia en los tiempos de procesado de cada módulo.

En la detección de rutas, el tiempo se incrementa cuando se eleva el número de puntos, N, ya que el cálculo de la distancia de Hausdorff requiere comparación punto a punto. Además, en escenarios donde el número de objetos a procesar es alto se generan más trayectorias cerradas, con lo que va a ser necesario el cálculo de distancias para fusionar con posibles rutas semejantes.

El procesado semántico depende básicamente del número de objetos de la escena, aunque escenas con pocos objetos pueden ver incrementado el tiempo invertido en el modelado semántico si el número de puntos seleccionado para las trayectorias/rutas es elevado.

Por otro lado, para evaluar si el consumo de recursos es constante a lo largo de todo escenario, si es mayor en los primeros fotogramas o si existen picos en el procesado, en la Figura 7.19 a Figura 7.23 se recoge la evolución de los distintos tiempos para cada escenario.

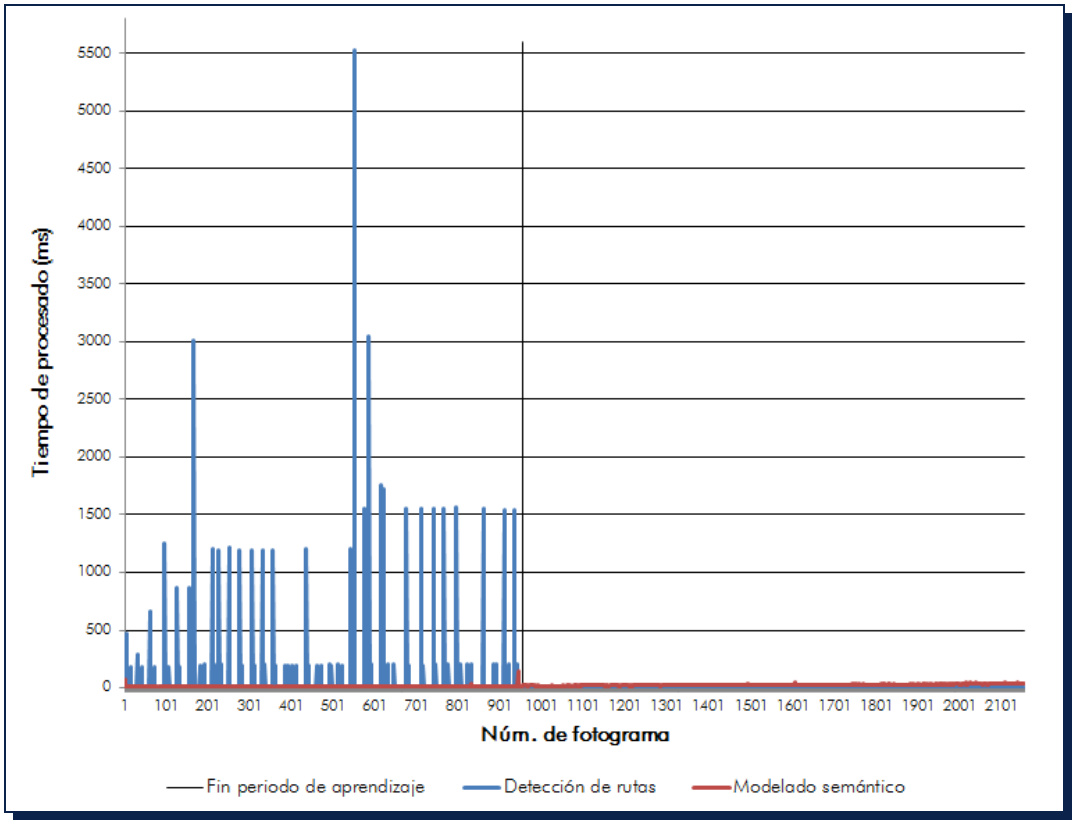


Figura 7.19. Tiempo de procesado por fotograma: Escenario 1.

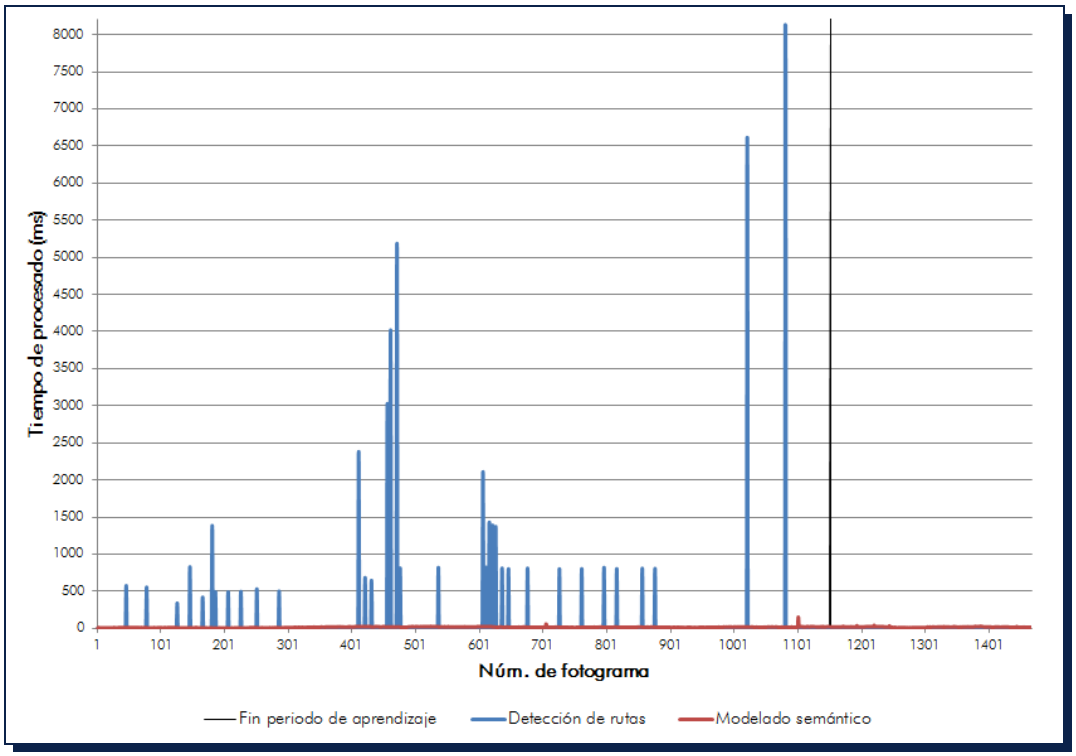


Figura 7.20. Tiempo de procesado por fotograma: Escenario 2.

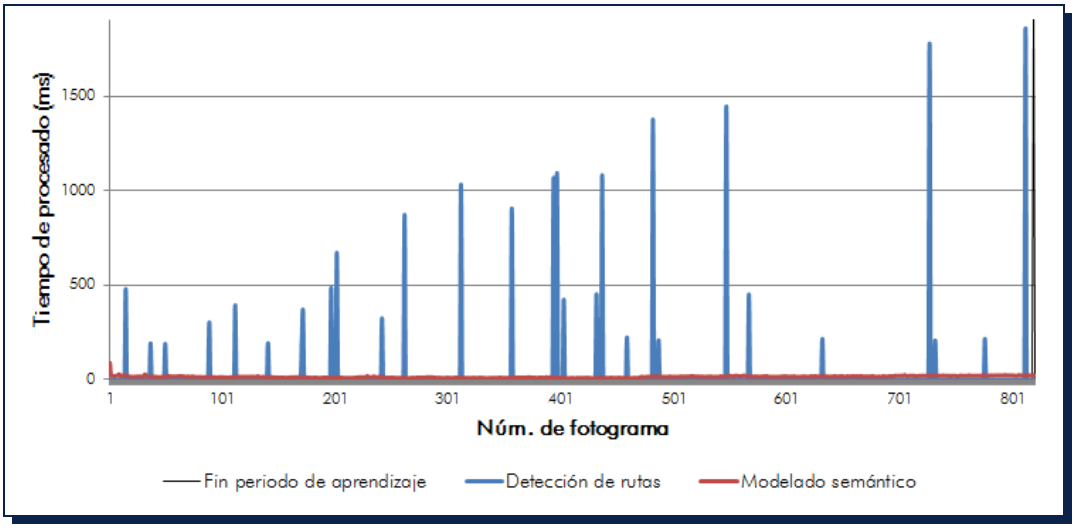


Figura 7.21. Tiempo de procesado por fotograma: Escenario 3.

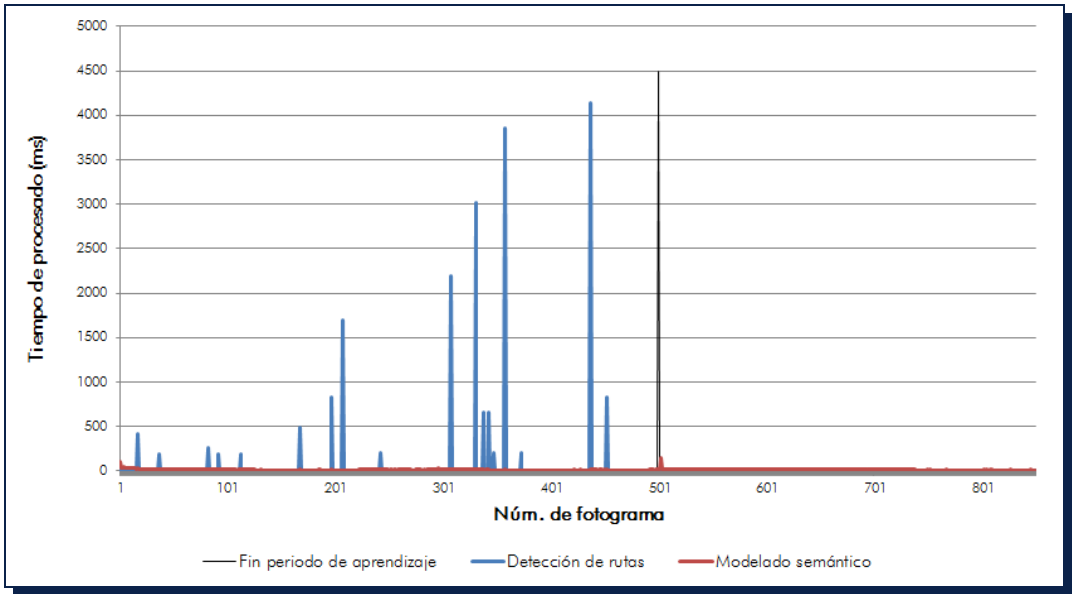


Figura 7.22. Tiempo de procesado por fotograma: Escenario 4.

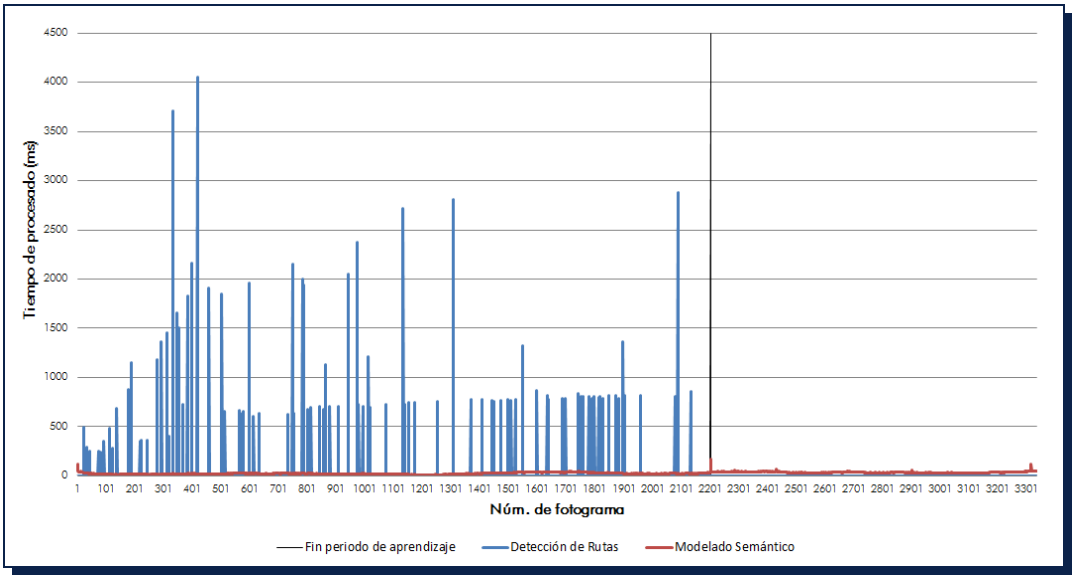


Figura 7.23. Tiempo de procesado por fotograma: Escenario 5.

Del análisis de las figuras se puede determinar que el módulo de detección de rutas es el que más tiempo de procesado consume durante el periodo de aprendizaje, produciéndose incrementos momentáneos importantes cuando se descubren nuevas trayectorias cerradas. Estos máximos no dependen de la complejidad del escenario sino que aumentan en número cuanto mayor cantidad de objetos se identifican.

Debido a estos picos en el procesado podría pensarse que el sistema no es capaz de funcionar en tiempo real. Sin embargo, la detección de rutas sólo se ejecuta durante el proceso de aprendizaje, siendo el modo operación el habitual, y por tanto el que establece las limitaciones en cuanto a tiempos de ejecución.

Cuando se está funcionado en modo operación, la mayor parte de los esfuerzos los consume el modelado semántico, por lo que se va a analizar cómo influye el número de objetos y rutas en el tiempo requerido por este módulo.

El modelado ontológico comprende la traducción semántica y subprocesos de razonamiento semántico (población, inclusión de reglas, razonado, consultas para la identificación de alarmas, etc.). En este punto, los datos que se tratan ya no son visuales, sino sólo individuos dentro de las ontologías. Ésto hace que no haya relación entre los esfuerzos necesarios para llevar a cabo estos procesos y las magnitudes visuales/espaciales de la escena como la geometría o la resolución. Por tanto, los resultados de la evaluación comparativa son directamente extrapolables para condiciones de funcionamiento reales e independientes del escenario.

Para cada prueba, se generan 50 fotogramas con objetos y rutas en ubicaciones aleatorias que se procesan con el módulo semántico. La Figura 7.24 presenta el promedio de tiempo de procesamiento de cada fotograma.

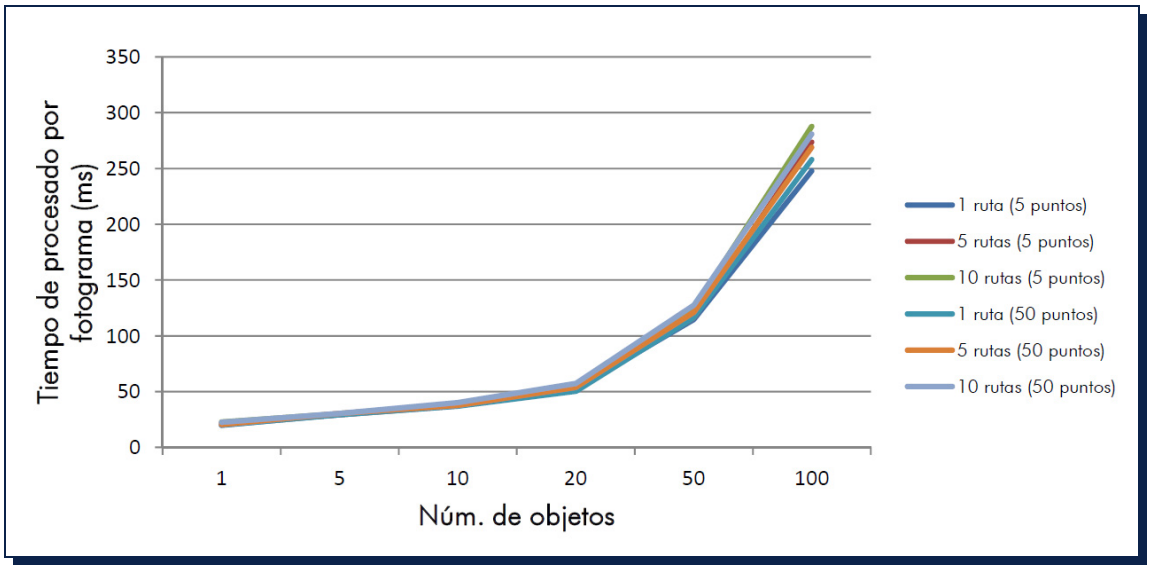


Figura 7.24. Tiempo de procesado de cada fotograma en función del número de objetos, rutas de la imagen y los puntos de las mismas.

Los esfuerzos necesarios dependen principalmente del número de objetos. Las variaciones en el número de rutas o el número de puntos por trayectoria dan resultados muy similares. Los valores obtenidos son satisfactorios, desde un fotograma con menos de 20 objetos que se procesa en 25-50 ms (permitiendo un procesado en tiempo real para una captación de hasta 20 fps) a un escenario muy concurrido, con 100 objetos, que se procesa en menos de 300 ms.

7.3.2 Precisión en la caracterización de escenarios

En esta sección se va a analizar la precisión del sistema. Para ello se determina la exactitud en la identificación y clasificación de rutas, de objetos y de alarmas.

7.3.2.1 Acierto en la determinación de escenas y objetos

Para conocer la fidelidad en la caracterización de escenarios se comprueba, de forma visual, el número de rutas reales del escenario, divididas de tres categorías: rutas, pasos de peatones y aceras. Estos valores se van a comparar con los determinados por el detector de rutas y etiquetados por el módulo de modelado semántico en la última escena mostrada por la interfaz gráfica

Núm. escenario	Núm. rutas	Núm. *Road*	Núm. *Road* identificadas	Núm. *Sidewalk*	Núm. *Sidewalk* identificadas	Núm. *Crosswalk*	Núm. *Crosswalk* identificadas	Núm. rutas no identificadas	Núm. identificadas erróneamente	% identificadas	% erróneas
1	4	4	4	0	0	0	0	0	0	100%	0%
2	5	2	2	2	2	1	1	0	0	100%	0%
3	5	2	2	1	1	1	1	0	0	100%	0%
4	7	4	3	0	0	3	2	2	0	71,5%	0%
5	9	9	7	0	0	5	3	4	0	71,5%	0%

Tabla 7.2. Acierto en la determinación de rutas.

Núm. escenario	Núm. objetos	Núm. peatones	Núm. peatones identificados	Núm. vehículos	Núm. vehículos identificados	Núm. objetos no identificados	Núm. identificados erróneamente	% identificados	% erróneos (peatones/ vehículos)
1	164	0	0	164	163	0	0	100%	0%
2	58	18	18	40	40	0	0	100%	0%
3	25	14	14	11	11	0	0	100%	0%
4	37	29	29	18	18	0	0	100%	0%
5	151	45	43	106	101	0	7	100%	4,4% / 4,7%

Tabla 7.3. Precisión en la identificación de objetos.

La Tabla 7.2 recoge los valores obtenidos, incluyendo los falsos positivos (rutas identificadas erroneamente) y las rutas no detectadas. Indicar que los datos se han tomado tras una ejecución normal del sistema, es decir, el módulo detector de rutas deja de procesar durante el tiempo de operación. Analizando los resultados puede observarse que, para escenarios sencillos, en cuanto a la identificación y clasificación de rutas se obtienen precisiones del 100%, valor que se reducen según se complica la escena.

Al igual que en el caso anterior, se observa el video identificando el número de peatones y vehículos reales del escenario. A continuación se monitoriza la interfaz gráfica, que muestra el proceso de clasificación, apuntando los errores en la clasificación, objetos no detectados, etc., recogiendo los mismos en la Tabla 7.3.

A pesar de que al aumentar la complejidad del escenario los resultados obtenidos no son los ideales, se pueden considerar que un 95,6% de precisión en el peor de los casos, es un buen dato, teniendo en cuenta que en estudios como el de Lin *et al.* [202] o Lee *et al.* [203] se consiguen valores para vehículos del 89,1 y 94% y para peatones del 88,2 y 83,72% respectivamente.

7.3.2.2 Rigor en la identificación de situaciones de alerta

La Tabla 7.4 recoge los resultados obtenidos para los diferentes escenarios en la detección e identificación de situaciones anómalas.

Núm. escenario	Evento a detectar	Momento en que sucede en evento (s)	Evento detectado	Momento en que se detecta en evento (s)	Núm. falsos positivos	Núm. eventos no detectados
1	Vehículo fuera de vía	1:50	Out Of Road	1:50	0	0
	Vehículo circulando en dirección inapropiada	3:36	Wrong Direction Vehicle	3:36	0	0
2	Peatón cruzando la calle de forma inapropiada	2:27	Pedestrian Crossing Inapropiately	2:27	0	0
4	Peatón cruzando la calle de forma inapropiada	1:02	Pedestrian Crossing Inapropiately	1:02	0	0

Tabla 7.4. Rigor en la identificación de situaciones anómalas.

Las pruebas sobre la exactitud en la identificación alarma y tasas de falsas alarmas son de poco valor cuando se aplican a la fase de modelado ontológico del sistema. Debido a su naturaleza semántica (es decir, las condiciones para detectar eventos se definen en términos del significado de los objetos y su comportamiento), si la escena se caracteriza adecuadamente por el detector de ruta y la ontología considera las condiciones apropiadas, son habituales tasas de identificación del 100% y 0% de falsas alarmas. Por lo tanto, la precisión del sistema depende directamente de las tasas de detección de ruta (que a su vez está influenciado en gran medida por la calidad de los algoritmos de detección de movimiento).

7.4 Aplicación a otros dominios

Indicar que, el sistema no está pensado para ambientes muy concurridos donde, es muy difícil, incluso visualmente por un operador humano, identificar el robo de una cartera, por ejemplo. Además, al basarse únicamente en el movimiento para realizar las detecciones, no detecta caras u otros rasgos significativos de las personas para poder seguirlos. El objetivo no es tras un acto vandálico encontrar al responsable, sino determinar que se está llevando a cabo para informar las autoridades y que actúen con la mayor celeridad.

A pesar de todo, no está limitado para un funcionamiento exclusivo en escenarios de control de tráfico sino que, la metodología expuesta, es adaptable fácilmente a otros entornos. Manteniendo el software y los módulos de hardware y simplemente cambiando la ontología y las reglas empleadas en el razonamiento semántico, la adaptación a un nuevo ámbito es inmediata.

Siguiendo el movimiento de un determinado objeto por la escena, sus paradas, trayectorias repetitivas durante el tiempo, etc., se podrían detectar personas sospechosas que quizá observan el comportamiento del resto para realizar hurtos. O, conociendo los tiempos de funcionamiento de un semáforo, por ejemplo, se conseguirían detectar vehículos que no realizan la parada como se espera.

Continuando con las posibilidades que ofrece la propuesta, a continuación se describen dos escenarios de aplicación adicionales: el control de acceso a un recinto o la detección de objetos que caen en las vías del tren o metro.

7.4.1 Control de acceso

Durante el desarrollo del proyecto HuSIMS, en Ness Ziona (Israel), se hizo un despliegue para detectar el acceso a un recinto cerrado con el objetivo de controlar la entrada de vehículos y/o personas a través de una puerta trasera.

La primera acción a realizar es determinar cuando se produce la apertura de la puerta determinando si han coseguido acceder, y en ese caso especificar si es una persona o vehículo el intruso. Por otro lado, como se producen inclursiones no autorizadas por encima de la valla, desean estar informados también de estos sucesos. Ambos eventos se muestran en la Figura 7.25.

Figura 7.25. Control de acceso: puerta abierta, acceso de una persona al recinto.

La ontología de la Figura 7.26 detalla los objetos de este escenario, los comportamientos a detectar y los periodos de tiempo en los que, estos eventos podrían considerarse realizarlos personal autorizado o acceso ilegal.

Al igual que en los escenarios de control de tráfico, la clasificación de los diferentes objetos se efectúa utilizando sus parámetros de movimiento (dimensiones y velocidad). Así, las dimensiones de la puerta serán elevadas y su movimiento lento, comparado con el peatón, mucho más pequeño y rápido.

Así para detectar la apertura de la puerta, habitualmente estática, sólo hay que determinar que el objeto de la escena que se mueve es de la clase "puerta". Si, además, se identifica un peatón que la cruza, movimiento no definido por el detector de rutas durante el aprendizaje, se registra una alerta "accesoPersona" que en función del horario y del día de la semana (laborable o no), se convierte en alarma.

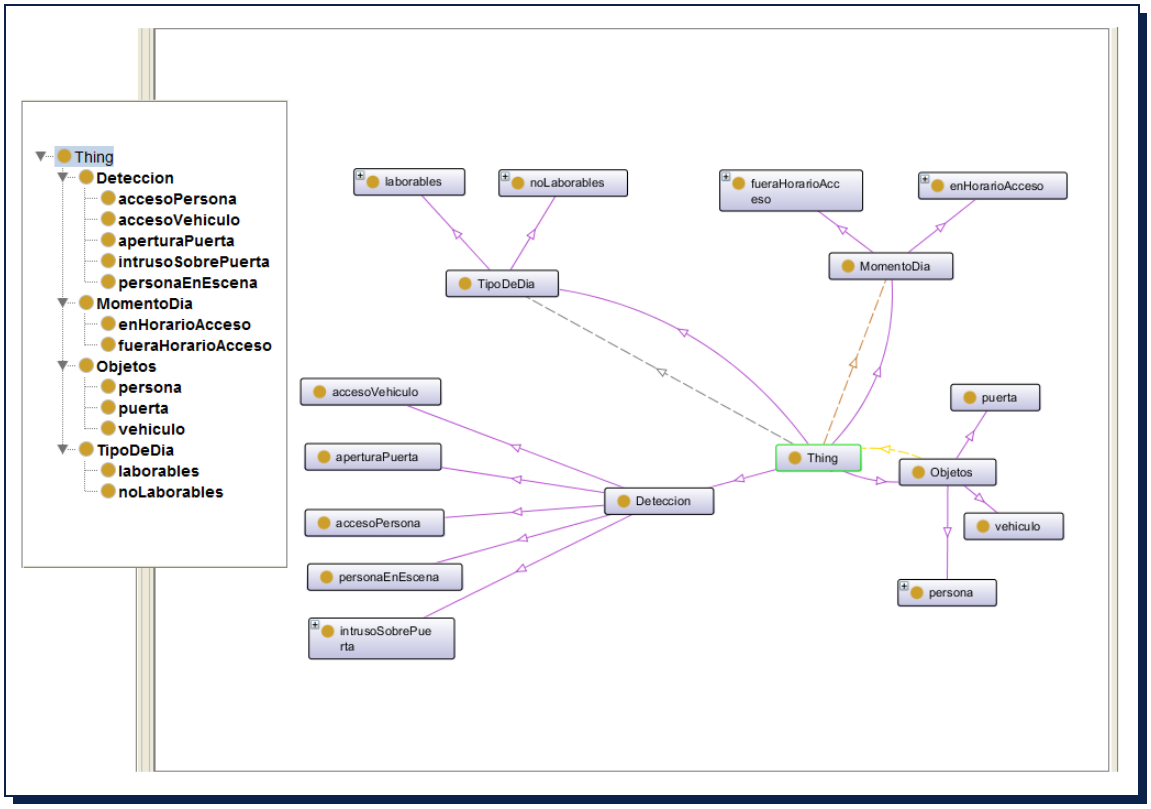


Figura 7.26. Ontología para el control de acceso.

Para el segundo caso, la persona saltando la puerta para acceder al recinto, se detecta una persona con un movimiento no habitual, de hecho, supera los límites de la escena establecidos en las reglas como adecuados para el desplazamiento de los individuos. En este caso, la identificación supone el lanzamiento de una alarma independientemente del horario y día de trabajo, ya que no es un comportamiento realizado por el personal municipal.

7.4.2 Caída de peatón en el metro

En este escenario, la metodología se aplica como un medio de seguridad para detectar cuándo una persona cae a las vías del tren en una estación (Figura 7.27). En esa situación, la ontología clasifica al individuo como un peatón que camina en la plataforma. Cuando el viandante se cae y se encuentra en las vías, el sistema genera una alarma.

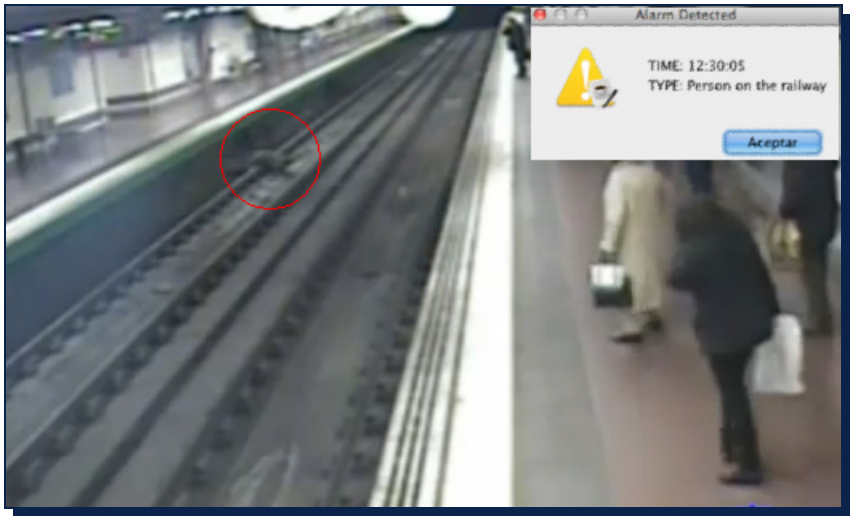


Figura 7.27. Caída de peatón en el metro.

Vale la pena señalar que si un objeto se desploma, cuando aterriza ya no es divisado por la cámara, ya que permanece estático. Sin embargo, el sistema entiende que un objeto que se movía ha desaparecido en un lugar que no es un sumidero, por lo que se lanza una alarma.

Un motor de vigilancia basado en el análisis estadístico también podría detectar que se ha producido una anomalía ya que hay un objeto detectado en movimiento en una zona y con una dirección no habituales. Sin embargo, en ese caso, no es posible diferenciar automáticamente entre una persona que cae a las vías o una que sube sobre un banco, por ejemplo, ya que ambas acciones se saldrían de la norma. Esto significa que un operador humano tiene que comprobar la señal de video para identificar las condiciones de alarma y, en este caso, llamar al conductor de metro para advertirle acerca de la caída de la persona. Este proceso puede tardar algún tiempo, unos segundos que en situaciones críticas como ésta, puede suponer no llegar a tiempo.

Por el contrario, el sistema semántico propuesto en este trabajo presenta la ventaja de ser capaz de discriminar el tipo de alarma y configurarse (especificando un comportamiento en la ontología) para enviar automáticamente la alerta "*Person on the railway*" a los conductores del metro, o incluso activar los frenos de emergencia del tren entrando en la estación. La literatura [7][8][23][34] presenta alternativas de propósito específico, basados en la semántica, que también pueden ofrecer esta función de automatización de respuesta de alarma. Sin embargo, en esos casos el dominio de conocimiento está codificado en el propio sistema, por lo que sólo es funcional en ese campo de aplicación concreto. Por el contrario, en la metodología

que se presenta a lo largo de esta Tesis, la semántica se separa de la implementación, así que cambiando la ontología y las reglas de inferencia es suficiente para tener exactamente el mismo sistema operativo en un dominio diferente.

7.5 Conclusiones

Para verificar que el mecanismo propuesto es adecuado para el fin descrito, se ha testeado en distintos escenarios urbanos e interurbanos para el control de tráfico y la detección de infracciones en estos entornos. Con ello, se ha determinado tanto la eficiencia del sistema para su funcionamiento en tiempo real en diferentes escenarios, como la precisión en la caracterización de las escenas e identificación de alarmas, obteniendo resultados satisfactorios en ambos casos. Así pues, el algoritmo descrito permite la interpretación de diferentes escenas de video de diferente grado de complejidad en tiempo real.

Por otro lado, la descripción de situaciones de alerta se consigue gracias a la semántica, consiguiendo acercarse lo máximo posible al lenguaje formal para que un operador o los servicios de emergencia conozcan el evento concreto, no sólo que se produce un comportamiento que se sale del habitual.

Para finalizar, indicar que el sistema puede adaptarse de forma sencilla para su funcionamiento en diferentes campos de aplicación. Además, hace uso, no sólo de la información de movimiento proporcionada por los sensores visuales, sino que también puede incluir otros datos del entorno, muy adecuado para su uso dentro de las *Smart Cities*.

8

CONCLUSIONES Y LÍNEAS FUTURAS

Como se expuso en el Capítulo 1, el principal objetivo de esta Tesis Doctoral ha sido el diseño e implementación de un sistema semántico flexible, capaz de llevar a cabo la caracterización de escenarios y que pueda ser fácilmente adaptado para funcionar adecuadamente en diferentes dominios. La utilización de semántica y ontologías para la caracterización de escenas presenta varias ventajas, principalmente que es posible dotar al sistema de un modelo de conocimiento humano con el que definir las escenas en función de unos conceptos con sentido. Una aplicación directa de la descripción de escenas son los sistemas de videovigilancia, sobre todo enfocados a las *Smart Cities,* ya que proporcionan seguridad personal y material sin intromisión a la privacidad. La ejecución de este objetivo global se ha realizado a partir de los distintos objetivos específicos identificados en el Capítulo 1 que se han ido completando a lo largo de los Capítulos precedentes.

Este Capítulo se encuentra estructurado de la siguiente manera. Las principales aportaciones originales presentes en la Tesis se exponen en la Sección 8.1. En la Sección 8.2 se detalla la validación de los resultados obtenidos, incluyendo la participación en proyectos de investigación, tanto europeos como nacionales, así como la elaboración de diferentes publicaciones. Las principales conclusiones de la Tesis se exponen en la Sección 8.3. Finalmente, en la Sección 8.4 se introducen las futuras líneas de investigación que se derivan del trabajo realizado.

8.1 Aportaciones de la Tesis

Las principales aportaciones originales de la Tesis ya se expusieron de forma detallada en la Sección 1.3 del Capítulo 1, si bien, a modo de conclusión pueden resumirse de la siguiente manera:

- Desarrollo de una metodología para la caracterización automatizada de escenarios puramente semántica utilizando las ontologías y reglas diseñadas.

  Tipificación de los objetos concretos que en un momento determinado aparecen en la imagen y su localización dentro de la misma, pero no sólo espacial, sino dentro del propio contexto. Para ello se hace uso de los patrones de movimiento y las características de los objetos (complementados, si se dispone, con los datos procedentes de otros sensores) tanto del momento actual como de momentos previos. Además, el procesado es puramente semántico y se obtiene la información necesaria para conseguir la definición de escenas.

- Utilización de ontologías persistentes para el modelado de escenarios.

  Para mejorar los resultados de los procesos de inferencia se usan ontologías persistentes. Los historiales de objetos previos de la escena y las conclusiones a las que se ha llegado en razonados anteriores se van incluyendo en la ontología para contar con una información más completa y conseguir mejores resultados, evitando errores de identificación.

- Diseño e implementación de un sistema integrado que a partir de imágenes, incluso de baja resolución, indique, en lenguaje natural, la situación de alarma.

  Diseño y desarrollo de un sistema integral capaz de:

  - Clasificar, utilizando semántica, los objetos en movimiento que en cada momento aparecen en la escena, empleando únicamente los parámetros del desplazamiento y características de los mismos que se obtienen mediante un tratamiento sencillo de imágenes. Este mecanismo de identificación no necesita disponer a priori de modelos previos con los que comparar las formas de los mismos y es válido incluso para imágenes de baja calidad.

  - Identificar, usando un conjunto de ontologías y reglas, los comportamientos de los elementos móviles identificados para conocer qué está sucediendo en ese momento en la escena, permitiendo así distinguir situaciones especiales. Además, el sistema integra la información procedente de diferentes sensores, si los hubiera, como complemento para distinguir las situaciones con mayor precisión.

  - Adecuar la metodología, de forma sencilla, para que sea fácilmente adaptable para funcionar en diferentes dominios.

La Tabla 8.1 recoge una comparativa entre los mecanismos actuales y las aportaciones de esta Tesis Doctoral.

Comparativa	Estado del arte	Aportaciones
Caracterización de escenarios puramente semántica utilizando las ontologías y reglas desarrolladas.	Recurren a la semántica sólo para realizar el modelado de la escena utilizando otros mecanismos para identificar los objetos.	Es semántico el razonamiento, identificación de objetos y lanzamiento de alarmas.
Utilización de ontologías persistentes para el modelado de escenarios.	No utilizan ontologías persistentes sino que emplean el razonamiento sobre los datos de la escena en el momento actual.	Se vale de la aplicación de persistencia a las ontologías para disponer de información de los históricos y poder realimentar el sistema con los resultados de procesos de inferencia previos.
Diseño e implementación de un sistema integrado que a partir de imágenes, incluso de baja resolución, indique, en lenguaje natural, la situación de alarma.	Los métodos existentes **no incluyen todas las características**, en algunos casos sólo alguna de ellas, habitualmente: -Para la identificación de objetos suelen utilizar la forma, color, etc., para compararlos con imágenes preestablecidas que sirven como modelo. -Utilizan básicamente información visual y no pueden procesar la información heterogénea procedente de sensores del entorno. -Suelen determinar los comportamientos de los objetos que aparecen en la escena no las situaciones de alarma. -Los sistemas están centrados para su funcionamiento en escenarios controlados y su adaptación a nuevas escenas suele ser compleja.	El sistema posee **todas** las siguientes características: -Sólo utiliza los parámetros de movimiento de los objetos. No utiliza parámetros adicionales como la forma, color, etc., por lo que puede funcionar con imágenes de baja resolución mejorando la privacidad. -Puede incluir información no sólo procedente de cámaras de video sino también es capaz de utilizar la procedente de otros sensores que la complementen. -Identifica las situaciones anómalas en lenguaje natural. -El sistema se adapta fácilmente para funcionar en diferentes dominios.

Tabla 8.1. Aportaciones originales.

8.2 Validación de los Resultados

Los trabajos incluidos en esta Tesis Doctoral han sido desarrollados y validados a través de la participación activa en diversos proyectos de investigación así como mediante la publicación de los resultados más relevantes en diversos foros de divulgación científico-técnica.

8.2.1 Proyectos de Investigación

Si bien el trabajo realizado en esta Tesis Doctoral se ha llevado a cabo dentro del proyecto de investigación HuSIMS, otros proyectos, tanto nacionales como internacionales, han servido como base, introduciendo tecnologías y conceptos utilizados en este trabajo.

	**HuSIMS – Human Situation Monitoring System** Ref: TSI-020400-2010-102. EUREKA-CELTIC, Ministerio de Industria, Turismo y Comercio – FEDER, 2010-2012
	**OPUCE – Open Platform for User-centric Service Creation and Execution** Ref: FP6-34101. Comisión Europea IST FP6 2006-2009
	**VISION – Comunicaciones de Video de Nueva Generación** Ref: CENIT 2006-2010. Ministerio de Industria, Turismo y Comercio
	**mIO! – Technologies for Service Delivery in the future intelligent universe** Ref: CENIT 2008-2011. Ministerio de Industria, Turismo y Comercio.
	**WIMSAT – Wimax, IMS and SATellite Convergence** Ref: TSI-020100-2010-103. Ministerio de Industria, Turismo y Comercio – FEDER, 2008-2011
	**V-ER – Virtualización en la nube de Escritorio Remoto** Ref: TSI-020100-2011-145. Ministerio de Industria, Turismo y Comercio – FEDER, 2011-2013

8.2.2 Publicaciones

8.2.2.1 Artículos científicos en revistas indexadas

Al igual que en el caso de los proyectos de investigación, hay varias publicaciones que recogen los diseños, implementaciones y resultados derivados de esta Tesis Doctoral:

- Calavia, L., Baladrón, C., Aguiar, J. M., Carro, B., & Sánchez-Esguevillas, A. (2012). **A Semantic Autonomous Video Surveillance System for Dense Camera Networks in Smart Cities**. *Sensors*, 12(8), 10407-10429. ISSN 1424-8220. Digital Object Identifier: 10.3390/s120810407.
    - Índice de impacto: 1.953 (*Journal Citation Report ISI*)
    - Área: Instruments & Instrumentation
    - Posición: #9/57 (Q1)
    - Año: 2012
- Fernández, J., Calavia, L., Baladrón, C., Aguiar, J. M., Carro, B., Sánchez-Esguevillas, A., Alonso-López, J. A., & Smilansky, Z. (2013). **An Intelligent Surveillance Platform for Large Metropolitan Areas with Dense Sensor Deployment.** *Sensors*, 13(6), 7414-7442. ISSN 1424-8220. Digital Object Identifier: 10.3390/s130607414.
    - Índice de impacto: 1.953 (*Journal Citation Report ISI*)
    - Área: Instruments & Instrumentation
    - Posición: #9/57 (Q1)
    - Año: 2012

Sin embargo, otros artículos incluyen investigaciones transversales que sirven como base para la toma de decisiones y selección de tecnologías:

- Baladrón, C., Aguiar, J. M., Calavia, L., Carro, B., Sánchez-Esguevillas, A., & Hernández, L. (2012). **Performance Study of the Application of Artificial Neural Networks to the Completion and Prediction of Data Retrieved by Underwater Sensors**. *Sensors*, 12(2), 1468-1481. ISSN 1424-8220. Digital Object Identifier: 10.3390/s120201468
    - Índice de impacto: 1.953 (*Journal Citation Report ISI*)
    - Área: Instruments & Instrumentation
    - Posición: #9/57 (Q1)
    - Año: 2012

- Baladrón, C., Aguiar, J. M., Carro, B., Calavia, L., Cadenas, A., & Sanchez-Esguevillas, A. (2012). **Framework for Intelligent Service Adaptation to User's Context in Next Generation Networks**. *IEEE Communications Magazine*, 50(3), 18-25. ISSN: 0163-6804. Digital Object Identifier: 10.1109/MCOM.2012.6163578.
    - Índice de impacto: 3.661(*Journal Citation Report ISI*)
    - Área: Telecommunications
    - Posición: #3/77 (Q1)
    - Año: 2012
- Hernández, L., Baladrón, C., Aguiar, J. M., Calavia, L., Carro, B., Sánchez-Esguevillas, A., Cook, D. J., Chinarro, D., & Gómez, J. (2012). **A Study of the Relationship between Weather Variables and Electric Power Demand inside a Smart Grid/Smart World Framework**. *Sensors*, 12(9), 11571-11591. ISSN 1424-8220. Digital Object Identifier: 10.3390/s120911571.
    - Índice de impacto: 1.953 (*Journal Citation Report ISI*)
    - Área: Instruments & Instrumentation
    - Posición: #9/57 (Q1)
    - Año: 2012
- Baladrón, C., Aguiar, J. M., Cadenas, A., Calavia, L., Carro, B., & Sánchez, A. (2012). **User Oriented Environment for Management of Convergent Services.** *IEEE Communications Magazine*, 50(11), 142-149. ISSN: 0163-6804. Digital Object Identifier: 10.1109/MCOM.2012.6353694.
    - Índice de impacto: 3.661(*Journal Citation Report ISI*)
    - Área: Telecommunications
    - Posición: #3/77 (Q1)
    - Año: 2012
- Hernández, L., Baladrón, C., Aguiar, J. M., Calavia, L., Carro, B., Sánchez-Esguevillas, A., García, P., & Lloret, J. (2013). Experimental Analysis of the Input Variables' Relevance to Forecast Next Day's Aggregated Electric Demand Using Neural Networks. Energies, 6(6), 2927-2948. ISSN 1996-1073. Digital Object Identifier: 10.3390/en6062927.
    - Índice de impacto: 1.844 (*Journal Citation Report ISI*)
    - Área: Energy & Fuels
    - Posición: #38/81 (Q2)
    - Año: 2012

- Hernández, L., Baladrón, C., Aguiar, J. M., Calavia, L., Carro, B., Sánchez-Esguevillas, A., Sanjuán, J., González, A., & Lloret, J. (2013). Improved Short-Term Load Forecasting Based on Two-Stage Predictions with Artificial Neural Networks in a Microgrid Environment. *Energies*, 6(9), 4489-4507. ISSN 1996-1073. Digital Object Identifier: 10.3390/en6094489.
    - Índice de impacto: 1.844 (*Journal Citation Report ISI*)
    - Área: Energy & Fuels
    - Posición: #38/81 (Q2)
    - Año: 2012

8.2.2.2 Otros artículos científicos

Entre los artículos no indexados en el *Journal Citation Report ISI* de temática transversal a esta Tesis Doctoral se encuentran:

- Martínez, A., Baladrón, C., León, A., García, C., Calavia, L., Aguiar, J. M., & Caetano, J. (2009). New Business Models: User Generated Services. *IEEE Latin America Transactions*, 7(3), 395-399. ISSN: 1548-0992. Digital Object Identifier: 10.1109/TLA.2009.5336640.
- Calavia, L., Baladrón, C., Aguiar, J. M., Carro, B., & Sánchez-Esguevillas, A. (2011). QoS Traffic Mapping between WiMAX and DiffServ Networks. *Network Protocols and Algorithms*, 3(3), 67-79. ISSN 1943-3581. Digital Object Identifier: 10.5296/npa.v3i3.1063.

8.2.2.3 Conferencias

Con respecto a las Conferencias, la primera de las relacionadas a continuación derivada directamente de las investigaciones realizadas en esta Tesis, siendo el resto publicaciones relacionadas:

- Baladrón, C., Calavia, L., Aguiar, J. M., Carro, B., Sánchez Esguevillas, A., & Alonso, J. (2011). Sistema de Detección de Alarmas de Videovigilancia Basado en Análisis Semántico. *XXI Jornadas Telecom I+D*, Santander (España), 28, 29 y 30 Septiembre 2011. ISBN: 978-84-694-7808-0.
- Martínez, A., Baladrón, C., León, A., García, C., Caetano, J., Calavia, L., & Aguiar, J. M. (2008). Nuevos Modelos de Negocio: Servicios Generados por el Usuario. *XVIII Jornadas Telecom I+D*, Bilbao (España), 29-31 Octubre 2008. ISBN-13: 978-84-9860-135-0.

- Pérez, E., Calavia, L., Gobernado, J., Aguiar, J. M., Baladrón, C., & Carro, B. (2010). Plataforma para Búsqueda de Servicios en Entornos Móviles. *XX Jornadas Telecom I+D*, Valladolid (España), 27, 28 y 29 Septiembre 2010. ISBN 978-84-89900-38-7.

- Ruano, M. A., Baladrón, C., Aguiar, J. M., Calavia, L., Carro, B., & Sánchez Esguevillas, A. (2010). Servicios Innovadores Sobre Televisión Digital Terrestre. *XX Jornadas Telecom I+D*, Valladolid (España), 27, 28 y 29 Septiembre 2010. ISBN 978-84-89900-38-7.

- Calavia, L., Baladrón, C., Aguiar, J. M., Carro, B., & Sánchez, A. (2011). Mapeo de Calidad de Servicio entre redes DiffServ y WiMAX. *XXI Jornadas Telecom I+D*, Santander (España), 28, 29 y 30 Septiembre 2011. ISBN: 978-84-694-7808-0.

- Calavia, L., Baladrón, C., Aguiar, J. M., Carro, B., & García, M. (2011). Sistema de Búsqueda Semántica Basado en Triple Space. *XXI Jornadas Telecom I+D*, Santander (España), 28, 29 y 30 Septiembre 2011. ISBN: 978-84-694-7808-0.

- Baladrón, C., Aguiar, J. M., Calavia, L., Carro, B., Cadenas, A., de las Heras, R., & Sanchez-Esguevillas, A. (2011). Platform for ubiquitous mobile service composition, management and delivery. *Conference on Next Generation Web Services Practices (NWeSP), 2011 7th International*, 43–48, Salamanca (España), 19-21 Octubre 2011. ISBN: 978-1-4577-1125-1.

8.2.2.4 Notas de prensa

Como resultado de la Tesis se han publicado varias noticias tanto en periódicos digitales como prensa impresa. Se listan a continuación:

- Un proyecto internacional desarrolla un sistema de videovigilancia inteligente para grandes ciudades, *Tribuna de la Ciencia*, No. 61, Año 7, pp. 9, Marzo 2012.

- Un vigía todopoderoso de la Ciudad, *El Mundo (Castilla y León)*, No. 102, pp. 4-5 (Suplemento Innovadores), 28 de Mayo de 2012.

- Un equipo de la UVA desarrolla un sistema de videovigilancia inteligente, *El Norte de Castilla*, pp. 6-7, 11 de Junio de 2012.

8.2.2.5 Premios

- Premios Innovadores El Mundo, *El Mundo (Castilla y León)*, 15 de Febrero de 2013.

8.3 Conclusiones

Si bien a lo largo de los Capítulos precedentes se han ido presentado de forma exhaustiva las conclusiones derivadas de los diferentes análisis y estudios realizados, se exponen a continuación de forma resumida los principales resultados obtenidos a lo largo del presente trabajo.

El trabajo que se presenta a lo largo de esta Tesis Doctoral se ha realizado dentro del proyecto europeo CELTIC HuSIMS. El proyecto pretende profundizar en esta línea de investigación diseñando un sistema de vigilancia altamente distribuido que dé una respuesta tecnológica a los problemas asociados a un despliegue de gran envergadura en una ciudad, y que sea lo suficientemente flexible como para detectar alarmas de todo tipo, desde accidentes de tráfico en carreteras a actos de vandalismo.

Para maximizar la escala a la que se puede efectuar el despliegue, HuSIMS trabaja principalmente sobre la idea de minimizar el coste de los sensores inteligentes, minimizar el ancho de banda utilizado por los mismos, facilitar su conectividad e instalación por medio de tecnologías avanzadas sin hilos, y ofrecer detección de alarmas automática en tiempo real con información de alto nivel útil para los equipos de emergencia. El resultado es que es posible adquirir más sensores para llegar a más lugares debido a su bajo precio y su facilidad de conexión, que el coste en comunicaciones se mantiene controlado y que es posible confiar en el sistema para llevar a cabo el análisis de la situación a lo largo de toda la base de sensores.

Para lograr este objetivo, el paradigma en el que se basa HuSIMS consiste en utilizar un procesado sencillo en las cámaras que únicamente detecte objetos en movimiento, transmitir el resultado de este procesado especificando una serie de parámetros de los objetos detectados (posición, tamaño, velocidad, dirección, etc.), y trasladar gran parte de la inteligencia al centro de control, donde se efectuará un análisis combinado de dichos parámetros en busca de comportamientos anómalos.

Con la realización de esta Tesis Doctoral se trata de diseñar y desarrollar un sistema apto para caracterizar diferentes escenarios aplicado para la detección de

anomalías de forma automatizada en videovigilancia. Este mecanismo es adecuado para despliegues en espacios inteligentes, capaces de trabajar con cámaras pequeñas y baratas, con requerimientos de ancho de banda reducidos y procesamiento optimizado.

El enfoque seguido por el sistema propuesto en esta Tesis Doctoral se basa en un esquema de procesamiento de tres etapas. Primero, la detección de objetos en movimiento en la propia cámara para evitar el envío de datos de video grandes, y al mismo tiempo mantener baja la potencia de procesamiento requerida por las cámaras para evitar la aplicación de algoritmos complejos. En segundo lugar, la construcción, de forma automática, de un modelo del recorrido de los objetos en movimiento en las escenas utilizando los parámetros identificados por las cámaras. Tercero, el razonado semántico sobre los parámetros del modelo de rutas y los objetos en movimiento para identificar las alarmas a nivel conceptual, es decir, no sólo la detección de una situación inusual sino también la identificación de la naturaleza de ese evento (un accidente automovilístico, un incendio, una intrusión, etc.). El objetivo es que cuando se dispare una alarma exista gran cantidad de información disponible relativa a la emergencia, en un formato de muy alto nivel directamente interpretable por un operador humano (colisión, atropello, disturbios, incendio, etc.), no un simple aviso de que ha ocurrido un suceso no habitual. Esta información adicional puede resultar muy ventajosa a la hora de ahorrar tiempo durante la gestión de la alarma, puesto que el operador humano sabe inicialmente a qué se enfrenta y se evita la inspección inicial de los archivos de video para identificar la situación. Incluso más aún, es posible automatizar en cierta medida las reacciones a diferentes tipos de alarmas, distribuyendo de forma inteligente las mismas a los operadores implicados.

La mayoría de los estudios dedicados a los sistemas de vigilancia se basan en el análisis estadístico de características de la imagen o en la identificación de objetos prefijados. La detección estadística de alarmas solamente detecta comportamientos anormales, entendiendo anormales como las cosas que no suceden "por lo general", de acuerdo a ciertos criterios matemáticos. Los sistemas no modificables trabajan sobre la base de un motor de reglas de codificación fija y no se suelen emplear las tecnologías semánticas formales. Un ejemplo de este tipo de sistemas de vigilancia son los sistemas de gestión de tráfico que operan a través de la identificación de los automóviles en la imagen.

Como alternativa al análisis estadístico, la solución propuesta en esta Tesis Doctoral presenta la ventaja de enviar información enriquecida sobre las alarmas

identificadas, totalmente controladas ya que en cualquier momento la ontología puede ser modificada por un experto para variar las condiciones de la alarma. Las *Smart Cities* son capaces de aprovechar esta ventaja, ya que esta información enriquecida se puede utilizar para tramitar las respuestas automáticas a las alarmas.

La aplicación de tecnologías semánticas formales es mucho más fácil en entornos donde las cámaras envían videos de alta resolución a un centro de control en el que se le aplican algoritmos de identificación de objetos complejos. En el caso de esta Tesis Doctoral el desafío es cómo identificar conceptos en la imagen mediante el análisis únicamente de los patrones de movimiento y los parámetros simples de objetos móviles desconocidos. Esta limitación se impone por el hecho de que HuSIMS opera con un gran número de cámaras: la incrustación de procesadores de gran alcance en todas ellas es demasiado cara, y el envío de toda la señal de video al centro de control requiere de enormes cantidades de ancho de banda. Los sensores en HuSIMS realizan un análisis simple de la imagen para identificar objetos en movimiento y sus parámetros.

Esta Tesis se centra en la aplicación de razonamiento semántico formal para sustituir las tareas de etiquetado de objetos mediante procesamiento de imagen, la introducción de una ontología persistente que modela un dominio de conocimientos y la aplicación de razonamiento semántico sobre ella para identificar conceptos empleando los parámetros de movimiento enviados por los sensores.

Aunque en ciertas ocasiones se necesitan detalles adicionales, como la apariencia, para determinar actividades complejas (para diferenciar, por ejemplo un motorista de una persona corriendo, sólo el patrón de movimiento no es suficiente), este enfoque ofrece una serie de ventajas:

- Admite un gran número de sensores baratos.
- Fácil escalabilidad e integración de los datos de entrada nuevos.
- Reduce la carga de trabajo de los operarios encargados de hacer cumplir la ley gracias al procesamiento automático.
- El sistema "habla" un lenguaje conceptual abstracto fácil de entender por los operadores humanos.
- Es adaptable a diferentes dominios mediante un simple cambio / adición de la ontología y reglas apropiadas.

8.4 Líneas Futuras

Salvo la captación de información visual y su procesado para determinar los principales parámetros de los objetos en movimiento llevadas a cabo por un sensor visual comercial, todos los diseños arquitecturales y mecanismos propuestos a lo largo de los capítulos previos se han implementado y son aportaciones de esta Tesis Doctoral.

Sin embargo, dada la gran cantidad de campos de aplicación de la caracterización de escenarios y la evolución constante de los mecanismos utilizados para tal fin, parte de la metodología presentada en este trabajo puede ser extendida y generalizada, indicando algunas posibles líneas de interés:

- Integración de múltiples cámaras con un procesamiento coordinado.

  La tecnología presentada centra su procesamiento en la imagen procedente de una única cámara, sin embargo, cada vez son más crecientes los entornos con grandes despliegues de cámaras baratas. En este sentido se contemplan dos alternativas. Por un lado, definir la misma escena utilizando la imagen procedente de cámaras situadas en distintas localizaciones y ángulos con lo que se consiguen imágenes desde distintas perspectivas que pueden ser procesadas de manera coordinada. Como alternativa, imágenes provenientes de cámaras que enfocan a escenas diferentes pero dentro de una misma área. En este caso los objetos cuando desaparecen del campo de visión de una cámara, aparecen en la cámara vecina. La fusión de los resultados de cada procesado individual o el procesado teniendo en cuenta las identificaciones de las cámaras que han capturado al objeto previamente pueden mejorar los resultados y conseguir identificaciones coherentes de áreas amplias.

  En cualquiera de los casos el procesado conjunto de la información proveniente de las múltiples cámaras podría complementar el sistema.

- Validación del sistema cuando se dispone de información complementaria procedente de sensores no visuales.

  La arquitectura y la implementación están diseñadas para aceptar datos procedentes de sensores alternativos a los visuales. El desarrollo está preparado para la que se incluyan en la ontología y solo es necesario tenerlo en cuenta en el diseño de las reglas para utilizar, en el proceso de inferencia, la información que aportan. Hasta ahora solo se ha podido trabajar con

imágenes por lo que en los procesos de verificación no se ha validado está parte del sistema quedando pendiente para trabajos futuros.

- Inclusión de información de audio.

  Una de las tendencias actuales es incorporar información de audio como complemento a la información visual y a la procedente de otros sensores distribuidos a lo largo de la escena. La inclusión de los sonidos del entorno como complemento del mecanismo propuesto de caracterización requiere un procesado previo que no es necesario para otros sensores por lo que no se ha incluido en el diseño actual. Sin embargo, una vez realizado el pre-procesado de la información de audio, el diseño propuesto es fácilmente adaptable, no sólo para incluir esta información adicional, sino también para identificar los distintos ruidos en función de la información recibida por el micrófono.

- Generación automática de ontologías y reglas.

  La adaptación del sistema a diferentes dominios, aunque sencilla, requiere de un operador humano que adapte el modelo ontológico y las reglas de forma manual. Para complementar el sistema y facilitar la adaptación se podría investigar un sistema para la generación automática de modelos de conocimiento semánticos basado en el análisis inteligente de las señales de video durante el periodo de aprendizaje.

  Este sistema podría tomar la información de señales de video y la procesarla con una cascada de algoritmos inteligentes para extraer la ontología, permitiendo obtener conocimiento abstracto conceptual de su experiencia visual. La intervención humana se limitaría entonces a etiquetar los conceptos para conseguir la definición semántica.

  Los algoritmos desarrollados llevarían a cabo todas las tareas necesarias para llegar a un modelo semántico de la escena visualizada. Utilizando el procesamiento de video de bajo nivel ya realizado (para extraer información atómica de la escena, como los parámetros de movimiento, objetos reconocidos, posiciones, etc.) se conseguiría la construcción de la ontología, con varios pasos inteligentes en el medio, como la agrupación de objetos reconocidos en las clases, la extracción de hechos y de aprendizaje inductivo. Por ejemplo, en una cámara dedicada al control del tráfico, la señal de video se procesa primero para extraer los parámetros de movimiento y la

trayectoria de los objetos, que luego se agruparon utilizando su tamaño, velocidad y características de posición para identificar las clases (primeramente conceptos sin etiquetar y después, coches, peatones, caminos, etc., definidos por el operador). Después se identificarían los conjuntos de hechos individuales (por ejemplo: "coche1 está en carretera2") para generalizarlos como axiomas ("los coches están en las carreteras"), que procesados, construirían dicho modelo de conocimiento.

- Aplicación de la metodología semántica a otros campos de aplicación distintos a la caracterización de escenas.

  El sistema propuesto utiliza ontologías persistentes para la caracterización de escenas utilizando imágenes de video y la información procedente de sensores que sirven como complemento. Sin embargo, se consigue aplicar la misma metodología de modelado semántico, sin utilizar la información visual, por ejemplo, para otros campos de aplicación como pueden ser la gestión de tráfico de red. Utilizando ontologías persistentes es posible hacer un modelado formal de la red y del tráfico de la misma para identificar congestión, perdida de paquetes, etc.

- Mejora e inclusión de nuevas estadísticas sobre los escenarios.

  En el sistema presentado se ha utilizado la identificación de objetos para determinar situaciones de alertas pero en ocasiones puede ser interesante utilizar la identificación de los objetos de los videos para obtener estadísticas. El sistema actual incluye estadísticas para el control y gestión del tráfico como identificar el número total de vehículos que han circulado por una vía y la velocidad media de los objetos que circulan por la misma. Sin embargo puede ser interesante, utilizando la misma base de datos, almacenar nuevos valores de interés para posteriormente ofrecérselos al usuario en forma de informes. Completando el sistema se podría permitir al usuario seleccionar distintas opciones como comparar los valores obtenidos para distintos momentos del día, seleccionar los valores obtenidos entre dos fechas concretas, etc.

GLOSARIO DE ABREVIATURAS

A

ABox	Assertional Box
ADI	Alternative Dunn Index
AI	Artificial Intelligence
API	Application Programming Interface
A.U.	Arbitrary Units

B

BIC	Bayesian Information Criterion

C

CAGR	Compound Annual Growth Rate
CCTV	Closed Circuit Television
CE	Classification Entropy
CIF	Common Intermediate Format
CMOS	Complementary Metal-Oxide-Semiconductor
CP	Convex Projections
CRF	Conditional Random Field
CVER	Continuous Visual Event Recognition

D

DAML	DARPA Agent Markup Language
DBSCAN	Density-based spatial clustering of applications with noise
DI	Dunn's Index
DOM	Document Object Model
DSP	Digital Signal Processor
DTW	Dynamic Time Warping

E

EM	Expectation–maximization
E/R	Entity/Relationship

F

FCM	Fuzzy c-Means
FEDER	Fondo Europeo de Desarrollo Regional
FOV	Field Of View
FPGA	Field Programmable Gate Array
fps	Frames Per Second

G

GG	Gath-Geva
GK	Gustafson-Kessel
GM	Gaussian Model
GMM	Gaussian Mixture Model
GPS	Global Positioning System

H

HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
HuSIMS	Human Situation Monitoring System

I

ICT	Information and Communication Technologies

J

JAXB	Java Architecture for XML Binding

L

LCSS	Longest Common Subsequence

M

MCR	MATLAB Compiler Runtime
MIT	Massachusetts Institute of Technology

O

OCL	Object Constraint Language
OIL	Ontology Interface Layer
OWL	Web Ontology Language

P

PC	Partition Coeficient
PCA	Principal Components Analysis
pdf	Probability Density Function
PDM	Point-Distribution Model
PLC	Power Line Communications

Q

QoS	Quality of Service

R

RDBMS	Relational DataBase Management System
RDF	Resource Description Framework
ROC	Receiver Operating Characteristic

S

S	Separation Index
SAX	Simple API for XML

SC	Partition Index
SOM	Self-Organizing Map
SON	Self-Organizing Network
SPARQL	SPARQL Protocol and RDF Query Language
Stax	Streaming API for XML
SWRL	Semantic Web Rule Language

T

TBox	Terminological Box
TSC	Tightness and Separation Criterion

U

UML	Unified Modelling Language
URI	Uniform Resource Identifier

V

VGA	Video Graphic Array
VMD	Video Motion Detection
VSAM	Video Surveillance and Monitoring
VQ	Vector Quantization

W

W3C	World Wide Web Consortium

X

XB	Xie and Beni's Index
XML	eXtensible Markup Language
XSD	XML Schema Definition

BIBLIOGRAFÍA

[1] Williamson, A., Lombardi, D. A., Folkard, S., Stutts, J., Courtney, T. K., & Connor, J. L. (2011). The link between fatigue and safety. *Accident Analysis & Prevention, 43*(2), 498-515.

[2] Makris, D., & Ellis, T. (2005). Learning semantic scene models from observing activity in visual surveillance. *IEEE Transactions on Systems, Man and Cybernetics B, 35*(3), 397–408.

[3] Piciarelli, C., & Foresti, G. L. (2006). On-line trajectory clustering for anomalous events detection. *Pattern Recognition Letters*, *27*(15), 1835–1842.

[4] Li, X., Hu, W., & Hu, W. (2006). A coarse-to-fine strategy for vehicle motion trajectory clustering. In *18th International Conference on Pattern Recognition (ICPR 2006).* Hong Kong, China, 22-24 August 2006 (Vol. 1, pp. 591-594).

[5] Morris, B. T., & Trivedi, M. M. (2008). Learning, modeling, and classification of vehicle track patterns from live video. *Intelligent Transportation Systems, IEEE Transactions on, 9*(3), 425-437.

[6] Foresti, G. L., Micheloni, C., Snidaro, L., Remagnino, P., & Ellis, T. (2005). Active Video-Based Surveillance System: The Low-Level Image and Video Processing Techniques Needed for Implementation. *IEEE Signal Processing Magazine*, *22*(2), 25-37.

[7] Hu, W., Tan, T., Wang, L., & Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors. *Transactions on Systems, Man, and Cybernetics, part C: Applications and Reviews, 34*(3), 334–352.

[8] Rota, N., & Thonnat, M. (2000). Video Sequence Interpretation for Visual Surveillance. In *IEEE Workshop Visual Surveillance*. Dublin, Ireland, 1 July 2000 (pp. 325-332).

[9] Assfalg, J., Bertini, M., Colombo, C., Del Bimbo, A., & Nunziati, W. (2003). Semantic annotation of soccer videos: automatic highlights identification. *Computer Vision and Image Understanding, 92*(2), 285-305.

[10] Aguilera, J., Thirde, D., Kampel, M., Borg, M., Fernandez, G., & Ferryman, J. (2006). Visual surveillance for airport monitoring applications. In *11th*

Computer Vision Winter Workshop. Telč, Czech Republic, 6-8 February 2006 (pp. 6-8).

[11] Geiger, A., Lauer, M., & Urtasun, R. (2011). A generative model for 3d urban scene understanding from movable platforms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Colorado Springs, CO, USA, 20-25 June 2011 (pp. 1945-1952).

[12] Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., ... & Thrun, S. (2008). Junior: The stanford entry in the urban challenge. *Journal of Field Robotics, 25*(9), 569-597.

[13] Scharstein, D., & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision, 47*(1-3), 7-42.

[14] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 32*(9), 1627-1645.

[15] Ess, A., Leibe, B., Schindler, K., & Van Gool, L. (2009). Moving obstacle detection in highly dynamic scenes. In *IEEE International Conference on Robotics and Automation (ICRA'09).* Kobe, Japan, 12-17 May 2009 (pp. 56-63).

[16] Gavrila, D. M., & Munder, S. (2007). Multi-cue pedestrian detection and tracking from a moving vehicle. *International journal of computer vision, 73*(1), 41-59.

[17] Wojek, C., Roth, S., Schindler, K., & Schiele, B. (2010). Monocular 3d scene modeling and inference: Understanding multi-object traffic scenes. In *Computer Vision–ECCV 2010.* Crete, Greece, 5-11 September 2010 (pp. 467-481).

[18] Wojek, C., & Schiele, B. (2008). A dynamic conditional random field model for joint labeling of object and scene classes. In *Computer Vision–ECCV 2008.* Marseille, France, 12-18 October 2008 (Vol. 5305, pp. 733-747).

[19] Sturgess, P., Alahari, K., Ladicky, L., & Torr, P. (2009). Combining appearance and structure from motion features for road scene understanding. In *British Machine Vision Conference (BMVC).* London, UK, 7-10 September 2009.

[20] Technavio Analytic Forecast. *Global Video Surveillance Market 2011–2015*. Disponible online: http://www.technavio.com/content/global-video-surveillance-market-2011–2015 (Última visita: Abril 2013).

[21] DataMonitor. (July 2004). *Global digital video surveillance markets: Finding future opportunities as analog makes way for digital*. Market research report.

[22] Skraba, P., & Guibas, L. (2007). Energy efficient intrusion detection in camera sensor networks. In *Distributed Computing in Sensor Systems*. Santa Fe, NM, USA, 18-20 June 2007 (Vol. 4549, pp. 309-323).

[23] Ferryman, J. M., Maybank, S. J., & Worrall, A. D. (2000). Visual surveillance for moving vehicles. *International Journal of Computer Vision*, *37*(2), 187–19731.

[24] Bodsky, T., Cohen, R., Cohen-Solal, E., Gutta, S., Lyons, D., Philomin, V., & Trajkovic, M. (2001). Visual surveillance in retail stores and in the home. In P. Remagnino, G. A. Jones, N. Paragios & C. S. Regazzoni (Eds.), *Video-based Surveillance Systems.* (pp. 51-61). Springer US.

[25] Liu, C. B., & Ahuja, N. (2004). Vision based fire detection. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004).* Cambridge, UK, 23-26 August 2004 (Vol. 4, pp. 134-137).

[26] Xu, M., Orwell, J., Lowey, L., & Thirde, D. (2005). Architecture and algorithms for tracking football players with multiple cameras. *IEE Proceedings-Vision, Image and Signal Processing, 152*(2), 232-241.

[27] Remagnino, P., Velastin, S. A., Foresti, G. L., & Trivedi, M. (2007). Novel concepts and challenges for the next generation of video surveillance systems. *Machine Vision and Applications, 18*(3), 135-137.

[28] Tian, Y. L., Brown, L., Hampapur, A., Lu, M., Senior, A., & Shu, C. F. (2008). IBM smart surveillance system (S3): event based video surveillance system with an open and extensible framework. *Machine Vision and Applications*, *19*(5-6), 315-327.

[29] Nghiem, A. T., Bremond, F., Thonnat, M., & Valentin, V. (2007). ETISEO, performance evaluation for video surveillance systems. In *IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS 2007).* London, UK, 5-7 September 2007 (pp. 476-481).

[30] Oh, S., Hoogs, A., Perera, A., Cuntoor, N., Chen, C. C., Lee, J. T., ... & Desai, M. (2011). A large-scale benchmark dataset for event recognition in

surveillance video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* Colorado Springs, CO, USA, 20-25 June 2011(pp. 3153-3160).

[31] Gorelick, L., Blank, M., Shechtman, E., Irani, M., & Basri, R. (2007). Actions as space-time shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 29*(12), 2247-2253.

[32] Laptev, I., & Pérez, P. (2007). Retrieving actions in movies. In *IEEE 11th International Conference on Computer Vision (ICCV 2007).* Rio de Janeiro, Brazil, 14-21 October 2007 (pp. 1-8).

[33] Liu, J., Luo, J., & Shah, M. (2009). Recognizing realistic actions from videos "in the wild". In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009).* Miami, FL, USA, 20-25 June 2009 (pp. 1996-2003).

[34] Lloret, J., García, M., Bri, D., & Sendra, S. (2009). A Wireless Sensor Network Deployment for Rural and Forest Fire Detection and Verification. *Sensors*, *9*(11), 8722-8747.

[35] Lloret, J., Bosch Roig, I., Sendra Compte, S., & Serrano Cartagena, A. (2011). A wireless Sensor Network that use Image Processing for Vineyard Monitoring. *Sensors*, *11*(6), 6165-6196.

[36] Stauffer, C., & Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* Fort Collins, CO, USA, 23-25 June 1999 (Vol. 2).

[37] Behrad, A., Shahrokni, A., Motamedi, S. A., & Madani, K. (2001). A robust vision-based moving target detection and tracking system. In *the Proceeding of Image and Vision Computing Conference.* Dunedin, New Zealand, 26-28 November 2001.

[38] Lipton, A. J., Fujiyoshi, H., & Patil, R. S. (1998). Moving target classification and tracking from real-time video. In *Proceedings of the Fourth IEEE Workshop on Applications of Computer Vision (WACV'98).* Princeton, NJ, USA, 19-21 October 1998 (pp. 8-14).

[39] Meyer, D., Psl, J., & Niemann, H. (1998). Gait classification with HMMs for trajectories of body parts extracted by mixture densities. In *British Machine Vision Conference (BMVC).* Southampton, UK, 10 September 1998 (pp. 459-468).

[40] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering, 82*(1), 35-45.

[41] Isard, M., & Blake, A. (1998). Condensation—conditional density propagation for visual tracking. *International Journal of Computer Vision, 29*(1), 5-28.

[42] Ghahramani, Z. (1997). Learning dynamic Bayesian networks. In *Adaptive processing of sequences and data structures*, *International Summer School on Neural Networks.* Salerno, Italy, 6-13 September 1997 (Vol. 1387, pp. 168-197).

[43] Wren, C. R., Azarbayejani, A., Darrell, T., & Pentland, A. P. (1997). Pfinder: Real-time tracking of the human body. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 19*(7), 780-785.

[44] Haritaoglu, I., Harwood, D., & Davis, L. S. (1998). W^4: Who? When? Where? What? A real time system for detecting and tracking people. In *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition.* Nara, Japan, 14-16 April 1998 (pp. 222-227).

[45] McKenna, S. J., Jabri, S., Duric, Z., Rosenfeld, A., & Wechsler, H. (2000). Tracking groups of people. *Computer Vision and Image Understanding, 80*(1), 42-56.

[46] Malik, J., Russell, S., Weber, J., Huang, T., & Koller, D. (1994). A machine vision based surveillance system for California roads. PATH project MOU-83 Final Report, University of California.

[47] Freedman, D., & Zhang, T. (2004). Active contours for tracking distributions. *Image Processing, IEEE Transactions on, 13*(4), 518-526.

[48] Yilmaz, A., Li, X., & Shah, M. (2004). Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 26*(11), 1531-1536.

[49] Paragios, N., & Deriche, R. (2000). Geodesic active contours and level sets for the detection and tracking of moving objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22*(3), 266-280.

[50] Aggarwal, J. K., Cai, Q., Liao, W., & Sabata, B. (1998). Nonrigid motion analysis: Articulated and elastic motion. *Computer Vision and Image Understanding, 70*(2), 142-156.

[51] Gardner, W. F., & Lawton, D. T. (1996). Interactive model-based vehicle tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 18*(11), 1115-1121.

[52] Kuno, Y., Watanabe, T., Shimosakoda, Y., & Nakagawa, S. (1996). Automated detection of human for visual surveillance system. In *Proceedings of the 13th International Conference on Pattern Recognition.* Vienna, Austria, 25-29 August 1996 (Vol. 3, pp. 865-869).

[53] Collins, R. T., Lipton, A., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., ... & Wixson, L. (2000). A system for video surveillance and monitoring. *Pittsburg: Carnegie Mellon University, the Robotics Institute*, 2.

[54] Cutler, R., & Davis, L. S. (2000). Robust real-time periodic motion detection, analysis, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *22*(8), 781-796.

[55] Lipton, A. J. (1999). *Local application of optic flow to analyse rigid versus non-rigid motion.* In *IEEE International Conference on Computer Vision Workshop Frame-Rate.* Kerkyra, Greece, 20-27 September 1999.

[56] Sivic, J., Russell, B., Efros, A., Zisserman, A., & Freeman, W. (2005). Discovering objects and their location in images. In *ICCV International Conference on Computer Vision*. Beijing, China, 17-21 October 2005 (Vol. 1, pp. 370–378).

[57] Torralba, A., Murphy, K. P., Freeman, W. T., & Rubin, M. A. (2003). Context-based vision system for place and object recognition. *Conference on Computer Vision ICCV International*. Nice, France, 13-16 October 2003 (Vol. 1, pp. 273–280).

[58] Tan, T. N., Sullivan, G. D., & Baker, K. D. (1998). Model-based localization and recognition of road vehicles. *International Journal of Computer Vision*, *29*(1), 22–25.

[59] Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., & Poggio, T. (2007). Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(3), 411–426.

[60] Chen, X., & Zhang, C. (2006). An Interactive Semantic Video Mining and Retrieval Platform – Application in Transportation Surveillance Video for Incident. In *The 2006 IEEE International Conference on Data Mining (ICDM)*. Hong Kong, China, 18-22 December 2006 (pp. 129-138).

[61] Raman, R. M., Chandran, M. S., & Vinotha, S. R. (2011). Motion Based Security Alarming System for Video Surveillance. In *International Conference on Computational Techniques and Artificial Intelligence (ICCTAI'2011).* Pattaya, Thailand, 7-8 October 2011.

[62] SanMiguel, J. C., & Martínez, J. M. (2012). A semantic-based probabilistic approach for real-time video event recognition. *Computer Vision and Image Understanding*, 116(9), 937-952.

[63] Craven, M., & Kumilien, J. (1999). Constructing Biological Knowledge Bases by Extracting Information from Text Sources. In *7th International Conference on Intelligent Systems for Molecular Biology.* Heidelberg, Germany, 6-10 August 1999.

[64] Nguyen, N. T., Bui, H. H., Venkatsh, S., & West, G. (2003). Recognizing and monitoring high-level behaviors in complex spatial environments. In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* Madison, WI, USA, 16-22 June 2003 (Vol. 2, pp. II-620).

[65] Ivanov, Y. A., & Bobick, A. F. (2000). Recognition of visual activities and interactions by stochastic parsing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22*(8), 852-872.

[66] Remagnino, P., Shihab, A. I., & Jones, G. A. (2004). Distributed intelligence for multi-camera visual surveillance. *Pattern recognition, 37*(4), 675-689.

[67] Ko, M. H., West, G., Venkatesh, S., & Kumar, M. (2008). Using dynamic time warping for online temporal fusion in multisensor systems. *Information Fusion, 9*(3), 370-388.

[68] Kim, Y. T., & Chua, T. S. (2005). Retrieval of news video using video sequence matching. In *Proceedings of the 11th International Multimedia Modelling Conference (MMM 2005).* Melbourne, Australia, 12-14 January 2005 (pp. 68-75).

[69] Morris, B., & Trivedi, M. M. (2009). Learning Trajectory Patterns by Clustering: Experimental Studies and Comparative Evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition*. Miami, FL, USA, 20-25 June 2009 (pp. 312–319).

[70] Zhang, Z., Huang, K., & Tan, T. (2006). Comparison of Similarity Measures for Trajectory Clustering in Outdoor Surveillance Scenes. In *18th International*

Conference on Pattern Recognition (ICPR 2006). Hong Kong, China, 20-24 August 2006 (Vol. 3, pp. 1135-1138).

[71] Sacchi, C., Regazzoni, C., & Vernazza, G. (2001). A neural network-based image processing system for detection of vandal acts in unmanned railway environments. In *Proceedings of the 11th International Conference on Image Analysis and Processing.* Palermo, Italy, 26-28 September 2001 (pp. 529-534).

[72] Baladrón, C., Aguiar, J. M., Calavia, L., Carro, B., Sánchez-Esguevillas, A., & Hernández, L. (2012). Performance study of the application of artificial neural networks to the completion and prediction of data retrieved by underwater sensors. *Sensors, 12*(2), 1468-1481.

[73] Cristani, M., & Cuel, R. (2005). A survey on ontology creation methodologies. *International Journal on Semantic Web & Information Systems*, *1*(2), 49-69.

[74] Vargas-Vera, M., Domingue, J., Kalfoglou, Y., Motta, E., & Buckingham, S. (2001). Template-Driven Information Extraction for Populating Ontologies. *In proceedings of the Workshop Ontology Learning IJCAI.* Seattle, WA, USA, 4 August 2001.

[75] McKenna, S. J., & Nait Charif, H. (2004). Summarising contextual activity and detecting unusual inactivity in a supportive home environment. *Pattern Analysis and Applications, 7*(4), 386–401.

[76] Tsow, F., Forzani, E., Rai, A., Rui Wang, Tsui, R., Mastroianni, S., Knobbe, C., Gandolfi, A. J., & Tao, N. J. (2009). A Wearable and Wireless Sensor System for Real-Time Monitoring of Toxic Environmental Volatile Organic Compounds. *IEEE Sensors Journal*, *9*(12), 1734-1740.

[77] Xinguo Yu. (2008). Approaches and principles of fall detection for elderly and patient. In *10th International Conference on e-health Networking, Applications and Services (HealthCom 2008).* Singapore, 7-9 July 2008 (pp. 42-47).

[78] Tung, F., Zelek, J. S., & Clausi, D. A. (2010). Goal-based trajectory analysis for unusual behaviour detection in intelligent surveillance. *Image and Vision Computing*, *29*(4), 230-240.

[79] Zhang, C., Chen, X., Zhou. L., & Chen, W. (2009). Semantic retrieval of events from indoor surveillance video databases. *Pattern Recognition Letters*, *30*(12), 1067-1076.

[80] Fensel, D. (2000). *Ontologies: A silver bullet for knowledge management and electronic commerce*. Springer.

[81] Buitelaar, P., Cimiano, P., & Magnini, B. (2005). *Ontology Learning from Text: Methods, Evaluation and Applications. Frontiers in Artificial Intelligence and Applications.* IOS Press.

[82] Whitehouse, K., Liu, J., & Zhao, F. (2006). Semantic Streams: a Framework for Composable Inference over Sensor Data. In *The Third European Workshop on Wireless Sensor Networks (EWSN), Springer-Verlag Lecture Notes in Computer Science*. Zurich, Switzerland, 13-15 February 2006 (pp. 5-20).

[83] Arslan, U., Emin Dönderler, M., Saykol, E., Ulusoy Ö., & Güdükbay, U. (2002). A Semi-Automatic Semantic Annotation Tool for Video Databases. In *Proceedings of the Workshop on Multimedia Semantics*. Milovy, Czech Republic, 24-29 November 2002 (pp. 1-10).

[84] Nakamura, E. F., Loureiro, A. A. F., & Frery, A. C. (2007). Information fusion for Wireless Sensor Networks: Methods, models and classifications. *ACM Computing Surveys*, *39*(3).

[85] Friedlander, D., & Poha, S. (2002). Semantic information fusion for coordinated signal processing in mobile sensor networks. *International Journal of High Performance Computing Applications*, *16*(3), 235–241.

[86] Marraud, D., Cepas, B., & Reithler, L. (2009). Semantic Browsing of Video Surveillance Databases through Online Generic Indexing. In *Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2009), IEEE Conference on Advanced Video and Signal Based Surveillance*. Como, Italy, 30 August – 2 September 2009 (pp. 1-8).

[87] Francois, A. R, Nevatia, R., Hobbs, J., & Bolles, R. C. (2005).VERL: An Ontology Framework for Representing and Annotating Video Events. *IEEE MultiMedia*, *12*(4), 76-86.

[88] Poppe, C., Martens, G., De Potter, P., & De Walle, R. V. (2012). Semantic web technologies for video surveillance metadata. *Multimedia Tools and Applications*, *56*(3), 439-467.

[89] Faure, D., & N'Edellec, C. (1998). ASIUM: Learning sub-categorization frames and restrictions of selection. In *Proceedings of the 10th Conference on Machine Learning– Workshop on Text Mining*. Chemnitz, Germany, 21-23 April 1998.

[90] Tanev, H., & Magnini, B. (2006). Weakly Supervised Approaches for Ontology Population. In *Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006)*. Trento, Italy, 3-7 April 2006 (pp. 129-143).

[91] Cimiano, P., & Völker. J. (2005). Towards large-scale,open-domain and ontology-based named entity classification. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2005)*. Borovets, Bulgaria, 24 September 2005 (pp. 166–172).

[92] Pavlidis, I., Morellas, V., Tsiamyrtzis, P., & Harp, S. (2001). Urban surveillance systems: from the laboratory to the commercial world. *Proceedings of the IEEE, 89*(10), 1478-1497.

[93] Cai, Q., & Aggarwal, J. K. (1996). Tracking human motion using multiple cameras. In *Proceedings of the 13th International Conference on Pattern Recognition.* Vienna, Austria, 25-29 August 1996 (Vol. 3, pp. 68-72).

[94] Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., & Shafer, S. (2000). *Surveillance* Multi-camera multi-person tracking for easyliving. In *Proceedings of the Third IEEE International Workshop on Visual Surveillance.* Dublin, Ireland, 1 July 2000 (pp. 3-10).

[95] Javed, O., Khan, S., Rasheed, Z., & Shah, M. (2000). Camera handoff: tracking in multiple uncalibrated stationary cameras. In *Proceedings of the Workshop on Human Motion.* Austin, TX, USA, 7-8 December 2000 (pp. 113-118).

[96] Baladrón, C., Cadenas, A., Aguiar, J. M., Carro, B., & Sánchez-Esguevillas, A. (2010). Multi-Level context management and inference framework for smart telecommunication services. *Journal of Universal Computer Science*, 16, 1973–1991.

[97] Lo, B. P. L., Sun, J., & Velastin, S. A. (2003). Fusing visual and audio information in a distributed intelligent surveillance system for public transport systems. *Acta Automatica Sinica, 29*(3), 393-407.

[98] Velastin, S. A., Khoudour, L., Lo, B. P. L., Sun, J., & Vicencio-Silva, M. A. (2004). PRISMATICA: A multi-sensor surveillance system for public transport networks. In *12th IEE International Conference on Road Transport Information and Control (RTIC 2004).* London, UK, 20-22 April 2004 (pp. 19-25).

[99] Dee, H. M., Fraile, R., Hogg, D. C., & Cohn, A. G. (2008). Modelling scenes using the activity within them. In *Proceedings of the International Conference on Spatial Cognition VI: learning, reasoning, and talking about space*. Freiburg, Germany, 15-19 September 2008 (pp. 394-408).

[100] Mallot, H. A., Biilthoff, H. H., Little, J. J., & Bohrer, S. (1991). Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological Cybernetics*, *64*, 177-185.

[101] Roberts, L. (2004). History of Video Surveillance and CCTV. *WE C U Surveillance.* Disponible online: http://www.wecusurveillance.com/cctvhistory (Última visita: Abril 2013).

[102] Belbachir, A. N., & Göbel, P. M. (2010). Smart Cameras: A Historical Evolution. In A. N. Belbachir (Ed.), *Smart Cameras* (pp. 3-17). Springer US.

[103] Thompson, M. (1985). Maximizing CCTV Manpower. *Security World, 22*(6), 41-44.

[104] Rodger, R. M., Grist, I. J., & Peskett, A. O. (1994). Video motion detection systems: a review for the nineties. In *Proceedings of the Security Technology. 28th Annual 1994 International Carnahan Conference on Institute of Electrical and Electronics Engineers.* Albuquerque, NM, USA, 12-14 October 1994 (pp. 92-97).

[105] Michalopoulos, P., Wolf, B., & Benke, R. (1990). Testing and Field Implementation of the Minnesota Video Detection System (AUTOSCOPE). *Transportation Research Record*, 1287, 176-184.

[106] Kaneda, K., Nakamae, E., Takahashi, E., & Yazawa, K. (1990). An unmanned watching system using video cameras. *Computer Applications in Power, IEEE, 3*(2), 20-24.

[107] Hampapur, A., Brown, L., Connell, J., Ekin, A., Haas, N., Lu, M., ... & Pankanti, S. (2005). Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking. *Signal Processing Magazine, IEEE, 22*(2), 38-51.

[108] Rinner, B., & Wolf, W. (2008). An introduction to distributed smart cameras. *Proceedings of the IEEE, 96*(10), 1565-1575.

[109] Rinner, B., Winkler, T., Schriebl, W., Quaritsch, M., & Wolf, W. (2008). The evolution from single to pervasive smart cameras. In *Second ACM/IEEE*

International Conference on Distributed Smart Cameras (ICDSC 2008). Stanford, CA, USA, 7-11 September 2008 (pp. 1-10).

[110] Quaritsch, M., Kreuzthaler, M., Rinner, B., Bischof, H., & Strobl, B. (2007). Autonomous multicamera tracking on embedded smart cameras. *EURASIP Journal on Embedded Systems, 2007*(1), 35-35.

[111] Wang, Y., Velipasalar, S., & Casares, M. (2010). Cooperative object tracking and composite event detection with wireless embedded smart cameras. *Image Processing, IEEE Transactions on, 19*(10), 2614-2633.

[112] Mucci, C., Vanzolini, L., Deledda, A., Campi, F., & Gaillat, G. (2007). Intelligent cameras and embedded reconfigurable computing: a case-study on motion detection. In *2007 International Symposium on System-on-Chip.* Tampere, Finland, 20-21 November 2007 (pp. 1-4).

[113] Hengstler, S., Prashanth, D., Fong, S., & Aghajan, H. (2007). MeshEye: a hybrid-resolution smart camera mote for applications in distributed intelligent surveillance. In *6th International Symposium on Information Processing in Sensor Networks (IPSN 2007).* Cambridge, MA, USA, 25-27 April 2007 (pp. 360-369).

[114] Casares, M., Velipasalar, S., & Pinto, A. (2010). Light-weight salient foreground detection for embedded smart cameras. *Computer Vision and Image Understanding, 114*(11), 1223-1237.

[115] Dworak, V., Selbeck, J., Dammer, K. H., Hoffmann, M., Zarezadeh, A. A., & Bobda, C. (2013). Strategy for the Development of a Smart NDVI Camera System for Outdoor Plant Detection and Agricultural Embedded Systems. *Sensors, 13*(2), 1523-1538.

[116] Fernández, J., Calavia, L., Baladrón, C., Aguiar, J. M., Carro, B., Sánchez-Esguevillas, A., Alonso-López, J. A., & Smilansky, Z. (2013). An Intelligent Surveillance Platform for Large Metropolitan Areas with Dense Sensor Deployment. *Sensors, 13*(6), 7414-7442.

[117] Broggi, A. (1995). Robust real-time lane and road detection in critical shadow conditions. In *Proceedings of the International Symposium on Computer Vision.* Coral Gables, FL, USA, 21-23 November 1995 (pp. 353-358).

[118] He, Y., Wang, H., & Zhang, B. (2004). Color-based road detection in urban traffic scenes. *Intelligent Transportation Systems, IEEE Transactions on*, *5*(4), 309-318.

[119] Tsai, L. W., Hsieh, J. W., Chuang, C. H., & Fan, K. C. (2008). Lane detection using directional random walks. In *2008 IEEE Intelligent Vehicles Symposium.* Eindhoven, the Netherlands, 4-6 June 2008 (pp. 303-306).

[120] Fernyhough, J. H., Cohn, A. G., & Hogg, D. C. (1996). Generation of Semantic Regions from Image Sequences. In B. Buxton & R. Cipolla (Eds.), *Computer Vision* (pp. 475-478). Springer Berlin Heidelberg.

[121] Howarth, R. J., & Buxton, H. (1992). Analogical Representation of Spatial Events for Understanding Traffic Behavior. In B. Neumann (Ed.), *10th European Conference on Artificial Intelligence.* Vienna, Austria, 3-7 August 1992 (pp. 785-789).

[122] Makris, D., & Ellis, T. (2003). Automatic learning of an activity-based semantic scene model. In *IEEE Conference on Advanced Video and Signal Based Surveillance*. Miami, FL, USA, 21-22 July 2003 (pp. 183-188).

[123] MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In L. Lecam & J. Neyman (Eds.), *Proceedings of the Fifth Symposium on Math, Statistics, and Probability*. Berkeley, CA, USA, 21 June, 18 July 1965 (Vol. 1, pp. 281-297).

[124] Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1-38.

[125] Brandle, N., Bauer, D., & Seer, S. (2006). Track-based finding of stopping pedestrians-a practical approach for analyzing a public infrastructure. In *IEEE Intelligent Transportation Systems Conference (ITSC'06).* Toronto, Canada, 17-20 September 2006 (pp. 115-120).

[126] Morris, B. T., & Trivedi, M. M. (2008). A survey of vision-based trajectory learning and analysis for surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, *18*(8), 1114–1127.

[127] Hu, W., Xiao, X., Fu, Z., & Xie, D. (2006). A system for learning statistical motion patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(9), 1450-1464.

[128] Hu, W., Xiao, X., Xie, D., Tan, T., & Maybank, S. (2004). Traffic accident prediction using 3-D model-based vehicle tracking. *Vehicular Technology, IEEE Transactions on*, *53*(3), 677-694.

[129] Morris, B. T., & Trivedi, M. M. (2008). Learning and classification of trajectories in dynamic scenes: A general framework for live video analysis. In *IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance (AVSS'08).* Santa Fe, NM, USA, 1-3 September 2008 (pp. 154-161).

[130] Makris, D., & Ellis, T. (2002). Path detection in video surveillance. *Image and Vision Computing*, *20*(12), 895–903.

[131] Hu, W., Xie, D., Fu, Z., Zeng, W., & Maybank, S. (2007). Semantic-based surveillance video retrieval. *Image Processing, IEEE Transactions on*, *16*(4), 1168-1181.

[132] Junejo, I., Javed, O., & Shah, M. (2004). Multi Feature Path Modelling for Video Surveillence. In *Proceedings of the* 17th *International Conference on Pattern Recognition (ICPR'04)*. Cambridge, UK, 23-26 August 2004 (Vol. 2, pp. 716-719).

[133] Zhong, H., Shi, J., & Visontai, M. (2004). Detecting unusual activity in video. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004).* Washington, DC, USA, 27 June – 2 July 2004 (Vol. 2, pp. II-819).

[134] Naftel, A., & Khalid, S. (2006). Classifying spatiotemporal object trajectories using unsupervised learning in the coefficient feature space. *Multimedia Systems*, *12*(3), 227-238.

[135] Porikli, F. (2004). Learning object trajectory patterns by spectral clustering. In *2004 IEEE International Conference on Multimedia and Expo (ICME'04).* Taipei, Taiwan, 27-30 June 2004 (Vol. 2, pp. 1171-1174).

[136] Biliotti, D., Antonini, G., & Thiran, J. P. (2005). Multi-layer hierarchical clustering of pedestrian trajectories for automatic counting of people in video sequences. In *Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTIONS'05).* Breckenridge, CO, USA, 5-7 January 2005 (Vol. 2, pp. 50-57).

[137] Bashir, F., Qu, W., Khokhar, A., & Schonfeld, D. (2005). HMM-based motion recognition system using segmented PCA. In *IEEE International Conference on Image Processing (ICIP 2005).* Genoa, Italy, 11-14 September 2005 (Vol. 3, pp. III-1288).

[138] Atev, S., Masoud, O., & Papanikolopoulos, N. (2006). Learning traffic patterns at intersections by spectral clustering of motion trajectories. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems.* Beijing, China, 9-15 October 2006 (pp. 4851-4856).

[139] Fu, Z., Hu, W., & Tan, T. (2005). Similarity based vehicle trajectory clustering and anomaly detection. In *IEEE International Conference on Image Processing (ICIP 2005).* Genoa, Italy, 11-14 September 2005 (Vol. 2, pp. II-602).

[140] Keogh, E. J., & Pazzani, M. J. (2000). Scaling up dynamic time warping for datamining applications. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. Boston, MA, USA, 20-23 August 2000 (pp. 285-289).

[141] Rabiner, L., & Juang, B. H. (1993). *Fundamentals of speech recognition*. Prentice Hall.

[142] Vlachos, M., Kollios, G., & Gunopulos, D. (2002). Discovering similar multidimensional trajectories. In *Proceedings of the 18th International Conference on Data Engineering.* San Jose, CA, USA, 26 February – 1 March 2002 (pp. 673-684).

[143] Buzan, D., Sclaroff, S., & Kollios, G. (2004). Extraction and clustering of motion trajectories in video. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004).* Cambridge, UK, 23-26 August 2004 (Vol. 2, pp. 521-524).

[144] Lou, J., Liu, Q., Tan, T., & Hu, W. (2002). Semantic Interpretation of Object Activities in a Surveillance System. In *Proceedings 16th International Conference on Pattern Recognition (ICPR'02)*. Québec City, Canada, 11-15 August 2002 (Vol. 3, pp. 777-780).

[145] Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, *31*(3), 264-323.

[146] Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data* (pp. 25-71). Springer Berlin Heidelberg.

[147] Lin, J., Vlachos, M., Keogh, E., & Gunopulos, D. (2004). Iterative incremental clustering of time series. In *Advances in Database Technology-EDBT 2004*. Crete, Greece, 14-18 March 2004 (pp. 106-122).

[148] Kohonen, T. (1990). The self-organizing map. In *Proceedings of the IEEE*, *78*(9), 1464-1480.

[149] Johnson, N., & Hogg, D. (1996). Learning the Distribution of Object Trajectories for Event Recognition. *Image and Vision Computing*. *14*(8), 609-615.

[150] Sumpter, N., & Bulpitt, A. (2000). Learning Spatio-Temporal Patterns for Predicting Object Behavior. *Image and Vision Computing*. *18*(9), 697-704.

[151] Jiao, L., Wu, Y., Wu, G., Chang, E. Y., & Wang, Y. F. (2004). Anatomy of a multicamera video surveillance system. *Multimedia systems*, *10*(2), 144-163.

[152] Basharat, A., Gritai, A., & Shah, M. (2008). Learning object motion patterns for anomaly detection and improved object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Anchorage, AK, USA, 23-28 June 2008 (pp. 1-8).

[153] Anjum, N., & Cavallaro, A. (2008). Multi-Feature Object Trajectory Clustering for Video Analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, *18*(11), 1555-1564.

[154] Reiss, M., & Taylor, J. G. (1991). Storing temporal sequences. *Neural networks*, *4*(6), 773-787.

[155] Boyd, J. E., Meloche, J., & Vardi, Y. (1999) Statistical Tracking in Video Traffic Surveillance. In *Proceedings of the Seventh International Conference on Computer Vision.* Kerkyra, Greece, 20-27 September 1999 (pp. 163-168).

[156] Sudderth, E., Hunter, E., Kreutz-Delgado, K., Kelly, P. H., & Jain, R. (1998). Adaptive video segmentation: theory and real-time implementation. In *DARPA Image Understanding Workshop*. Monterey, CA, USA, 20-23 November 1998 (Vol. 1, pp. 177-181).

[157] Wang, X., Tieu, K., & Grimson, E. (2006). Learning Semantic Scene Models by Trajectory Analysis. In *Proceedings of the 9th European Conference on Computer Vision*. Graz, Austria, 7-13 May 2006, (pp. 110–123).

[158] Huttenlocher, D. P., Klanderman, G. A., & Rucklidge, W. J. (1993). Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*(9), 850-863.

[159] Balasko, B., Abonyi, J., & Feil, B. Fuzzy Clustering and Data Analysis Toolbox for use with MATLAB. Department of Process Engineering University of Veszprem, Hungría. http://www.abonyilab.com/software-and-data/fclusttoolbox (Última visita: Noviembre 2012).

[160] Kaufman, L., & Rousseeuw, P. J. (1987). Clustering by means of medoids. In Y. Dodge (Ed.), *Statistical data analysis based on the L1norm* (pp. 405-416). Amsterdam: North-Holland.

[161] Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithm*. New York: Plenum Press.

[162] Gustafson D. E., & Kessel, W. C. (1979). Fuzzy clustering with a fuzzy covariance matrix. In *Proceedings of the 1978 IEEE Conference on Decision and Control Including the 17th Symposium on Adaptive Processes*. San Diego, CA, USA, 10-12 January 1979 (pp. 761-766).

[163] Gath, I., & Geva A. B. (1989). Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *11*(7), 773–780.

[164] Bezdek, J. C. (1974). *Cluster validity with fuzzy sets. Journal of Cybernetics*, *3* (3), 58-73.

[165] Bezdek, J. C. (1975). Mathematical models for systemics and taxonomy. In G. Estabrook (Ed.), *Proceedings Eight Annual International Conference on Numerical Taxonomy.* Oeiras, Portugal, 1974 (pp. 143-164).

[166] Abonyi, J., & Feil, B. (2007). Aggregation and Visualization of Fuzzy Clusters Based on Fuzzy Similarity Measures. In J. Valente de Oliveira & W. Pedrycz (Eds.), *Advances in fuzzy clustering and its applications* (pp. 95-122). New York: John Wiley & Sons, Ltd.

[167] Xie, N. L., & Beni, G. A. (1991). Avalidity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*(8), 841–847.

[168] Dunn, J. C. (1973). A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics, 3*(3), 32–57.

[169] Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On Clustering Validation Techniques. *Journal of Intelligent Information Systems, 17*, 107–145.

[170] Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings 2nd International Conference on Knowledge Discovery and Data Mining*. Portland, OR, USA, 2-4 August 1996 (Vol. 96, pp. 226-231).

[171] Daszykowski, M., Walczak, B., & Massart, D.L. (2001). Looking for Natural Patterns in Data. Part 1: Density Based Approach. *Chemometrics and Intelligent Laboratory Systems*, *56*(2), 83-92.

[172] Minsky, M. (1975). A framework for representing knowledge. In P.H. Winston (Ed.), *The Psychology of Computer Vision* (pp. 211-217). McGraw Hill.

[173] Quillian, M. R. (1967). Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral Science*, *12*(5), 410-430.

[174] Newell, A. (1973). Production Systems: Models of Control Structures. *Visual Information Processing*. New York: Academic Press.

[175] Hammer, E. M. (1998). Semantics for existential graphs. *Journal of Philosophical Logic*, *27*(5), 489-503.

[176] Fensel, D., McGuiness, D. L., Schulten, E., Ng, W. K., Lim, G. P., & Yan, G. (2001). Ontologies and electronic commerce. *Intelligent Systems, IEEE, 16*(1), 8-14.

[177] Weigand, H. (1997). A multilingual ontology-based lexicon for news filtering| the TREVI project. In *IJCAI Workshop on Ontologies and Multilingual NLP. International Joint Conference on Artificial Intelligence.* Nagoya, Japan, 3 August 1997.

[178] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition, 5*(2), 199-220.

[179] Hendler, J. (2001). Agents and the semantic web. *Intelligent Systems, IEEE, 16*(2), 30-37.

[180] Chandrasekaran, B., Josephson, J. R., & Benjamins, V. R. (1999). What are ontologies, and why do we need them?. *Intelligent Systems and Their Applications, IEEE, 14*(1), 20-26.

[181] Guarino, N., & Poli, R. (1995). Formal ontology, conceptual analysis and knowledge representation. *International Journal of Human Computer Studies, 43*(5), 625-640.

[182] McGuinness, D. L., Fikes, R., Hendler, J., & Stein, L. A. (2002). DAML+ OIL: an ontology language for the Semantic Web. *Intelligent Systems, IEEE, 17*(5), 72-80.

[183] Gomez-Perez, A., Corcho-Garcia, O., & Fernandez-Lopez, M. (2004). Ontological engineering. *Computing Reviews, 45*(8), 478-479.

[184] Zhou, J., Ma, L., Liu, Q., Zhang, L., Yu, Y., & Pan, Y. (2006). Minerva: A scalable OWL ontology storage and inference system. In *The Semantic Web–ASWC 2006.* Beijing, China, 3-7 September 2006 (pp. 429-443).

[185] del Mar Roldan-Garcia, M., & Aldana-Montes, J. F. (2005). A Tool for Storing OWL Using Database Technology. In *Proceedings of the OWLED 2005 Workshop on OWL: Experiences and Directions.* Galway, Ireland, 11-12 November 2005.

[186] Pan, Z., & Heflin, J. (2004). DLDB: *Extending relational databases to support semantic web queries.* Lehigh University, Bethlehem PA, Department of Computer Science and Electrical Engineering. http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA451847 (Última visita: Noviembre 2012).

[187] Khalid, A., Shah, S. A. H., & Qadir, M. A. (2009). OntRel: An Ontology Indexer to store OWL-DL Ontologies and its Instances. In *International Conference of Soft Computing and Pattern Recognition (SOCPAR'09).* Malacca, Malaysia, 4-7 December 2009 (pp. 478-483).

[188] Jeong, D., Choi, M., Jeon, Y. S., Han, Y. H., Yang, L. T., Jeong, Y. S., & Han, S. K. (2007). Persistent storage system for efficient management of OWL web ontology. In *4th International Conference of Ubiquitous Intelligence and Computing (UIC 2007)*. Hong Kong, China, 11-13 July 2007 (pp. 1089-1097).

[189] Apache Jena. *Semantic Web Framework for Java*. Disponible online: http://jena.sourceforge.net/ontology/index.html (Última visita: Abril 2013).

[190] Fensel, D., Van Harmelen, F., Horrocks, I., McGuinness, D. L., & Patel-Schneider, P. F. (2001). OIL: An ontology infrastructure for the semantic web. *Intelligent Systems, IEEE, 16*(2), 38-45.

[191] Decker, S., Melnik, S., Van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., ... & Horrocks, I. (2000). The semantic web: The roles of XML and RDF. *Internet Computing, IEEE, 4*(5), 63-73.

[192] McGuinness, D. L., & Van Harmelen, F. (2004). OWL web ontology language overview. *W3C recommendation, 10*(2004-03), 10.

[193] Motik, B., Sattler, U., & Studer, R. (2005). Query answering for OWL-DL with rules. *Web Semantics: Science, Services and Agents on the World Wide Web, 3*(1), 41-60.

[194] Horrocks, I., & Patel-Schneider, P. F. (2004). A proposal for an OWL rules language. In *Proceedings of the 13th international conference on World Wide* Web. New York, NY, USA, 17-22 May 2004 (pp. 723-731).

[195] Horrocks, I., Patel-Schneider, P. F., Bechhofer, S., & Tsarkov, D. (2005). OWL rules: A proposal and prototype implementation. *Web Semantics: Science, Services and Agents on the World Wide Web, 3*(1), 23-40.

[196] Bertini, M., Del Bimbo, A., & Serra, G. (2008). Learning ontology rules for semantic video annotation. In *Proceedings of the 2nd ACM workshop on Multimedia semantics*. Vancouver, Canada, 26- 31 October 2008 (pp. 1-8).

[197] Carroll, J. J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., & Wilkinson, K. (2004). Jena: implementing the semantic web recommendations. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*. New York, NY, USA, 17-22 May 2004 (pp. 74-83).

[198] Meditskos, G., & Bassiliades, N. (2010). DLEJena: A practical forward-chaining OWL 2 RL reasoner combining Jena and Pellet. *Web Semantics: Science, Services and Agents on the World Wide Web, 8*(1), 89-94.

[199] Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y. (2007). Pellet: A practical owl-dl reasoner. *Web Semantics: science, services and agents on the World Wide Web, 5*(2), 51-53.

[200] Newell, A., & Simon, H. A. (1972). *Human problem solving* (Vol. 14). Englewood Cliffs, NJ: Prentice-Hall.

[201] Salminen, A., & Tompa, F. (2011). Why Use XML?. *Communicating with XML* (pp. 69-91). Springer US.

[202] Lin, D. T., & Chen, Y. T. (2011). Pedestrian and Vehicle Classification Surveillance System for Street-Crossing Safety. In *The 2011 International Conference on Image Processing, Computer Vision, and Pattern Recognition*. Las Vegas, NV, USA, 18-21 July 2011.

[203] Lee, P. H., Chiu, T. H., Lin, Y. L., & Hung, Y. P. (2009). Real-time pedestrian and vehicle detection in video using 3d cues. In *IEEE International Conference on Multimedia and Expo (ICME 2009).* New York, NY, USA, 28 June - 3 July 2009 (pp. 614-617).