UNIVERSIDAD DE VALLADOLID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

TRABAJO FIN DE MÁSTER

MÁSTER UNIVERSITARIO EN INVESTIGACIÓN

EN TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

Sistema de Video Vigilancia Semántico basado en movimiento. Aplicación a la Seguridad y Control de Tráfico

Autor:

D. Jorge Fernández Gutiérrez

Tutores:

Dra. Dña. Belén Carro Martínez

Dr. D. Javier Manuel Aguiar Pérez

Valladolid, 18 de Julio de 2013

TÍTULO:	**Sistema de Video Vigilancia Semántico basado en movimiento. Aplicación a la Seguridad y Control de Tráfico**
AUTOR:	**D. Jorge Fernández Gutiérrez**
TUTORES:	**Dra. Dña. Belén Carro Martínez** **Dr. D. Javier Manuel Aguiar Pérez**
DEPARTAMENTO:	TEORÍA DE LA SEÑAL Y COMUNICACIONES E INGENIERÍA TELEMÁTICA

TRIBUNAL

PRESIDENTE:	**Dr. D. Alonso Alonso Alonso**
VOCAL:	**Dra. Dña. Helena Castán Lanasta**
SECRETARIO:	**Dra. Dña. Noemí Merayo Álvarez**
FECHA:	**18 de Julio de 2013**
CALIFICACIÓN:	

Resumen de TFM

Se realiza el diseño y la arquitectura de un sistema de videovigilancia semántico orientado al control de tráfico. A partir de los datos provenientes de una red de sensores visuales inteligentes y basándose en el conocimiento definido en una ontología, el sistema automáticamente detecta e identifica las alarmas ocurridas en la escena. Este trabajo se ha desarrollado dentro del proyecto Europeo Celtic HuSIMS.

Palabras clave

Videovigilancia, Detección inteligente, Semántica, Sensores visuales inteligentes, Seguridad

Abstract

This work defines the design and the definition of the architecture of a semantic video surveillance system aimed at controlling traffic. Based on data from a network of intelligent visual sensors, the system automatically detects and identifies the alarms occurring in the scene based on knowledge that is defined into ontology. This work has been carried out within the European project Celtic HuSIMS.

Keywords

Surveillance, Intelligent detection, Semantic, Smart visual sensors, Security

AGRADECIMIENTOS

"***De derrota en derrota hasta la victoria final***"

Winston Churchill

A mi mujer que, no sólo ha estado a mi lado y me ha dado fuerzas en los malos momentos, sino que ha utilizado sus pocos ratos libres para compartir su conocimiento conmigo permitiéndome lograr superar todos los obstáculos.

A mis padres, mi hermano, mis abuelos y a toda mi familia por tener la paciencia suficiente para soportar mi ausencia durante este año.

A mis tutores, Belén y Javier, por darme la oportunidad de realizar este proyecto y crecer como investigador, ingeniero y persona.

A mis compañeros de laboratorio del grupo SRC, Carlos, Marian y Daniel por proporcionarme su apoyo durante la consecución de este TFM.

Por último me gustaría agradecer sus contribuciones a todas las empresas participantes en este proyecto, Alvarion, Afcon, B-I Industrial, C-B4, C2Tech, Gigle, EMZA, Ericsson y SQS, así como al Ministerio de Industria, Turismo y Comercio, las cuales han ayudado a llevar a buen fin este trabajo.

TABLA DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO 1

INTRODUCCIÓN

1.1. Introducción

Los constantes y cada vez más rápidos avances tecnológicos que actualmente se están produciendo están dotando a la sociedad de la capacidad de poder controlar y ser consciente de toda la información que la rodea y no sólo eso, sino también analizarla y sacar conclusiones de ella. Uno de los términos más utilizados en el ámbito de las TIC (Tecnologías de la Información y las Comunicaciones) es Smart. Éste se asocia a hogares, edificios, ciudades, etc., lugares donde gracias a la incorporación de sensores se monitoriza la información deseada dotando a su gestor de la capacidad de adaptar "de forma inteligente" ciertos parámetros para ajustarse a los requerimientos del beneficiario.

Esto está, en cierta manera, relacionado con la seguridad. Y es que esta sociedad cada vez está más interesada en conocer lo que la rodea y a su vez tener una cierta sensación de protección. Para ello se utilizan los sistemas de videovigilancia.

La integración de estos sistemas formados por redes de sensores, especialmente de video, dentro de estos Smart Places o espacios inteligentes proporcionan a dicho sistema de la capacidad de monitorizar el medio en el que se encuentra y, gracias a la inclusión de técnicas de inteligencia artificial (IA), analizar dicha información e identificar qué situaciones pueden ser de interés o potencialmente peligrosas.

La variedad de equipos y sistemas de videovigilancia en el mercado es muy variada. Éstos se utilizan como medio disuasorio y método de localización de intrusos en edificios y como método de vigilancia en carreteras.

Sin embargo los sistemas de videovigilancia actuales cuentan con una serie de inconvenientes. El primero de ellos reside en el hardware utilizado. La mayoría de estos sistemas consta de redes de cámaras de vigilancia de alta definición. Su uso implica por una parte, un cierto problema en cuanto a la privacidad de las personas que se encuentran en la escena, por lo que su instalación en lugares públicos puede ser complicada. El sistema aquí propuesto evita este tipo de cuestiones al realizar el procesado de la imagen localmente evitando su envío. Además dicho sensor visual trabaja con imágenes de baja resolución solventado así el dilema del tratamiento de la privacidad al evitar identificar identidades de personas.

Unido a lo anterior, el hecho de utilizar este tipo de imágenes de alta resolución conlleva también que la transmisión de las mismas al centro de procesado necesite un gran ancho de banda y, una vez recibidas éstas, su análisis y procesado requiera una gran carga computacional. Todo esto hace que los costes del sistema sean muy elevados. Para solucionarlo, el sistema propuesto utiliza unos sensores visuales que son capaces de procesar la imagen localmente y como dicha imagen es de baja resolución la carga computacional de los mismos no es muy alta. Por otra parte al no enviarse las imágenes sino archivos en formato XML (*eXtensible Markup Language*) se evita la necesidad de grandes anchos de banda, permitiéndose sistemas con un mayor número de redes de sensores. Esto implica una gran reducción de los costes.

Otro de los principales problemas de los sistemas actuales es que son muy rígidos, es decir, su adaptación a nuevos entornos o nuevas situaciones es muy problemático siendo necesario una restructuración o nuevo diseño de dicho sistema. Gracias a inclusión de la semántica en el sistema aquí propuesto, se ha conseguido que con unas pequeñas modificaciones el sistema sea capaz de realizar varios tipos de análisis para una misma escena.

No obstante no acaban aquí las limitaciones de este tipo de sistemas. Una de las cuestiones más importantes es la eficiencia de los mismos y ésta se ve reducida en gran parte debido a la dependencia de los mismos de un operador humano, que es quien se encarga de determinar qué es lo que está ocurriendo en la escena. Sin embargo es una realidad que éstos se encuentran limitados por el cansancio y la falta de atención con lo que, si el número de cámaras es muy alto, la eficiencia del usuario pasado un tiempo decrece sustancialmente. Además el uso de varios operadores encarece considerablemente el coste del sistema. Para solventar este problema se propone la automatización del mismo, ya que en este trabajo se propone un sistema autónomo que permite la identificación, previo a un aprendizaje de la escena, de las situaciones anómalas evitando la intervención humana.

Con lo comentado anteriormente este sistema logra solventar con gran acierto los problemas de los que constan actualmente los sistemas de videovigilancia dotando al mismo de una gran adaptación a la escena, una reducción de coste y un aumento de

la eficiencia. Destacar también que este trabajo se ha realizado bajo las directivas del proyecto Europeo Celtic HuSIMS (*Human Situation Monitoring System*) [1].

1.2. Motivación

Los sistemas de videovigilancia actuales basan su funcionamiento en la utilización de potentes equipos de video que se encargan de enviar las imágenes, en la mayoría de las situaciones de alta resolución, a centros de control. Esto implica una necesidad de una gran capacidad de transmisión y un aumento de la carga de procesado.

Una vez en el centro del control un operario humano se encarga de determinar si ha ocurrido alguna situación de riesgo o alarma. Estos sistemas se diseñan con el objetivo de trabajar en una situación concreta impidiendo o condicionando su uso para nuevos enfoques de detección.

Los costes implicados de su implementación son uno de los temas a tratar ya que la instalación de grandes redes de cámaras de videovigilancia conllevaría unas inversiones de un valor demasiado elevado.

Por otra parte los algoritmos de análisis utilizados en la mayoría de estos sistemas no identifican la situación de alarma producida, lo que implica la necesidad adicional de un operador humano que, una vez determinado que hay una situación de alarma, identifique la misma. Esto evidencia una falta de automatización de los mismos aumentando en gran medida el coste.

El sistema de videovigilancia propuesto pretende eliminar o por lo menos mitigar en lo posible todos estos problemas diseñando un sistema flexible, capaz de detectar e identificar las situaciones potencialmente peligrosas como accidentes de tráfico, salidas de vía, etc., mediante un procesado sencillo y que a su vez pueda ser aplicado a otras áreas como vandalismo de una forma rápida.

Además este sistema será capaz de trabajar de forma autónoma identificando dichas situaciones anómalas en tiempo real y siendo capaz de comunicarse con un centro de control para notificar las incidencias. Por otra parte, los sensores aquí propuestos destacan por su bajo precio y por su capacidad de procesado de la imagen, disminuyendo el ancho de banda necesario para la transmisión de la información, y por lo tanto rebajando el coste global del sistema y permitiendo un mayor despliegue del mismo a un precio competitivo.

1.3. Objetivos

El objetivo general de este trabajo es diseñar e implementar un sistema de videovigilancia inteligente, que a partir de los datos enviados por sensores visuales de los objetos en movimiento de la escena, sea capaz de interpretarlos mediante un análisis semántico y determinar situaciones anómalas que ocurran en la escena.

Para lograr este objetivo se definen varios objetivos intermedios, que mediante su realización, logren ayudar a conseguir el principal. Estos se encuentran detallados a continuación:

- Crear un sistema capaz de trabajar con redes de varios sensores. Para ello debe ser capaz de procesar la información procedente de dichos dispositivos en tiempo real.
- Evitar en todo momento identificar a los individuos de la escena para evitar conflictos legales de privacidad.
- Diseñar un sistema y modelo de conocimiento de la escena para el control de tráfico que a su vez identifique la situación anómala que está ocurriendo. Todo esto se debe realizar mediante la utilización de tecnología semántica para dotar al sistema de un razonado lo más cercano al humano posible.
- Implementar un sistema adaptable a diferentes situaciones (vandalismo, incendio, etc.) con un bajo tiempo y coste de adaptación.
- Dotar al sistema de la capacidad de comunicación con el servicio responsable o más adecuado para la atención de las situaciones anómalas detectadas.

Todo lo comentado anteriormente debe ser compatible con las tecnologías utilizadas por el resto de motores de análisis de la escena incluidos dentro del proyecto HuSIMS.

1.4. Metodología

La metodología utilizada a la hora de afrontar la resolución de este Trabajo Fin de Máster (TFM) se ha basado en cinco fases.

La primera de estas fases se ha basado en la búsqueda de información relativa a la tecnología semántica para poder comprender en qué pilares se fundamenta y comprender posteriormente su enfoque en la videovigilancia.

A continuación, la segunda de las fases ha sido similar a la fase anterior y ha consistido en el estudio en profundidad de la bibliografía existente sobre sistemas de videovigilancia con el fin de analizar los sistemas existentes, centrándose en sus principios, limitaciones y posibles usos. En dicha búsqueda se ha prestado interés en los sistemas que tuviesen un sistema de procesado de datos semántico, como es el que se pretende realizar en este trabajo.

Una vez comprendida la tecnología a usar y los sistemas implementados hasta ahora, se ha pasado a proponer la solución del sistema a implementar que evite las limitaciones y cumpla los requisitos impuestos al inicio del trabajo. Se definen también las tecnologías a utilizar y las fases de la misma. Esto ha sido tratado en la tercera de las fases.

Con las ideas claras se ha pasado, en la cuarta fase, a implementar el sistema definido previamente, siguiendo las pautas definidas anteriormente, volviendo a la fase anterior siempre que se encontrase un punto que llevase a un desarrollo no viable o solución no óptima del sistema.

Por último se ha utilizado el sistema en varios escenarios, comprobando su funcionamiento y comparando, en la medida de lo posible, dichos datos con los planteados en trabajos encontrados previamente en la literatura, comprobando la eficiencia y la mejora que supone el sistema implementado en este trabajo.

1.5. Estructura del trabajo

Este Trabajo Fin de Máster o TFM se encuentra dividido en 6 capítulos. En el primero de ellos se realiza una introducción al tema desarrollado junto con una descripción de los objetivos del trabajo así como la metodología utilizada para su realización.

En el capítulo 2 se muestra un estudio sobre los sistemas de videovigilancia existentes en la literatura, así como una pequeña revisión de las partes individuales de las que están formados estos sistemas, concretamente sensores, red de transmisión y de algoritmos de análisis de datos.

Una vez definido lo anterior se pasa a describir el trabajo desarrollado en este TFM en el capítulo 3. En él se detalla la arquitectura y los componentes del sistema.

En el capítulo 4 se realiza la validación de los datos obtenidos mediante el testeo del sistema en diferentes casos de uso así como una validación del modelo de conocimiento. Se realizará de forma adicional una comparativa entre el sistema implementado y diversos sistemas de videovigilancia existentes.

A continuación, en el capítulo 5 se detallan las conclusiones obtenidas de la realización del trabajo y las líneas futuras que pueden tomarse a partir de éste.

Por último se incluye la bibliografía utilizada para la confección de este trabajo y un anexo donde se adjunta el artículo publicado en la revista “Sensors” elaborado y publicado en el desarrollo de este Trabajo Fin de Máster.

CAPÍTULO 2

CONOCIMIENTOS PREVIOS

2.1. Sistemas de Videovigilancia

El constante avance de la tecnología ha permitido, no sólo que económicamente sea más accesible tanto el software como el hardware, sino que la eficiencia, capacidad y adaptación a nuevas situaciones por parte de los sistemas con procesamiento de la información sea cada vez más rápida y fácil.

Por su parte los sistemas de comunicaciones también han evolucionado, gracias principalmente a la inclusión dentro de ellos de nuevas tecnologías de comunicación como la fibra óptica y al crecimiento de Internet. Esto ha permitido que los datos generados por los sistemas anteriormente mencionados puedan ser transmitidos de una manera más rápida y que la gestión de los mismos pueda ser prácticamente en tiempo real.

En el caso de los sistemas de videovigilancia se han producido dos tipos de avances. El primero es el tecnológico, ya que estos sistemas avanzan a la par de los desarrollos hardware del mercado, notándose una gran progresión en los mismos debido en parte también a la inclusión de nuevos enfoques para el procesado de los datos e imágenes.

Esto ha hecho que la investigación en sistemas de videovigilancia, y principalmente en sus componentes, se haya convertido en un tema de interés. Algunos ejemplos de esto son los trabajos realizados en hardware [2-4] (cámaras y sensores),

infraestructuras de comunicaciones [5-8], software [9-10] (análisis inteligente de vídeo).

Por otro lado está el hecho de que cada vez la sociedad requiere la instalación y uso de este tipo de sistemas. Este aumento se relaciona con la creciente necesidad por parte de la sociedad de mantener una cierta vigilancia sobre, no sólo sus hogares, sino también determinar un nivel de seguridad en espacios y edificios públicos. Esto se ve reflejado en las estadísticas ya que según la *Compound Annual Growth Rate* o CAGC [11] este crecimiento de las ventas será de un 14,33% entre los años 2011 y 2015.

Los primeros sistemas de videovigilancia se establecieron en lugares privados como bancos e instalaciones militares debido a la importancia de la seguridad en las mismas. Sin embargo lugares como parques, cruces de carreteras, comercios, etc., han incluido este tipo de sistemas correspondiendo a la preocupación de la sociedad en determinar un cierto grado de garantía durante su uso.

Sin embargo el nivel de seguridad exigida en estos entornos varía considerablemente. No se establece la misma escala de riesgo para un parque en las afueras que en un cruce situado en las cercanías de un colegio.

No obstante en los sistemas mencionados anteriormente, aquellos en los cuales la necesidad de seguridad tiene una gran importancia llegando incluso a ser crítica, el número de cámaras, y por lo tanto de información que han de procesar estos operadores humanos, es muy grande.

Para paliar este inconveniente, o por lo menos para hacer que el sistema sea más eficiente, se utilizan elementos intermedios entre las cámaras y los operadores (detectores de movimiento, sensores previos de alarma, etc.) que filtren dicha información y sólo deje aquella que pueda ser procesada por dicho operador.

Existen a su vez escenarios donde no es posible el uso de filtros o donde el nivel de seguridad debe de ser tan alto debido al riesgo que convierte la seguridad en una cuestión de primer nivel. Sin embargo todas estas situaciones siguen estando bajo una total dependencia del operador humano.

Este operador está condicionado por el cansancio y su capacidad de concentración. En la mayoría de los casos comentados anteriormente, éste debe vigilar varias escenas y durante largos periodos de tiempo, lo que conlleva a que el rendimiento disminuya considerablemente según avanza el tiempo que se dedica a esta actividad.

Aquí es donde entra en escena la utilización de tecnologías o sistemas basados en Inteligencia Artificial o IA. Estas técnicas se fundamentan en la utilización de ciertos algoritmos que tratan las imágenes procedentes de las cámaras y realizan un análisis e interpretación de la misma en función de ciertos modelos de conocimiento previamente establecidos. Estos sistemas suponen una gran innovación dentro del

campo de la videovigilancia ya que en primer lugar permiten la utilización de un mayor número de dispositivos de videovigilancia (sensores, cámaras, etc.) y en segundo lugar estos sistemas son autónomos, disminuyendo en gran medida la dependencia de éstos del operador humano. Ejemplos de estos sistemas son los presentados en [12-15].

Dentro de estos sistemas de videovigilancia, la mayoría de ellos se centran en la utilización de análisis estadísticos con el fin de determinar características especiales de las imágenes. Nuevas generaciones de sensores incluyen en su software este tipo de algoritmos con el objetivo de dotar a éstos de la capacidad de rastrear movimiento [2] mediante el uso de modelo Bayesiano, el análisis del fondo de imagen para su posterior análisis [4] y nuevas herramientas pensadas para minimizar el coste de la red como configuración de orientación y campo de visión [3].

Este tipo de análisis crea un patrón de comportamiento estándar y detecta aquellos que se salen de éste, etiquetándolas como situaciones anómalas o comportamientos anómalos. Sin embargo el problema de este tipo de sistemas es que no pueden identificar qué situación concreta es la que ha activado la alarma.

Otro tipo de sistemas de videovigilancia son los sistemas codificados o rígidos. Éstos realizan el análisis de la escena en función de un modelo de reglas creadas previamente. Este tipo de observación evita el uso de tecnologías semánticas. El problema de este tipo de sistemas es que necesitan modificar el algoritmo en el cual están basados de forma manual para poder introducirles en otras escenas u otros ámbitos, siendo esto complejo y costoso.

Los sistemas de videovigilancia basados en el control del tráfico deben tener un cierto grado de flexibilidad a la hora de afrontar el análisis de la escena, puesto que ésta puede variar en gran medida. Sin embargo una de las cuestiones más importantes a tener en cuenta es que deben ser capaces de identificar la situación concreta que se está dando, haciendo así posible una mejor y más satisfactoria resolución de la misma. Una cuestión adicional pero que puede ser de gran interés es la posibilidad de realizar varias funciones, no sólo la de identificar situaciones anómalas como puede ser un accidente, sino acciones de vandalismo, incendios, etc.

Otro de los grandes hándicaps a la hora de diseñar sistemas de videovigilancia, es la implementación de las redes de sensores y las limitaciones del ancho de banda disponible para el envío de los datos. Además estos sistemas deben enviar la información en tiempo real lo que hace que esa conexión sea más importante aún.

Existen varios trabajos que tratan de hacer frente a estos problemas utilizando imágenes de baja resolución, lo que hace que baje el ancho de banda necesario para enviar dicha información. Ejemplo de esto es el trabajo descrito en [6].

En cuanto a las limitaciones de los protocolos de transporte de datos, en [7] se proporciona un diseño de aplicación de MAC (*Media Access Control*) sincronizadas en el tiempo, el cual es capaz de operar en la capa superior de protocolos IEEE 802.11 y mejorar el rendimiento al eliminar el retardo. La idea es la de mejorar el uso de los recursos de la red, priorizando conexiones mediante la detección de situaciones críticas de la red. Así se asigna a las unidades que lo requieran los recursos que estarían por otra parte desperdiciados. Esa es la fundamentación presentada en [8].

2.2. Mecanismos de captura y procesado de imagen

La forma en la que el ser humano adquiere la gran mayoría de la información es a través de métodos visuales, concretamente más del 90%. Y es en el procesado de dicha información en lo que, aproximadamente la mitad del cerebro dedica su ocupación.

Sin embargo la mayoría de los sistemas de visión artificial actuales dependen de la grabación de la imagen y no realizan el procesado de la misma. Además sus características físicas son problemáticas ya que tienen un alto coste y suelen ser voluminosos.

La introducción de cámaras de video como método de seguridad se remonta a mediados del siglo XX, concretamente a la década de los 40 [16]. Sin embargo los sistemas de videovigilancia analógicos no llegaron a producirse comercialmente hasta llegados los 70. Una de las situaciones que no ha cambiado mucho desde entonces es que un operario humano tenía que estar continuamente visualizando la imagen en búsqueda de una situación de alarma. La posterior creación de los dispositivos de almacenamiento (grabadoras de video) y la posibilidad de obtener las imágenes desde diferentes cámaras centralizando éstas en un solo sistema de control, dotó a éstos de una mayor flexibilidad a la hora de afrontar este tipo de situaciones [17]. Cabe destacar que la tendencia a la utilización de cámaras analógicas sólo se ha visto superada hace pocos años por la intrusión en el mercado de las cámaras IP (*Internet Protocol)*.

Determinando que el hecho de que la visualización por parte de un operador humano de este tipo de sistemas no es eficiente [18,19], el tema de la automatización de los sistemas de videovigilancia se convirtió en un tema de gran importancia. Las primeras aplicaciones basadas en sistemas automáticos fueron las de tráfico y detección de intrusos [20-21], aunque estos sistemas se calificaron como ineficientes [22] dotando a éstos de una mala publicidad y permitiendo que se siguiese utilizando a los operadores humanos como baza principal a la hora de analizar una situación.

El fundamento de estos primeros sistemas automáticos se basaba en la utilización de cámaras de video que se encargaban de transmitir la información en tiempo real a una centralita que era la que procesaba la información mediante operaciones y

algoritmos matemáticos [23,34]. El siguiente paso evolutivo de estas cámaras, fue la inclusión de capacidades de grabación de video de forma local en vez de forma remota.

Como se comentó anteriormente existen otros problemas en este tipo de sistemas, como los costes o la infraestructura donde se ubican. Con el objetivo de reducir la dependencia de esta última, se utilizan sensores de vigilancia híbridos, que sólo se activan cuando un sensor de detección de movimiento detecta éste. Esto reducía el coste de la red de sensores, en cuanto a la alimentación de los mismos, considerablemente.

Sin embargo todo el procesamiento de la imagen, para este tipo de cámaras, se enfocaba en la estación de control. Existen actualmente trabajos en la literatura que versan sobre "cámaras inteligentes" [17,25-28]. Estos proporcionan alternativas como arquitecturas específicas para la escena [28,30], equipos de híbridos que combinan sensores de baja y alta resolución [31] o el desarrollo de cámaras que introduzcan la capacidad de análisis de la escena mediante algoritmos que necesiten bajo procesamiento, como es el caso de [26,32]. Esto último será la fundamentación de los sistemas utilizados en el sistema.

2.3. Procesamiento de imágenes

La utilización de algoritmos de visión artificial, normalmente de un alto grado de complejidad está siendo una de las hojas de ruta en los sistemas de videovigilancia, como puede verse en [33-36]. Su problema reside en la necesidad de una gran capacidad de cálculo por lo que el hecho de utilizar un procesado en el sensor, eleva sustancialmente el precio de los mismos, por lo que se sigue utilizando el envío de vídeo a una central de procesado quien se encarga del mismo. Además esto vuelve a implicar la utilización de un gran ancho de banda en las comunicaciones.

Otro enfoque es la utilización de sistemas más ligeros [37] que permitan el procesado de esta imagen en local. El problema es que estos algoritmos dejan mucha información sin procesar que puede ser de interés como la forma de los objetos.

Uno de los enfoques utilizados como solución es el probabilístico [38,39]. Dentro de este enfoque se utilizan varias herramientas estadísticas como el teorema de Bayes. Éste es usado con el fin de determinar la probabilidad de que una situación específica sea una alarma. Esto se consigue con la comparación entre la información explícita proporcionada por el sistema y unas variables definidas previamente.

Sin embargo no es el único método estadístico que se utiliza. Existen trabajos como [40] que utilizan técnicas más complejas como HMM (*Hidden Markov Model)* con el fin de reconocer parámetros anómalos dentro de unos patrones definidos por medio del análisis de la información en bruto proporcionada por los sensores. Siguiendo con esta metodología en [41,42] se usan otras dos técnicas para el reconocimiento de patrones.

Éstas son DTW *(Dynamic Time Warping*) y LCSS (*Longest Common Sub-Sequence*) respectivamente. La aplicación de estos algoritmos está siendo un éxito a la hora de realizar agrupamiento de las diferentes trayectorias existentes en una escena. Esto se ve reflejado en la literatura [43,44].

En la literatura se encuentran otras técnicas de análisis basadas en la implementación de redes neuronales [45,46] o algoritmos dedicados al agrupamiento o *clustering* [47] con el fin de determinar y clasificar los comportamientos y el contexto de los objetos en la escena. Sin embargo este tipo de análisis es bastante costoso en cuanto a recursos y además bastante complicado de realizar en tiempo real.

No obstante a lo largo de la última década se han desarrollado tecnologías que tratan de aplicar a la detección nuevos enfoques con el fin de mejorar el rendimiento y las capacidades de los mismos. Una de estas técnicas es la semántica. En ella se representa la información mediante un modelo formal de conocimiento u ontología. Ejemplos de este uso son [48,49]. Entre las ventajas de estos sistemas se encuentra la facilidad de aplicación a diferentes dominios mediante la inclusión o modificación de estos modelos y la capacidad de trabajo conjunto con otros sistemas.

Por último se encuentran las técnicas orientadas a la combinación de información procedente de varios de estos sistemas de análisis. Y es que los sistemas de videovigilancia actuales están formados por grandes redes de sensores y la necesidad o requerimiento por parte del usuario de unos mínimos de certeza a la hora de precisar las situaciones de alarma. Para ello es posible que los sistemas de videovigilancia integren varios de los algoritmos anteriores con el fin de dotar a éste de una gran precisión. Con el objetivo de filtrar esas informaciones y eliminar los falsos positivos se utilizan las técnicas de fusión [50,51].

2.4. Reconocimiento de objetos y análisis de comportamientos

El despliegue de los sistemas de videovigilancia está orientado a su uso en grandes zonas públicas o grandes eventos como Juegos Olímpicos [52], detección de caídas de personas mayores [53], etc. Existen sistemas como [54,55] que se han implementado en ciudades como Manhattan con el objetivo de identificar matrículas o comportamientos de personas que puedan ser anómalos. Sin embargo existe una preocupación en la sociedad de que estos tipos de sistemas no son lo suficientemente precisos a la hora de determinar este tipo de situaciones críticas [56,57]. Por lo tanto actualmente se está desarrollando nuevos sistemas que intentan realizar análisis inteligentes de la escena para aprovechar al máximo la capacidad de estos mismos a la hora de determinar los patrones de comportamiento de los objetos que circulan o se mueven por la misma.

El análisis de los comportamientos de la escena se dirige a formar un sistema de comprensión automático para “entender” o saber determinar qué está pasando en la escena. La idea es contar con sistemas que no sean sólo capaces de grabar vídeo, es decir, sistemas inteligentes que puedan utilizarse para determinar situaciones de peligro [10]. Para ello existen varios sistemas que enfocan su trabajo hacia el seguimiento de los objetos que se encuentran dentro de la escena [2,9,12].

Existen empresas que están apostando por esta metodología. Un ejemplo de ellas es IBM. Ésta ha diseñado un sistema de videovigilancia [58] en la que además de estudiar la escena se extraen patrones de comportamiento de los objetos de los mismos, mediante la utilización de algoritmos estadísticos, a la par que determina eventos en tiempo real.

Con el objetivo de ver funcionar este tipo de sistemas, existe una herramienta de prueba denominada ETISEO [59]. Junto con ella se encuentra los datos y las acciones determinadas en base a su estudio [60].

Una de las cuestiones más importantes a la hora de afrontar el reconocimiento de los objetos, al igual que su comportamiento, es la eficacia y precisión con la que se determinen. En el caso de los sistemas de videovigilancia centrados en el tráfico, esto es de gran importancia y se ha ido incrementando en los últimos años. Existen varios detectores de objetos que no se centran en un ámbito específico como el propuesto en [61]. En cuanto al ámbito del control de tráfico en entornos urbanos [62,63] utilizan el encuadramiento de peatones en escenas complicadas analizando fotograma a fotograma a los mismos.

El modelado en 3D es una de las técnicas más usadas por las personas que se dedican a realizar seguimiento de objetos. Unidos a los anteriores, en [64] se propone un modelo probabilístico para realiza el seguimiento de múltiples objetos.

Otro tipo de metodologías es la división de la escena en zonas. Estas zonas son declaradas zonas de interés. Un ejemplo de estos sistemas son los representados en [65, 66].

La detección de objetos y su posterior seguimiento se dividen en 5 fases principales. Dichas fases varían un poco en función del método utilizado para el análisis de la imagen, pero en general, siguen las mismas. Éstas son:

- Segmentación del video: utilizando algoritmos específicos para ello se divide dicho video en *frames* o imágenes para su posterior análisis y determinación de objetos.
- Determinación de los objetos en la escena: una vez dividida la imagen y mediante el uso de algoritmos de comparación se determinan dónde y cuántos objetos existen en dicha escena.

- Seguimiento o *tracking* de los objetos: una vez identificado dónde se encuentran éstos se revisa la imagen vislumbrando el movimiento de los mismos.
- Identificación de los objetos: en función de sus características físicas y de su movimiento se encasilla a los objetos en un tipo concreto.
- Análisis del comportamiento: es la fase más importante y permite determinar el comportamiento de los objetos permitiendo identificar los patrones que se determinarán como normales y, más importante aún, las situaciones anómalas.

CAPÍTULO 3

TRABAJO DESARROLLADO

3.1. Requisitos del sistema

El objetivo de este trabajo es proporcionar un sistema capaz de detectar y determinar situaciones anómalas mediante la utilización de tecnología semántica. A su vez este sistema forma parte de un ente más global que condiciona ciertas características de éste.

El primer requisito del sistema tiene que ver con la adquisición de los datos. Los sensores visuales se encargan del procesado de la imagen proporcionando al motor semántico la información de interés. Esta información es enviada en formato XML por lo que el sistema deberá ser capaz de crear un canal de entrada de datos para recibir dicha información y extraer de cada XML la información que sea de su interés. Todo esto se ha de diseñar de tal manera que, aumentando el número de sensores, el sistema sea capaz de seguir procesando en tiempo real.

Una vez recibida y filtrada la información se deberá crear una ontología o base de conocimiento donde se van a incluir y posteriormente razonar dichos datos. En este caso el requisito consiste en la creación de dicho modelo para un sistema de videovigilancia de tráfico en el cual se incluyan las alarmas más frecuentes que puedan darse así como la identificación de las rutas y la categorización de los objetos en movimiento (vehículos y peatones).

El siguiente requisito es la realización del procesado de los datos en tiempo real. Es decir, se ha de analizar los datos provenientes de los sensores e inferir las alarmas existentes.

Crear un canal de salida para el envío al centro de control de las alarmas y los datos que puedan ser de relevancia es otro de los requisitos. En este caso la conectividad está condicionada por la red, previamente definida, de HuSIMS, la cual combina varias tecnologías como WiFi (*Wireless Fidelity)*, WiMAX (*Worldwide Interoperability for Microwave Access*) y el PLC (*Power Line Communications*), que permiten al sistema conectarse a redes densas de sensores tanto en redes de área local (LAN, *Local Area Network*) y redes de área metropolitana (MAN, *Metropolitan Area Network*).

El último de estos requisitos es que el sistema sea capaz de trabajar en conjunto con el resto de motores pudiendo analizar información procedente de ellos.

3.2. Arquitectura y componentes del sistema

La arquitectura del sistema HuSIMS se muestra en la Figura 1. En ella se puede ver la localización del motor semántico (*Semantic Engine)* dentro del sistema de control y monitorización (*Monitoring & Control System*), así como la composición del mismo formada por los otros dos motores, el sistema de video, el sistema de gestión de las alarmas y la red de comunicación entre todo el sistema.

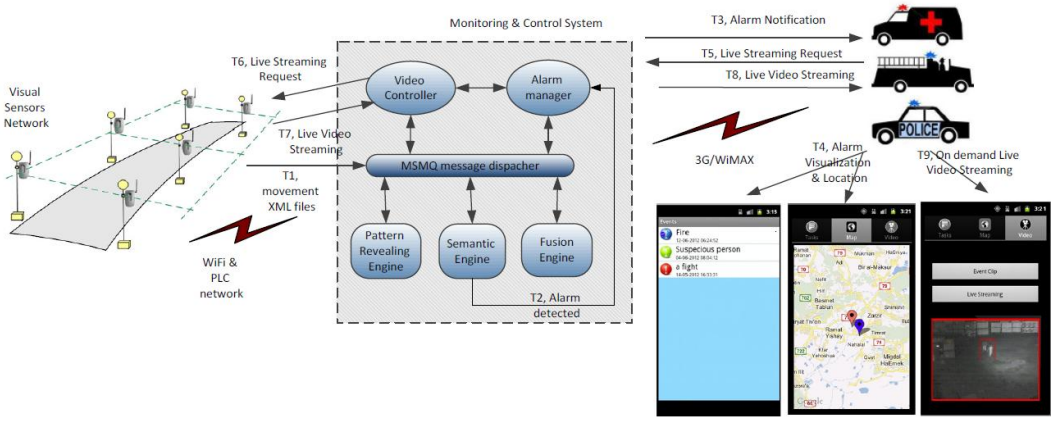


Figura 1. Arquitectura y funcionamiento del sistema HuSIMS

Sin embargo el propósito de este TFM es el diseño e implementación de un sistema de videovigilancia semántico que posteriormente se adaptará a dicha arquitectura. Este sistema se basa en un diseño formado por tres bloques.

El primero de ellos se centra en la obtención de la información de la escena. Este bloque denominado Sensorización, consta de una red de sensores visuales que procesan la imagen localmente y convierte el flujo de video en un conjunto de información en formato XML que contiene las características de los objetos de la escena y que posteriormente será enviada al centro de control para su procesado.

El segundo se dedica a la detección de las rutas existentes en el emplazamiento a partir de los datos enviados por el bloque de sensorización. Este bloque se encarga de determinar las trayectorias, los patrones de movimiento, lugares de parada, etc., de los objetos de la escena. El funcionamiento de este bloque se centra en el periodo de aprendizaje, ya que una vez determinado todo lo anterior, se determina que el sistema ya es funcional y se comienza a analizar la escena en búsqueda de alarmas.

Por último, el tercer bloque realiza el razonado semántico y posterior envío de la alarma. Una vez terminada la fase anterior, están determinadas las rutas y los elementos que, en condiciones normales, circulan por ellas. Esta fase transforma esa información al dominio semántico identificando los objetos como vehículos o peatones y determinando qué situación de alarma se ha producido en función de la ontología y las reglas semánticas definidas.

Todo lo anterior puede verse gráficamente en la Figura 2 [67].

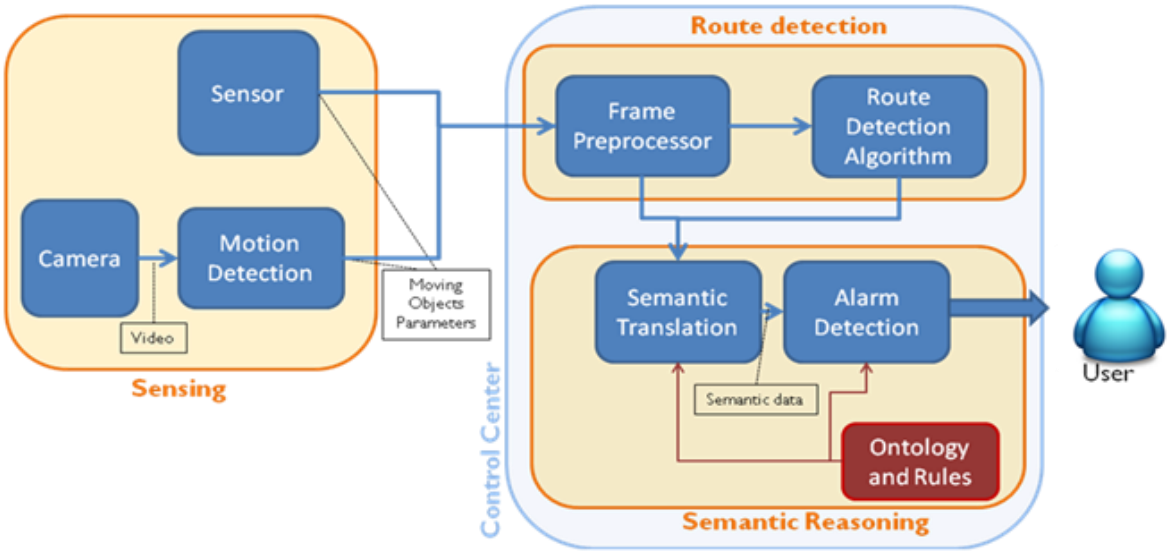


Figura 2. Arquitectura del sistema de videovigilancia semántico

En primer lugar se explicará la red de monitorización implementada para este sistema, las características de los sensores y la red de conexión utilizada por el sistema.

Posteriormente se pasará a describir de manera general el sistema de monitorización y control implementado donde se incluirá el motor semántico. Después se realizará una breve descripción de los dos motores que trabajarán en conjunto con él para, a continuación, explicar de una manera más precisa el motor semántico.

3.3. Sensorización

La toma de datos de la escena se realiza mediante la instalación de una red de sensores visuales inteligentes. El término inteligente se le asigna porque este sensor dispone de la capacidad de análisis de la imagen que captura. La salida de estos

sensores se compone de una cadena de XML donde se incluye la información de interés determinada por el algoritmo de análisis.

Una de las principales ventajas del uso de este tipo de sensores es que se evita el envío de dichas imágenes, manteniéndose la privacidad y eliminando costes, sobre todo a la hora de diseñar la red de comunicaciones. Además se permite reducir el consumo de los mismos a la vez que disminuir la complejidad de instalación.

A diferencia de otros sistemas planteados en la literatura, este tipo de sensores se basa en la captura de imágenes de baja resolución para lo cual se utilizan sensores del tipo VGA (*Video Graphics Array*) que captura las imágenes a unos 15-30 fotogramas por segundo (FPS).

Como se ha comentado, estos dispositivos realizan un procesado de la escena. Para ello cuentan con un procesador ARM9 (*Advanced RISC Machine 9*) que es el que se encarga de ejecutar los algoritmos de detección de movimiento. Hay que tener en cuenta que la capacidad de procesamiento de este tipo de procesadores es limitada y además que se ha de conseguir que dicho procesamiento sea lo más eficiente posible puesto que cuestiones como la duración de la batería son de gran interés e importancia.

Para ello lo que se ha realizado en este proyecto es la simplificación de los cálculos. En los sistemas tradicionales se utilizan herramientas matemáticas como análisis de Fourier, técnicas Gaussianas, etc. El problema es que se requiere hardware específico y un procesado de datos considerable. Con el fin de sustituir este enfoque el tratamiento de la imagen se inspira en la detección de colores, contrastes, movimiento, direcciones, etc. Se realiza un análisis de los píxeles de la imagen (hay que considerar que ésta es fija) estableciendo una franja de intensidad, por medio de un límite inferior y superior, en función de un histórico de comportamiento que se ha ido previamente desarrollando. El algoritmo encargado de la realización de este análisis se denomina *Hot pixel algorithm.* Por su parte, el almacenamiento de este histórico no supone ningún problema puesto que los datos guardados en él tienen un tamaño muy pequeño. Este almacenamiento se realiza en tablas que son consultadas cuando se detecta un cambio en el píxel. Ejemplo de esto es lo mostrado en la Figura 3.

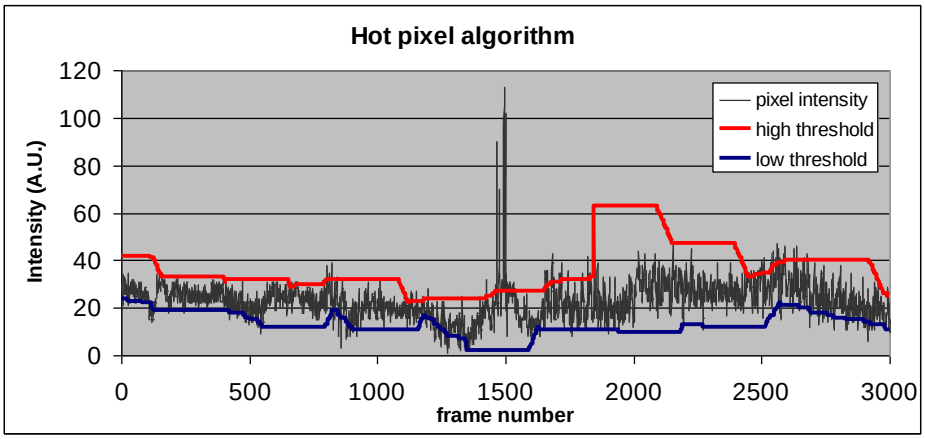


Figura 3. Algoritmo *Hot Pixel*

En dicha figura se puede ver como existe, entorno al número de *frame* 1500 un píxel que excede el valor del límite superior. Éste es marcando como caliente. Comprobando la conexión existente entre varios píxeles calientes se puede realizar una detección de un determinado objeto.

Sin embargo existen dos cuestiones importantes a la hora de evitar falsas alarmas. La primera de ellas es la resolución de la imagen, ya que cuanto mayor sea ésta, menos alarmas falsas se producirán pero mayor carga computacional necesitará. Adicionalmente a lo anterior, la sensibilidad con la que se configure el sistema (umbral definido por los límites superiores e inferiores comentados anteriormente) es la segunda de ellos, ya que si la sensibilidad es muy alta se evita que se determinen falsas alarmas pero puede que se elimine cierta información que pueda ser de interés. Debido a la necesidad de diseñar el sistema con una baja carga computacional, el parámetro en este caso a tener más en cuenta es la sensibilidad.

Mediante la realización de varios estudios se ha convenido que el establecimiento en el algoritmo de un marco de sensibilidad 1:100 consigue un bajo número de alarmas falsas, evitando la necesidad de aumentar el número de píxel y por lo tanto aprovechando en mayor medida la vida de la fuente de energía.

Todo esto hace que se puedan realizar grandes despliegues de redes de sensores con bajos costes económicos y con una mejor carga computacional del sistema global consiguiendo una mayor eficiencia y siendo económicamente más viables.

3.4. Componentes de red

La red implementada para este sistema está compuesta por dos subredes. La primera de ellas se encarga de la transmisión de los datos desde los sensores al centro de procesado, mientras que la segunda se encarga de distribuir las alarmas a los centros encargados de gestionarlas.

3.4.1. Red de adquisición de datos

Como se ha comentado anteriormente, la función de esta red es la de trasladar la información procedente de los sensores al lugar de procesamiento que en este caso será el centro de control o MCS. Esta información serán archivos XML.

En el caso de que la instalación del sistema de videovigilancia se diese en lugares de interior como hogares u oficinas, es posible utilizar la propia infraestructura del edificio como soporte para la transmisión de estos datos, ya que debido al pequeño tamaño de los mismos, no se necesita grandes capacidades de transmisión.

Si la instalación de la red se diese para situaciones de exterior, la comunicación se establecería mediante tecnologías como WiFi (802.11) o WiMAX (802.16). Este tipo de sistemas de comunicaciones inalámbricas permiten, gracias a alta capacidad y calidad de servicio, implementar grandes redes de sensores.

El diseño de la red utilizado en este sistema es flexible y permite dotar de una comunicación constante a los sensores con el MCS. Este diseño se basa en nodos de acceso inalámbricos (para el caso de la instalación en exteriores) fundamentado en una configuración multipunto de forma que los sensores visuales serán capaces de llegar a más de un nodo de acceso con el fin de proporcionar una mayor fiabilidad al sistema. Con esto se evita que en el caso de que una sensor no logre la comunicación con una célula pueda transmitir su información por medio de otro nodo. Todo esto se ha fundamentado en la comunicación basada en el protocolo 802.11n.

Por su parte en el caso de interiores se utilizarán dos tipos de tecnologías, WiFi y unas comunicaciones basadas en PLC *(Power Line Communications*). La fundamentación de utilizar dos tipos de tecnologías es la de que una es la base para las transmisiones, mientras que la otra se utiliza como medio de seguridad a la hora de mantener una conectividad fiable y continua al MCS.

3.4.2. Red de distribución de alarmas

En este caso el objetivo de la esta red es conectar el MCS con los servicios de emergencia encargados de la alarma. A diferencia del caso anterior, aquí sí que va a ser necesario un gran ancho de banda. Esto es debido a que, aunque el sistema está configurado en primer lugar para enviar mensajes de avisos a dichos servicios de emergencia, si estos lo consideran necesario se les enviará video en tiempo real de la escena.

Para ello se ha configurado una red basada en WiMAX (802.16e) que, gracias a su gran capacidad de transmisión de datos y sus características de servicio, permite la transmisión en tiempo real de dicha señal.

3.5. Sistema de Monitorización y Control

El Sistema de Monitorización y Control o MCS tiene como objetivo controlar el flujo de datos que se produce entre las diferentes partes del sistema. Éste se divide en varios módulos. La arquitectura de este sistema puede verse en la Figura 4.

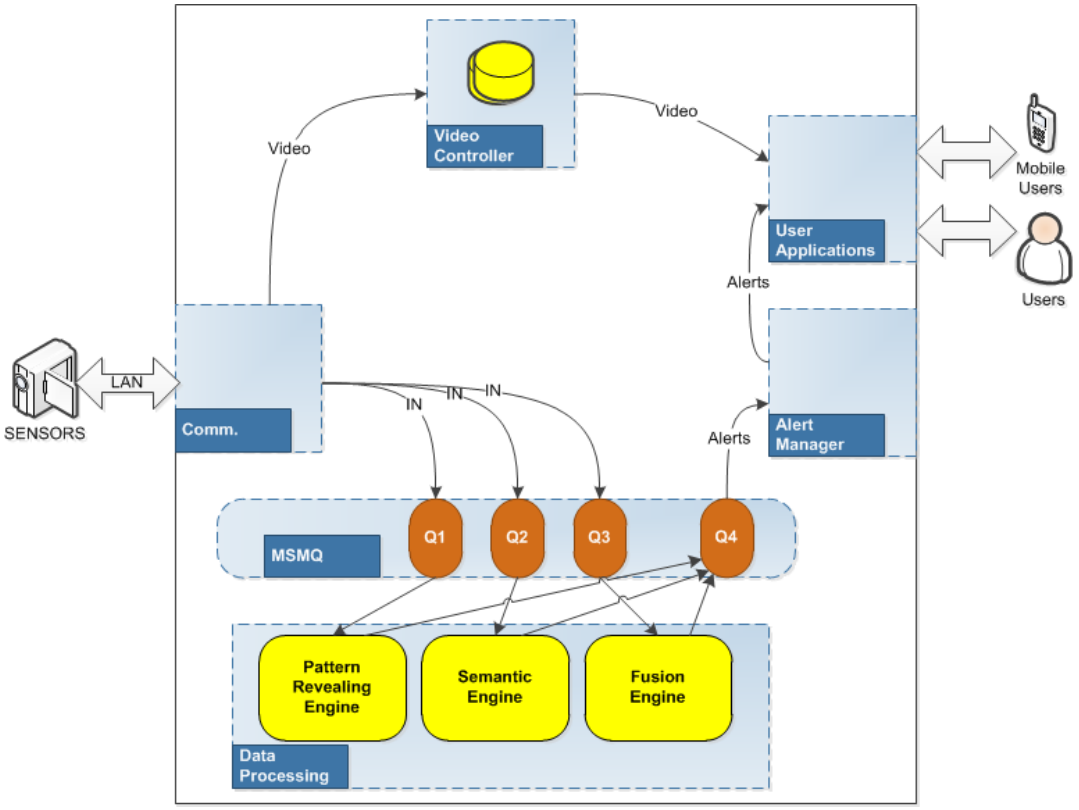


Figura 4. Arquitectura del Sistema de Monitorización y Control (MCS)

En éste destacan los siguientes módulos:

- Comunicación: recibe la información de los sensores visuales y reenvía ésta al módulo de procesamiento de datos.
- MSMQ (*Microsoft Message Queue Server*): tecnología utilizada para gestionar esta información y enviársela a los diferentes motores de análisis de datos.
- Módulo de procesamiento de datos: está compuesto por tres motores que se encargan de analizar la información enviada por los sensores visuales con el fin de determinar patrones de comportamientos y situaciones anómalas. Estos motores son los siguientes:
    - Motor de reconocimiento de patrones: basado en técnicas estadísticas, éste se encarga de determinar patrones normales. Se determina que existe una alarma cuando una situación se sale del patrón definido como normal.

    - Motor de fusión: este sistema se encarga de, mediante un conjunto de reglas, fusionar los datos procedentes de varias fuentes con el fin de determinar de una manera más segura la situación que está ocurriendo. Estas reglas son ampliables por parte de las usuarios permitiendo una cierta flexibilidad del sistema.
    - Motor semántico: a partir de una base de conocimiento formada por una ontología se caracteriza la escena dando significado y etiquetando los objetos y las relaciones entre ellos. Este sistema es capaz de identificar la alarma que se produce.
- Gestor de alerta: recibe las alarmas determinadas por el módulo de procesamiento y las envía al usuario para su gestión. Este sistema se encarga de satisfacer la petición de video realizada por el usuario mediante la comunicación con el centro del control.

3.5.1. Motor de Reconocimiento de Patrones

El motor de reconocimiento de patrones o *Pattern Revealing Engine* basa su funcionamiento en la familia de algoritmos [68]. Destacar que el algoritmo utilizado en este motor está patentado por la empresa C-B4, con lo que aquí sólo se mostrará un pequeño resumen de su funcionamiento dentro del proyecto HuSIMS.

Este motor consigue un modelado automático de las secuencia de datos KPI (*Key Performance Indicators* o Indicadores Clave de Desempeño). Este motor basa su modelo de funcionamiento en una estructura codificada por una red de árboles, la cual representa todos los patrones de datos que son significativos desde el punto de vista estadístico. Mediante predicciones basadas en el contexto se optimiza la eficiencia del modelo y se reestructura el tamaño del mismo [69,70].

Una vez construido el modelo, se consigue detectar anomalías dentro de las secuencias de datos. Los pasos para la detección de alarmas son las siguientes:

- Generación de un patrón: los nuevos datos se almacenan como posible ampliación de los patones ya generados. Esto representa la generación de una nueva rama adicional dentro de las existentes en el árbol del conocimiento.
- Clasificación del patrón: a cada nuevo patrón se le asocia una determinada probabilidad de suceso. Esta clasificación se realiza gracias a un algoritmo específico de clasificación el cual se separa del anterior (algoritmo de generación de patrones) con el objetivo de trabajar en tiempo real y mejorar el rendimiento del sistema.
- Toma de decisiones: al igual que en los dos casos anteriores existe un algoritmo encargado de realizar dicha función. Éste es fundamental a la hora de determinar si una situación es o no alarma. También se encarga de

agrupar los datos dentro de un patrón ya existente o crear uno nuevo en el caso de que no existan coincidencias.

- Agrupamiento de patrones: para facilitar el envío de los mismos, se realiza un agrupamiento por parte de otro algoritmo.

Este sistema es capaz de detectar situaciones anómalas en función de un cambio de los KPI o mediante la comparación de los patrones. Todo esto después de un periodo de aprendizaje, ya que en principio el sistema debe aprender qué situación se considera como normal.

3.5.2. Motor de Fusión

Al igual que en el caso anterior, el objetivo de este motor es analizar la información enviada por varios sensores y, mediante la ejecución de varios algoritmos (en este caso algoritmos de fusión) detectar situaciones anómalas automáticamente. El funcionamiento de este motor puede verse en la siguiente figura.

Figura 5. Funcionamiento del motor de fusión

Dicha figura muestra tres ventanas. En la superior se muestran los datos, ya seleccionados, procedentes de los sensores. En la ventana del centro se une toda la información obtenida por los diferentes sensores y se representa ésta en la escena. Procesada esta información se generan las alarmas correspondientes, las cuales se visualizan en la ventana de abajo.

Todo este procesado se realiza mediante el procesado de la información almacenada en una base de datos a través de reglas. Este procesado se realiza en tres niveles. En el primero y más bajo de ellos, el operario, desarrollador o administrador puede incluir las funciones matemáticas que se consideren necesarias mediante Java. Estas nuevas funciones se utilizan para aplicar nuevas normas en la segunda capa de manera que sean utilizadas como modelos de comportamiento. Éstos se utilizan en el nivel superior con el fin de dotar al objeto de comportamientos específicos y así evitar alarmas indeseadas o generar unas nuevas en función de ciertos aspectos.

En cuanto a su arquitectura, ésta se fundamenta en la combinación de una base de datos junto con tres submotores que trabajan de manera coordinada. Éstos son:

• Motor de Identificación de Datos Genéricos (GDIE): éste permite obtener la información de las diferentes fuentes de datos (XML, bases de datos, etc.) para su posterior verificación, formateo e inclusión de los mismos dentro de la base de datos.

• Motor de Fusión de Datos Genéricos (GDFE): se encarga de fusionar los nuevos datos con los existentes en la base de datos.

• Motor de Salida de Datos Genéricos (GDOE): su función es la de generar y enviar los informes de situación, los cuales pueden ser personalizados ya que el sistema admite la inclusión de plantillas específicas en función de lo requerido por el usuario.

Todo lo comentado anteriormente puede verse de manera gráfica en la Figura 6.

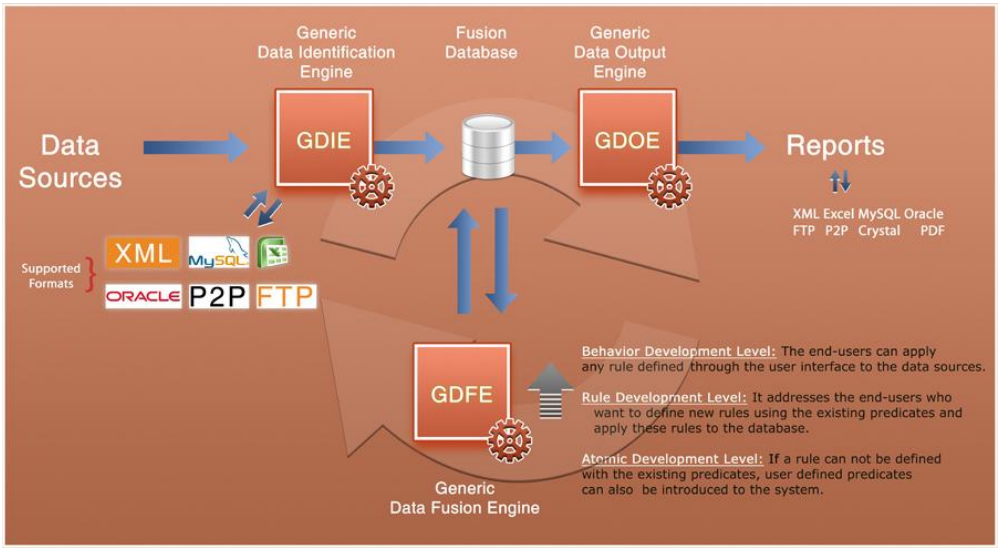


Figura 6: Estructura interna del motor de fusión

3.5.3. Motor semántico

Este motor proporciona un análisis semántico de los datos que recibe de los sensores visuales. Este tipo de procesado se basa en una interpretación de los datos similar a la que realizaría una persona. A diferencia del funcionamiento de sistemas estadísticos que lo que harían es determinar cuándo se ha producido una situación anómala, el semántico se dedica a caracterizar la información procedente de los sensores en función de una ontología. Ésta representa el conocimiento del sistema y se utiliza para clasificar a los objetos, principalmente los que se encuentran en movimiento, de la escena dentro de una categoría así como a dotarles de unas ciertas propiedades características de cada uno de ellos. Adicionalmente este modelo cuenta con un paquete de reglas, las cuales ayudan a definir de una manera más concisa las situaciones o a realizar ciertos razonamientos que no era posible realizarlos de otra forma. Esto proporciona, no sólo la detección de una situación anómala, sino su identificación y etiquetación, dotando al sistema de una mayor capacidad y eficiencia a la hora de afrontar dicha alarma.

De manera general la estructura interna (en cuanto a software o tecnologías utilizadas) del sistema es la mostrada en la siguiente figura.

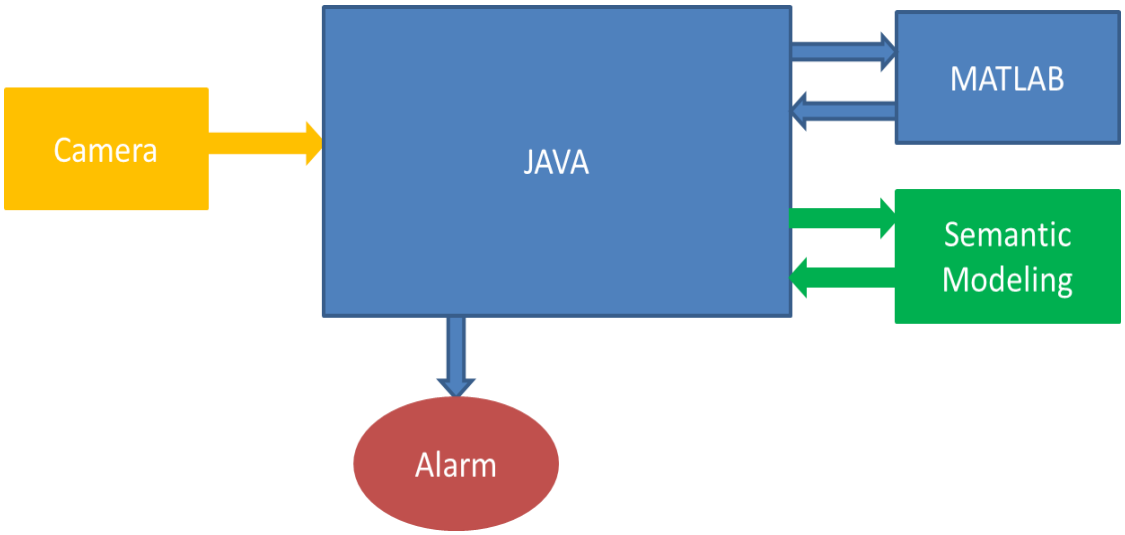


Figura 7. Estructura básica interna del motor semántico

En ella destacan tres bloques principales. El bloque de Java, que se encarga de gestionar el envío/recepción de los datos, el bloque de MATLAB que se encarga del procesamiento de las rutas y el bloque de Modelado Semántico que es el que se encarga de determinar las situaciones anómalas. A continuación se van a describir de una manera más detallada la estructura interna de cada uno de los bloques y sus fundamentos.

3.5.3.1. Bloque Java

El primero de ellos es el bloque Java. Como se ha dicho anteriormente, este bloque se encarga de la recepción y envío de los datos. En la Figura 8 se muestra la estructura interna de este bloque.

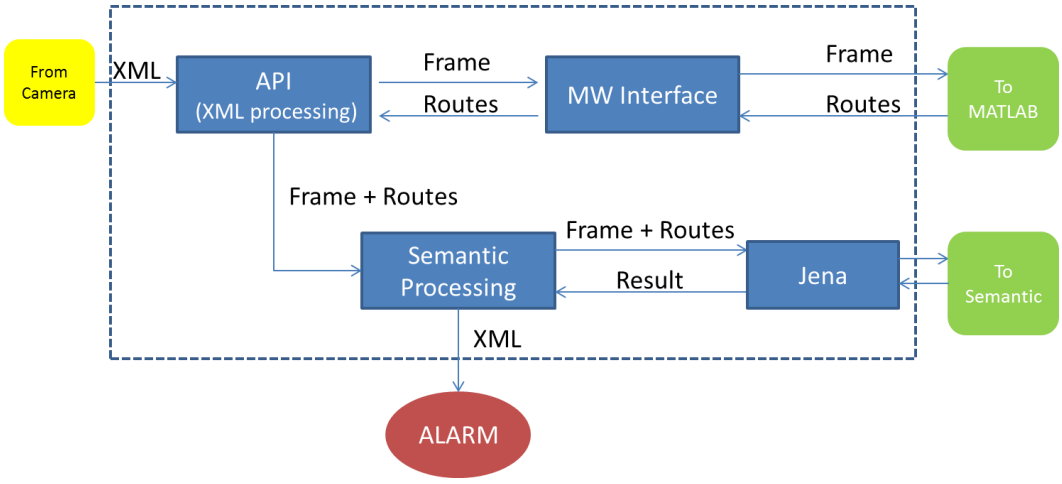


Figura 8. Estructura interna del bloque Java

Este bloque contiene una API (*Application Programming Interface*) que recibe los datos de la cámara (en formato XML), y que posteriormente son procesados con el objetivo de extraer de ellos sólo la información que es de interés para este sistema

concreto. La información de cada uno de los frames de video, ya filtrada se envía al bloque de MATLAB para su posterior procesado.

Una vez procesada la información, lo que este módulo recibe son las rutas generadas por los objetos de la escena. Estas rutas, junto con la información procedente de la cámara es enviada, previo a un procesado semántico para dotarla de la estructura correcta, al bloque de razonado semántico que es el que se encargará de determinar las situaciones anómalas. Esto será devuelto al bloque de Java, el cual se encargará posteriormente de enviar las alarmas.

3.5.3.2. Bloque MATLAB

El segundo de estos bloques es el bloque de MATLAB. En él se realiza la detección de rutas. La función de este bloque es la de implementar un modelo de la escena en el que aparezcan las rutas y los puntos de entrada y salida de los objetos, a los cuales se les denominarán como fuentes y sumideros respectivamente. El funcionamiento de este bloque se puede ver en la Figura 9.

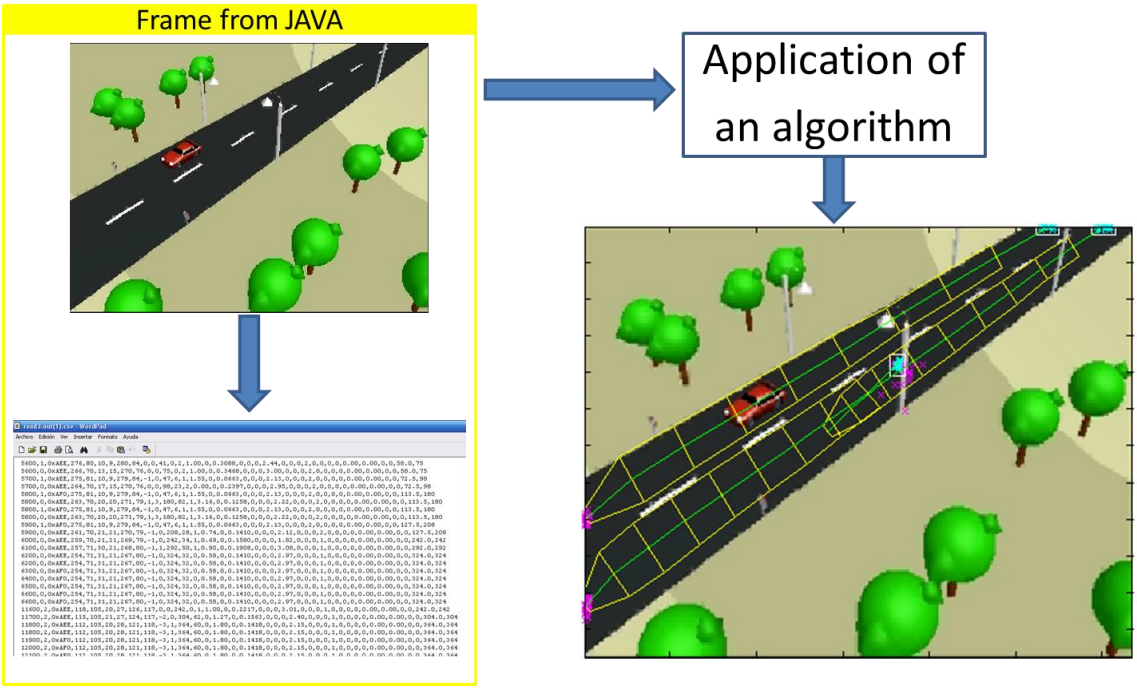


Figura 9. Funcionamiento del bloque de detección de rutas

Se consideran rutas aquellas zonas de la escena por donde los objetos se mueven habitualmente. En primer lugar se determinan las trayectorias que realizan los objetos individualmente que posteriormente se unen aquellas que coinciden y forma una ruta.

En cuanto a las zonas de entrada y salida, éstas son zonas donde aparecen y desaparecen los objetos. Es importante determinar este tipo de zonas para determinar de una forma más precisa los comienzos y finales de las rutas.

Por supuesto todo esto depende del ámbito en el que nos encontremos. Si por ejemplo nos encontrásemos monitorizando una carretera, una fuente podría ser el comienzo de una calle o determinarse como sumidero un paso de peatones. En el caso de estar en una videovigilancia de interiores, una fuente puede ser una puerta de acceso a un edificio. Determinar este tipo de localizaciones de la escena es trabajo del módulo semántico.

Volviendo al sistema de detección de rutas, el funcionamiento del mismo es similar a los encontrados en [43,47,71-73]. En ellos se utiliza la distancia de Hausdorff como medio de agrupamiento de las trayectorias en rutas dotando a cada una de las rutas de un parámetro que indica cuantas de estas trayectorias se han fusionado en ellas. Cuanto mayor sea este número más consistencia tiene ésta.

La Figura 10 contiene la representación de estas rutas formadas por una trayectoria central y dos envolventes (superior e inferior) que determinan el ancho de las mismas y los límites en los cuales se pueden encontrar los objetos para estar incluidos en dichas rutas. Las zonas marcadas de azul representan las fuentes mientras que las marcas moradas corresponden a los sumideros o zona donde desaparecen los objetos. Por otra parte puede verse una cierto sentido de movimiento en las envolventes, ya que éstas indican hacía donde se mueven los entes situados en ellas. Esto puede verse en la Figura 10.

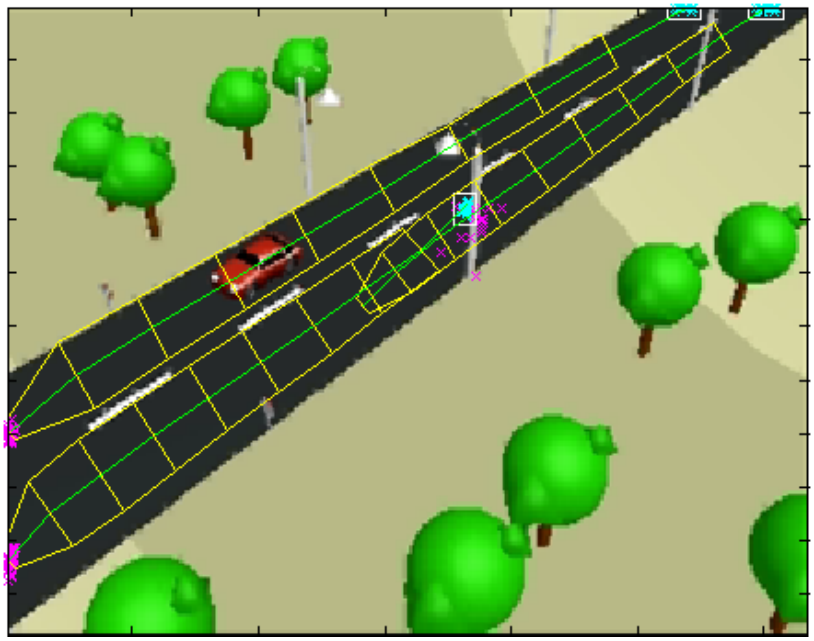


Figura 10. Ejemplo de detección de rutas

El funcionamiento del algoritmo a la hora de determinar la ruta puede resumirse en tres pasos.

- Se considera una trayectoria como el desplazamiento de un objeto desde que aparece en la escena hasta que desaparece. Pero sólo se tiene en cuenta ésta una vez que dicho objeto no se encuentre en ésta.
- Se compara la trayectoria que se ha determinado con las rutas existentes. Si no coincide con ninguna de las existentes se crea una ruta nueva. En el caso de que exista cierta similitud se fusionan o se crea otra nueva. Esto se determina mediante el uso de la distancia de Hausdorff y la distancia angular existentes entre las direcciones de la trayectoria y de la ruta. Se determinan unos ciertos parámetros y si se cumplen se fusionan.
- Por último se estudian los puntos iniciales y finales de dicha trayectoria. Para ello se crea una matriz donde se agrupan éstos y que posteriormente se agrupan mediante el algoritmo de agrupamiento DBSCAN. Se ha elegido este algoritmo ya que una de sus características principales es que no se necesita conocer a priori el número de agrupamientos, es decir, sólo se necesita saber la distancia máxima entre las trayectorias y el número de puntos mínimos para realizar la fusión de las mismas.

Este sistema, además de determinar las trayectorias y rutas, analiza parámetros de los objetos que se encuentran en la escena como la velocidad media de los mismos, tamaños medios, etc. Éstos se incluirán junto a la información de las rutas a la hora de enviarlos para su posterior análisis semántico.

3.5.3.3. Bloque de Modelado Semántico

La estructura interna del bloque de análisis semántico puede verse en la Figura 11.

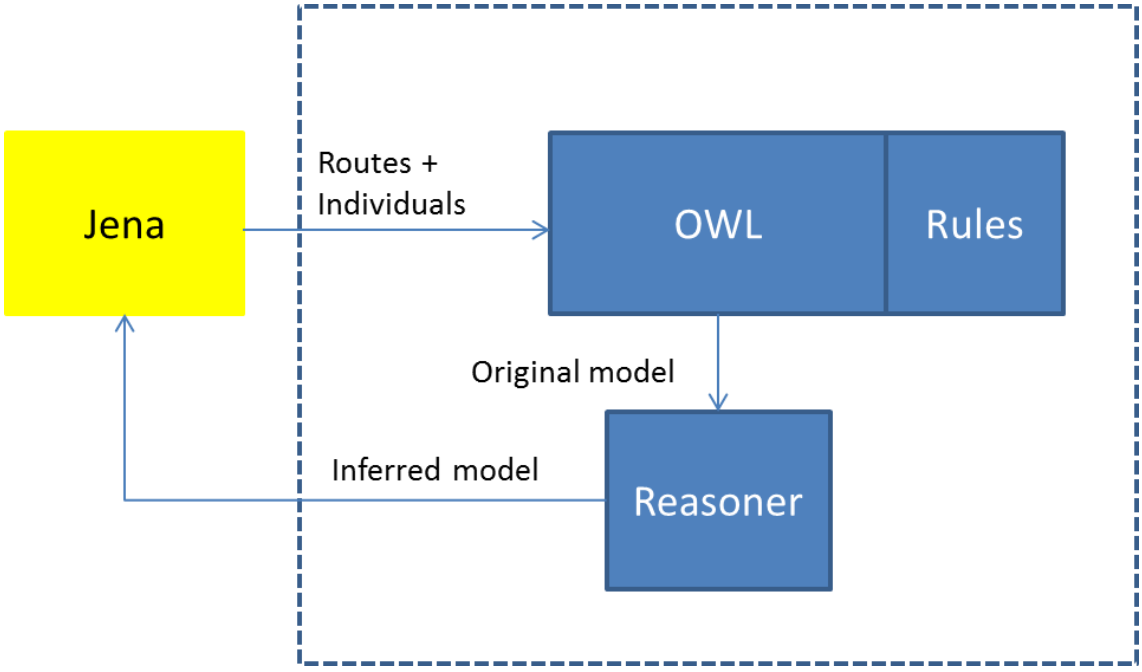


Figura 11. Estructura interna del modelado semántico

Destacar en primer lugar que es el Jena, *framework* de Java, el que se encarga de la transmisión de las rutas, de introducir la información en la ontología, de permitir el razonado y la inclusión de las reglas en éste con el objetivo de lograr una mejor inferencia de los datos. En este bloque se de distinguen otros tres sub-bloques:

El primero de estos bloques es llamado OWL, el cual contiene la ontología. Ésta es el modelo del conocimiento y representa una estructura jerárquica que permite la caracterización o encasillamiento de los diferentes entes que forman la escena. En la Figura 12 puede verse cómo es el funcionamiento de los datos.

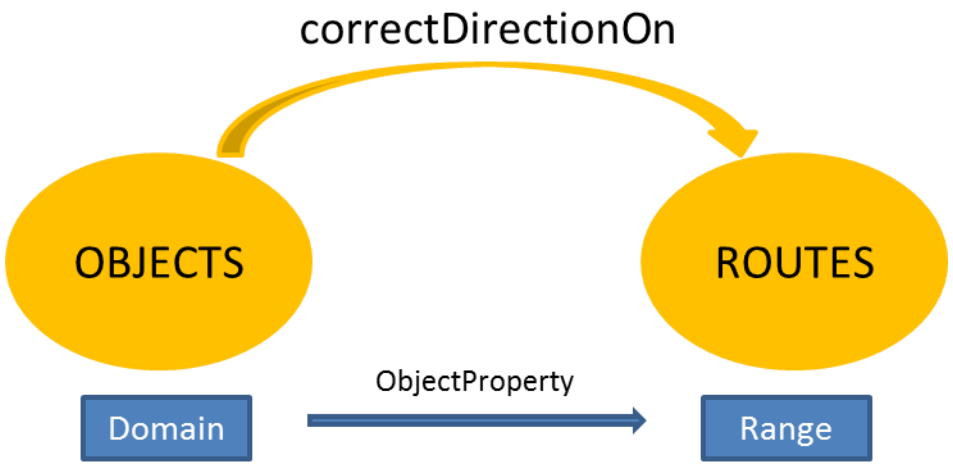


Figura 12. Fundamentación de los modelos de conocimiento semántico

Su funcionamiento se basa en un modelo de sujeto-predicado-objeto o tripletas. En el caso de la Figura 12 se comprende un sujeto (OBJECTS) el cual está relacionado con un objeto (ROUTES) mediante un predicado o propiedad (*correctDirectionOn*).

Cada una de las entidades se representa mediante una URI (*Uniform Resource Identifier)*. El conocimiento expresado en la ontología sigue la sintaxis mostrada en la Figura 13.

```
<!--
    http://www.semanticweb.org/ontologies/2011/5/wrongPlace.owl#correctDirectionOn
  -->
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2011/5/wrongPlace.owl#correctDirectionOn">
    <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2011/5/wrongPlace.owl#OBJECTS"/>
    <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2011/5/wrongPlace.owl#ROUTES"/>
  </owl:ObjectProperty>
```

Figura 13. Estructura interna de la ontología

En este caso para poder definir dicho modelo de conocimiento se ha utilizado el software Protégé [74]. Éste permite, de una manera sencilla, la definición de la estructura general de la ontología así como su testeo al incluir dentro de él un razonador. Para el testeo de estas ontologías se ha utilizado el razonador Pellet, el cual tiene un funcionamiento muy similar al que se terminado usando en el motor semántico. Dentro de los razonadores disponibles en este software, se ha elegido éste

por los siguientes motivos. Éstos son importantes a la hora poder validar el modelo de conocimiento.

- Código libre.
- Soporta reglas semánticas.
- Soporta diferentes versiones del lenguaje OWL (OWL-DL and Full).
- Soporta el razonamiento de datos en formato XML.

Un ejemplo de este tipo de estructura es el ilustrado en la Figura 14.

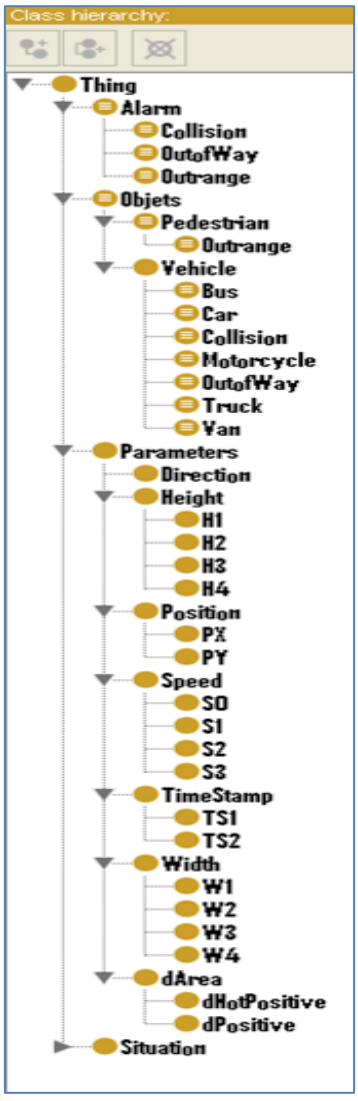


Figura 14. Definición de una ontología en Protégé

La ontología creada para este sistema de videovigilancia de tráfico se basa en el lenguaje ontológico OWL, concretamente en su versión Full, la cual permite una mayor expresividad a la hora de representar el conocimiento semántico. Este tipo de modelos es fácilmente ampliable mediante la fusión de una o varias ontologías dentro de una de ellas o creando una nueva. Con ello se dotaría al sistema de la flexibilidad

necesaria a la hora de detectar situaciones anómalas de ámbitos diferentes. Un ejemplo de esto podría ser la inclusión de una ontología específica sobre vandalismo a una que se encarga de la detección de accidentes en una calle transitada.

El segundo de estos bloques son las reglas semánticas. Éstas consisten en un complemento de conocimiento a la ontología y se utilizan para realizar operaciones lógicas y relaciones que no son posibles en la ontología y así servir de ayuda al razonador para crear las nuevas relaciones o conocimiento y así poder determinar las situaciones anómalas. En este caso se han utilizado reglas escritas en formato SWRL (*Semantic Web Rule Language*) con el fin de obtener compatibilidad con el *framework* de Java que posteriormente se encargara del procesado de la información. Un ejemplo de la sintaxis de estas reglas utilizadas en el motor semántico puede verse la Figura 15.

```
@prefix owl: http://www.w3.org/2002/07/owl#
@prefix wrongPlace: http://www.semanticweb.org/ontologies/2011/5/wrongPlace.owl#
@prefix xsd: http://www.w3.org/2001/XMLSchema#
@prefix rdfs: http://www.w3.org/2000/01/rdf-schema#
@prefix rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#


[r1:
        (?s rdf:type wrongPlace:OBJECTS)
        (?s wrongPlace:hasDirection ?od)
        equal(?od,13)
        ->(?s rdf:type wrongPlace:Notification)
        (?s rdf:type wrongPlace:StoppedVehicle)]

[r2:

        (?s rdf:type wrongPlace:OBJECTS)
        (?r rdf:type wrongPlace:ROUTES)
        (?s wrongPlace:ObjectsRoutes ?r)
        (?s wrongPlace:hasDirection ?od)
        (?r wrongPlace:routeDirection ?rd)
        min(?od, ?rd, ?cmin)
        max(?od, ?rd, ?cmax)
        difference(?cmax, ?cmin, ?result)
        greaterThan(?result,2)
        noValue(?s rdf:type wrongPlace:StoppedVehicle)
        ->(?s rdf:type wrongPlace:Alarm)
        (?s wrongPlace:hasAlarm wrongPlace:wrongDirection)]
```

Figura 15. Reglas semánticas

El último de estos sub-bloques es el razonador. Este módulo se encarga de realizar la inferencia de los datos creando nuevas relaciones y propiedades a los entes definidos dentro de la ontología. Su funcionamiento trata de imitar el pensamiento humano, generando relaciones semánticas y no sintácticas.

En la actualidad existen una gran variedad de razonadores (Pellet, Racer, etc.) que realizan perfectamente las funciones exigidas en este bloque. En el caso de este proyecto se ha utilizado un razonador genérico que permitía el uso de reglas SWRL en Jena.

El procesamiento de los datos es sencillo y se divide en dos partes, una primera que consiste en la traducción de los datos. Ésta se dedica a introducir los datos provenientes de los sensores en el modelo semántico categorizando el mismo dentro

de las clases definidas en él. La segunda de las fases es la encargada de realizar la inferencia de los datos por medio de la aplicación de las reglas (si las hubiese) y el razonador semántico. Esta fase es la que detecta las anomalías en la escena.

Por su parte el proceso de detección de las alarmas también se divide en dos partes, aprendizaje y detección.

- Aprendizaje: durante esta fase se detectan las trayectorias y se determinan qué objetos circulan por ellas y los comportamientos considerados como normales de los mismos y así poder asignarles a las rutas sus características (dirección, anchura, etc.).
- Detección: una vez que se han determinado los datos "normales" y las características de las rutas, se introducen éstos en la ontología para su posterior razonado y detección de situaciones anómalas.

Con todo lo definido hasta ahora, se puede vislumbrar una clara ventaja del sistema, su flexibilidad. Para adaptarlo a una nueva situación, la estructura del mismo no cambia, sólo habría que diseñar una nueva ontología específica para el mismo o modificar la actual para poder hacer frente a nuevas situaciones peligrosas.

Una vez determinada la alarma, ésta es enviada al MCS o centro de control donde será procesada. El formato de la información enviada al centro de información, es decir del XML enviado con la alarma producida puede verse en la siguiente figura.

```
<Alarm>
    <Source>1</Source>
    <SensorID>0001</SensorID>
    <Dest>4</Dest>
    <Alarm_Type>1</Alarm_Type>
    <Alarm_Severity>2</Alarm_Severity>
    <Orig_NO>
        <Type>
            <Orig_Field>Speed</Orig_Field>
            <Orig_Value>70</Orig_Value>
        </Type>
        <Type>
            <Orig_Field>Direction</Orig_Field>
            <Orig_Value>5</Orig_Value>
        </Type>
    </Orig_NO>
    <XML_DB_Ref>0000</XML_DB_Ref>
</Alarm>
```

Figura 16. Estructura del XML

Como resumen de lo anterior y para visualizar mejor la arquitectura y el funcionamiento del sistema, se incluye la Figura 17.

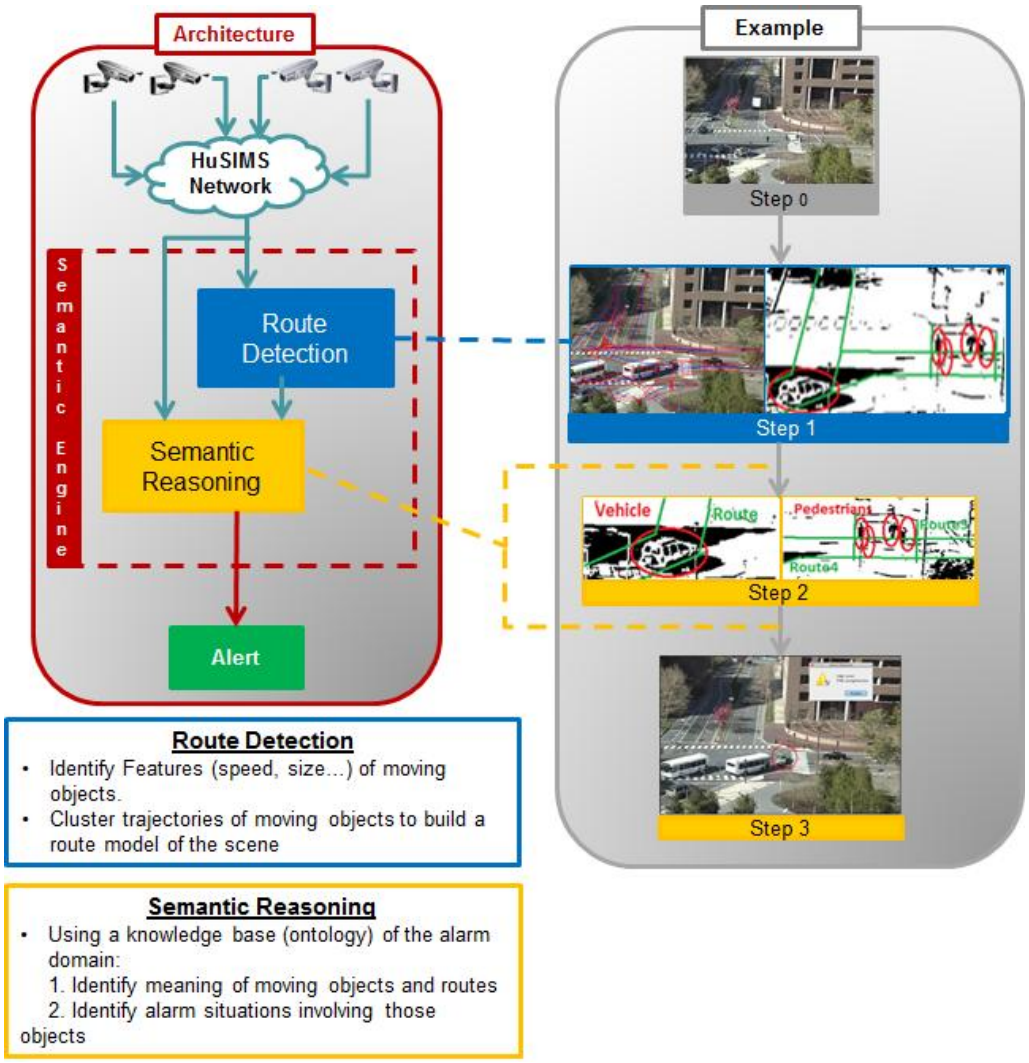


Figura 17. Arquitectura y funcionamiento del motor semántico

CAPÍTULO 4

ANÁLISIS Y VALIDACIÓN DE LOS DATOS

4.1. Validación del modelo semántico

El primer paso a la hora de comprobar el correcto funcionamiento del sistema es determinar que el modelo semántico es válido o que está correctamente formado. Para ello lo que se ha realizado es la implementación de varias ontologías que describen diferentes situaciones. Hay que tener en cuenta que una de los objetivos de este proyecto es que el sistema fuese flexible con lo que es interesante tener ciertos modelos de conocimiento que puedan aplicarse a diferentes situaciones.

Para testear estas ontologías se ha utilizado el software Protégé. Como se ha comentado anteriormente este programa permite la definición de una estructura semántica e incluye un razonador que además de conseguir inferir las nuevas relaciones en función de los datos y de las reglas semánticas, proporciona una realimentación al usuario al indicar, si los hubiese, errores en la confección de la ontología, como clases duplicadas o incongruencias.

Los datos proporcionados a este programa se han generado de manera sintética con el único fin de comprobar si los razonamientos obtenidos eran los correctos.

En la Figura 18 se encuentra representado el primero de estos ejemplos. En ella se define una base de conocimiento orientada al tráfico. Ésta es la base que se va a usar para lograr los objetivos del proyecto.

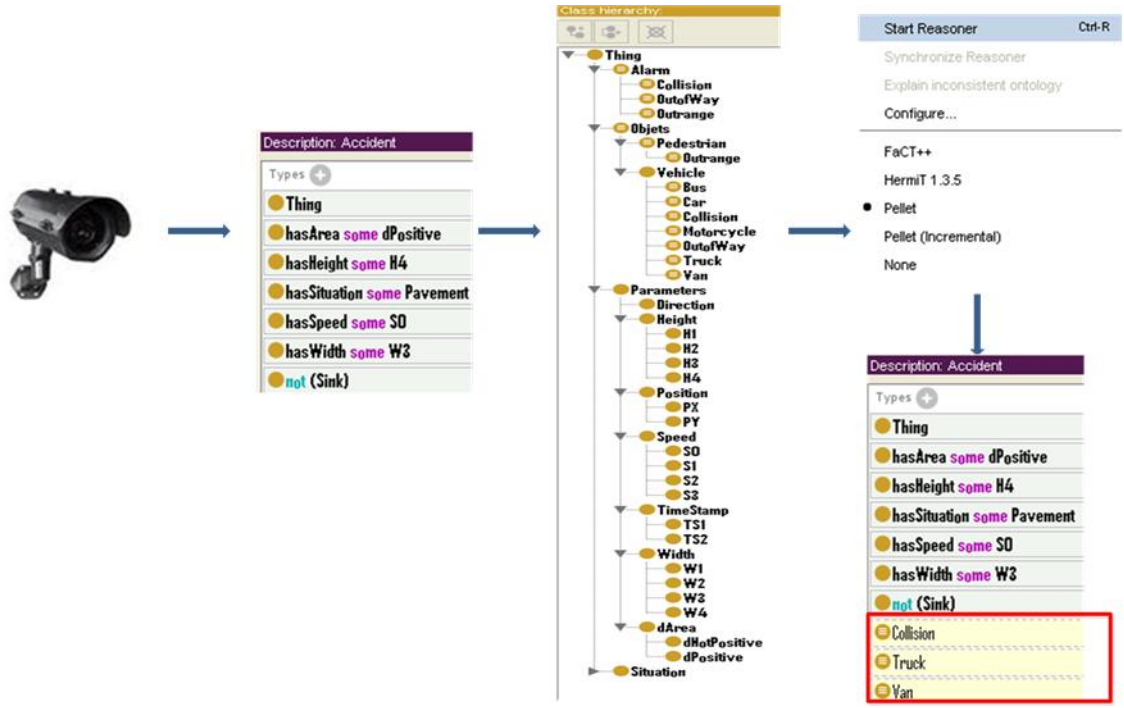


Figura 18. Ejemplo de funcionamiento para el análisis del tráfico

El funcionamiento es el siguiente. Una vez obtenidos los datos de la cámara, éstos son introducidos en la ontología. En ella se encuentra determinada una situación de alarma para dicha situación, concretamente una colisión entre dos coches. En primer lugar se determina que los objetos son vehículos, para ello se introducen ciertos parámetros que ayuden a determinar que lo son, como una velocidad, una altura y una anchura. Además se determina que dichos objetos deben de estar en la carretera.

Con los datos de la escena introducidos en la ontología se pasa a utilizar el razonador Pellet para obtener una inferencia de la escena. Una vez hecho esto se obtiene lo marcado en el cuadro rojo, que existe una situación de alarma denominada como colisión en la que se han visto involucrados dos vehículos, un camión y una furgoneta.

El segundo de los ejemplos se encuentra orientado hacia una situación de incendio. En el caso concreto de este proyecto, los sensores visuales no disponen de la capacidad de detectar focos de calor, con lo cual el siguiente ejemplo no podría implementarse. No obstante el hecho de tener un modelo específico de conocimiento para estas situaciones puede ayudar a determinar qué tipo de sensores son los necesarios a la hora de detectar este tipo de alarmas.

En este ejemplo concreto se ha determinado que cuando se determine una cierta temperatura y a la vez exista un viento con una determinada velocidad y una humedad en el ambiente determinada (estos parámetros han de estar previamente estudiados) existe una gran probabilidad de incendio y el sistema lanza una alarma. Esto puede verse en la Figura 19.

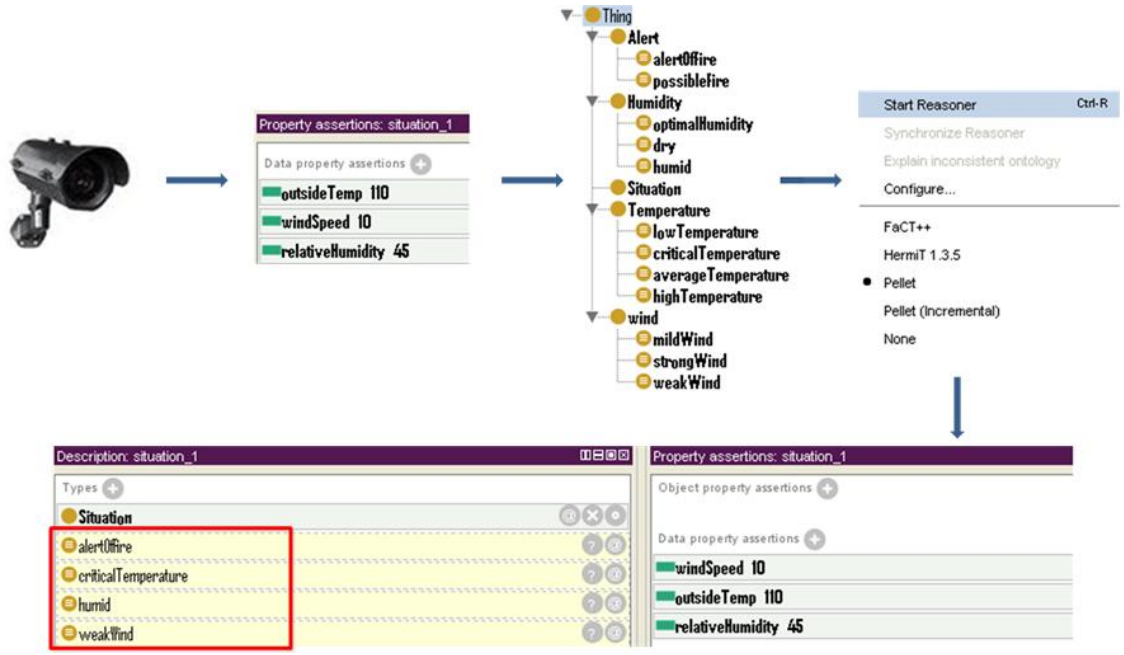


Figura 19. Ejemplo de funcionamiento para una situación de incendio

El último de los ejemplos es el concerniente a una situación de vandalismo. Esta situación se da cada vez más en la actualidad y cobra cierto interés determinar con una cierta antelación alarmas de este tipo. Para ello lo que se ha considerado es que se dispone de ciertos detectores de sonido. La ontología aquí modelada considera que una situación es de posible acto de vandalismo cuando el día actual es un día no laborable, se han determinado ciertos sonidos de alarmas y en el momento actual existen personas que corren. Esto puede verse en la Figura 20.

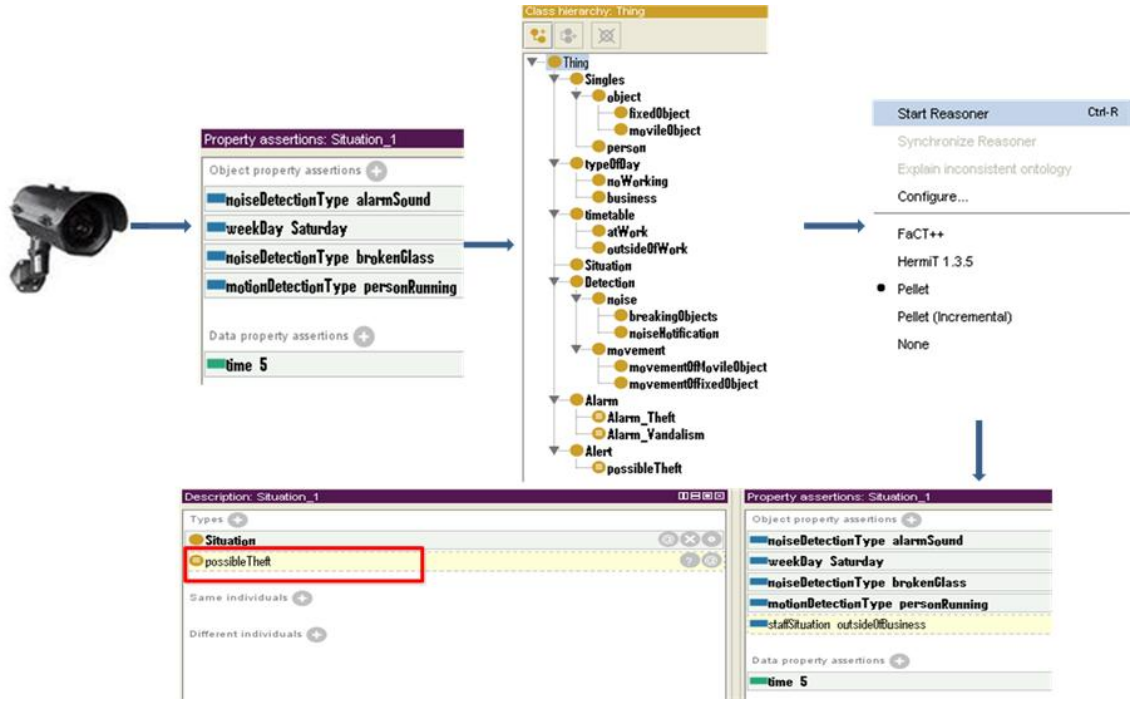


Figura 20. Ejemplo de funcionamiento para una situación de vandalismo

4.2. Casos de uso

El funcionamiento del sistema ha sido validado mediante la utilización de dos videos, ambos sintéticos. El primero de ellos se centra en la validación del motor semántico mientras que el segundo se ha utilizado en el sistema completo de HuSIMS. A continuación se pasará a describir ambos escenarios.

4.2.1. Vehículo en dirección contraria

Este video ha sido proporcionado por la empresa SQS (*Software Quality Systems*) que a lo largo del proyecto se ha encargado de la implementación de este tipo de animaciones para proporcionar al resto de socios un método de testeo de los motores de análisis.

En la Figura 21 puede verse el escenario a analizar. Consta de 4 carriles por donde circulan coches. En los carriles de la parte superior de la escena los vehículos se mueven de derecha a izquierda y en la parte inferior al contrario.

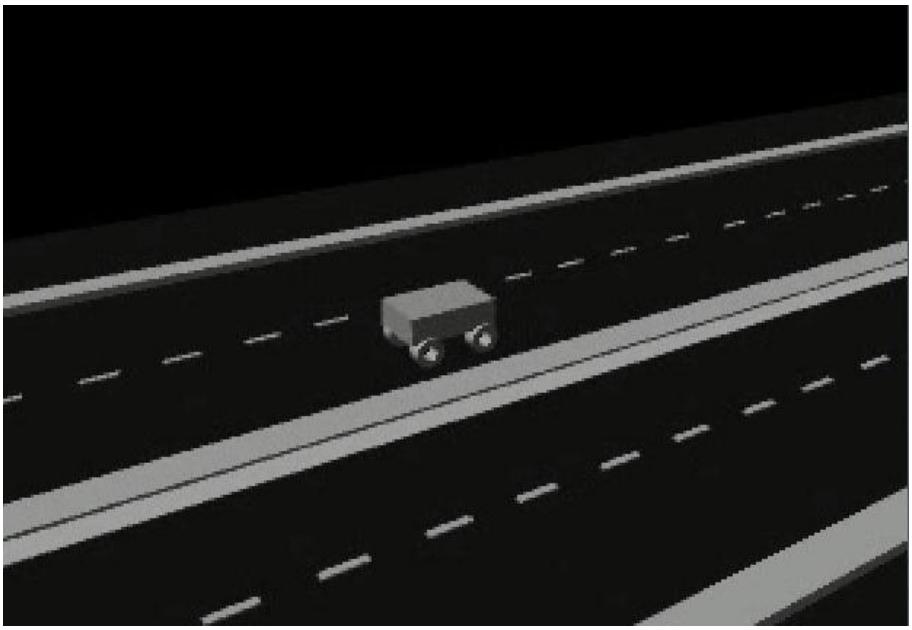


Figura 21. Visualización del escenario a analizar

Mediante un periodo de aprendizaje se van determinando las rutas de la escena. Esto puede verse en la Figura 22.

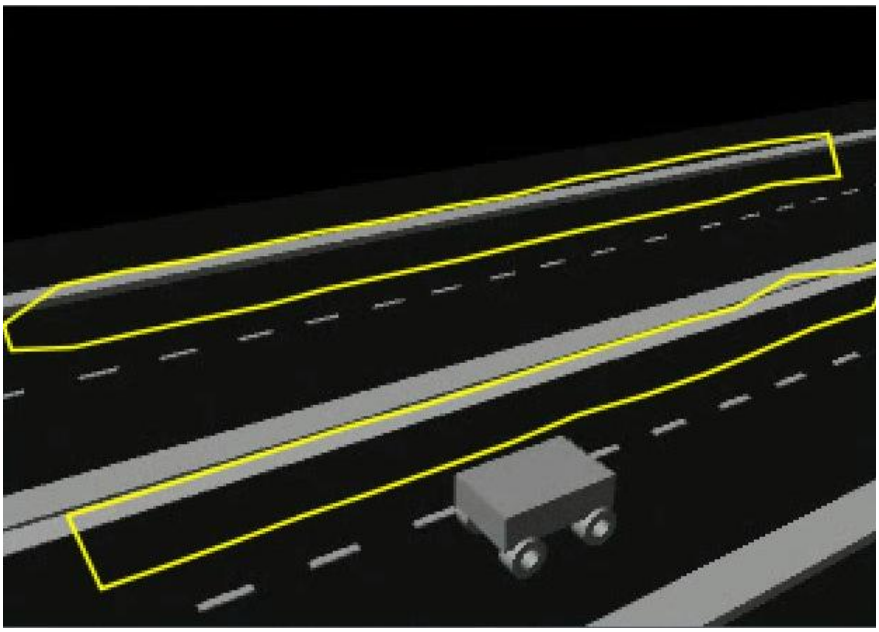


Figura 22. Determinación de las rutas

Determinadas las rutas y asignadas las direcciones propias de cada una de ellas se pasa a la fase de detección. En ella se encuentra que existe un objeto en una de las rutas el cual tienen una dirección opuesta a la que, en condiciones normales, se ha asignado a ésta. Esto implica que se considere una situación de alarma. En primer lugar, ésta es mostrada al usuario y posteriormente enviada al centro de control para su posterior procesado. La representación de este evento se puede visualizar en la Figura 23.

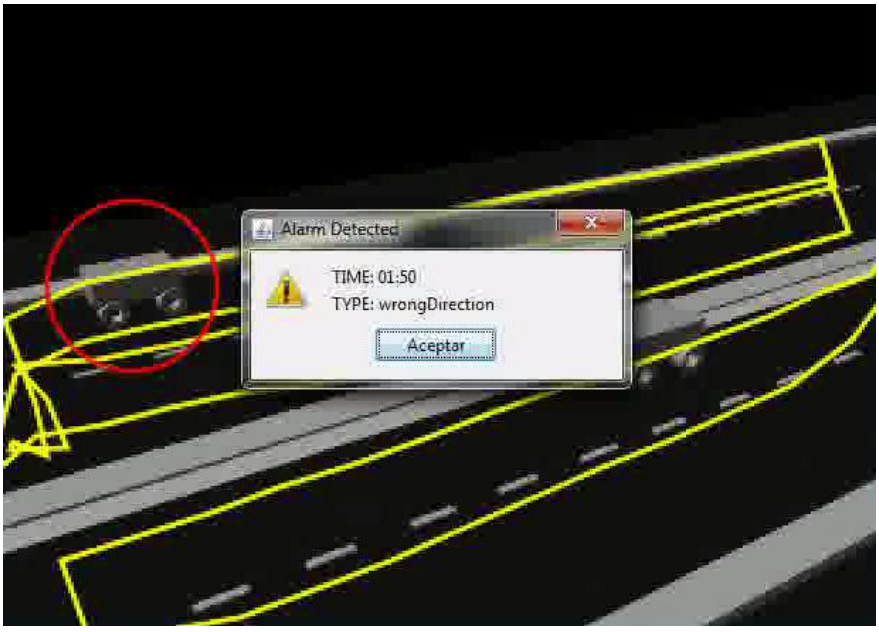


Figura 23. Determinación de una alarma

4.2.2. Gestión del tráfico

Este caso presenta cómo funciona el sistema implementado en HuSIMS y concretamente el funcionamiento de los tres motores en conjunto.

En primer lugar se encuentran los sensores visuales. Éstos procesan la información y cuando determinan que existe un objeto en movimiento dentro de la escena se genera un archivo XML que será enviado al MCS o centro de control. Un ejemplo de esto puede verse en la Figura 24.

```
<?xml version="1.0"?>
- <Objects>
  - <Object>
      <ID>9.0</ID>
      <Timestamp>24700.0</Timestamp>
      <ObjectType>0xAEE</ObjectType>
      <Left>302.0</Left>
      <Top>114.0</Top>
      <Width>12.0</Width>
      <Height>24.0</Height>
      <Xcog>194.0</Xcog>
      <Ycog>199.0</Ycog>
      <dX>0.0</dX>
      <dY>0.0</dY>
      <Area>288.0</Area>
      <dArea>0.0</dArea>
      <MotionState>0.0</MotionState>
      <Speed>2.0</Speed>
      <Direction>0.0</Direction>
      <CorrR2>0.5584</CorrR2>
      <TriggerPix>0.0</TriggerPix>
      <MaskedPix>0.0</MaskedPix>
      <Brightness>0.0</Brightness>
      <Thinness>2.25</Thinness>
      <Fire1>0.0</Fire1>
      <Fire2>0.0</Fire2>
      <Fire3>0.0</Fire3>
      <TotalObjects>7.0</TotalObjects>
      <TotalMovingObjects>0.0</TotalMovingObjects>
      <TotalStaticObjects>0.0</TotalStaticObjects>
      <MovedEccentric>0.0</MovedEccentric>
      <AreaEccentric>0.0</AreaEccentric>
      <MedianSpeed>0.0</MedianSpeed>
      <MaxSpeed>0.0</MaxSpeed>
      <TotalTooSlow>0.0</TotalTooSlow>
      <TotalTooFast>0.0</TotalTooFast>
      <MedianArea>13.0</MedianArea>
      <MaxArea>158.0</MaxArea>
    </Object>
</Objects>
```

Figura 24. Estructura de un XML enviado por los sensores

La tasa de envío de estos XML es configurable, al igual que la información enviada. Esta información será posteriormente filtrada por los motores en función de sus necesidades. Cabe destacar que cada objeto tiene un identificador por lo que se enviará un XML por cada uno de los objetos que se encuentren en la escena cada vez que se realice el procesado.

Una vez recibidos y filtrados los datos comienza el periodo de aprendizaje. En éste se determina qué comportamiento o parámetros son considerados como normales para posteriormente en el proceso de detección, determinar los anómalos.

En primer lugar se encuentra el motor de revelado de patrones. El funcionamiento de este motor consiste en identificar todos los patrones significativos basándose en la

posición, velocidad y dirección de los objetos. Éstos se representan por líneas azules. Cuando el sistema sale de la fase de aprendizaje pasa a la de detección, en la que dota a cada uno de estos patrones de una puntuación marcada por los puntos amarillos. Si es muy alta implica que el patrón es anómalo. Adicionalmente se incluye un umbral que determina cuándo esta situación fuera de lo común es una alarma. Éste es representado por una línea roja. Todo esto puede verse en la Figura 25.

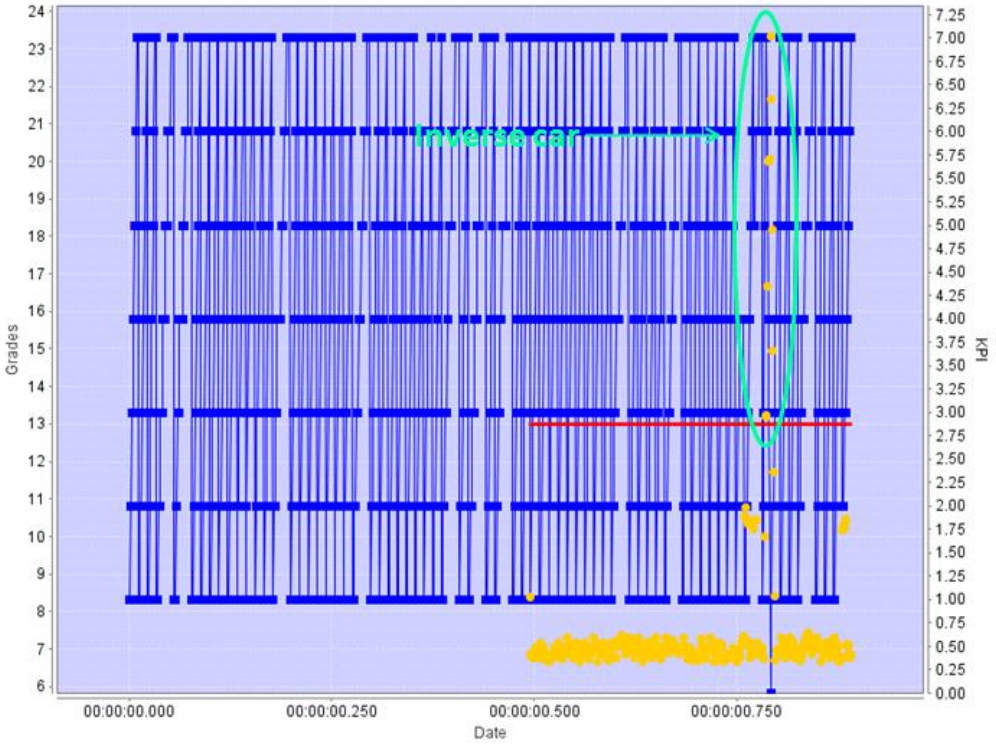


Figura 25. Detección de un patrón anómalo

En este caso el periodo de aprendizaje coincide con el espacio de tiempo antes de la determinación del umbral o línea roja. Una vez introducido en el periodo de detección comienzan a aparecer los puntos amarillos. Cuando dichas puntuaciones superan el umbral se determina que ha ocurrido una situación de alarma.

El caso del motor semántico difiere en algunos aspectos al anterior. En primer lugar ha creado una ontología específica para el control del tráfico. En ella se han de determinar los aspectos que se quieren determinar de la escena, como rutas, objetos, tipos de objetos, alarmas, etc. Sin embargo el funcionamiento del motor semántico, al igual que en el caso anterior, tiene un proceso previo a la detección de la alarma donde se construye el modelo de la escena. Éste es el aprendizaje. En él se determina las rutas por las que circulan los objetos y las propiedades de éstos, como dirección, velocidad, etc. Otra cuestión importante es determinar qué tipos de objetos circulan por cada una de las rutas para poder distinguir qué tipo de ruta es (carretera, acera, paso de peatones, etc.). Este periodo de aprendizaje puede verse en la Figura 26.

Figura 26. Aprendizaje del motor semántico

En dicha imagen se puede ver cómo en la ventana de la izquierda se muestra la escena actual, mientras que en la ventana de la derecha se muestran las rutas detectadas. Debajo de la ventana de la izquierda se muestran los datos razonados, es decir, que el *objeto14* es un vehículo y que la ruta denominada como *R5* es una carretera al circular vehículos por ella. Por último se encuentra la ventana situada en la esquinada inferior derecha. En ella se mostrarían las alarmas detectadas por el sistema.

A cada una de las rutas se les asigna una dirección que corresponde con la dirección, determinada como normal, de los objetos que circulan por ella. El valor de ésta corresponde a un número del 0 al 11 correspondiente con la posición de las agujas del reloj.

En el caso concreto en el que nos encontramos, a la ruta *R5* se la ha añadido la propiedad de que tienen una dirección igual al valor 5. Sin embargo un vehículo circula en dirección 11, contraria a la 5 por lo que se determina que es una alarma y se identifica a ésta como vehículo en dirección contraria o *wrongDirectionCar.*

Ambos motores funcionan de una manera similar. Destaca que el motor semántico, además de determinar que existe una situación de alarma, es capaz de etiquetar dicha situación.

Por otra parte se encuentra el motor de fusión. Éste se encarga de combinar la información obtenida por parte de los otros dos motores y así eliminar falsos positivos. Por ejemplo puede ser que uno de los motores detecte una situación que no sea una alarma mientras que el otro sí que lo detecte. Este motor, basándose en unas reglas establecidas, podrá determinar comparando las informaciones procedentes de los

motores si la alarma es correcta o no. También proporcionará la identificación de alguna anomalía que no esté definida en la ontología del motor semántico comunicándosela al MCS para su procesado.

4.2.3. Aplicación a otros escenarios

Uno de los objetivos definidos al principio del documento era la capacidad de este sistema de adaptarse a diferentes ámbitos de detección, en uno o varios escenarios. Uno de estos ámbitos podría ser la detección de actos de vandalismo o el control de multitudes.

En estos casos el sistema se encargará de determinar en primer lugar el comportamiento normal de las personas de la escena y posteriormente identificar comportamientos anormales. Para ello se deberá incluir ciertos sensores que sean capaces de determinar ruido o focos de calor. Aquí el motor semántico dispondrá de una ontología adaptada al lugar y, una vez acabado su proceso de aprendizaje, detectará las alarmas ocurridas. En ese proceso de aprendizaje se determinarán las áreas en las cuales las personas puedan moverse con total libertad, mientras que se identificarán ciertos sumideros o fuentes de objetos, los cuales acarrearían una alarma. Una vez determinada la alarma, ésta será enviada al MCS para su posterior análisis.

Por su parte el motor de detección de patrones actuará de forma similar, identificando los patrones “normales” durante el entrenamiento y detectando variaciones en los mismos para determinar alarmas.

En este caso el motor de fusión será capaz de rastrear al usuario o usuarios que hayan generado las alamas, indicando en todo momento su posición a los servicios encargados de su gestión.

4.3. Análisis del sistema implementado

A continuación se encuentran dos tablas que contienen la comparativa entre el sistema implementado y otros procedentes del mercado actual. En la primera de estas tablas se realiza una comparación entre los sensores visuales utilizados en este sistema en comparación a otros tres tipos de sensores. Estos sensores son los siguientes:

- VideoIQ: incluye análisis y grabación de la escena.
- Mobotix: se configura con cámaras de alta resolución y es capaz de enviar la información directamente a una base de datos.
- Axis: se basa en la utilización de cámaras IP (*Internet Protocol*).

La comparativa de los datos puede verse en la siguiente tabla.

Tabla 1. Comparación entre sensores de videovigilancia

	VideoIQ	Mobotix	Axis	HuSIMS Visual Sensors
Aim	Analyze and record video in place, allow remote access	Record video and transmit directly to storage	Record video	Describe a dynamic scene with thin XML data
Front end	Video camera, analytics and storage	Day/night megapixel camera + microphone	Camera	Visual sensor, interprets the scene and transmits thin XML description
Back end	Video servers performing image analysis	Storage device/ analytics channel	Video recorder/ storage/analytics channel	Statistical engines analyzing and correlating activity data from hundreds of visual sensors
Analytics	Intruder detection	None	None	Automatic detection of anomalous events, per sensor, per time of day
Infrastructure	Fiber-optics for transmitting live video, high power consumption			Low wireless bandwidth, low power consumption

La segunda comparación se centra en las características del sistema general. Para ello se han elegido varios sistemas de videovigilancia comerciales y se ha realizado un estudio de los propuestos en la literatura. Todo esto puede verse en la Tabla 2.

Tabla 2. Comparación entre los sistemas de videovigilancia

	HuSIMS	Current State of the Art	ADVISOR [75]	ARGOS [76]	DETER [77]	AVITRACK [78]
Objective	Alarm Detection	Recording video	Send warnings to human operators	Boat traffic monitoring	Alarm detection	Monitor and recognize activities
Resolution	Low	High	384 × 288 pixels	320 × 240 pixels	High	720 × 576
Bandwidth	Low	High	Ethernet IP Multicast	Local PC connection. (No specified connection with the control center)	Coaxial cable	1 Gb Ethernet
Storage	Rarely, only important video streams	Always	Yes (video + annotations)	Yes	Yes	Yes
Privacy	Gentle	Aggressive	Aggressive	Gentle	Gentle	Aggressive
Cost	Low	High	High	High	High	High
Type of Data Analyzed	Motion parameters	Video Signals	Video Signals	Video Signal	Video Signal	Video Signal
Domain	Multi-domain		Centered on metro stations	Maritime traffic detections (designed for Venice)	Vehicle and people detection	Airport Security

CAPÍTULO 5

CONCLUSIONES Y LÍNEAS FUTURAS

5.1. Conclusiones

La determinación de una situación de alarma implica el análisis de una gran cantidad de información, principalmente de alto nivel. El uso de la semántica y su modelo de conocimiento permite crear un sistema que, basado en el razonamiento humano, pueda utilizar esos datos e identificar esa situación anómala (en el caso de videovigilancia de tráfico situaciones como atropello, colisión, etc.).

La rápida y correcta resolución de este tipo de situaciones puede ser fundamental a la hora de gestionar la alarma e indicar a los servicios responsables qué ha ocurrido. Una automatización inteligente, que emule el conocimiento humano a la hora de identificar las alarmas ocurridas en estos sistemas, disminuye el tiempo de determinación de dicha situación anómala.

No obstante la adaptación de este tipo de sistemas a la hora de afrontar, dentro del mismo dominio de la escena, nuevas situaciones no es inmediata. Esto es debido a que las tecnologías semánticas basan su conocimiento en los términos introducidos en la ontología, por lo que si se define una ontología en un sistema que se encarga de la vigilancia del tráfico existente en una calle, el sistema no es capaz de determinar un incendio o actos de vandalismo sin la previa modificación de dicha ontología.

Con el objetivo de mitigar esto se ha dotado al sistema general de la capacidad de detectar, en función de métodos estadísticos, comportamientos irregulares o fuera de

lo normal en función a un histórico generado con anterioridad. Esto se consigue mediante la inclusión de otros dos motores de análisis.

Sin embargo el hecho de que el sistema pueda imitar el razonamiento humano gracias al uso de la semántica, proporciona al sistema general de grandes ventajas que no están al alcance de otro tipo de sistemas actuales. La facilidad de creación de ontologías y su capacidad de fusión dentro de un mismo sistema mitiga esa falta de flexibilidad.

La eliminación del procesado de las imágenes por parte de los sensores visuales ha permitido eliminar los costes, no sólo de equipos de procesamiento, sino de equipos de red y permitir un mejor procesado de la información, dotando al sistema de la capacidad de tratar la información de una red más amplia de sensores. A su vez, como el sistema evita el envío de las imágenes y el sistema semántico relaciona propiedades sin necesidad de identificar al objeto concreto, se eliminan los problemas de privacidad existentes actualmente con otros sistemas.

Por otra parte la capacidad de adaptación dentro de la plataforma de videovigilancia HuSIMS del motor semántico ha permitido crear, junto a las características adicionales proporcionadas por los otros dos motores, un sistema robusto capaz de detectar y determinar situaciones y comportamientos anómalos, a la vez que competente a la hora de comunicar al centro de control las alarmas generadas para su rápida atención.

5.2. Líneas futuras

El sistema aquí implementado proporciona unas bases interesantes para posteriores desarrollos. En este caso concreto se han considerado ciertos aspectos a la hora de continuar con esta metodología.

El primero de ellos es la adaptación del sistema completo a otros dominios. Conociendo la flexibilidad dada por el sistema, la creación de ontologías específicas para ellos es un desafío a la hora de permitir al sistema trabajar en diferentes escenas.

Esto conllevaría otro de los desafíos o de los caminos posibles para este tipo de sistemas y es la inclusión de nuevos sensores que permitan distintas mediciones.

Una vez construido el modelo de conocimiento e introducidos los nuevos sensores se ha de pasar a desplegar el sistema en situaciones reales. Esto puede darse sin necesidad de incluir nuevos escenarios puesto que lo que se ha proporcionado aquí son casos de validación en los que los datos eran sintéticos.

Otro de los caminos a seguir es la mejora de la interfaz de usuario. En ella se representan los caminos seleccionados pero se podría incluir más información como periodos de aprendizaje, localización, rutas aprendidas, etc.

Hasta ahora todo el sistema implementado recibe la información de un solo sensor. Una línea futura es comprobar su funcionamiento con varios de estos dispositivos.

Por último la creación de una base de datos local para el almacenamiento de estadísticas de los resultados de los análisis, así como de los objetos que han pasado por la escena y de las rutas implementadas es una de las vías de desarrollo más interesantes.

BIBLIOGRAFÍA

[1] HuSIMS Web Page. Accesible vía online desde http://projects.celtic-initiative.org/ HuSIMS/. (Último acceso Julio 2013).

[2] Zhu, J., Lao, Y., & Zheng, Y. F. (2010). Object tracking in structured environments for video surveillance applications. *Circuits and Systems for Video Technology, IEEE Transactions on, 20*(2), 223-235.

[3] Osais, Y. E., St-Hilaire, M., & Fei, R. Y. (2010). Directional sensor placement with optimal sensing range, field of view and orientation. *Mobile Networks and Applications, 15*(2), 216-225.

[4] Brutzer, S., Hoferlin, B., & Heidemann, G. (2011, June). Evaluation of background subtraction techniques for video surveillance. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (pp. 1937-1944). IEEE.

[5] Buttyán, L., Gessner, D., Hessler, A., & Langendoerfer, P. (2010). Application of wireless sensor networks in critical infrastructure protection: challenges and design options [Security and Privacy in Emerging Wireless Networks]. *Wireless Communications, IEEE, 17*(5), 44-49.

[6] Chen, M., González, S., Cao, H., Zhang, Y., & Vuong, S. T. (2013). Enabling low bit-rate and reliable video surveillance over practical wireless sensor network. *The Journal of Supercomputing*, 1-14.

[7] Kandhalu, A., Rowe, A., Rajkumar, R., Huang, C., & Yeh, C. C. (2009, April). Real-time video surveillance over IEEE 802.11 mesh networks. In *Real-Time and Embedded Technology and Applications Symposium, 2009. RTAS 2009. 15th IEEE* (pp. 205-214). IEEE.

[8] Durmus, Y., Ozgovde, A., & Ersoy, C. (2012). Distributed and online fair resource management in video surveillance sensor networks. *Mobile Computing, IEEE Transactions on, 11*(5), 835-848.

[9] Dore, A., Soto, M., & Regazzoni, C. S. (2010). Bayesian tracking for video analytics. *Signal Processing Magazine, IEEE, 27*(5), 46-55.

[10] Regazzoni, C. S., Cavallaro, A., Wu, Y., Konrad, J., & Hampapur, A. (2010). Video analytics for surveillance: Theory and practice [from the guest editors]. *Signal Processing Magazine, IEEE, 27*(5), 16-17.

[11] Technavio Analytic Forecast. Global Video Surveillance Market 2011–2015. Accesible vía online desde http://www.technavio.com/content/global-video-surveillance-market-2011–2015 (Último acceso Abril 2013).

[12] 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. *Signal Processing Magazine, IEEE, 22*(2), 25-37.

[13] Hu, W., Tan, T., Wang, L., & Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 34*(3), 334-352.

[14] Rota, N., & Thonnat, M. (2000). Video sequence interpretation for visual surveillance. In *Visual Surveillance, 2000. Proceedings. Third IEEE International Workshop on* (pp. 59-68). IEEE.

[15] 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.

[16] Roberts, L. History of Video Surveillance and CCTV. WE C U Surveillance 2004. Accesible vía online desde http://www.wecusurveillance.com/cctvhistory (Último acceso Abril 2013).

[17] Belbachir, A. N., & Göbel, P. M. (2010). Smart Cameras: A Historical Evolution. In *Smart Cameras* (pp. 3-17). Springer US.

[18] Thompson, M. (1985). Maximizing CCTV Manpower. *Security World, 22*(6), 41-44.

[19] Rodger, R. M., Grist, I. J., & Peskett, A. O. (1994, October). Video motion detection systems: a review for the nineties. In *Security Technology, 1994. Proceedings. Institute of Electrical and Electronics Engineers 28th Annual 1994 International Carnahan Conference on* (pp. 92-97). IEEE.

[20] Michalopoulos, P., Wolf, B., & Benke, R. (1990). Testing and field implementation of the Minnesota video detection system (AUTOSCOPE). *Transportation Research Record*, (1287).

[21] 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.

[22] Honovich, J. (2008). Top 3 Problems Limiting the Use and Growth of Video Analytics.

[23] 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.

[24] 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. *Signal Processing Magazine, IEEE, 22*(2), 25-37.

[25] Rinner, B., & Wolf, W. (2008). An introduction to distributed smart cameras. *Proceedings of the IEEE, 96*(10), 1565-1575.

[26] Rinner, B., Winkler, T., Schriebl, W., Quaritsch, M., & Wolf, W. (2008, September). The evolution from single to pervasive smart cameras. In *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on* (pp. 1-10). IEEE.

[27] 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.

[28] 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.

[29] Mucci, C., Vanzolini, L., Deledda, A., Campi, F., & Gaillat, G. (2007, November). Intelligent cameras and embedded reconfigurable computing: a case-study on motion detection. In *System-on-Chip, 2007 International Symposium on* (pp. 1-4). IEEE.

[30] 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.

[31] Hengstler, S., Prashanth, D., Fong, S., & Aghajan, H. (2007, April). MeshEye: a hybrid-resolution smart camera mote for applications in distributed intelligent surveillance. In *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on* (pp. 360-369). IEEE.

[32] 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.

[33] Sivic, J., Russell, B. C., Efros, A. A., Zisserman, A., & Freeman, W. T. (2005, October). Discovering objects and their location in images. In Computer Vision, 2005. ICCV 2005. *Tenth IEEE International Conference on* (Vol. 1, pp. 370-377). IEEE.

[34] Torralba, A., Murphy, K. P., Freeman, W. T., & Rubin, M. A. (2003, October). Context-based vision system for place and object recognition. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on* (pp. 273-280). IEEE.

[35] Tan, T. N., Sullivan, G. D., & Baker, K. D. (1998). Model-based localisation and recognition of road vehicles. *International Journal of Computer Vision*, *27*(1), 5-25.

[36] Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., & Poggio, T. (2007). Robust object recognition with cortex-like mechanisms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *29*(3), 411-426.

[37] 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.

[38] Nguyen, N. T., Bui, H. H., Venkatsh, S., & West, G. (2003, June). Recognizing and monitoring high-level behaviors in complex spatial environments. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on* (Vol. 2, pp. II-620). IEEE.

[39] 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.

[40] Remagnino, P., Shihab, A. I., & Jones, G. A. (2004). Distributed intelligence for multi-camera visual surveillance. *Pattern recognition*, *37*(4), 675-689.

[41] 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.

[42] Kim, Y. T., & Chua, T. S. (2005, January). Retrieval of news video using video sequence matching. In *Multimedia Modelling Conference, 2005. MMM 2005. Proceedings of the 11th International* (pp. 68-75). IEEE.

[43] Morris, B., & Trivedi, M. (2009, June). Learning trajectory patterns by clustering: Experimental studies and comparative evaluation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (pp. 312-319). IEEE.

[44] Zhang, Z., Huang, K., & Tan, T. (2006, August). Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on* (Vol. 3, pp. 1135-1138). IEEE.

[45] Sacchi, C., Regazzoni, C., & Vernazza, G. (2001, September). A neural network-based image processing system for detection of vandal acts in unmanned railway environments. In *Image Analysis and Processing, 2001. Proceedings. 11th International Conference on* (pp. 529-534). IEEE.

[46] 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.

[47] Piciarelli, C., & Foresti, G. L. (2006). On-line trajectory clustering for anomalous events detection. *Pattern Recognition Letters, 27*(15), 1835-1842.

[48] Liu, J., & Ali, S. (2010, August). Learning Scene Semantics Using Fiedler Embedding. In *Pattern Recognition (ICPR), 2010 20th International Conference on* (pp. 3627-3630). IEEE.

[49] Fernández, C., Baiget, P., Roca, X., & Gonzàlez, J. (2008). Interpretation of complex situations in a semantic-based surveillance framework. *Signal Processing: Image Communication, 23*(7), 554-569.

[50] Nakamura, E. F., Loureiro, A. A., & Frery, A. C. (2007). Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Computing Surveys (CSUR), 39*(3), 9.

[51] Friedlander, D. S., & Phoha, S. (2002). Semantic information fusion for coordinated signal processing in mobile sensor networks. *International Journal of High Performance Computing Applications, 16*(3), 235-241.

[52] Vellacott, O. (2010). The Olympic Challenge–Securing Major Events using Distributed IP Video Surveillance. *Edison, NJ: IndigoVision Inc*.

[53] Rougier, C., Meunier, J., St-Arnaud, A., & Rousseau, J. (2011). Robust video surveillance for fall detection based on human shape deformation. *Circuits and Systems for Video Technology, IEEE Transactions on, 21*(5), 611-622.

[54] Buckley, C. (2007). New York plans surveillance veil for downtown. *New York Times, 9*(3).

[55] Coaffee, J. (2004). Recasting the "Ring of Steel": designing out terrorism in the City of London?. *Cities, War, and Terrorism: Towards an Urban Geopolitics*, 276-296.

[56] Hughes, M. (2009). CCTV in the Spotlight: one crime solved for every 1,000 Cameras. *The Independent*, 25.

[57] Evans, I. Report: London No Safer for All its CCTV Cameras. The Christian Science Monitor 2012. Accesible vía online desde http://www.csmonitor.com/World/Europe/2012/0222/Report-London-no-safer-for-all-its-CCTV-cameras (Último acceso 2013).

[58] 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.

[59] Nghiem, A. T., Bremond, F., Thonnat, M., & Valentin, V. (2007, September). ETISEO, performance evaluation for video surveillance systems. In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on* (pp. 476-481). IEEE.

[60] Oh, S., Hoogs, A., Perera, A., Cuntoor, N., Chen, C. C., Lee, J. T., & Desai, M. (2011, June). A large-scale benchmark dataset for event recognition in surveillance video. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (pp. 3153-3160). IEEE.

[61] 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.

[62] Ess, A., Leibe, B., Schindler, K., & Van Gool, L. (2009, May). Moving obstacle detection in highly dynamic scenes. In Robotics and Automation, 2009. ICRA'09. IEEE International Conference on (pp. 56-63). IEEE.

[63] 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.

[64] 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* (pp. 467-481). Springer Berlin Heidelberg.

[65] Wojek, C., & Schiele, B. (2008). A dynamic conditional random field model for joint labeling of object and scene classes. In *Computer Vision–ECCV 2008* (pp. 733-747). Springer Berlin Heidelberg.

[66] Sturgess, P., Alahari, K., Ladicky, L., & Torr, P. (2009). Combining appearance and structure from motion features for road scene understanding.

[67] 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.

[68] Ben-Gal, I., Morag, G., & Shmilovici, A. (2003). Context-based statistical process control: a monitoring procedure for state-dependent processes. *Technometrics, 45*(4), 293-311.

[69] Ben-Gal, I., Shmilovici, A., Morag, G., & Zinger, G. (2008). *U.S. Patent No. 7,424,409*. Washington, DC: U.S. Patent and Trademark Office.

[70] SHMILOVICI, A., MORAG, G., & ZINGER, G. (2002). *WIPO Patent No. 2002067075*. Geneva, Switzerland: World Intellectual Property Organization.

[71] Roberts, L. (2004). History of Video Surveillance and CCTV. *WE C U Surveillance*.

[72] 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.

[73] Rinner, B., & Wolf, W. (2008). An introduction to distributed smart cameras. *Proceedings of the IEEE, 96*(10), 1565-1575.

[74] Protégé Web Page. Accesible vía online desde http://protege.stanford.edu/. (Último acceso Julio 2013).

[75] Siebel, N. T., & Maybank, S. (2004, May). The advisor visual surveillance system. In *ECCV 2004 workshop Applications of Computer Vision (ACV)* (Vol. 1).

[76] Bloisi, D., & Iocchi, L. (2009). Argos—A video surveillance system for boat traffic monitoring in Venice. *International Journal of Pattern Recognition and Artificial Intelligence, 23*(07), 1477-1502.

[77] 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.

[78] 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*.

ANEXO

ARTÍCULO PUBLICADO

Fernández J, Calavia L, Baladrón C, Aguiar JM, Carro B, Sánchez-Esguevillas A, Alonso-López JA, Smilansky Z; An Intelligent Surveillance Platform for Large Metropolitan Areas with Dense Sensor Deployment. *Sensors*. 2013, 13(6), 7414-7442. ISSN 1424-8220. Digital Object Identifier: 10.3390/s130607414.

- Índice de impacto: 1.953 (*Journal Citation Report* ISI) (2012)
- Área: *Instruments & Instrumentation*
- Posición: #9/57 (Q1)

Sensors **2013**, *13*, 7414-7442; doi:10.3390/s130607414

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

An Intelligent Surveillance Platform for Large Metropolitan Areas with Dense Sensor Deployment

Jorge Fernández [1,*], Lorena Calavia [1], Carlos Baladrón [1], Javier M. Aguiar [1], Belén Carro [1], Antonio Sánchez-Esguevillas [1], Jesus A. Alonso-López [2] and Zeev Smilansky [3]

[1] Dpto. TSyCeIT, ETSIT, Universidad de Valladolid, Paseo de Belén 15, Valladolid 47011, Spain; E-Mails: lcaldom@ribera.tel.uva.es (L.C.); cbalzor@ribera.tel.uva.es (C.B.); javagu@tel.uva.es (J.M.A.); belcar@tel.uva.es (B.C.); antsan@tel.uva.es (A.S.-E.)

[2] Alvarion Spain SL, Parque Tecnológico de Boecillo, Edificio CEEI, 3.15, Valladolid 47151, Spain; E-Mail: jesus.alonso@alvarion.com

[3] Emza Visual Sense Ltd., 3 Hayozma st., Kfar Sava 44422, Israel; E-Mail: zeev@emza-vs.com

***** Author to whom correspondence should be addressed; E-Mail: jfergut@ribera.tel.uva.es; Tel.: +34-983-423-704; Fax: +34-983-423-667.

Received: 22 April 2013; in revised form: 17 May 2013 / Accepted: 22 May 2013 / Published: 7 June 2013

Abstract: This paper presents an intelligent surveillance platform based on the usage of large numbers of inexpensive sensors designed and developed inside the European Eureka Celtic project HuSIMS. With the aim of maximizing the number of deployable units while keeping monetary and resource/bandwidth costs at a minimum, the surveillance platform is based on the usage of inexpensive visual sensors which apply efficient motion detection and tracking algorithms to transform the video signal in a set of motion parameters. In order to automate the analysis of the myriad of data streams generated by the visual sensors, the platform's control center includes an alarm detection engine which comprises three components applying three different Artificial Intelligence strategies in parallel. These strategies are generic, domain-independent approaches which are able to operate in several domains (traffic surveillance, vandalism prevention, perimeter security, *etc.*). The architecture is completed with a versatile communication network which facilitates data collection from the visual sensors and alarm and video stream distribution towards the emergency teams. The resulting surveillance system is extremely suitable for its deployment in metropolitan areas, smart cities, and large facilities, mainly because cheap visual sensors and autonomous alarm detection facilitate dense sensor network deployments for wide and detailed coverage.

Keywords: smart visual sensors; surveillance; intelligent detection; security

1. State of the Art in Intelligent Video Surveillance Systems

The concern for personal safety and security in public places is rising everywhere. Sales of video surveillance systems are expected to grow at a Compound Annual Growth Rate (CAGR) of 14.33% over the period 2011–2015 [1]. Research on video surveillance systems is therefore a hot topic, with many different areas being addressed at different levels. Some examples of typically very active areas are the cameras and visual sensors [2–4], the infrastructure to provide a network of sensors [5–8] and the field of intelligent video analytics [9,10].

New generations of visual sensors are constantly being explored. Some new sensors are enhanced with movement tracking capabilities based on Bayesian models [2], with effective background subtraction that enables later automatic video analysis [4] and with new planning tools that allow to configure their optimal sensing range, field of view and orientation in order to minimize the cost of the network of sensors [3].

Deploying dense networks of sensors is challenging because of power and bandwidth limitations. Both issues are due to the fact that the network of sensors need to multiplex hundreds of video streams in real-time, and in the uplink, while most of the network equipment is rather optimized to support more traffic in the downlink. To deal with this problem, several solutions have been proposed. An energy efficient image transportation strategy based on motion detection has been designed in [6] to tackle both the power limitations and the bandwidth limitation issues since it reduces the amount of frames to be transmitted. The limitations of 802.11 DCF MAC protocol specially in multi-hops scenarios has been addressed in [7] by designing a Time-Synchronized Application level MAC protocol (TSAM) capable of operating on top of existing 802.11 protocols; in addition, it can be used with off-the-shelf hardware and is capable of improving throughput and providing bounded delay. Another interesting approach to the performance of the network of sensors proposes a new schema for fair resource allocation operating at the application layer [8]. This system identifies critical network-wide resources and allocates them to the application level messaging units (called events) as opposed to regular flow of frames. In general, the idea is to provide some kind of scheduling capabilities to get some fairness and prioritization that complement a contention-based technology like 802.11.

Video analytics, which targets the autonomous understanding of events occurring in a monitored scene, is one of the main research trends in video surveillance systems. The idea is to have intelligent systems that are not only able to record video that can be later used as forensic evidence of crime or accidents but to help to avoid crime, terrorism and accidents in a proactive fashion [10]. Much of the research efforts in this field focuses on automatic tracking of objects and persons in motion [2,9]. However, video analytics systems process potentially sensitive information when using person tracking, behavior analysis or person identification [11].

Some companies are delivering sophisticated platforms to the market. A relevant player like IBM has presented a smart surveillance system which provides not only the capability to automatically

monitor a scene but also the capability to manage the surveillance data, perform event based retrieval, receive real time event alerts and extract long term statistical patterns of activity [12].

Testing tools like ETISEO [13], a system for automatic testing of video surveillance systems, are already available as well as benchmarks [14] with realistic datasets that include actions by non-actor subjects.

Video surveillance systems are mainly intended to be used in big open public areas, such as municipalities, major events (such as Olympic Games [15], popular marathons) or in critical infrastructure protection [5] but they can also be applied in eHealth systems (e.g., in surveillance for fall detection of elderly people [16]). There are many examples of this trend, such as the 3,000 cameras of the Lower Manhattan Security Initiative [17] (aimed at picking up activities such as package delivery, and completed with car plate recognition, radiation detectors and mobile roadblocks) or London's "ring of steel" [18].

However, while the current trend is to continue deploying dense and wide area video surveillance sensor networks in order to provide all-time, all-location security, there are concerns that these systems are currently not reaching the promised levels, which has resulted in an increasing criticism across the society and media [19,20]. Therefore, it is becoming apparent that while this kind of systems has a huge potential, it is currently being underutilized, mainly because there is an overabundance and overflow of data which does not directly translates into information: video streams are properly captured, but intelligent analysis algorithms are not taking full advantage from them, and it is not feasible to have human operators watching them in real time. In the end, many of the videos are used for obtaining evidence after the crime has happened, but not for prevention or emergency management.

The focus of European Eureka Celtic Human Situation Monitoring System (HuSIMS) project is in two spots: first, in the video analytics systems to improve intelligent event detection; and second, in employing cheap intelligent sensors which reduce the amount of raw data sent to analysis to facilitate deployment of hundreds of thousands of units to public bodies, but also the implementation of private surveillance networks in smaller areas.

The visual sensors used in the project do not send in a regular basis video streaming but XML files containing perceived movements that are then processed by a powerful backend application in order to identify potential alarm situations. To the best of our knowledge, there is so far no attempt to build a system that combines the utilization of three different search engines with complementary approaches in the analysis of the visual sensors output. Our approach combines statistical analysis in the pattern revealing search engine, semantic web technology in the semantic search engine and relies on a rules-based system for the fusion of events and detected alarms. The utilization of text-based information in the analysis enables HuSIMS to be gentler with privacy matters.

After this summary of the state of the art, Section 2 presents the new requirements for surveillance systems and their overall system design; Section 3 deals with the intelligent visual sensors employed; Section 4 describes the network components to send the information to the Monitoring and Control System, including the intelligent alarm detection engines which will be described in Section 5; Section 6 presents two use cases; Section 7 compares the global HuSIMS system against the other solutions. Finally, Section 8 summarizes the conclusions of this work.

2. Vision, Principles and Architecture

2.1. New Surveillance Systems Requirements

HuSIMS intends to provide an intelligent video surveillance system for deployments in wide urban areas. The HuSIMS approach tries to differentiate from the main trend in current video surveillance systems that use High Definition cameras that required lots of bandwidth to transmit the video to control nodes. Instead of that, HuSIMS employs low-cost analytic visual sensors that are able to track objects in motion and send low-weight XML files instead of heavy video streaming to a backend application. Of course, this reduces the amount of required bandwidth and therefore the cost of the required network. This low cost feature will enable large and dense deployments in municipalities as part of the future Smart Cities, can be a good solution for the coverage of crowed scenarios like concerts and sport events, and facilitates the adoption and deployment of the system by other smaller private initiatives to secure critical private facilities [21].

HuSIMS' main target is to provide real-time alerts for *irregular activity* in both indoors and outdoors scenarios. The latency of sending XML files is much lower than that of sending video streaming in a regular basis. The files will be processed by the backend application that includes three alarm detection engines working in parallel in order to bring rich alarm detection.

HuSIMS defines as *irregular activity* an activity that is important to identify and that can actually be identified. For example a parameter violation situation in which a wrong object is detected in a wrong place at a wrong time and/or moving in a wrong way: this may translate, depending on the domain, into a car accident in which a vehicle is out of its way and has invaded the sidewalk, an individual breaking a security perimeter, or a crowd running away from a fire. How these situations are detected by the different intelligent engines will be detailed in Section 5.

The system will transmit video only during an alarm situation. When a situation has been confirmed as alarm state, the system operators and the first responders will be able to request live video streaming from the alarmed place. The same approach is applied to video storage, where only short periods of a few minutes will be stored—those related to confirmed alarm situations and that will be specifically requested to the visual sensors upon alarm detection.

The fact that the system is using low resolution visual sensors provides an additional advantage regarding governance and politics: the provided image does not usually allow recognition of people's faces, which makes the system more respectful of privacy. This enables HuSIMS to provide a novel and compelling trade-off between privacy and enhanced security in a public space.

Summarizing, the key points of HuSIMS are:

- To minimize the processing and intelligence required in the visual sensors and therefore its cost. The visual sensors used in HuSIMS are intended to be simple, low resolution ones with a limited processing capability which makes impossible the utilization of advance techniques to recognize faces or objects. The processing made at the visual sensors is limited to objects/persons detection and movement tracking.
- To minimize the amount of data to be transferred in the network of sensors. It the visual sensors are not expected to provide the main processing then the intelligence has to be transferred to a control center where the processing will take place. However, in such a case, the required

bandwidth in the network would be huge, particularly in very dense networks of visual sensors. In HuSIMS, the visual sensors send to the processing node only the parameters resulting of the movements' tracking concerning the objects in the observed scene. The video signal will still be available to the human operators that ask to manually analyze a given scene but this transmission is done only on demand, this wouldn't be the default situation.

- Alarm detection based on the objects parameters. Since the video signal does not progress to the control center and the visual sensors don't make on their own the alarm detection, the processing made in the control center is based on the objects parameters, e.g., size, direction or speed.
- Distribution of alarms and relevant data in real time. The results of the intelligent analysis will progress using a network infrastructure that will include the public authorities, security corps and first responders. This network will be used to transmit the alarms notification, their location, and scripts for first assistance depending on the type of alarm detected to the human operators in real time. The alarm subscribers could be in motion when approaching the alarm place and can request on demand video signal using client apps in their smartphones, laptops and tablets.
- To use highly flexible connectivity solutions. The network infrastructure used in HuSIMS combines several technologies like WiFi (802.11), WiMAX (802.16) and PLC (Power Line Communications) that enable the system to connect dense networks of sensors in both Local Area Networks (LAN) and in Metropolitan Area Networks (MAN). HuSIMS provides a Self-Organized Network (SON) whose nodes have self-configuration and self-healing capabilities.

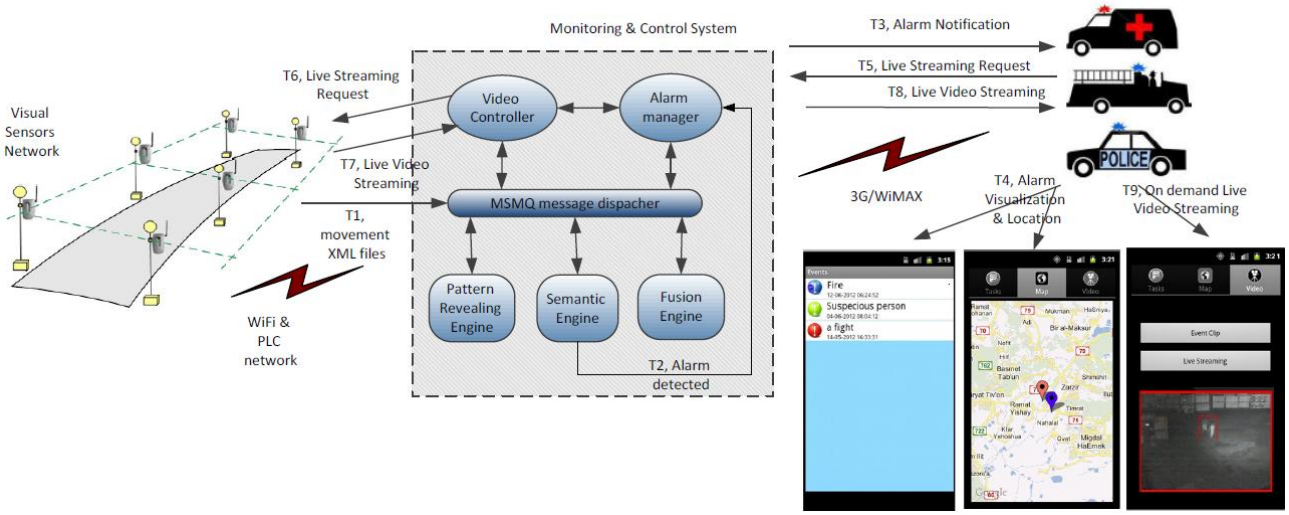
2.2. HuSIMS System Design

HuSIMS operates at three functional levels: movement tracking, alarm detection and alarm notification, as seen in Figure 1 (T1–T4). A dense network of visual sensors is on charge of the movement detection and tracking. Each visual sensor that detects movement in the scene that is watching sends a XML file with the object's motion parameters to the Monitoring and Control System (MCS). The visual sensors are able to filter shadows, weather conditions and many uninteresting noise-movements like those of the trees' leaves. The network of visual sensors is based on a combination of WiFi (802.11) and Power Line Communications (PLC) for covering both indoors and outdoors scenarios.

The XML files sent from the visual sensors are processed in the MCS by three different search engines that work in parallel detecting alarm situations at the MCS. Each search engine uses a different search strategy. The Pattern Revealing Engine aims at learning recurrent patterns in the motion parameters and raises alarms when the detected pattern does not correspond to one of the learned ones. The Semantic Engine translates to formal semantic formats the content of the XML files and uses ontologies for understanding what is happening on the scene and subsequent alarm detection. The Fusion Engine is a rule-based system that is able to detect alarm situations and is also able to correlate alarms detected by the other two search engines with the appropriate rules. When an alarm is detected the First Responders (police patrols, ambulances, firemen vehicles) are notified and receive in a mobile application information about the detected alarm, a script with tasks to be done to tackle the alarm situation and the alarm's location.

When an alarm state has been determined and notified, the first responders can request a live streaming session using their mobile application. The relevant visual sensor will receive the request and provide the streaming session that will be properly encoded by the Video Controller component at the MCS (see Figure 1, T5-T9).

Figure 1. HuSIMS: alarm detection, notification and video streaming request upon alarm detection.



3. Intelligent Visual Sensors

3.1. State of the Art in Security Cameras

The natural modality of obtaining information on the world around us is visual—we obtain more than 90% of information about the world surrounding us with our eyes, and about half of the human brain is busy with interpretation of this information. Even small animals, birds and insects can easily interpret the visual world surrounding them—this with a fraction of the computational power of an ordinary computer. Unfortunately, current artificial vision systems are usually bulky, expensive, and instead of having cognitive capabilities are often limited to image recording.

The concept of using motion cameras for security dates back to the 1940s, with the first commercial analogue video surveillance systems becoming available in the 1970s [22]. Initially, a person was required to continually monitor the video stream. Introduction of video recorders and later video multiplexing allowed greater flexibility in viewing, storing, and retrieving the (analog) video. Digital, or IP video cameras were introduced in the 1990s. Only recently have the sales of IP cameras outgrown those of analog cameras in the surveillance world. A recent, thorough and historical review can be found in [23].

It was quickly recognized the human observation of surveillance streams is ineffective [24]. As computers became prevalent, automated analysis of video became a topic of interest [25]. Such algorithms often used simple frame-to-frame differencing and thresholding, so called Video Motion Detection (VMD). Among the first applications were traffic monitoring [26] and intruder detection [27]. Automated analysis of video sequences has captured academic interest, which has earned the name "video analytics", has been unsuccessful. While the size of the video surveillance market is above $14 billion, the size of the "analytics" market barely reaches $100 million, less than 1% of the total

market; numerous companies that were active in this field disappeared, and none of them reached sales of over $10 million. The selling hype, together with under-performance, so discouraged users that today "analytics" has become synonymous with "non-functionality" [28] and the term has practically disappeared.

Automated analysis of video streams usually relied on a separated architecture, where the cameras relay live video to a central video server facility. Such servers often attempt to perform costly operations such as edge detection, tracking, object recognition and even gesture recognition, applying complex computational and mathematical operations [29,30]. There are very few products where the analysis is tightly coupled with the acquisition camera. These are sometimes termed "edge cameras", although this includes cameras that record their video on-board, rather than remotely. Another approach meant to reduce the reliance of video surveillance on expensive infrastructure is the hybrid surveillance sensor, where a camera is in deep sleep mode, being awakened by a low-power sensor (e.g., a Passive IR, PIR). Again, this is simply a recording or transmitting instrument, with no analytical capabilities. Finally, while exaggerated claims are being made as to what automated analysis of video can achieve ("identifying suspicious people", "left luggage at a busy subway station"), such products have yet to penetrate the market and gain commercial traction. Thus, there are no current solutions to real-time alerting of irregular events in metropolitan areas, particularly such that require low-cost, low-complexity infrastructure.

In the research community, the topic of "smart cameras" and "embedded smart cameras" has gained considerable interest (cf. reviews in [23,31,32]). Some recent studies considered tracking performed by a network of embedded smart cameras [33,34] and specific hardware architectures [35]. Recent designs for low-energy surveillance systems include a hybrid low-resolution stereo "sensor" coupled with higher resolution color camera [36], development of light-weight algorithms for embedded smart cameras [32,37], and smart camera networks for agricultural applications [38].

Figure 2. Example of an irregular event with a strong visual signature: single person loitering near the community center main entrance door (resulted in a foiled arson event, courtesy: security department of the city of Nes Ziyona, Israel).

3.2. HuSIMS Intelligent Sensor

The intelligent visual sensor developed for HuSIMS is novel in several respects. First, it normally produces no video output—only a digital (XML) description of the activity observed in its monitored

scene. Second, it aims to reduce drastically the device's profile—in terms of size, cost, required bandwidth, power consumption, and installation complexity, while keeping high end performance in varying weather and illumination conditions. In contrast with some of the publications mentioned above, the HuSIMS visual sensor relies on standard components and architecture, with a CIF or VGA CMOS sensor (mobile phone type) and ARM9 processor running proprietary, low computational cost algorithms, at 10–15 frames per second (FPS). These features make it a true visual sensor, and not a camera. All this allows the sensors to be densely deployed, each one monitoring a limited city area (e.g., a road junction, a pub entrance, or a bus stop). Finally, the visual sensor is meant to serve the HuSIMS system, where *irregular events with a strong visual signature* (Figure 2) are detected at real time and relevant verification video is streamed to policemen or other security professionals.

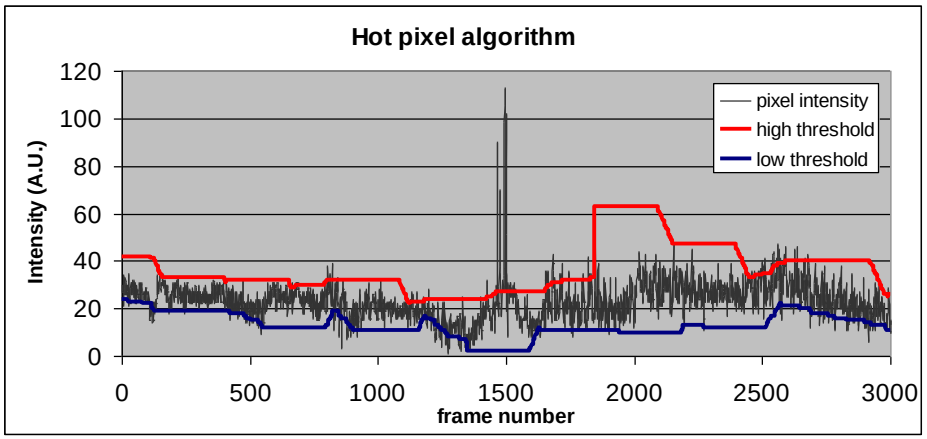
The AVS architecture is meant to maximize the data obtained from each pixel, to allow reduction of the visual sensor resolution and processor power. The algorithms are built in three layers: pixel layer, segment layer and object/motion layer. A main innovation is the design of the low-level, or pixel layer. This part is the most computationally intensive, and effort has been taken to make it as efficient as possible. Our approach calls for simplification of the computational blocks, contrasting with traditional image processing approaches, which tend to consider the image acquisition device as a measurement tool and the processing of its data as an implementation of *rigorous mathematical tools* such as Gaussian edge detectors, Fourier analysis, and convolutions. Such operations require floating point arithmetic and complex architecture or application specific hardware such as DSP or FPGA which can be power hungry. Our approach to the pixel level processing is inspired by nature's visual processing architecture, which utilizes a large number of simple analog receptors, each sensitive to a particular aspect of vision: color, contrast, motion, direction, spatial resolution, *etc.*

Since we focus on static sensors that stare at a fixed scene, each pixel must determine whether, at a given frame, the light intensity impinging on it is regular, or irregular, with respect to the historical intensities it experienced over a prescribed interval (several seconds to a few minutes). The classical implementation of such an operator requires digitally storing the historical intensity levels and computing the relevant statistics. However this implies significant storage and computational resources. Our approach captures pixel historical behavior using a lower and an upper adaptive threshold. These thresholds define the envelope of expected intensities of a "regular" incoming pixel signal.

While this approach in which new pixel values are compared to historical statistics is already known, its light-weight, extremely efficient implementation (making use of the two threshold approach and an elementary, almost analog update schema) is in fact a sensible innovation in the field. This algorithm uses only integer arithmetic, incrementations/decrementations, comparisons and table lookups, making this sensor able to analyze real-time video on a low-performance ARM9 processor.

An example of thresholds' behavior as a function of pixel intensity profile is shown in Figure 3. When pixel intensity exceeds the value of the high threshold (e.g., frame 1500 in Figure 3) or falls below the low threshold, it becomes "hot".

Figure 3. Hot pixel algorithm, showing the high and low thresholds and their adaptive behavior as a function of pixel intensity profile.



The following stage, segmentation, identifies connected components of hot pixels. A basic merit figure of any surveillance system is the minimal detectible object size, in pixels. For example, for a minimal object size of 8 pixels, the expectation of a random occurrence of eight specific pixels being simultaneously hot is p8 where p is the random expectation of a pixel to be hot. Setting p = 1/100 and assuming VGA format and 30 FPS, a single event of 2 ×4 random pixels (anywhere in the frame) will occur once in $100^8 = 10^{16}$ events. This is equivalent to 30 FPS × 31,536,000 seconds/year × 307,200 VGA pixels × 34.5 years. Thus if the pixels are tuned to sensitivity of 1:100 frames, good detection sensitivity can be achieved with low inherent false alarms and a small detectable form factor, maximizing the use of each pixel and avoiding the necessity for a large number of pixels, reducing further the power consumption.

Analysis and experience show that a p factor of 0.01, while could be initially regarded as simplistic, is very general, allowing the (static) sensors to operate well in a large variety of applications, including intruder detection, safe city or home security. This can be compared to the natural process of adaptive retinal sensitivity, which is a universal process that performs similarly indoors and outdoors, night and day, in urban, vegetated or desert regions of the globe. Reducing this value (say to 0.001) will reduce drastically the sensitivity of the system, while increasing it (say to 0.1) will result in numerous false alarms.

3.3. Innovations and Comparison

Table 1 below summarizes the innovation of the HuSIMS visual sensor compared to available systems. We compare the HuSIMS visual sensor with three of the current market leading systems. The first is the VideoIQ system that includes analytics and recording onboard the edge unit. The second is Mobotix, a manufacturer of high-end megapixel cameras and video management systems. One key feature of Mobotix is that the video can be transmitted directly to storage. The third comparable is Axis, the market leader in IP surveillance cameras.

Table 1. Comparison of current video surveillance systems with the HuSIMS visual sensors.

	VideoIQ	Mobotix	Axis	HuSIMS Visual Sensors
Aim	Analyze and record video in place, allow remote access	Record video and transmit directly to storage	Record video	Describe a dynamic scene with thin XML data
Front end	Video camera, analytics and storage	Day/night megapixel camera + microphone	Camera	Visual sensor, interprets the scene and transmits thin XML description
Back end	Video servers performing image analysis	Storage device/ analytics channel	Video recorder/ storage/analytics channel	Statistical engines analyzing and correlating activity data from hundreds of visual sensors
Analytics	Intruder detection	None	None	Automatic detection of anomalous events, per sensor, per time of day
Infrastructure	Fiber-optics for transmitting live video, high power consumption			Low wireless bandwidth, low power consumption

4. Network Components

HuSIMS requires two different networks from a functional point of view: a network of visual sensors for data collection and an alarm distribution network for the notification to first responders of the confirmed alarms.

Both are private networks. It is becoming increasingly common to physically separate emergency application networks from carrier networks—both cellular and landline. When emergencies arise, carrier and operator networks fast become overloaded and mission-critical emergency applications can cease operating. In a privately-owned and operated safe city network, the city can control traffic, define priorities and make sure that the network is always available for the critical applications when they are needed. Finally, private networks cost less to operate than leased networks.

While VPNs (Virtual Private Networks) could have been a viable solution (even presenting some advantages like increased flexibility), there are some concerns regarding their application in real environments, mainly the problems that may arise due to sharing bandwidth with other applications which might result in delays or information loss. For critical security applications, this is sometimes unacceptable. Ultimately, the choice between real and virtual private networks will bring different features into the system.

4.1. Outdoor/ Indoor Network for Alarm Collection

The data collection network will connect the sensors to the MCS application. Its main functionality will be to transport the XML files sent by the visual sensors to the MCS where they will be processed. When constructing the visual sensors network, cost-effective and easily deployable technologies are used. Houses and Buildings infrastructure includes coaxial cables, power lines and phone line cables.

Reusing this infrastructure for surveillance allows a massive fast deployment of the visual sensors in indoor scenarios.

In outdoor scenarios, Wireless networks can be deployed quickly and are very flexible. Wireless technologies like WiFi (802.11) and WiMAX (802.16) make it possible to add and place a very dense amount of cameras and visual sensors in locations previously inaccessible, and offer Quality of Service (QoS) management, high-capacity, high-availability, built-in data encryption mechanisms and low latency connectivity essential for real-time high-resolution video streaming overlarge geographic areas.

However wireless equipment is in general designed to support most of the traffic in the Down Link (DL). In video-surveillance applications most of the traffic goes in the Up Link (UL). Therefore the wireless equipment used in HuSIMS in the alarm collection network was re-designed to support a flexible balancing of the traffic. One of the main targets of HuSIMS is to be a cost-effective system that may be deployed in large and heterogeneous areas. The deployment of Wireless + Wired solutions (e.g., WiFi-enabled visual sensors linked to the power line network using Power Line Communications—PLC technology) allows wireless, fast and low cost deployment with "zero cost" using existing wiring and providing a double linked network to guarantee QoS, capacity and availability.

The new generation of wireless equipment developed for the project includes new Self-Organization Networks (SON) features like self-configuration and self-healing. Self-configuration allows the quick deployment of sensors following a 'plug-and-play' paradigm and enables to download new configuration parameters and new software versions. This is achieved using TR-069 enabled CPEs and Alvarion's Automatic Configuration Server called StarACS.

Self-healing helps to reduce the impact of failure in a given network element allowing the sensors that were connected to the failing node to find connectivity via adjacent cells and enabling quick addition of new sensors and replacement of the damaged ones. HuSIMS network has a flexible architecture that enables the visual sensors to be in constant communication with the MCS. Rather than using mesh networks whose performance quickly degrades in multi-hop scenarios, the system employs point to multi point wireless access nodes based on 802.11n protocol. Regarding network planning, the visual sensors will always be able to reach more than one access node in order to provide redundant paths for the visual sensors to reach the MCS. On the other hand, in indoors scenarios, two technologies like WiFi and PLC communications will be used one as backup infrastructure of the other in order to get always-on connectivity to the MCS.

Self-healing features will allow quick addition of new sensors and quick withdrawal and replacement of damaged ones. Self-healing helps to reduce the impact of failure in a given network element allowing the sensors that were connected to the failing node to find connectivity via adjacent cells.

4.2. Outdoor Wireless Network for Alarm Distribution

The alarms distribution network will connect the MCS application with the first responders (police patrols or emergency vehicles).

While the sensor network emphasizes the low-bandwidth required, in alert or emergency situations, the bandwidth requirements increase significantly due to the real time distribution of the video signal

to the emergency teams. This network needs to be based on a technology that supports broadband and mobility for enabling mobile users to receive the bandwidth demanding video necessary to have a clear idea of what is happening on-site. A network based on WiMAX (802.16e) thanks to its QoS management features has been chosen and enhanced with optimized video transmission capabilities for that purpose.

The video optimization feature allows that video streaming sessions using MPEG-4 codec get a special treatment that ensures the quality of the video transmission in situations of air resources shortage. In MPEG-4 codec the video frames can be classified into Group of Pictures (GOP) in which three different types of frames can be found. Key frames, or I frames that are coded independently; P frames, which include delta updates of I frames; and B frames, which are bi-directional frames. Losing I frames impacts the entire GOP. P frames are second in priority and B frames would have the lowest priority. We use three different queues to classify each type of frame. Using this approach to classify the video frames, when not all video packets can be transmitted (e.g., air resources shortage), we will prioritize I packets and drop P/B packets as necessary.

5. Monitoring and Control System

5.1. State of the Art in Intelligent Alarm Detection

Nowadays, intelligent surveillance is a field of research that is constantly growing. New intelligent cameras, sensors, multi-camera environments, *etc.* require the development of new technologies to take advantage of the raw data retrieved by these components and transform it in useful, high level information for the operators. For this, several data processing alternatives are being developed, such as scene understanding, face/plate/object recognition, or alarm detection.

The application of complex machine vision algorithms is one of the main trends for video surveillance [39–42]. However, they normally have strong requirements on computing power, so either the sensor itself packs a powerful processing unit, making it expensive, or the high definition video signal is sent to a central processing unit, a solution with high bandwidth consumption (especially in a scenario with large numbers of cameras). Those limitations are solved by systems that employ lighter paradigms to process the image, which normally imply reducing the video stream to a set of second level parameters (such as object movement [43]. These approaches present all the advantages of being lighter, but usually left out much information in the image (such as color or shape of objects) when reducing the video stream to parameters. Therefore, they require advanced analysis tools to maximize the high level information that can be extracted and its subsequent interpretation.

One of the solutions is the probabilistic approach. Systems such as the ones presented in [44,45] use Bayesian network based solutions. Bayes' Theorem is very useful to determinate the probabilities of an alarm by using the relations between all the variables in a specific situation. Other authors [46] prefer more complex probabilistic approaches like Hidden Markov Models (HMM) (typically employed in many pattern recognition domains), to extract unknown but meaningful parameters from raw data provided by the cameras. Other techniques, used in pattern recognition too, are the Dynamic Time Warping (DTW) [47] and Longest Common Subsequence (LCSS) [48]. Those mechanisms compare

groups of variables to find similarities among them, and they are successfully employed, for instance, to group similar trajectories in surveillance scenes [49,50].

The presented Bayesian network solution and similarity based methods use the explicit information provided by the system to detect alarms. The HMM-based systems go beyond. They extract new implicit information, hidden in the raw data provided by the cameras and sensors. Some deductive techniques use Neural Networks [51,52] or Clustering Algorithms [53] to classify behaviors and contexts but they are normally resource-greedy and data processing is slow.

On the other hand, along the last 10 years have witnessed the development of the Semantic knowledge technologies, a new approach for formally representing and processing knowledge (using knowledge models known as ontologies) which was first applied in the World Wide Web (giving birth to the Semantic Web, or Web 3.0), but which was quickly extended to other fields, including intelligent surveillance, with good results [54,55]. Semantic technologies offer several advantages, like easy interoperability among heterogeneous systems and easy adaptation to different application domains by replacing ontologies.

Modern surveillance systems normally comprise big numbers of cameras and sensors. Typically, video signals from these sensors have been treated independently, but there are many cases in which their outputs can be combined in order to get a better understanding of what is happening, and even for detecting events which might slip undetected through the analysis of a single scene. In order to take advantage of the increased situational awareness that emerges from the combined interpretation of several sensors outputs, data fusion techniques have been also applied [56,57] with good results.

The HuSIMS alarm detector aims at combining all the advantages of the different techniques exposed by the application of three different analysis engines in parallel. The result is system which can cover a dense network of cameras and sensors, reliably detecting anomalous situations.

5.2. HuSIMS Monitor and Control System

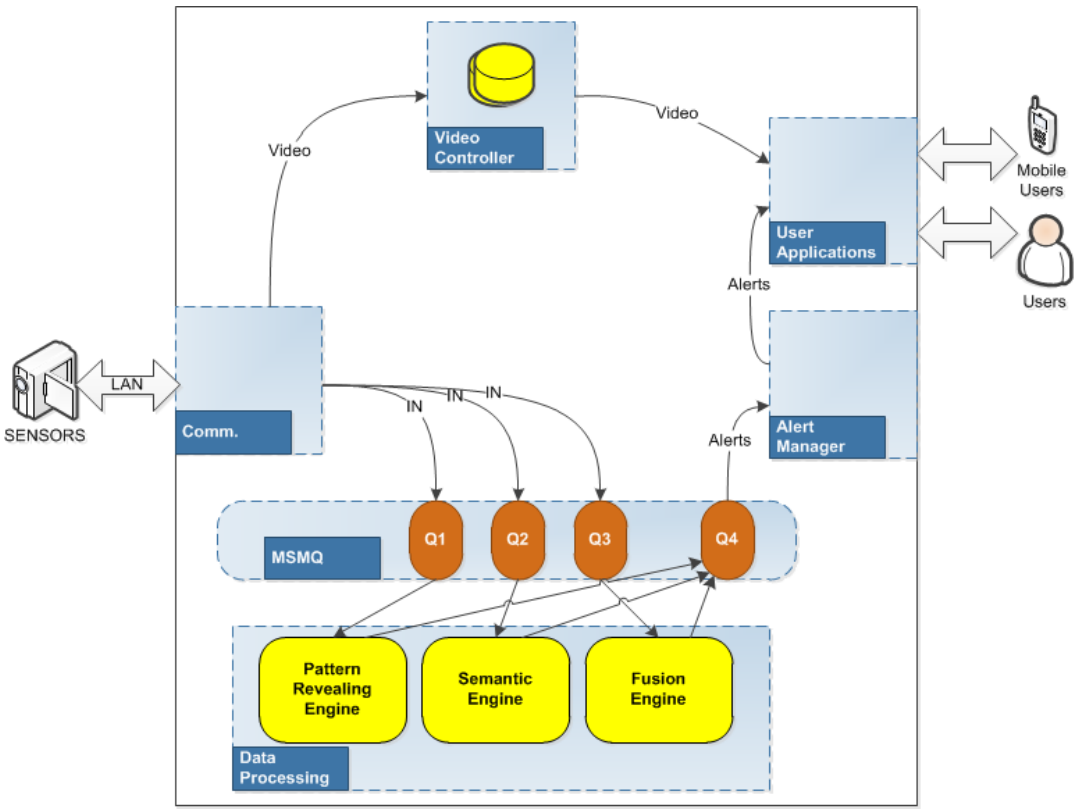
The objective of the HuSIMS' Monitor & Control System (MCS) module is to control the data flow through the different parts of the system. In order to implement that function, the MCS system includes separate modules for communication, data analysis, and user applications—as Figure 4 shows. The MCS communication module receives the information from the visual sensors through the Meshed Sensory Network (MSN) and forwards the raw data to the system core, the Data Processing Module, using a MSMQ (Microsoft Message Queuing) technology. This data processing module is composed by three complementary engines which follow different data processing strategies which are able to distinguish between normal and abnormal situations (patterns) at the monitored area:

- The Pattern Revealing Engine converts the moving object data in Key Performance Indicators (KPI) information and learns their typical patters. An alarm is then raised when the pattern starts to drift out from a normal one.
- The Semantic Engine performs the semantic characterization of the information sent from the MSN. The system is based on the interpretation of the watched scene in terms of the motion parameters of the objects, giving a semantic meaning to them, and using Semantic Web technologies. A semantic reasoner process identifies when an emergency situation is developing.

- Finally, the Fusion Engine is an advanced rule engine dealing with the problem of how to fuse data from multiple sources in order to make a more accurate estimation of the environment. Users can add their own mathematical or semantic functions in order to create new rules. These rules are used to create a behavior later. In a behavior, a case is defined by functions, real data fields and rules. Output of a running behavior is the expected result set of scenery.

When the Data Processing detects an alert the Alert Manager receives the details and forwards them to the user through the User Applications. If the user needs to watch the situation, this application also requests the video of the scene to the MSN using the Video Controller Module.

Figure 4. MCS architecture.



5.3. Pattern Revealing Engine

The Pattern Revealing Engine is based on a family of algorithms [58] for automated modeling and characterizing of any sequence of KPI values in the operational dataset. The model is a data structure that specifies the conditional probability of any KPI value, given past or current values of other KPIs in the sequence (called the context). The model is coded by a network of trees that represents all the patterns in the data that are statistically significant. A specific context-Based Forecasting (C-B4) algorithm is employed to optimize the size and the statistical efficiency of the model, depending on the triggered application [59,60].

Once the model is constructed, it captures all the significant dynamics and dependencies in the data for each KPI. The Pattern Revealing Engine detects anomalies in data sequences, based on patterns rather than the data values, while maintaining relevant and readily interpretable results. Whereas traditional performance-management techniques are usually limited to using KPI control limits to

identify problems, HuSIMS Pattern Revealing Engine can detect pattern's anomalies before KPIs exceed their control limits.

The HuSIMS' algorithm is as follows:

a) Algorithm for pattern generation: each new data received can be stored as a possible extended pattern of a previously detected pattern, if indeed this extension is justified by certain information-theoretic measures. The way the pattern is being constructed, as an additional branch in an existing tree, is actually a patented algorithm.
b) Pattern grading: each new pattern is associated with given likelihood grade based upon information theory metrics (as the amount of new information contained in the new pattern). For HuSIMS needs, the grading algorithm is separated from the pattern generation algorithm to enable fast and near real-time performance.
c) Algorithm for decision: this module can identify if the new pattern, with its grade, is indeed a new pattern, or in fact it is similar to some previously detected patterns. This algorithm is crucial for the monitoring application, since it provides the ability to detect new patterns in a "true-true" needed confidence level.
d) Clustering algorithm: builds clusters of patterns to facilitate handing.
    i. All those algorithms include sub-algorithms, to ensure a high level of confidence in the received results. For example, these algorithms can be used to find anomalies in KPI correlations, even when the KPIs themselves behave normally, pattern matching provides measures of the similarity (or difference) between data sequences that can be used to compare and classify them. This can be used to classify errors in operational datasets.
    ii. To aggregate different KPIs to support pattern generation and identification.
    iii. For root cause analyses for faults and errors in the system.

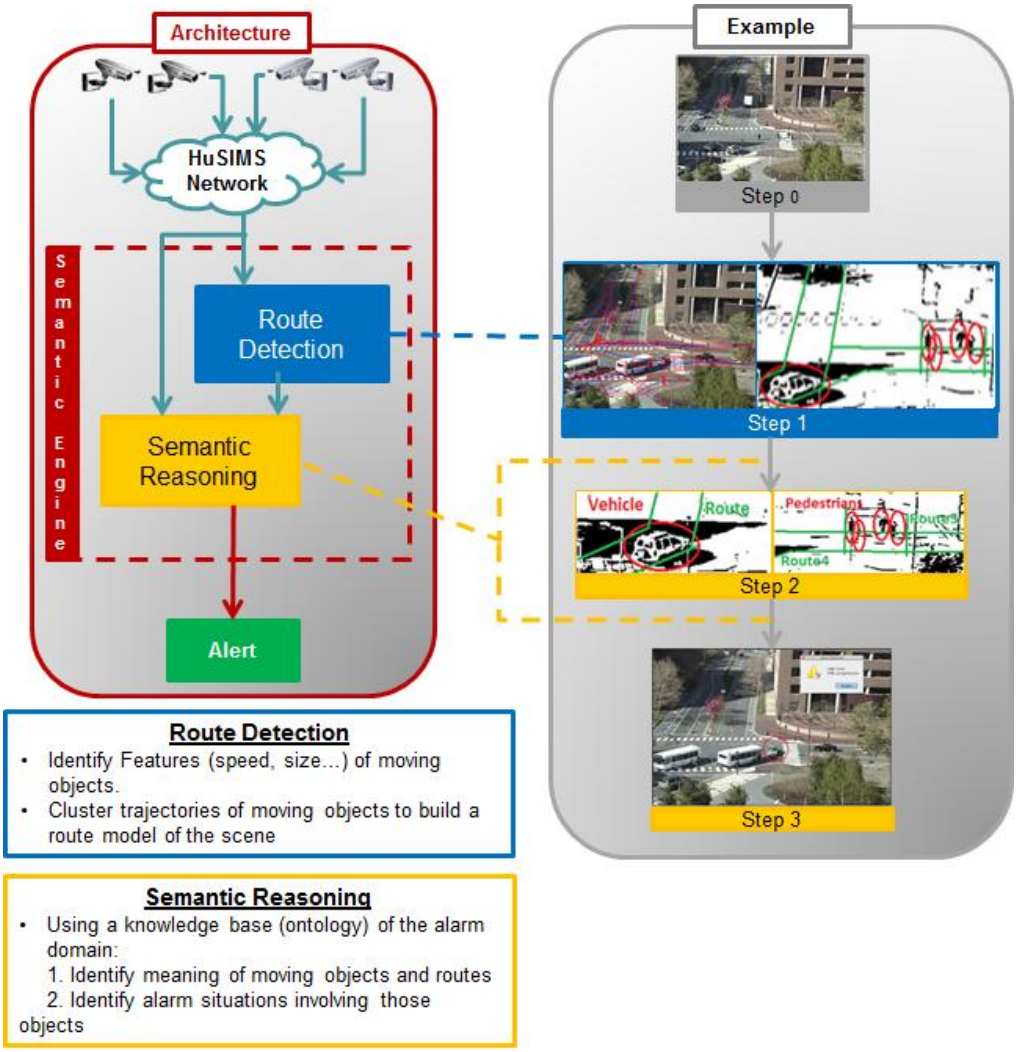
After a training period the C-B4 system is capable of detecting, for example, a normal state of movement direction inside a specific area in the monitored area. If there is a change in these KPI resulting from an abnormal movement direction in that area, an alert is automatically. It is worth noting that no rules are set a priori, the specific normal situation is learnt automatically by the algorithm.

5.4. Semantic Engine

The Semantic Engine is based on projecting the sensed data into a knowledge model representing a human interpretation of the data domain. Specifically, the Semantic Engine developed for HuSIMS [61] operates with only with the simple object motion parameters provided by the visual sensors while other semantic systems use advanced image processing techniques like object and shape recognition in order to identify meaning.

Figure 5 represents the Semantic Engine's internal architecture. There are two main modules called "Route Detection" and "Semantic Reasoning".

Figure 5. Architecture of Semantic Engine.



The Semantic Engine receives from the MSN a file with the data received by each visual sensor and forwards it to the "Route Detection" module. This module processes every frame and using a set of algorithms, it determines the routes, *i.e.*, the zones of the scene where objects habitually move in, by clustering their trajectories. After Step 0 representing image capturing, Step 1 in Figure 5 shows an example with the result of the Route Detection process. The right side of the image shows how the pedestrian and vehicles detected in the scene are the inputs to the Route Detection module to determine the scene trajectories (in green). In the left side, the trajectories detected in the real scene after training are presented. Route Detection is performed only when the system is in training mode, and when finished, this model is fed to a second block, called Semantic Reasoning, so as to be inserted in the ontology and used during the operation mode.

When the system is in operation mode, the information extracted from the frames received is sent to the Semantic Reasoning too. This block uses Java and Jena (an extension of Java which implements a semantic framework) to process the previous information and populate the ontology with the semantic information about the individuals appearing in the image. Finally, the semantic reasoner (a Generic Rule Reasoner is used for this HuSIMS implementation), which is the core of the Semantic Reasoning block, processes the ontology, recently populated with the new data, to infer properties about the

objects in the image and label them according with the type of object identified. In Figure 5, Step 2 shows how the labels Vehicle or Pedestrian are given to moving objects, and the detected routes discriminated with tags like Road or Sidewalk. Then, with this new inferred information, the Semantic Reasoning specifically identifies if an alarm situation is going on (see Step 3 in Figure 5). If it is the case, an appropriate Alarm is sent towards the MCS.

5.5. Fusion Engine

The main purpose of Fusion Engine is to analyze the information collected by several visual sensors and runs pre-existing fusion algorithms, mining the data with the purpose of identifying surveillance anomalies automatically.

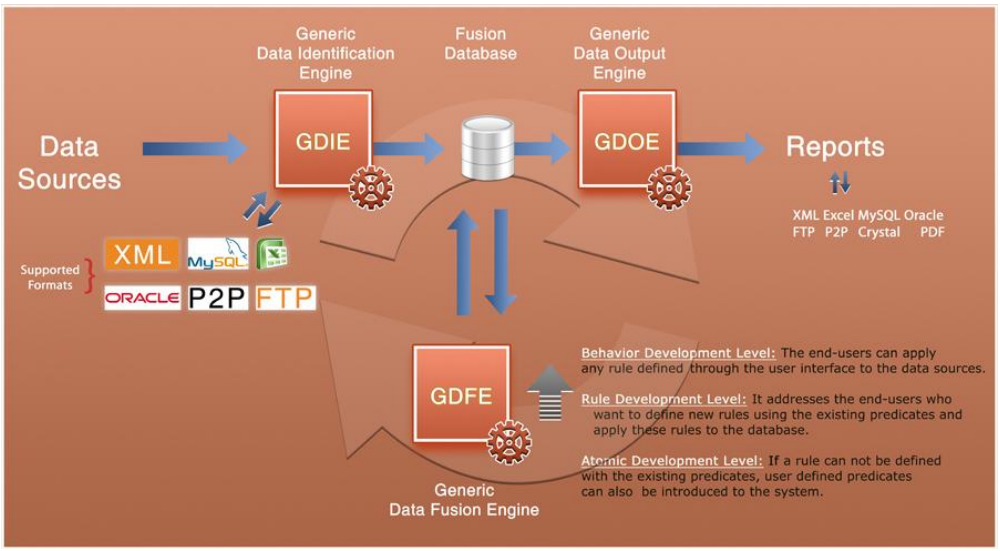
The Fusion Engine executes a generic, user-friendly information fusion process via a database using rules, which can be defined at run time. There are three different layers at which rules can be defined: Behavior Development Level, Rule Development Level and Atomic Development Level. At the Atomic level, a super-user can introduce new mathematical/atomic functions in the platform, at run time using Java. These atomic functions can be used to implement new rules at the second layer that can act as simple behavior templates. Then, these templates can be composed together at the top level in order for the user to implement specific behaviors.

The Fusion Engine consists of a central database and three coordinated and simultaneously working engines:

- Generic Data Identification Engine (GDIE).
- Generic Data Fusion Engine (GDFE).
- Generic Data Output Engine (GDOE).

The relationship between these components and the different composition levels are shown in Figure 6.

Figure 6. General structure of Generic Data Fusion Engine.



The first of these components is a GDIE, which allows to connect different data sources (XML files, Excel files, information stored in a MySQL database, *etc.*) and sensors and to collect information

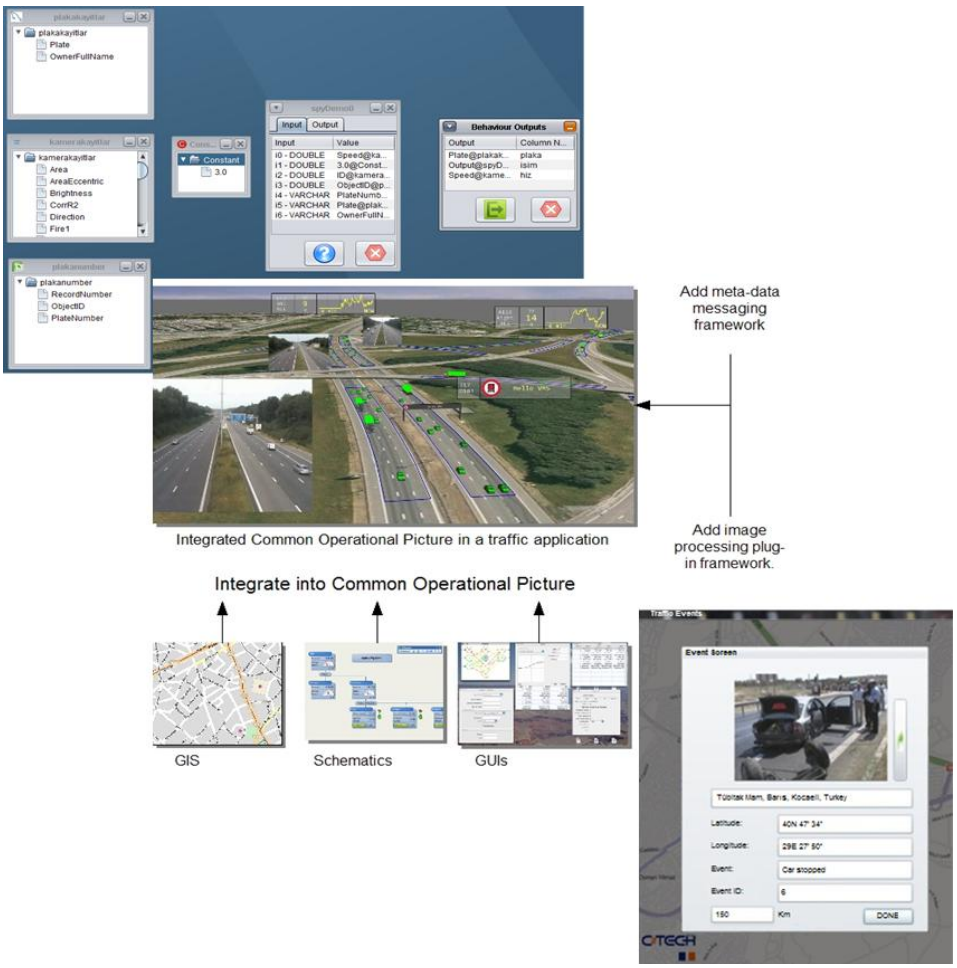
from them. After data verification, format definition and mapping the fields into MySQL, the data is inserted into the central database.

The GDFE associates the gathered data in central database with each other and fuses them as rule-based.

The last component, the GDOE exports and reports the fused data in several formats. It can generate not only dynamic reports but also custom reports by using predesigned report templates. On the other hand, if the fusion results include location data, results can also be shown on a visual map. In HuSIMS project, Fusion Engine templates are sent using Web Services to a Mobile Event Manager that shows the event information to the user in his terminal, for instance using Google Maps, Bing Maps or Apple Maps.

Figure 7 shows how the Fusion Engine throws an event using the information sent by several visual sensors. The left side of the image presents the information provided by the different sensors and sources (as represented in the engine's user interface). In the middle one, all that information reported by different sensors is merged in the same scene and processed to identify the alarm situation show at the right side of the image.

Figure 7. Fusion Engine operation.



6. Alarm Detection Use Cases

This section presents several use cases where the operation of the system is shown. The system has been tested with video data processed using the visual sensors software.

6.1. Traffic Management

This use case shows how the HuSIMS system is employed in a traffic management scenario and is capable of detecting vehicles driving in the wrong direction.

First, the real video is processed by one of the visual sensors that provides information about the movement objects. Whenever a moving object is detected in a frame, an XML file (See Figure 8) is sent towards the MCS reporting the main features of all the objects detected. These means that reports are not sent on a continuous basis, but only when meaningful information is available. Frame rates are also configurable, and the use case has been tested for instance with frame rates of 10 FPS. For each object the visual sensor reports width, height, position (x, y), area, speed, direction of the movement, brightness and thinness, and maximum historical values for area, width, height, speed and brightness. In addition, each object has an id to allow tracking along the scene.

Figure 8. Example of XML file.

During the training process, the engines learn the correct/normal values for the scene, each of them using their own abstractions.

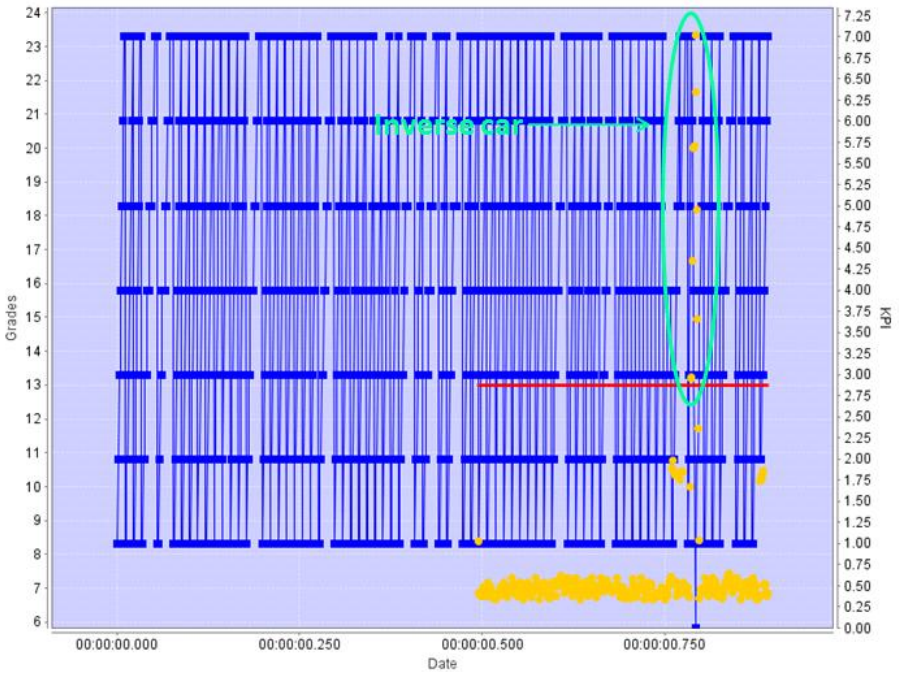
Figure 9 represents the behavior of the Pattern Revealing Engine in this scenario, with the *x*-axis representing time, and the *y*-axis representing at the same time the values of the position KPI and the grading of patterns.

First, during the training period the Pattern Revealing Engine identifies all the significant patterns in the position, speed, and direction KPIs for the objects reported, building a scene-specific pattern model. The values reported are represented in the Figure 9 as blue points. Once the system enters the operation mode, each newly detected pattern is given a score (in grades), which are represented in

Figure 9 with a yellow dot. Those grades indicate how much the new pattern is similar to the previously learned "normal" set of patterns. The grade received is a measure of how different is the pattern being analyzed when compared of the normal patterns: if the grade is low, the current pattern detected in the KPI evolution is quite similar to one of the learnt ones. The red line is a threshold line: if the score (yellow dot) is above this line, the difference of the current pattern with the learnt ones is meaningful, and an alert is issued.

Figure 9 is an example of detection by the C-B4 grading algorithm. During the times before the red line appears in the image the Pattern Revealing engine is working in training mode, receiving information from a camera watching a one-direction highway where cars go south. When it enters the operation mode, the figure shows yellow dots for each graded pattern, which for the most part represent cars travelling in the right direction. However, at some point, the grades obtained start to surpass the threshold: looking at the blue line it is easy to see that at this point there is a change in the trend exhibited by the calculated KPI. This change in the trend is caused by a car that was detected travelling in the wrong direction.

Figure 9. A car driving in the wrong direction, a change in the pattern of direction of movement is detected, even before car completed the full change of direction. The grades referred to the directional KPI indicates a new pattern of the direction KPI.



The Semantic Engine follows a different approach to deal with the same case. First, a human ontology engineer designs a traffic management ontology, which states that "road" objects may have a preferred direction, and defining that detecting a car travelling in the opposite direction should be identified as an alarm state. This ontology is loaded in the Semantic Engine.

Then the Semantic Engine has to learn a route model for each specific sensor. To build the model for this use case, the Engine is set up in training mode, which in the end learns the usual routes of the objects and the direction of the movement in those routes, and according to the parameters "area" and "speed" of the objects populating them (as specified in the ontology), they are labeled as a "road" or a "sidewalk".

Figure 10 shows an example of how this route model is built inside the Semantic Engine during the training mode. The left side of the picture presents the original scene, while it right side also includes the labels assigned to routes (road, crosswalk, *etc*.) and objects (pedestrian or vehicle). While it is not shown in the image, the system stores many internal parameters about the routes, such as the direction of movement and the type of objects usually moving inside it.

Figure 10. Example of a Semantic Engine's learning process.

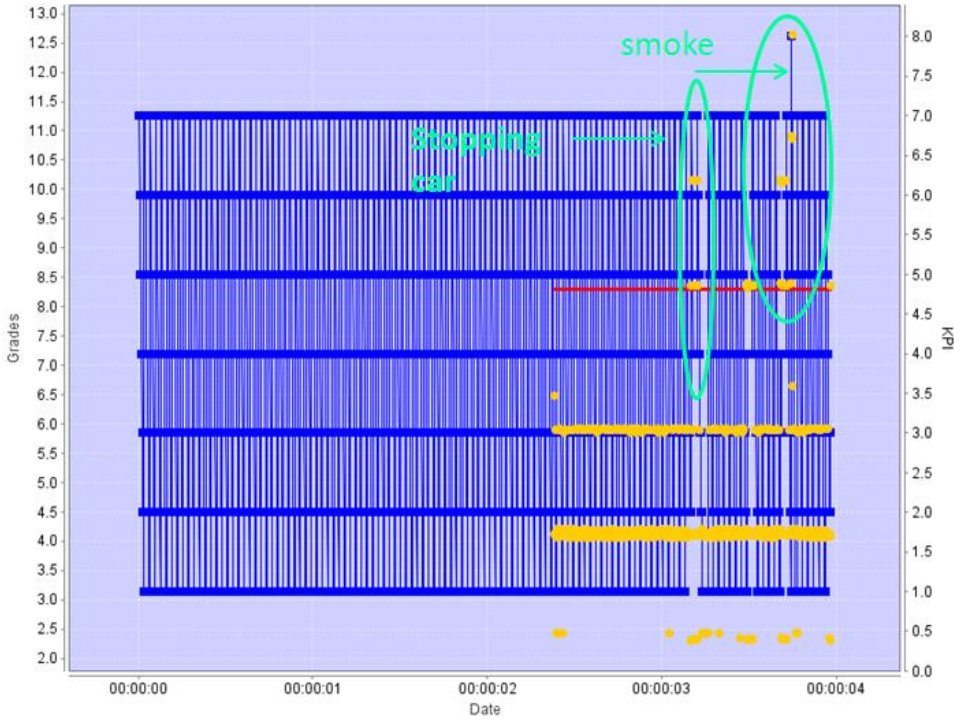
When it enters the operation mode, the Semantic Engine no longer learns routes, but assigns labels to the moving objects depending on their parameters. Route R5 (the lane at the left side of the road, where cars move downwards), for example, was identified as "road" and the direction of the vehicles that move through it was 5 (using the hours in a clock as a reference). When a vehicle with a direction of 11, that vehicle is identified as moving in the opposite direction and an alarm is thrown.

Both engines will work similarly for detecting cars abnormally stopped in the road. Figure 11 shows the KPI and grading scheme for the Pattern Revealing Engine, also presenting a clear change in the KPI trend for two cars stopped in the middle of the road, which is correctly detected by the grading algorithm. The second car is actually facing a breakdown which causes smoke to come out of the engine. The movement pattern of the "smoke" object is so different from the learnt ones that the value of the KPI comes out of the positional range observed up to that point: the smoke is moving in a zone of the image where no object has been moving before.

The Semantic Engine will also detect that a vehicle is stopped in a zone where there should not be any stopped object: on a pedestrian crossing. This abnormal behavior has to be coded inside the ontology.

It is worth mentioning that for a car stopping before a red traffic light or a stop signal, for instance, no alarm is issued in any of the cases. The Pattern Revealing engine would have learnt positional patterns that represent cars stopping at those points in the image, and the Semantic Engine would have labeled that part of the image as a "road before a traffic light" because objects labeled as "cars" would have previously stopped in that area of the scene.

Figure 11. Stopping car—the normal movement KPI were recognized as a stable pattern. Once a change in the velocity data pattern was recognized—an alert is issued.



6.2. Vandalism Detection and Crowd Control

The HuSIMS system can be easily deployed in many different scenarios, such as vandalism detection and crowd control. When applied to these domains, the system will identify abnormal behavior of people, and the movement parameters monitored by the cameras will be enriched with microphones reporting noise levels. In the case of a fire in a pub, the cameras will report abnormal values of the brightness parameters for the flames, and some minutes after abnormal motion parameters, because the people in panic trying to escape. During training, the Semantic Engine will have identified forbidden areas and places where people move freely, which will be characterized in the crowd control ontology together with parameters such as average speed. First, a "flames" alarm will be issued to the MCS when an object with an abnormal brightness parameter is detected in a place where a light is not identified, and afterwards—when people starts moving at an unusually high speed in unusual places as they try to escape—they are labeled as people running. As the ontology specifies that many persons running represent a crowd in panic, and an alarm of this type is issued.

The Pattern Revealing Engine will act similarly, identifying brightness and movement patterns extremely different from those learnt during training, and therefore identifying and abnormal event.

Another scenario where the system may result useful is vandalism detection. In this example, a person punches another during a fight in a subway station. It should be noted that identifying this kind of abnormal situation solely on the basis of the action *per se* is extremely difficult. Even an extremely focused human operator will have trouble discriminating this action from two friends shaking hands, for instance, specifically taking into account the low resolution of the sensors. However, the HuSIMS system relies on the assumption that vandalism actions are usually accompanied by erratic or strange

behavior before and after the event. In the case of the fight, it is extremely unlikely that the attacker will simply hit another person and continue walking normally. Instead, he will probably leave running.

Even more, HuSIMS supports the usage of additional, non-visual features and parameters in order to help in detecting this kind of visually-confusing situations. For instance, the previously mentioned noise detector could help identifying screams and shouts related to a fight; and in the case it is happening inside a subway station, a detector of the presence of a train will help discriminating if the noise is coming from a train entering the station or it is due to an abnormal/alarm situation.

Under these circumstances, the Pattern Revealing Engine and the Semantic Engine will both detect that there has been and abnormal event. The Semantic Engine will even specify a fight alarm.

6.3. Data Fusion and Engine Collaboration

The Fusion Engine will help in putting together all the multimodal information provided by the different sensors of the architecture and the alarms issued by the engines. For instance in the case of the fight described in the previous subsection, the Fusion Engine is capable of tracking the running suspect across the scenes as reported by the different sensors, effectively identifying the route followed. It will identify that the object moving at an abnormal speed which is sequentially reported by different sensors is the same individual, and will report his position in real time to the authorities.

But the Fusion Engine will also help in reducing the number of false positives, by combining the output of the other two engines. For instance, in the case that there is a sensor camera for surveillance in a street where there is a shop with a window display, during training it might happen that all pedestrians either enter the shop, or continue walking along the sidewalk. However, during the operation mode, a pedestrian stops in front of the window display in order to view some of the items in which he is interested. The Pattern Revealing Engine may understand this action as an abnormal event, since no other pedestrian has stopped there before, issuing an alarm which is false in this case. On the other hand, the Semantic Engine understands that it is not an alarm that a pedestrian is stopped, as long as he is on a sidewalk, and therefore does not issue an alarm. The Fusion Engine will, in this case, combine both outputs (using a set of rules defined by the operator) to identify that it is a false alarm (and differentiating it from the many situations in which the Pattern Revealing Engine will simply identify alarms that are not covered by the ontology in the Semantic Engine); therefore, in this case, the alarm identified by the Pattern Revealing Engine will not be progressed to the MCS.

7. System Comparison

HuSIMS presents several distinct features when compared to other available solutions in the literature and in the market. Most notably, the usage of cheap visual sensors, the conversion of low-resolution video streams into a set of object motion parameters making unnecessary the streaming of video in a regular basis, AI reasoning over those motion parameters for autonomous alarm detection, and generic, domain-agnostic approaches to scene processing (which effectively qualify the system for multi-domain operation—traffic surveillance, vandalism, perimeter security, *etc*.), represent a very particular philosophy which stands HuSIMS apart from alternatives. Table 2 summarizes a top-level feature comparison of HuSIMS and other reported initiatives following very different approaches.

Table 2. HuSIMS feature comparison.

	HuSIMS	**Current State of the Art**	**ADVISOR [62]**	**ARGOS [63]**	**DETER [64]**	**AVITRACK [65]**
Objective	Alarm Detection	Recording video	Send warnings to human operators	Boat traffic monitoring	Alarm detection	Monitor and recognize activities
Resolution	Low	High	384 × 288 pixels	320 × 240 pixels	High	720 × 576
Bandwidth	Low	High	Ethernet IP Multicast	Local PC connection. (No specified connection with the control center)	Coaxial cable	1 Gb Ethernet
Storage	Rarely, only important video streams	Always	Yes (video + annotations)	Yes	Yes	Yes
Privacy	Gentle	Aggressive	Aggressive	Gentle	Gentle	Aggressive
Cost	Low	High	High	High	High	High
Type of Data Analyzed	Motion parameters	Video Signals	Video Signals	Video Signal	Video Signal	Video Signal
Domain	Multi-domain		Centered on metro stations	Maritime traffic detections (designed for Venice)	Vehicle and people detection	Airport Security

8. Conclusions

In this work, the HuSIMS video surveillance platform has been presented. It has been designed with wide area, dense deployments in mind, using inexpensive sensors with low resource/bandwidth requirements in order to facilitate deployment and management of thousands of sensors, and generic, multi-domain reasoning and alarm detection engines. This makes HuSIMS a perfect candidate for integrated surveillance systems in smart cities and big facilities where it is necessary to count with large numbers of cameras to provide full coverage of the entire area.

In addition, the three alarm detection engines present innovative solutions on their own: the Pattern Revealing Engine enjoys the ability of learning meaning-agnostic patterns in the motion parameters sent by the cameras; the ontology domain knowledge model employed by the Semantic Engine allows to understand what is happening in the scene; and the data Fusion Engine is capable of extracting information from the combination of outputs. The parallel operation of three different strategies provides a wide range of different alarm detection features, because the system is capable of providing meaningful and rich information about the different known situations, as represented by the ontologies in the semantic engine, and at the same time detect abnormal behaviors which have not been explicitly specified in a knowledge model. As a result, the system is robust (alarms are detected according to an expert knowledge model) and dynamic/flexible (unexpected but abnormal situations are also detected) at the same time.

Acknowledgments

This work has been partially funded by the Ministerio de Industria, Turismo y Comercio del Gobierno de España and the Fondo de Desarrollo Regional (FEDER) and the Israeli Chief Scientist Research Grant 43660 inside the European Eureka Celtic project HuSIMS (TSI-020400-2010-102). The authors would like to thank the Companies C-B4 and C Tech for their valuable collaboration in this paper and in HuSIMS project.

Conflict of Interest

The authors declare no conflict of interest.

References

1. Technavio Analytic Forecast. *Global Video Surveillance Market 2011–2015*. Available online: http://www.technavio.com/content/global-video-surveillance-market-2011–2015 (accessed on 16 April 2013).
2. Zhu, J.; Lao, Y.; Zheng, Y.F. Object tracking in structured environments for video surveillance applications. *IEEE Trans. Circuits Syst. Video Technol.* **2010**, *20*, 223–235.
3. Osais, Y.E.; St-Hilaire, M.; Fei, R.Y. Directional sensor placement with optimal sensing range, field of view and orientation. *Mob. Netw. Appl.* **2010**, *15*, 216–225.
4. Brutzer, S.; Hoferlin, B.; Heidemann, G. Evaluation of Background Subtraction Techniques for Video Surveillance. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 20–25 June 2011; pp. 1937–1944.

5. Butty án, L.; Gessner, D.; Hessler, A.; Langendoerfer, P. Application of wireless sensor networks in critical infrastructure protection: Challenges and design options [Security and Privacy in Emerging Wireless Networks]. *IEEE Wirel. Commun.* **2010**, *17*, 44–49.
6. Chen, M.; Gonz áez, S.; Cao, H.; Zhang, Y.; Vuong, S.T. Enabling low bit-rate and reliable video surveillance over practical wireless sensor network. *J. Supercomput.* **2010**, doi:10.1007/s11227-010-0475-2.
7. Kandhalu, A.; Rowe, A.; Rajkumar, R.; Huang, C.; Yeh, C.-C. Real-time video surveillance over IEEE 802.11 mesh networks. In Proceedings of the 15th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2009), San Francisco, CA, USA, 13–16 April 2009; pp. 205–214.
8. Durmus, Y.; Ozgovde, A.; Ersoy, C. Distributed and online fair resource management in video surveillance sensor networks. *IEEE Trans. Mob. Comput.* **2012**, *11*, 835–848.
9. Dore, A.; Soto, M.; Regazzoni, C.S. Bayesian tracking for video analytics. *IEEE Signal Process. Mag.* **2010**, *27*, 46–55.
10. Regazzoni, C.S.; Cavallaro, A.; Wu, Y.; Konrad, J.; Hampapur, A. Video analytics for surveillance: Theory and practice [from the guest editors]. *Signal Process. Mag. IEEE* **2010**, *27*, 16–17.
11. Piatrik, T.; Fernandez, V.; Izquierdo, E. The Privacy Challenges of In-Depth Video Analytics. In Proceedings of the 2012 IEEE 14th International Workshop on Multimedia Signal Processing (MMSP), Banff, AB, Canada, 17–19 September 2012; pp. 383–386.
12. Tian, Y.-l.; Brown, L.; Hampapur, A.; Lu, M.; Senior, A.; Shu, C.-f. IBM smart surveillance system (S3): Event based video surveillance system with an open and extensible framework. *Mach. Vis. Appl.* **2008**, *19*, 315–327.
13. Nghiem, A.-T.; Bremond, F.; Thonnat, M.; Valentin, V. ETISEO, Performance Evaluation for Video Surveillance Systems. In Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, AVSS 2007, London, UK, 5–7 September 2007; pp. 476–481.
14. Oh, S.; Hoogs, A.; Perera, A.; Cuntoor, N.; Chen, C.-C.; Lee, J.T.; Mukherjee, S.; Aggarwal, J.; Lee, H.; Davis, L. A large-scale benchmark dataset for event recognition in surveillance video. In Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 20-25 June 2011; pp. 3153–3160.
15. Vellacott, O. *The Olympic Challenge – Securing Major Events using Distributed IP Video Surveillance*. IndigoVision, Inc.: Edinburgh, UK. Available online: http://www.indigovision.com/documents/public/articles/Securing%20Major%20Events%20using%20IP%20Video%20Surveillance-US.pdf (accessed on 18 April 2013).
16. Rougier, C.; Meunier, J.; St-Arnaud, A.; Rousseau, J. Robust video surveillance for fall detection based on human shape deformation. *IEEE Trans. Circ. Syst. Video Technol.* **2011**, *21*, 611–622.
17. Buckley, C. New York Plans Surveillance Veil for Downtown. *New York Times* **2007**, *9*, 3. Available online: http://www.nytimes.com/2007/07/09/nyregion/09ring.html (accessed on 18 April 2013).
18. Coaffee, J. Recasting the "Ring of Steel": Designing Out Terrorism in the City of London? In *Cities, War, and Terrorism: Towards an Urban Geopolitics*; Graham, S., Ed.; Blackwell: Oxford, UK, 2004; pp. 276–296.

19. Hughes, M. CCTV in the Spotlight: One Crime Solved for Every 1,000 Cameras. *The Independent* 2009. Available online: http://www.independent.co.uk/news/uk/crime/cctv-in-the-spotlight-one-crime-solved-for-every-1000-cameras-1776774.html (accessed on 18 April 2013).
20. Evans, I. Report: London No Safer for All its CCTV Cameras. *The Christian Science Monitor* 2012. Available online: http://www.csmonitor.com/World/Europe/2012/0222/Report-London-no-safer-for-all-its-CCTV-cameras (accessed on 18 April 2013).
21. Hernandez, L.; Baladron, C.; Aguiar, J.M.; Carro, B.; Sanchez-Esguevillas, A.; Lloret, J.; Chinarro, D.; Gomez-Sanz, J.J.; Cook, D. A Multi-Agent System Architecture for Smart Grid Management and Forecasting of Energy Demand in Virtual Power Plants. *IEEE Commun. Mag.* **2013**, *51*, 106–113.
22. Roberts, L. History of Video Surveillance and CCTV. *WE C U Surveillance* 2004. Available online: http://www.wecusurveillance.com/cctvhistory (accessed on 18 April 2013).
23. Belbachir, A.N., Göbel, P.M. Smart Cameras: A Historical Evolution. In *Smart Cameras*; Belbachir, A.N., Ed.; Springer, US: 2010, pp. 3 – 17.
24. Thompson, M. Maximizing CCTV Manpower. *Secur. World* **1985**, *22*, 41–44.
25. Rodger, R.M.; Grist, I.; Peskett, A. Video Motion Detection Systems: A Review for the Nineties. In Proceedings Institute of Electrical and Electronics Engineers 28th Annual 1994 International Carnahan Conference on Security Technology, Albuquerque, NM, 12–14 October 1994; pp. 92–97.
26. Michalopoulos, P.; Wolf, B.; Benke, R. Testing and field implementation of the minnesota video detection system (AUTOSCOPE). In *Traffic Flow, Capacity, Roadway Lighting, and Urban Traffic Systems* Transportation Research Board: Washington, DC, USA, 1990; pp. 176–184.
27. Kaneda, K.; Nakamae, E.; Takahashi, E.; Yazawa, K. An unmanned watching system using video cameras. *IEEE Comput. Appl. Power* **1990**, *3*, 20–24.
28. Honovich, J. Top 3 Problems Limiting the Use and Growth of Video Analytics. IPVM 2008. Available online: http://ipvm.com/report/top_3_problems_limiting_the_use_and_growth_of_video_analytics (accessed on 18 April 2013).
29. Hampapur, A.; Brown, L.; Connell, J.; Ekin, A.; Haas, N.; Lu, M.; Merkl, H.; Pankanti, S. Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking. *IEEE Signal Process. Mag.* **2005**, *22*, 38–51.
30. Foresti, G.L.; Micheloni, C.; Snidaro, L.; Remagnino, P.; Ellis, T. Active video-based surveillance system: The low-level image and video processing techniques needed for implementation. *IEEE Signal Process. Mag.* **2005**, *22*, 25–37.
31. Rinner, B.; Wolf, W. An introduction to distributed smart cameras. *Proc. IEEE* **2008**, *96*, 1565–1575.
32. Rinner, B.; Winkler, T.; Schriebl, W.; Quaritsch, M.; Wolf, W. The Evolution from Single to Pervasive Smart Cameras. In Proceedings of the Second ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2008), Stanford, CA, USA, 7–11 September 2008; pp. 1–10.
33. Quaritsch, M.; Kreuzthaler, M.; Rinner, B.; Bischof, H.; Strobl, B. Autonomous multicamera tracking on embedded smart cameras. *EURASIP J. Embed. Syst.* **2007**, *2007*, 35–35.
34. Wang, Y.; Velipasalar, S.; Casares, M. Cooperative object tracking and composite event detection with wireless embedded smart cameras. *IEEE Trans. Image Process.* **2010**, *19*, 2614–2633.

35. Mucci, C.; Vanzolini, L.; Deledda, A.; Campi, F.; Gaillat, G. Intelligent Cameras and Embedded Reconfigurable Computing: A Case-Study on Motion Detection. In Proceedings of the 2007 International Symposium on System-on-Chip, Tampere, Finland, 20–21 November 2007; pp. 1–4.
36. Hengstler, S.; Prashanth, D.; Fong, S.; Aghajan, H. MeshEye: A Hybrid-Resolution Smart Camera Mote for Applications in Distributed Intelligent Surveillance. In Proceedings of the 6th International Symposium on Information Processing in Sensor Networks, 2007 (IPSN 2007), Cambridge, MA, USA, 25-27 April 2007; pp. 360–369.
37. Casares, M.; Velipasalar, S.; Pinto, A. Light-weight salient foreground detection for embedded smart cameras. *Comput. Vision Image Underst.* **2010**, *114*, 1223–1237.
38. Dworak, V.; Selbeck, J.; Dammer, K.-H.; Hoffmann, M.; Zarezadeh, A.A.; Bobda, C. Strategy for the development of a smart NDVI camera system for outdoor plant detection and agricultural embedded systems. *Sensors* **2013**, *13*, 1523–1538.
39. Sivic, J.; Russell, B.C.; Efros, A.A.; Zisserman, A.; Freeman, W.T. Discovering Objects and Their Location in Images. In Proceedings of the Tenth IEEE International Conference on Computer Vision, ICCV 2005, Beijing, China,17–21 October 2005; pp. 370–377.
40. Torralba, A.; Murphy, K.P.; Freeman, W.T.; Rubin, M.A. Context-Based Vision System for Place and Object Recognition. In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; pp. 273–280.
41. Tan, T.-N.; Sullivan, G.D.; Baker, K.D. Model-Based localisation and recognition of road vehicles. *Int. J. Comput. Vis.* **1998**, *27*, 5–25.
42. Serre, T.; Wolf, L.; Bileschi, S.; Riesenhuber, M.; Poggio, T. Robust object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 411–426.
43. Cutler, R.; Davis, L.S. Robust real-time periodic motion detection, analysis, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 781–796.
44. Nguyen, N.T.; Bui, H.H.; Venkatsh, S.; West, G. Recognizing and Monitoring High-Level Behaviours in Complex Spatial Environments. In Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, WI, USA, 18–20 June 2003; volume 622, pp. II-620–II-625.
45. Ivanov, Y.A.; Bobick, A.F. Recognition of visual activities and interactions by stochastic parsing. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 852–872.
46. Remagnino, P.; Shihab, A.; Jones, G. Distributed intelligence for multi-camera visual surveillance. *Pattern Recognit.* **2004**, *37*, 675–689.
47. Ko, M.H.; West, G.; Venkatesh, S.; Kumar, M. Using dynamic time warping for online temporal fusion in multisensor systems. *Inf. Fusion* **2008**, *9*, 370–388.
48. Kim, Y.-T.; Chua, T.-S. 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.
49. Morris, B.; Trivedi, M. Learning Trajectory Patterns by Clustering: Experimental Studies and Comparative Evaluation. In Proceedings of IEEE Conference on the Computer Vision and Pattern Recognition, CVPR 2009, Miami, FL, USA, 20-25 June 2009; pp. 312–319.
50. Zhang, Z.; Huang, K.; Tan, T. Comparison of Similarity Measures for Trajectory Clustering in Outdoor Surveillance Scenes. In Proceedings of the 18th International Conference on Pattern Recognition, ICPR 2006, Hong Kong, China, 20-24 August 2006; pp. 1135–1138.

51. Sacchi, C.; Regazzoni, C.; Vernazza, G. A Neural Network-Based Image Processing System for Detection of Vandal Acts in Unmanned Railway Environments. In Proceedings of the11th International Conference on Image Analysis and Processing, Palermo, Italy, 26–28 September 2001; pp. 529–534.
52. Baladrón, C.; Aguiar, J.M.; Calavia, L.; Carro, B.; Sánchez-Esguevillas, A.; Hernández, L. Performance study of the application of artificial neural networks to the completion and prediction of data retrieved by underwater sensors. *Sensors* **2012**, *12*, 1468–1481.
53. Piciarelli, C.; Foresti, G. On-Line Trajectory Clustering for Anomalous Events Detection. *Pattern Recognit. Lett.* **2006**, *27*, 1835–1842.
54. Liu, J.; Ali, S. Learning Scene Semantics Using Fiedler Embedding. In Proceedings of the 20th International Conference on Pattern Recognition (ICPR), Istanbul, Turkey, 23–26 August 2010; pp. 3627–3630.
55. Fernández, C.; Baiget, P.; Roca, X.; Gonzàlez, J. Interpretation of complex situations in a semantic-based surveillance framework. *Signal Process. Image Commun.* **2008**, *23*, 554–569.
56. Nakamura, E.F.; Loureiro, A.A.; Frery, A.C. Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Comput. Surv.* **2007**, *39*, 9.
57. Friedlander, D.; Phoha, S. Semantic information fusion for coordinated signal processing in mobile sensor networks. *Int. J. High. Perform. Comput. Appl.* **2002**, *16*, 235–241.
58. Ben-Gal, I.; Morag, G.; Shmilovici, A. Context-Based Statistical Process Control: a Monitoring Procedure for State-Dependent Processes. *Technometrics* **2003**, *45*, 293–311.
59. Ben-Gal, I.; Shmilovici, A.; Morag, G.; Zinger, G. Stochastic modeling of time distributed sequences. Available online: http://www.google.com/patents/US20030061015 (accessed on 30 May 2013).
60. Ben-Gal, I.; Shmilovici, A.; Morag, G.; Zinger, G. Stochastic modeling of spatial distributed sequences. Available online: http://www.google.com/patents/WO2002067075A3?cl=en (accessed on 30 May 2013).
61. Calavia, L.; Baladrón, C.; Aguiar, J.M.; Carro, B.; Sánchez-Esguevillas, A. A semantic autonomous video surveillance system for dense camera networks in smart cities. *Sensors* **2012**, *12*, 10407–10429.
62. Siebel, N.T.; Maybank, S. The Advisor Visual Surveillance System. In Proceedings of the ECCV 2004 workshop Applications of Computer Vision (ACV), Prague, Czech Republic, 10–16 May 2004; pp. 103–111.
63. Bloisi, D.; Iocchi, L. Argos—A video surveillance system for boat traffic monitoring in Venice. *Int. J. Pattern Recognit. Artif. Intell.* **2009**, *23*, 1477–1502.
64. Pavlidis, I.; Morellas, V.; Tsiamyrtzis, P.; Harp, S. Urban surveillance systems: From the laboratory to the commercial world. *Proc. IEEE* **2001**, *89*, 1478–1497.
65. Aguilera, J.; Thirde, D.; Kampel, M.; Borg, M.; Fernandez, G.; Ferryman, J. Visual Surveillance for Airport Monitoring Applications. In Proceedings of the 11th Computer Vision Winter Workshop 2006, Telc, Czech Republic, 6-8 February 2006; pp. 6–8.