# DESARROLLO DE UN SISTEMA DE VIDEOVIGILANCIA INTELIGENTE PARA LA DETECCIÓN DE OBJETOS ABANDONADOS Y ROBADOS

**NILSON DAVID MARMOLEJO OSSA**

**UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS**
**EN CONVENIO**
**UNIVERSIDAD DE LA AMAZONIA**
**FACULTAD DE INGENIERÍA**
**PROGRAMA DE INGENIERÍA DE SISTEMAS**
**FLORENCIA, CAQUETÁ**
**2013**

# DESARROLLO DE UN SISTEMA DE VIDEOVIGILANCIA INTELIGENTE PARA LA DETECCIÓN DE OBJETOS ABANDONADOS Y ROBADOS

**NILSON DAVID MARMOLEJO OSSA**

Trabajo de grado para optar al título de:
Ingeniero de Sistemas

**Director:**
Ing. Milher Fabian Tovar Rubiano

**UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS**
**EN CONVENIO**
**UNIVERSIDAD DE LA AMAZONIA**
**FACULTAD DE INGENIERÍA**
**PROGRAMA DE INGENIERÍA DE SISTEMAS**
**FLORENCIA, CAQUETÁ**
**2013**

**Nota de Aceptación:**

_____

_____

_____

_____

_____

_____

_____

**Firma de jurado**

_____

**Firma de jurado**

_____

**Ciudad y fecha**

*Dedico este libro a Dios, a mi madre y a mi prometida Carolina Sarmiento*

***NILSON MARMOLEJO***

# Resumen

El interés de los sistemas de videovigilancia ha ido creciendo rápidamente en los escenarios académicos, públicos, militares y comerciales. Sin embargo, existen dos problemas principales en los sistemas de vigilancia tradicionales, en primer lugar, la dependencia de los operadores humanos en los sistemas de vigilancia tradicionales implica graves deficiencias, como son, los altos costos de mano de obra, las limitaciones para la monitorización de múltiples cámaras, la perdida de eventos importantes a causa de la fatiga provocada por la observación prolongada de las cámaras, etc. En segundo lugar está la incapacidad de los sistemas tradicionales para interpretar por sí mismos eventos o comportamientos que ocurren en el área vigilada.

Como solución a estos problemas surgen los sistemas de videovigilancia inteligentes, los cuales están destinados a detectar eventos específicos. Sin embargo estos sistemas se construyen con el único propósito de detectar los eventos para los cuales fueron diseñados. Por lo cual, surge la necesidad de construir sistemas flexibles y escalables, que permitan incluir nuevos módulos para la detección de nuevos eventos.

Frente a esta necesidad, en este trabajo se presenta una arquitectura flexible y escalable para sistemas de videovigilancia inteligentes. Esta arquitectura se basa en un modelo cliente-servidor y propone unos componentes denominados: módulos de análisis, motor de analítica y motor de reglas, los cuales permiten al sistema adaptarse a diferentes entornos e incluir nuevos detectores de eventos, en donde el primer detector implementado está destinado a la detección de objetos abandonados y robados.

Para determinar la arquitectura del sistema propuesto y para que éste cumpla con los requerimientos del sector, se analizaron las diferentes necesidades que deben solventar los sistemas de videovigilancia inteligentes y se estudiaron algunas arquitecturas de sistemas propuestos por la academia.

Los algoritmos que se implementaron en las diferentes etapas del sistema de videovigilancia se escogieron mediante un estudio del estado del arte de los diferentes algoritmos presentes en los sistemas de videovigilancia inteligentes, este estudio generó un conjunto de técnicas en donde posteriormente se evaluaron experimentalmente los algoritmos de sustracción de fondo y detección de objetos.

Finalmente, se presentó los resultados del sistema de videovigilancia construido, en donde se realizó una evaluación cualitativa y cuantitativamente de las diferentes capas presentes en la arquitectura propuesta.

# Abstract

The interest in surveillance systems has grown rapidly in different sectors, such as the academic, public, military and comercial ones. However, there are two critical problems in the traditional surveillance systems: The first one is its dependency on human operation, which by default, is subject to grave deficiencies, such as expensive worker wages, the inability to observe multiple cameras at the same time and missing important events on screen due to eye fatigue, and so on. Secondly, the systems themselves lack the awareness necessary to interpreter events or behavior occurring in their line of sight.

In response to these problems, the industry has developed intelligent video surveillance systems, which are capable of detecting specific events. Unfortunately, these can solely detect the very specific occurrences for which they were designed. This is why the need to build scalable and flexible systems has arisen, in which is possible to add new modules for the inclusion of new events.

In the face of this new reality, I am presenting in this thesis a new flexible and scalable architecture for video intelligent video surveillance systems. This new approach is based on a client-server model that proposes the following specific components: analysis modules, analytics engine and a rules engine; all of which will allow the system to adapt to its new surroundings and, at the same time, include new event detectors, of which the first implemented detector is destined to detect abandoned and stolen objects.

In order to determine the architecture of my system, and consequently assure that it will comply with the requirements of the sector; I have made an analysis of the different needs that must be met in order to address the pressing challenges of the present intelligent video surveillance systems. At the same time, I have studied some of the system architectures proposed by the academy.

The techniques implemented in the different stages of my system were selected by an in-depth study of the current algorithms present in the state of the art. This study generated multiple techniques by which I evaluated, in an experimental manner, the algorithms of background subtraction and object detection.

Finally, I present the results of my constructed video surveillance system, in which I perform an evaluation in qualitative and quantitative forms of the different layers built in proposed architecture.

# *Agradecimientos*

*En primer lugar me gustaría agradecer a mi madre por todo su amor y apoyo incondicional, por cada uno de los segundos que me ha dedicado, en gran parte es gracias a ella que soy lo que soy.*

*Mil gracias serían pocas para Carolina Sarmiento, porque me ha brindado todo su amor, su compañía, porque me ha aguantado durante todo el tiempo que duré realizando este trabajo, por su colaboración y su continuo sirili a continuar mejorando cada día.*

*Agradezco al ingeniero Milher Tovar, por aceptarme para realizar este trabajo de grado bajo su dirección, por su tiempo y orientación. También agradezco al ingeniero Luis Gabriel Marín por su continuo apoyo, por sus continuos consejos que me han ayudado y sin duda me seguirán ayudando a mejorar profesionalmente. A ambos les agradezco por su apoyo y su conocimiento.*

*Por último quiero darle las gracias a mi hermano Hugo Alberto, a mis amigos Jairo Viuche, Luis Angel Hernandez y Amilcar Rojas por la colaboración que cada uno de ellos me ha dado en el desarrollo de esta tesis.*

# Tabla de Contenido

pág.

# Lista de Figuras

pág.

# Lista de Tablas

pág.

# *CAPÍTULO 1*

# *CONTEXTUALIZACIÓN*

# 1. Contextualización

El interés de los sistemas de videovigilancia ha ido creciendo rápidamente en los escenarios públicos, militares y comerciales, sin embargo, la dependencia de los operadores humanos en los sistemas de vigilancia tradicionales implica graves deficiencias, como son: los altos costos de mano de obra, las limitaciones para la monitorización de múltiples cámaras, las pérdidas de eventos importantes, etc. Los sistemas de videovigilancia inteligentes pueden complementar o incluso remplazar los sistemas tradicionales.

Las principales etapas de un sistema de videovigilancia inteligente son: modelado del entorno, reconocimiento de objetos, seguimiento de objetos, detección de eventos y recuperación de contenido multimedia; en donde cada etapa requiere información de la etapa inmediatamente anterior, con el fin de identificar en tiempo real situaciones anómalas o indeseadas.

Esta tesis propone un sistema de videovigilancia inteligente, denominado Beholder, el cual es flexible y escalable, en el sentido de que es posible cambiar y combinar, con relativa facilidad, algoritmos en las diferentes etapas con el fin de que se ajusten a entornos específicos.

## 1.1. Contexto general

El término videovigilancia proviene de la traducción del inglés de "video surveillance", la Real Academia Española de la lengua lo define como: "Vigilancia a través de un sistema de cámaras, fijas o móviles". Los sistemas de videovigilancia existen desde hace más de tres décadas, pero en los últimos años se ha avanzado bastante en este campo. Esto se debe a tres motivos principales: el desarrollo tecnológico, la demanda de mayores niveles de seguridad y el estudio de técnicas de análisis de vídeo.

Desde la aparición de las primeras cámaras analógicas con tubo conectadas a VCR[1] hasta los nuevos sistemas semiautomáticos existe una gran diferencia. Las nuevas tecnologías y los avances de las últimas décadas han permitido conseguir sistemas con elevadas prestaciones. Las cámaras actuales poseen características como: alta resolución, zoom, visión infrarroja, múltiples lentes, comunicación mediante el protocolo TCP/IP e incluso características de detección de movimiento embebidas.

En un principio, las cintas de grabación y los primeros DVR[2] limitaban la duración de las grabaciones, lo que obligaba a grabar pocas imágenes por segundo (fps[3]). El enorme

[1] Del inglés Video Casette Recorder
[2] Del inglés Digital Video Recorder
[3] Imágenes por segundo, del inglés frames per second, es la medida de la frecuencia a la cual un reproductor de imágenes genera distintos fotogramas (frames).

desarrollo de los sistemas de almacenamiento hace que actualmente el espacio para la grabación ya no suponga mayor problema.

Los avances citados previamente han venido acompañados de una importante disminución en los costos de los sistemas, lo que ha provocado una continua subida de la demanda en seguridad [58]. Actualmente no solo se instalan sistemas de videoseguridad en entornos tradicionales como: bases militares, edificios gubernamentales, aeropuertos, estaciones, bancos, etc; sino que también en ambientes como vías públicas, estadios, instalaciones industriales, oficinas, locales comerciales, hogares y demás recintos privados [12].

Este aumento de la variedad de escenarios ha generado la necesidad de mejorar los sistemas y adaptarlos a las necesidades específicas de cada uno de ellos. Muchos sistemas de seguridad se han diseñado y desarrollado por la industria y la academia. En la literatura, estos sistemas se describen a un muy alto nivel, lo que impide distinguir las funcionalidades entre las diferentes aproximaciones.

En la última década varios proyectos de investigación se han desarrollado con el fin de lograr avances en el ámbito de la vigilancia inteligente. Algunos de los problemas que se abordan en estos proyectos están encaminados a la detección, seguimiento y clasificación de objetos y detección de eventos. Un proyecto de investigación relevante fue CROMATICA [39], concebido con el objetivo de mejorar la vigilancia de los pasajeros en el transporte público. Este proyecto combina tecnologías de análisis de video y transferencia inalámbrica de datos [52]. Posteriormente tal proyecto se amplió a PRISMATICA [71], con una arquitectura distribuida que permitía la comunicación entre dispositivos heterogéneos por medio de CORBA. Algunos de los eventos anómalos que detectaba el sistema eran: detección de personas accediendo a áreas prohibidas, gente haciendo fila de forma inusual o personas que caminan en la dirección equivocada. En estos sistemas, el análisis y la interpretación de los hechos y comportamientos fueron llevados a cabo en ambientes cerrados y se centró principalmente en las personas.

Otro proyecto de investigación relevante fue VSAM [60], financiado por el Departamento de Defensa de EE.UU. El sistema utilizaba flujos de video capturadas por cámaras de seguridad, para clasificar objetos en movimiento como: personas, grupos de personas o vehículos. La capa responsable de la identificación de eventos fue diseñada para analizar las trayectorias de los vehículos y los movimientos de las personas. El sistema era capaz de determinar si una persona estaba caminando, corriendo o realizando otras actividades.

W4 [27] fue un sistema enfocado a analizar el comportamiento de las personas en ambientes extramurales, ignorando el comportamiento de elementos tales como vehículos. El sistema detectaba y determinaba las siluetas de las personas mediante el uso de las imágenes obtenidas por las cámaras de seguridad y sensores infrarrojos. Una vez obtenida la silueta, un segundo módulo identificaba las partes relevantes del cuerpo humano (cabeza, tronco, brazos y piernas). La posición de estas partes fue utilizada para identificar

acontecimientos de interés, tales como personas corriendo, caminando con paquetes, alzando las manos, etc.

ASESOR [64] fue un proyecto desarrollado con el objetivo de analizar el comportamiento de las personas en las estaciones de los metros. Al igual que PRISMATICA, este sistema se compone de varias cámaras distribuidas alrededor de una estación, conectadas a un servidor central que lleva a cabo el análisis de imágenes. Este módulo almacena en una base de datos la secuencia de imágenes que representan una situación anómala.

AVITRACK [10] y ARGOS [11] son proyectos de investigación más recientes, que tienen como objetivo analizar la conducta en los aeropuertos de personas y vehículos, especialmente en las tareas de mantenimiento, tales como la carga y descarga de aviones, abastecimiento de combustible, reparaciones individuales, etc. Por otra parte, ARGOS fue concebido para hacer frente al control del tráfico marítimo.

La mayor parte de las obras anteriormente citadas se componen de múltiples módulos de análisis diseñadas de manera independiente sin seguir un esquema común. Estos sistemas pueden ser entendidos como un conjunto de piezas que actúan por separado para detectar eventos específicos en escenarios particulares, sin establecer relaciones entre ellos para obtener un análisis global. Sin embargo, el diseño de los sistemas de vigilancia inteligentes debería estar encaminado hacia el desarrollo de sistemas escalables que proporcionen mecanismos para incluir nuevos módulos de análisis y en lo posible que no afecten los componentes que ya están desplegados.

Teniendo en cuenta los trabajos mencionados, se puede percibir la tendencia general de diseñar e implementar, tanto a nivel industrial como académico, sistemas de videovigilancia inteligentes escalables, con carga computacional distribuida, procesamiento en tiempo real, bajo consumo computacional, comunicación mediante las redes estándar y reconfiguración en caliente.

## 1.2. Descripción del problema

Con los recientes avances computacionales, los sistemas de videovigilancia se han convertido en un campo muy popular de investigación y desarrollo [29]. Los lugares públicos son monitoreados por muchas cámaras con el fin de garantizar el orden y la seguridad; y el procesamiento digital de imagen es un medio eficaz para tratar con la gran cantidad de datos generada por las redes de cámaras.

Los sistemas de seguridad más utilizados son los CCTV[1] tradicionales, se estima que existen 40 millones de cámaras de seguridad instaladas en el mundo, de las cuales más del 95% son análogas, únicamente en Londres hay más de dos millones de cámaras análogas, y

[1] Circuito Cerrado de Televisión, del inglés Closed Circuit Television.

15 millones en Estados Unidos [18]. Cada día se instalan más cámaras CCTV para combatir el creciente sentimiento de inseguridad. La revista New Scientist afirma:

"Si avanza la tecnología, se podría poner fin a un antiguo problema que ha perseguido a los sistemas CCTV casi desde el principio. Es simple: hay muchas cámaras y muy pocos pares de ojos para atenderlas a todas. Con más de un millón de cámaras CCTV solamente en el Reino Unido, éstas se están volviendo un problema cada vez más difícil de manejar [35]."

Tan Kok Kheng, vicepresidente de la división OEM[1], de una de las principales distribuidoras de sistemas de vigilancia avanzadas, WPG Systems, afirma que tras 20 minutos de vigilancia, la atención humana a los detalles del vídeo disminuye hasta niveles inaceptables, y cada vez es más difícil cumplir con las demandas del sector. En consecuencia, usualmente estas cámaras son poco o nada monitoreadas, de hecho a menudo son usadas solamente como medio de almacenamiento y reproducción de eventos una vez se conoce que un incidente ha tenido lugar.

Los sistemas de videovigilancia inteligentes (ISS[2]), pueden complementar o incluso remplazar los sistemas de vigilancia tradicionales. En los ISS, las tecnologías de: visión artificial, reconocimiento de patrones e inteligencia artificial; son desarrolladas para identificar comportamientos anormales en video. Como resultado, menos operadores pueden monitorizar más escenarios con mayor precisión.

Por lo cual, si se implementaran sistemas de videovigilancia inteligentes, se tendría una seguridad proactiva, dado que el objetivo de estos sistemas es obtener una descripción e interpretación automática de los eventos y tomar las acciones apropiadas en tiempo real [29].

Una aplicación importante de la videovigilancia inteligente es la detección de objetos abandonados y robados, dado que facilitan la prevención de robos, y ayudan a prevenir los riesgos potenciales de atentados terroristas, que requieren un control particular e inmediato para garantizar la seguridad, especialmente en lugares altamente poblados como aeropuertos, edificaciones públicas, estaciones de transporte, etc.

Con el fin de masificar el uso de los sistemas de videovigilancia inteligentes, se necesitan algoritmos robustos para la detección de movimiento, clasificación, seguimiento de objetos y detección de eventos [5]. El avance en estos campos no solo se traduce en sistemas de vigilancia confiables, sino que otros campos también se benefician de estas investigaciones. Algunos ejemplos son: realidad virtual, compresión de video, interacción humano computador, bases de datos de contenido multimedia, etc.

[1] Fabricante de Equipamiento Original, del inglés Original Equipment Manufacturer.
[2] Del inglés Intelligent Surveillance Systems.

Sin embargo el desarrollo de los ISS así como la detección de objetos abandonados y robados son tareas altamente complejas, debido a problemas relacionados con el modelado y la actualización del entorno de la escena, variaciones de la apariencia del objeto, movimiento de objetos con respecto a la cámara, cambios de iluminación, velocidad de los objetos y oclusiones entre objetos en movimiento y estáticos [34].

Por lo anterior la mayoría de los sistemas de videovigilancia inteligente existentes sufren de problemas de procesamiento, debido al alto costo computacional de los algoritmos, o poca escalabilidad, ya que están generalmente diseñados para contextos específicos y no pueden adaptarse fácilmente a otros contextos.

## 1.3. Solución propuesta

Para dar solución al problema planteado el presente trabajo propone una aproximación para sistemas de videovigilancia inteligente. Esta aproximación se basa en un diseño flexible y escalable, que añade una baja carga computacional a los algoritmos que corren en el sistema. Debido a la fácil integración de los componentes, el sistema proporciona un buen entorno para que los investigadores desarrollen nuevos algoritmos de análisis y se combinen diferentes algoritmos en las diferentes etapas para mejorar el rendimiento en entornos concretos.

El sistema objetivo que se construye con el diseño planteado, permite, en la etapa de detección de eventos, detectar objetos abandonados y robados. Lo anterior es un apoyo para el personal de vigilancia, ya que emite una alarma temprana si ocurren los eventos citados.

La arquitectura del sistema se basa en un modelo cliente/servidor y está compuesto por cuatro capas independientes: capa de entrada, capa de procesamiento, capa de gestión de datos y la capa de presentación. Cada capa está diseñada de forma modular y cumple una función específica. La capa de presentación corre en el cliente mientras que las otras capas se ejecutan sobre el servidor.

La capa de entrada se compone del módulo de captura el cual adquiere el video de diferentes fuentes, éste envía la secuencia de video frame[1] a frame al motor de analítica; los frames de video se envían en el formato de compresión de imagen JPEG (ISO/IEC 10918-1) o sin comprimir. El sistema soporta nuevas fuentes de video con relativa facilidad. Actualmente, y por restricciones presupuestales, el sistema está desarrollado para trabajar con cámaras USB.

La capa de procesamiento se compone de uno o más motores de analítica, que su vez están integrados por seis módulos de análisis: preprocesado, modelado del entorno, detección de objetos, clasificación de objetos, seguimiento de objetos y reconocimiento de eventos. El

[1] Fotograma o cuadro, una imagen particular dentro de una sucesión de imágenes que componen una animación.

primer módulo en ejecución, especificado por el motor de analítica, se comunica con la capa de entrada para adquirir el video, después transfiere los frames procesados al siguiente a módulo de análisis y así sucesivamente.

Para permitir la fácil integración de nuevos módulos de análisis, se crearán plantillas genéricas que definen como se interrelacionan los módulos con el sistema, y permitan un rápido desarrollo de nuevos algoritmos.

La capa de gestión de datos se encarga del almacenamiento de las imágenes procesadas y los descriptores. Esta capa se compone de una base de datos relacional que gestiona los resultados de la capa de procesamiento y la configuración del sistema.

La capa de presentación es la encargada de mostrar en tiempo real las imágenes procesadas por la capa de procesamiento o de consultar las imágenes almacenadas en la base de datos. Esta capa proporciona una interfaz amigable al operador para que pueda configurar el sistema, ver los videos de las cámaras conectadas o los videos almacenados en la base de datos.

El sistema se desarrolló usando el lenguaje de programación C++. Este lenguaje es requisito dado que, como ya se ha expuesto, la velocidad de procesamiento es clave en las tareas de visión artificial y más aún en el de videovigilancia ya que requiere procesamiento en tiempo real. La librería multiplataforma *Qt* fue usada para la construcción de la interfaz gráfica de usuario, además permite portar fácilmente el sistema Beholder a otras plataformas. OpenCV[1] es la librería principal usada para el procesamiento digital de imágenes y para facilitar la implementación de algoritmos y métodos que se hayan especificado en Matlab®[2]. El sistema de gestión de base de datos relacional *MySQL* es usado para gestionar la base de datos encargada de guardar los datos y la configuración del sistema.

## 1.4. Objetivos

### 1.4.1. General

Desarrollar un sistema de videovigilancia inteligente para la detección de objetos abandonados y robados, que apoye al personal de vigilancia a detectar el robo o abandono de objetos sospechosos en el área vigilada.

[1] Librería libre de visión artificial.
[2] Abreviatura de MATrix LABoratory "laboratorio de matrices", es una herramienta de software matemático que ofrece un entorno de desarrollo integrado.

### 1.4.2. Específicos

- Estudiar y evaluar las técnicas para la detección de movimiento, seguimiento de objetos y clasificación de objetos abandonados y robados, presentes en el estado del arte, con el fin de seleccionar los algoritmos que serán implementados en el sistema.

- Estudiar las ventajas y desventajas de las arquitecturas de los sistemas de videovigilancia inteligentes presentes en el estado del arte, con el fin de proponer una arquitectura modular y escalable.

- Seleccionar un conjunto de datos para entrenar los algoritmos previamente seleccionados, con el fin de integrarlos en la arquitectura propuesta.

- Validar las técnicas implementadas con secuencias de video provenientes de bases de datos especializadas en el estudio de sistemas de videovigilancia, con el fin estimar la efectividad del sistema construido.

## 1.5. Contenido del trabajo

El capítulo 2 contiene el marco teórico de la investigación, donde se detallan los fundamentos necesarios para el desarrollo del sistema objetivo. Se explica inicialmente la estructura y evolución de los sistemas de videovigilancia inteligentes, después se explican las principales técnicas usadas en cada una de las etapas y finalmente se describen temas relacionados a la ingeniera de software.

En el capítulo 3 se presenta el desarrollo metodológico del sistema, en donde se presenta las diferentes tareas realizadas en cada fase de la metodología usada para el desarrollo del sistema. Seguidamente se presenta algunas consideraciones metodologías para la selección de la arquitectura del sistema. Por último se presenta el flujo metodológico para la selección de los algoritmos de análisis.

En el capítulo 4 se presenta un análisis del sistema objetivo, en donde se describen los usuarios del sistema y como éstos interactúan con el sistema. Después se presenta una descripción general del sistema, tanto a nivel físico como lógico. Por último se hace una descripción del proceso implementado para la detección de objetos abandonados y robados.

El capítulo 5 describe el proceso para hacer la evaluación de los algoritmos de análisis del sistema, para posteriormente mostrarse los resultados del sistema y de las diferentes etapas del mismo.

En el capítulo 6, se presenta un resumen y unas conclusiones obtenidas tras el desarrollo del sistema. Igualmente se exponen algunos trabajos futuros encaminados a la mejora del sistema.

# *CAPÍTULO 2*

# *MARCO TEÓRICO*

# 2. Marco teórico

Este capítulo presenta inicialmente la estructura y evolución de los sistemas de videovigilancia inteligentes. Seguidamente se hace una descripción de las etapas que intervienen en esta clase de sistemas: modelado del entorno, detección de objetos, clasificación de objetos, seguimiento de objetos e interpretación de eventos. Esto con el propósito de entender las técnicas que se emplearon en la construcción del sistema objetivo.

## 2.1. Sistemas de videovigilancia

El principal objetivo de los sistemas de videovigilancia inteligentes es dar una interpretación automática de las escenas, así como entender y predecir las acciones e interacciones de los objetos observados. Los requerimientos para el diseño de este tipo de sistemas han sido objeto de varias investigaciones [52, 33, 74, 51, 42 y 50] y en general pueden describirse en las siguientes características: Sistemas escalables con cargas computacionales distribuidas, procesamiento en tiempo real, baja carga computacional, estándares para las comunicaciones a través de la red y reconfiguración online.

### 2.1.1. Evolución de los sistemas de vigilancia

Valera y Velastin [52] clasificaron los sistemas de vigilancia en tres generaciones de acuerdo a las tecnologías empleadas. La primera generación está formada por los sistemas CCTV. Estos sistemas consisten en un número de cámaras posicionadas en distintas localizaciones y conectadas a monitores, comúnmente situados en una misma sala, y supervisados por uno o varios operadores.

Los sistemas CCTV utilizan señales análogas para la distribución y el almacenamiento de la imagen. Sin embargo, esto dificulta el mantenimiento del sistema, así como el acceso remoto o la integración con otros sistemas, además, los sistemas análogos provocan la degradación de la imagen y la señal de video se vuelve susceptible al ruido. Por otro lado, este sistema tradicional de videovigilancia se vuelve ineficiente ya que la observación prolongada de los monitores causa fatiga en los vigilantes y, en consecuencia, falta de atención; lo que da lugar a la probabilidad de que una situación anómala no sea detectada.

A pesar de las deficiencias comentadas anteriormente, los sistemas de video vigilancia CCTV son ampliamente utilizados en todo el mundo, sobre todo en ámbitos comerciales e industriales [52, 29, 72]. Los sistemas CCTV más modernos solventan algunas de estas deficiencias con algunas mejoras, entre las que se incluyen las tecnologías digitales, control remoto de las cámaras desde una sala de control, que permiten el ajuste de parámetros como la inclinación o el nivel de zoom, la visión nocturna y la detección de movimiento, que permite al sistema cambiar a un estado de alerta ante posibles intrusiones.

El progreso de la tecnología llevó a una evolución en los sistemas de vigilancia y permitió el desarrollo de los sistemas semiautomáticos, conocidos como sistemas de vigilancia de

segunda generación. Los sistemas de vigilancia de segunda generación combinan las tecnologías de los sistemas CCTV con algoritmos de visión por computador e inteligencia artificial. Es decir, los sistemas de segunda generación intentan reducir la dependencia que existe con la actividad humana, interpretando en la medida de lo posible los eventos y comportamientos que se producen en el entorno monitorizado.

Actualmente, la interpretación de sucesos en entornos reales no es un problema resuelto y existe un gran número de líneas de investigación abiertas. Ni siquiera existe un consenso lo suficientemente claro sobre las tecnologías y metodologías más adecuadas para ofrecer soluciones óptimas al problema. En cuanto a los principales retos que se plantean actualmente en este tipo de sistemas podríamos destacar tres de ellos [30].

El primero es la representación del conocimiento de cualquier entorno del mundo real, para poder interpretar las situaciones que ocurren en él, es decir, capacidad de conocer los elementos que participan en la escena y qué relaciones existen entre ellos. Normalmente, estos entornos suelen ser bastante complejos y la representación del conocimiento no es una tarea trivial.

El segundo reto es el tratamiento adecuado de la incertidumbre y la vaguedad que existe en cualquier escenario real. Para un sistema artificial es prácticamente imposible afirmar, con total certeza, qué ocurre en un entorno determinado en cualquier instante de tiempo.

El tercer reto es el diseño de algoritmos eficientes que proporcionen resultados en un tiempo cercano al real. Este aspecto es vital para solventar los dos primeros enfoques, dado que si el sistema no dispone de un algoritmo eficiente, no podrá interpretar el entorno correctamente.

Finalmente, los sistemas de video vigilancia de tercera generación se caracterizan sobre todo por ser altamente distribuidos. Estos sistemas utilizan los avances de las dos generaciones anteriores y están formados por un amplio repertorio de sensores, distribuidos geográficamente por todo el entorno observado, los cuales transmiten información de forma simultánea en tiempo real.

La naturaleza distribuida de estos sistemas supone un gran avance para los sistemas de seguridad por varias razones. Una de ellas es que la carga de procesamiento no se encuentra centralizada y, por tanto, el sistema ofrece mayores garantías de responder en un tiempo cercano al real. Una segunda razón seria la ganancia en solidez, es decir, el sistema puede seguir trabajando a pesar de que algunos componentes sean dañados.

En cuanto a los principales problemas a los que se enfrentan los sistemas de tercera generación, cabe destacar la dificultad de combinar múltiples dispositivos heterogéneos en una misma red. En este caso el uso de middlewares[1] es apropiado para tal propósito. Y, por

[1] Software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones

otra parte, la necesidad de relacionar la información procedente de diversos sensores para fortalecer el proceso de razonamiento e interpretación.

La tabla 2.1 resume las principales ventajas y problemas [52] de las diferentes generaciones de los sistemas de vigilancia.

**Tabla 2.1.** Descripción de las generaciones de los sistemas de vigilancia.

| **1º Generación** | |
|---|---|
| **Técnicas** | - Sistemas análogos CCTV |
| **Ventajas** | - Dan buen rendimiento en algunas situaciones<br>- Es una tecnología madura |
| **Problemas** | - Usa técnicas análogas para la distribución y el almacenamiento las imágenes.<br>- Dependen totalmente de los seres humanos para el funcionamiento. |
| **Investigaciones actuales** | - Tratamiento, distribución y almacenamiento de imágenes mediante técnicas digitales.<br>- Algoritmos de compresión de video.<br>- Recuperación eficiente del contenido multimedia. |
| **2º Generación** | |
| **Técnicas** | - Videovigilancia automática mediante técnicas de visión artificial con tecnologías de sistemas CCTV |
| **Ventajas** | - Incrementa la eficiencia de los sistemas de video – seguridad los sistemas CCTV.<br>- Reducen la dependencia de la actividad humana para detectar situaciones anómalas. |
| **Problemas** | - Actualmente no existe una solución que permita realizar un razonamiento general sobre cualquier situación. Existen soluciones parciales, para razonar e interpretar sobre situaciones muy concretas (análisis de velocidades, trayectorias seguidas por objetos observados, etc.).<br>- Falta de soluciones robustas para disminuir las falsas alarmas. |
| **Investigaciones actuales** | - Desarrollo de algoritmos de visión artificial eficientes con respuestas en tiempo real.<br>- Representación de los elementos físicos de un entorno real y la relación que existe entre ellos.<br>- Reconocimiento de eventos y actividades.<br>- Distinción entre situaciones normales y anormales.<br>- Algoritmos de aprendizaje que amplían el conocimiento que tiene el sistema sobre el entorno. |

| | |
|---|---|
| | - Anticipación a posibles acciones que podrían dañar el entorno.<br>- Toma de decisiones y gestión de crisis. |
| **3º Generación** | |
| **Técnicas** | - Sistemas de vigilancia altamente distribuidos |
| **Ventajas** | - Información más veraz por la combinación de diferentes sensores.<br>- Descentralización de la información.<br>- Sistema distribuido. |
| **Problemas** | - Información distribuida, por lo cual hay que tener en cuenta la integración y la comunicación.<br>- Metodología de diseño.<br>- Relación e interpretación de la información que procede de múltiples fuentes. |
| **Investigaciones actuales** | - Fusión de la información<br>- Técnicas de vigilancias multi-cámara.<br>- Cada uno de los problemas que presentan los sistemas de tercera generación da lugar a una línea de investigación. |

## 2.2. Modelado del entorno

Una activa construcción y actualización del modelo del entorno es indispensable para los sistemas de video – seguridad inteligentes. El modelado del entorno puede ser clasificado en modelos 2D en imágenes planas y en modelos 3D en coordenadas del mundo real. Debido a su simplicidad, los modelos 2D tienen mayor aplicación [72].

En los sistemas de videovigilancia lo que se pretende es obtener un modelo de fondo robusto y eficiente, que evite el mayor número de falsas alarmas posibles. En este sentido, un buen modelo debe adaptarse a diferentes factores desfavorables, como son:

– Perturbaciones o movimientos: desplazamientos leves de la cámara, debidos al viento, movimientos continuados como en las hojas de los árboles, ondulaciones en el agua, etc.

– Cambios en la escena de fondo: inclusión de nuevos objetos como coches aparcados, objetos abandonados, que después un tiempo razonable en escena se deben considerar como fondo y no como motivo de alarma.

– Adaptación a los cambios de iluminación: lentos y graduales, como la variación de luz en el día; y a ser posible también a cambios rápidos y bruscos.

Hay muchos algoritmos para resolver los problemas anteriormente expuestos, Piccardi [48] propone la clasificación de métodos de modelado de fondo presente en la tabla 5.2, en donde se tiene en cuenta solamente los algoritmos con una velocidad alta de procesamiento, factor determinante para los sistemas de videovigilancia.

**Tabla 2.2.** Técnicas de modelado del entorno.

| Velocidad de procesamiento | Modelos Básicos | Modelos Paramétricos | Modelos no Paramétricos |
|---|---|---|---|
| **Alta** | Diferenciación temporal, Promedio temporal, Mediana, Media móvil | | |
| **Intermedia** | | Mezcla de Gaussianas (MoG) | Kernel density estimators (KDE), Eigenbackgrounds |

### 2.2.1. Modelos básicos

En este caso el valor de cada píxel[1] del modelo de fondo se calcula en función de los valores recientes de dicho píxel. Se utilizan modelos matemáticos sencillos como la diferencia entre imágenes, valores promedios, máximos y mínimos, etc.

En la mayoría de los modelos propuestos, el valor histórico se calcula a partir de los N frames anteriores, donde N es una constante elegida para el problema. Algunos modelos utilizan una media ponderada del valor de los píxeles, donde las últimas imágenes tienen mayor peso.

#### 2.2.1.1. Promedio temporal (average)

En este modelo se calcula una imagen de fondo estática hasta que se produce algún cambio. En ese momento el valor del fondo se corresponde con el promedio de N frames consecutivos. El modelo de fondo ($B_t$) se calcula tal que:

$$|I_t - B_t| > Th \rightarrow B_t = \frac{1}{N} \sum_{n=t}^{n=N-t} I_n \qquad (2.1)$$

[1] Acrónimo del inglés picture element, "elemento de imagen", es la menor unidad homogénea en color que forma parte de una imagen digital

$$|I_t - B_t| \leq Th \rightarrow B_t = I_t \qquad (2.2)$$

donde,

$I_t$ = imagen de entrada en el instante *t*

$B_t$ = imagen de fondo estimada en el instante *t*

Este método no resulta eficiente cuando hay varios objetos, o estos se mueven lentamente. Esto se debe a que es la información de movimiento la que se utiliza para actualizar el fondo, por lo que si un elemento se mueve lentamente no se detectará.

#### 2.2.1.2. Mediana (median)

El filtro mediana es una de las técnicas de modelado de fondo más utilizadas. El fondo se estima como la mediana de los píxeles almacenados en un buffer de N frames anteriores. Uno de los problemas de este método es que cuando se produce algún cambio el fondo se adapta muy lentamente, como se menciona en [61].

La principal ventaja de los dos métodos anteriores es que son bastante rápidos a nivel de coste computacional. Sin embargo los requisitos de memoria son altos, del orden de *N* por el tamaño de la imagen, donde *N* es el número de imágenes.

#### 2.2.1.3. Filtro de media móvil (running average)

Este tipo de filtro evita los problemas de memoria de los métodos anteriores, ya que no es necesario un buffer de almacenamiento. En este método se calcula una media ponderada entre la última imagen del background[1] y la imagen actual ($I_t$). Se hace uso del parámetro α, denominado factor de actualización, con valores del orden de $10^{-2}$. El valor de cada píxel del fondo se calcula tal que:

$$B_t = \alpha * I_{t-1} + (1 - \alpha) B_{t-1} \qquad (2.3)$$

En numerosas aplicaciones se utiliza el modelo anterior acompañado de la propiedad de selectividad, come se describe en [48]. Para ello se clasifica cada uno de los píxeles como background si $|I_t - B_t| < Th$ o como foreground[2] ($F_t$) si $|I_t - B_t| \geq Th$ de manera que:

Si *p(x,y)* es background:

$$B_t(x, y) = \alpha * I_{t-1}(x, y) + (1 - \alpha) B_{t-1}(x, y) \qquad (2.4)$$

Si *p(x,y)* es foreground:

[1] Es la escena general o superficie de una representación pictórica contra la cual se ven o representan diseños, patrones o figuras.

[2] Es la parte de la escena que está más cerca o en frente al espectador, la cual tiende a variar mientras el fondo esta estático.

$$B_t(x,y) = B_{t-1}(x,y) \qquad (2.5)$$

Con la modificación anterior lo que se consigue es evitar que el modelo de fondo se corrompa con píxeles que no pertenecen realmente a la escena. Sin embargo, en algunas aplicaciones de videovigilancia puede resultar interesante e incluso necesario incluir información relativa al foreground. En este sentido se hace uso de un modelado de fondo con las siguientes características:

Si *p(x,y)* es background:

$$B_t(x,y) = \alpha_{back} * I_{t-1}(x,y) + (1-\alpha_{back})B_{t-1}(x,y) \quad (2.6)$$

Si *p(x,y)* es foreground:

$$B_t(x,y) = \alpha_{fore} * I_{t-1}(x,y) + \left(1-\alpha_{fore}\right)B_{t-1}(x,y) \quad (2.7)$$

Se utilizarán para este método valores típicos de $\alpha_{back} \approx 10^{-2}$ y $\alpha_{fore} \approx 10^{-3}$.

Se consigue en este caso una solución intermedia, de forma que el modelo de fondo no se corrompa fácilmente pero que sea adaptable a cambios. De esta forma, aquellos objetos en movimiento detectados que permanezcan un tiempo razonable en escena formarán parte del fondo.

### 2.2.2. Modelos paramétricos

Los modelos paramétricos basan el modelo de fondo en una distribución estadística estándar de la cual hay que estimar los parámetros. Se trata de métodos más complejos con mayores capacidades de adaptación frente a cambios leves, al ruido, etc. La complejidad computacional se traduce en una menor velocidad de cómputo en comparación con los modelos básicos, sin embargo estos métodos funcionan bien sin la necesidad de almacenar tantas imágenes como el filtro media o mediana [48].

#### 2.2.2.1. Mezcla de gaussianas (mixture of gaussians)

Estos métodos trabajan con fondos multimodales [48]. Se utilizan generalmente en escenarios donde hay píxeles cuya intensidad fluctúa constantemente, tales como hojas de árboles, cámaras de baja calidad, agua en movimiento, etc.
Debido a estos cambios de intensidad, hacer uso de una única gaussiana por píxel (media y varianza de las intensidades) no es suficiente. Lo que pretende este método es caracterizar cada píxel como una mezcla de gaussianas (MoG).

La distribución de cada píxel f $f(I_t=u)$ se modela con *k* gaussianas de la forma:

$$f(I_t = u) = \sum_{i=t}^{k} w_{i,t} * n(u{:}\,\mu_{i,t}, \sigma_{i,t}) \qquad (2.8)$$

Donde $n(u{:}\,\mu_{i,t}, \sigma_{i,t})$ es la componente gaussiana i-ésima, con media de intensidad $\mu_{i,t}$, cuya desviación estándar es $\sigma_{i,t}$y con un peso $w_{i,t}$ que determina la cantidad de distribución utilizada en dicha componente. Los valores típicos de *K* varían entre 3 y 5. Esta descripción de MoG se basa en el esquema de Cheung y Kamath [61].

Para actualizar el modelo de fondo, se compara la intensidad de cada píxel del frame actual con sus posibles distribuciones en la imagen de fondo. Si el valor del píxel no supera la media $\mu_{i,t}$ considerando la desviación $\sigma_{i,t}$, se considerará un píxel de fondo y se actualizarán los parámetros para ese píxel. Si el píxel no se parece a ninguna de sus distribuciones asociadas se actualizará el modelo sustituyendo la distribución de menor peso.

### 2.2.3. Modelos no paramétricos

Estos métodos se basan en una estimación no paramétrica de la función densidad de probabilidad de una variable aleatoria. Los modelos no paramétricos son métodos complejos en los que no se asumen distribuciones estándar de probabilidad.

#### 2.2.3.1. Kernel density estimators (KDE)

Este modelo estima la probabilidad de la intensidad de cada píxel en función de una serie de muestras de intensidad anteriores para dicho píxel. La función densidad de probabilidad viene dada por el histograma de los N últimos valores, que se han almacenado en el buffer.

Según Elgammal *et al.*, en [2] la pertenencia al fondo de un píxel concreto está dada por la muestra del valor histórico del píxel $x_1, x_2, \ldots,$x3, la función densidad de probabilidad de que ese píxel tome valor $x_t$ en el instante *t* puede ser estimada de forma no paramétrica utilizando un kernel estimator *K* tal que:

$$\Pr(x_t) = \frac{1}{n}\sum_{i=1}^{k} K(x_t - x_i) \qquad (2.9)$$

El píxel se considerará fondo si $\Pr(x_t) < Th$, en caso contrario será frente.

Este método soporta cambios leves en la escena, movimientos en las hojas de árboles y es robusto al ruido. Sin embargo tiene una alta carga computacional, por lo que será más lento que otros métodos de los citados anteriormente.

#### 2.2.3.2. Eigenbackgrounds

A diferencia de los otros métodos que modelan cada pixel del fondo independientemente, Eigenbackgrounds captura la correlación espacial mediante el análisis de componentes principales a un conjunto de $N_L$ frames que no contienen objetos del foreground. Lo que resulta en un conjunto de funciones de las cuales solo las primeras d funciones tienen que capturar las características de apariencia de los frames. Un nuevo frame puede entonces proyectarse al eigenespacio *d*, definido por las d funciones básicas, y proyectarse de vuelta al espacio original de la imagen. Como estas funciones básicas solo modelan la parte estática de la escena, cuando no hay objetos del foreground, la imagen proyectada de vuelta no contendrá ningún objeto del foreground y como tal puede usarse como modelo de fondo.

## 2.3. Eliminación de sombras

Uno de los principales desafíos en la detección de objetos consiste en la identificación de aquellas sombras proyectadas por objetos en movimiento, tanto sobre la escena como sobre éste mismo.

Una sombra es una región de oscuridad donde la luz es obstaculizada por un objeto parcial o completamente. Puede proveer información importante acerca de la forma del objeto que la induce, así como de la orientación de la luz. Sin embargo, es uno de los principales problemas en la detección y segmentación de objetos en movimiento, debido a que no se tiene información de la orientación ni cantidad de luz de la escena.
Específicamente, el efecto de las sombras puede provocar, en determinadas situaciones, la fusión de varios objetos independientes, la distorsión de la forma del objeto detectado o incluso la no detección de objetos, debido a la proyección de una sombra sobre dichos objetos, así como se puede observar en la figura 2.1.

**(a)** **(b)**

**Figura 2.1**. Distorsión de la forma del objeto: (a) Sin detección de sombras. (b) Con detección de sombras (Tomado de http://cvrr.ucsd.edu/aton/shadow/).

### 2.3.1. Clasificación de sombras

Generalmente, las sombras pueden dividirse entre sombras estáticas y sombras dinámicas. Sin embargo, las sombras estáticas no son detectadas en la etapa de detección de movimiento, ya que son modeladas como parte del background de la escena. Por tanto, tan sólo serán objeto de interés aquellas sombras dinámicas asociadas a objetos tales como vehículos o personas en movimiento.

Debido a las diferentes condiciones de iluminación posibles en una escena y las características espectrales de las regiones sombreadas, las sombras se pueden clasificar en dos grupos sombras invisibles y sombras visibles, tal como se puede ver en la figura 2.2.

**(a)** **(b)**

**Figura 2.2.** Reclasificación de sombras: (a) Sombras invisibles (b) Sombras visibles (Tomado de http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html).

### 2.3.2. Algoritmos de detención de sombras

Prati *et al*., [56] proponen una clasificación de algoritmos, tal como se puede observar en la Figura 2.3.

Se consideran dos procesos de decisión diferentes: métodos determinísticos y métodos estadísticos. Los métodos estadísticos se basan en procesos de decisión binarios (On/Off). Adicionalmente, los métodos determinísticos también pueden sub-dividirse, basándose en si la decisión binaria se toma a partir del conocimiento del modelo de la escena o no. Los métodos estadísticos utilizan funciones probabilísticas para tomar estas decisiones, donde la etapa de selección de parámetros se torna crítica para conseguir un correcto comportamiento en la detección de sombras, lo cual provoca que la escisión de éstos en dos sub-niveles: paramétricos y no paramétricos.

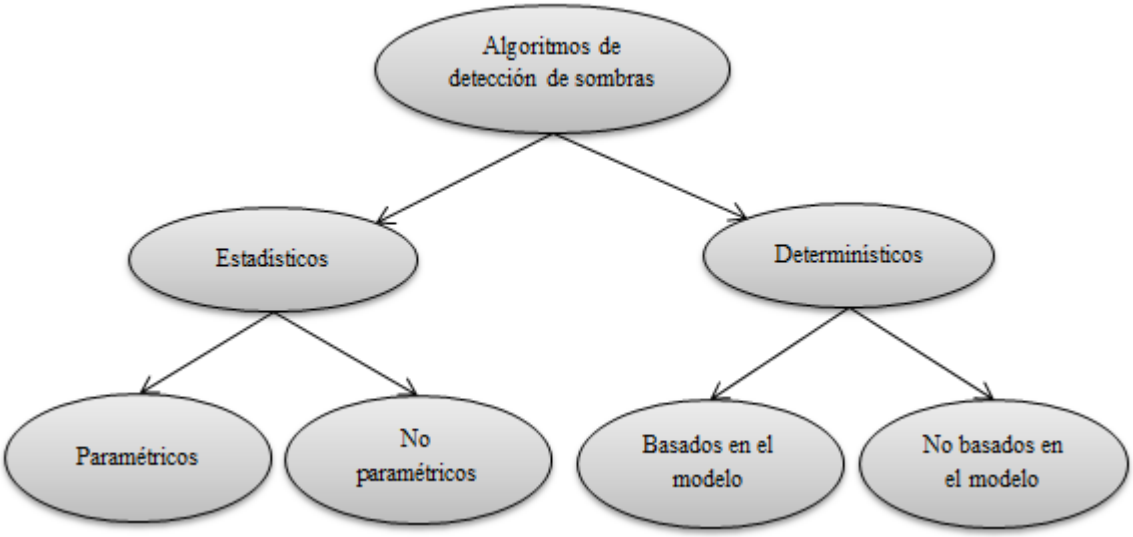



**Figura 2.3.** Clasificación de los algoritmos de detección de sombras basada en el proceso de decisión. (Adaptado de Prati *et al.*, [56])

A continuación, se describen brevemente aquellas propuestas representativas de tres de las cuatro categorías clasificadas en función del proceso de selección. El método determinístico basado en el modelo no se ha incluido en este estudio, pese a que ofrece indudablemente los mejores resultados, ya que es excesivamente complejo y con un alto coste computacional.

#### 2.3.2.1. Método estadístico no paramétrico

El trabajo propuesto por [68] es el ejemplo de método estadístico no paramétrico que va a analizarse.

En primer lugar, se utilizan los primeros $\boldsymbol{N}$ frames para calcular las medias y varianzas de los distintos canales de color para cada píxel (en este caso, en el espacio de color RGB[1]). Así, se obtiene $\boldsymbol{E_i} = [\boldsymbol{\mu_{R_i}}, \boldsymbol{\mu_{G_i}}, \boldsymbol{\mu_{B_i}}]$ (vector media), y $\boldsymbol{S_i} = [\boldsymbol{\sigma_{R_i}}, \boldsymbol{\sigma_{G_i}}, \boldsymbol{\sigma_{B_i}}]$ (vector varianza), para un píxel localizado en *i*.

La distorsión del brillo $\boldsymbol{\alpha_i}$ y la distorsión de la crominancia $\boldsymbol{CD_i}$ de la diferencia entre el valor esperado del píxel y su valor en la imagen actual $\boldsymbol{I_i} = [\boldsymbol{I_{R_i}}, \boldsymbol{I_{G_i}}, \boldsymbol{I_{B_i}}]$, se calcula de la siguiente manera:

[1] Composición del color en términos de la intensidad de los colores primarios con que se forma el rojo, el verde y el azul

$$\boldsymbol{\alpha_i} = \frac{\left(\frac{I_{R_i}\mu_{R_i}}{\sigma_{R_i}^2} + \frac{I_{G_i}\mu_{G_i}}{\sigma_{G_i}^2} + \frac{I_{B_i}\mu_{B_i}}{\sigma_{B_i}^2}\right)}{\left(\left[\frac{\mu_{R_i}}{\sigma_{R_i}}\right]^2 + \left[\frac{\mu_{G_i}}{\sigma_{G_i}}\right]^2 + \left[\frac{\mu_{B_i}}{\sigma_{B_i}}\right]^2\right)} \quad (2.10)$$

$$\boldsymbol{CD_i} = \sqrt{\left(\frac{I_{R_i} - \alpha_i\mu_{R_i}}{\sigma_{R_i}}\right)^2 + \left(\frac{I_{G_i} - \alpha_i\mu_{G_i}}{\sigma_{G_i}}\right)^2 + \left(\frac{I_{B_i} - \alpha_i\mu_{B_i}}{\sigma_{B_i}}\right)^2} \quad (2.11)$$

Al normalizar $\boldsymbol{CD_i}$ y $\boldsymbol{\alpha_i}$ se clasifica el píxel en una de estas tres categorías:

$$\boldsymbol{C_i} = \begin{cases} Foreground: \widehat{\boldsymbol{CD_i}} > \boldsymbol{\tau_{CD}} \; \boldsymbol{o} \; \widehat{\boldsymbol{\alpha_i}} < \boldsymbol{\tau_{\alpha l0}} \\ Background: \widehat{\boldsymbol{\alpha_i}} < \boldsymbol{\tau_{\alpha 1}} \; \boldsymbol{y} > \boldsymbol{\tau_{\alpha 2}} \\ Sombra: \widehat{\boldsymbol{\alpha_i}} < \boldsymbol{0} \end{cases} \quad (2.12)$$

Donde $\tau_{CD}, \tau_{\alpha l0}, \tau_{\alpha 1}$ y $\tau_{\alpha 2}$ son umbrales de decisión determinados empíricamente.

#### 2.3.2.2. Método estadístico no paramétrico

Para analizar el comportamiento de los métodos estadísticos paramétricos, se va a estudiar el algoritmo descrito en [47], utilizado en escenas de tráfico.

Se utilizan los primeros $\boldsymbol{N}$ frames para calcular, para cada píxel, las medias $\boldsymbol{E_i^B}\left[\boldsymbol{\mu_{R_i}^B}, \boldsymbol{\mu_{G_i}^B}, \boldsymbol{\mu_{B_i}^B}\right]$ y varianzas $\boldsymbol{S_i^B}\left[\boldsymbol{\sigma_{R_i}^B}, \boldsymbol{\sigma_{G_i}^B}, \boldsymbol{\sigma_{B_i}^B}\right]$, de los distintos canales de color (de nuevo, RGB) del background.

A continuación, y asumiendo que $\boldsymbol{v} = [\boldsymbol{R}, \boldsymbol{G}, \boldsymbol{B}]^{\boldsymbol{T}}$ es el valor de un píxel no sombreado, se realiza una transformación lineal $\boldsymbol{v} = \boldsymbol{D}\boldsymbol{v}$ (donde $\boldsymbol{D}$ es una matriz diagonal obtenida de manera empírica, relacionada con la reflectancia del background) para determinar cuál será el valor del píxel cuando se encuentre en una región sombreada.

$$\boldsymbol{D} = \begin{vmatrix} \boldsymbol{d_R} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{d_G} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{d_B} \end{vmatrix} \quad (2.13)$$

Si el background no es plano, o lo que es lo mismo, presenta diferentes texturas, deberán considerarse tantas matrices $\boldsymbol{D}$ como sean necesarias para representar el comportamiento de éstas.

Por tanto, y dadas las medias y varianzas de los distintos canales de color para un píxel en concreto, se puede considerar que los valores de dicho píxel, cuando éste se encuentre en una región sombreada, serán los siguientes:

$$\begin{aligned} \boldsymbol{\mu}_{R_i}^{S} &= \boldsymbol{\mu}_{R_i}^{B} \boldsymbol{d}_{R} & \boldsymbol{\sigma}_{R_i}^{S} &= \boldsymbol{\sigma}_{R_i}^{B} \boldsymbol{d}_{R} \\ \boldsymbol{\mu}_{G_i}^{S} &= \boldsymbol{\mu}_{G_i}^{B} \boldsymbol{d}_{G} & \boldsymbol{\sigma}_{G_i}^{S} &= \boldsymbol{\sigma}_{G_i}^{B} \boldsymbol{d}_{G} \\ \boldsymbol{\mu}_{B_i}^{S} &= \boldsymbol{\mu}_{B_i}^{B} \boldsymbol{d}_{B} & \boldsymbol{\sigma}_{B_i}^{S} &= \boldsymbol{\sigma}_{B_i}^{B} \boldsymbol{d}_{B} \end{aligned} \quad (2.14)$$

Finalmente, se estiman las probabilidades del píxel de pertenecer al background, foreground, o de ser clasificados como sombras.

### 2.3.2.3. Método determinístico no basado en el modelo

Este método se basa en la aplicación de umbrales sobre la reducción de intensidad y cromaticidad para evaluar si un determinado píxel se encuentra en una región sombreada o no.

En [16] se propone la aplicación de este método en el espacio de color HSV[1]. El motivo principal por el cual se inclinan por este espacio de color radica en que el comportamiento de dicho espacio se corresponde estrechamente con la percepción humana del color [24], y ofrece una mayor precisión en la detección de sombras, tal y como se verá más adelante.

El proceso de decisión que se sigue para detectar si un determinado píxel se encuentra en una región sombreada $\boldsymbol{S_k}$ $(\boldsymbol{x},\boldsymbol{y})$ es el siguiente:

$$\boldsymbol{S_k}\,(\boldsymbol{x},) = \begin{cases} \mathbf{1}, \boldsymbol{si}\ \boldsymbol{\alpha} \leq \frac{\boldsymbol{I_k^V}(\boldsymbol{x},\boldsymbol{y})}{\boldsymbol{B_k^V}(\boldsymbol{x},\boldsymbol{y})} \leq \boldsymbol{\beta} \\ \wedge \left(\boldsymbol{I_k^S}(\boldsymbol{x},\boldsymbol{y}) - \boldsymbol{B_k^S}\,(\boldsymbol{x},\boldsymbol{y})\right) < \boldsymbol{\tau_S} \\ \wedge \left(\boldsymbol{I_k^H}(\boldsymbol{x},\boldsymbol{y}) - \boldsymbol{B_k^H}\,(\boldsymbol{x},\boldsymbol{y})\right) < \boldsymbol{\tau_H} \\ \mathbf{0}, \boldsymbol{resto} \end{cases} \quad (2.15)$$

Donde $(\boldsymbol{x},)$ y $(\boldsymbol{x},)$ son los valores del píxel localizados en $(\boldsymbol{x},)$ en la imagen actual k-ésima y el background correspondiente en el *frame* k-ésimo, respectivamente; y $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, y $\boldsymbol{\tau_H}$ son los umbrales de decisión utilizados.

Otros métodos se centran en la discriminación entre los bordes de las sombras y los bordes o límites de los objetos [21]. Sin embargo, en diversas escenas se hace complicado extraer regiones de movimiento conectadas en su totalidad a partir del mapa de bordes resultante,

[1] Del inglés Hue Saturation Value – matriz, es un espacio de colores que trabaja con los componentes de matiz, saturación y valor.

que en ocasiones es irregular. Por otra parte, escenarios más complejos que contengan diversos objetos de pequeño tamaño presentan desventajas para estos modelos

## 2.4. Detección de objetos (segmentación)

La detección de movimiento, detección de objetos o segmentación, en los sistemas de vigilancia inteligentes consiste en encontrar los objetos deseados en la secuencia de entrada de imágenes. La aproximaciones convencionales para la detección de movimiento se pueden dividir en sustracción del fondo (background subtraction) [15, 27, 67], diferenciación temporal (temporal differencing) [4], y flujo óptico (optical flow) [31].

### 2.4.1. Sustracción del fondo (background subtraction)

Background subtraction o substracción del fondo, es una aproximación ampliamente usada debido a su precisión y a la alta velocidad en la computación a la hora de detectar el foreground. Con el objetivo de extraer los objetos del foreground, los algoritmos de sustracción de fondo detectan la diferencia entre la imagen actual y la imagen de referencia, usualmente llamada como imagen de fondo, background o modelado de fondo [29]. La figura 2.4 muestra un ejemplo de una la técnica de sustracción del fondo.


**Figura 2.4.** Ejemplo de una técnica de sustracción de fondo usada en la detección de movimiento (Tomado de Gómez [1]).

En la actualidad, los algoritmos de substracción de fondo se enfocan en un modelo robusto y actualización del fondo para adaptarse a los cambios de luz entre el día y la noche, la reconfiguración geométrica de la estructura del background, los cambios climáticos, y a los movimientos repetitivos y desordenados [29].

Si se encuentra un modelo Gaussiano en el valor de un píxel, entonces el píxel dado pasa a pertenecer al background y se actualiza el modelo Gaussiano mediante el uso del valor del píxel. En caso contrario, el píxel es clasificado como foreground y el modelo Gaussiano con menor peso es remplazado por uno nuevo, que toma el valor del píxel actual.

Aunque la decisión para establecer el número de modelos de Gaussianas y la inicialización de los modelos Gaussianos es ambigua y el modelado del entorno puede fallar cuando ocurran cambios drásticos en la iluminación, esta aproximación puede modelar y actualizar

robustamente el background de distribuciones multimodales, como son el movimiento de las nubes, y los cambios graduales en el fondo. Además la velocidad de procesamiento no requiere una memoria relativamente grande.

Haritaoglu *et al.*, [27] desarrollaron un método de modelado estadístico, en donde se entrena el background usando la historia del píxel. El modelo del background es representado por el mínimo valor de píxel *(M)*, el máximo valor del píxel *(N)*, y la máxima diferencia de intensidad entre los frames observados durante el periodo de entrenamiento *(D)*. El píxel actual se clasifica como background cuando la diferencia entre los valores del píxel actual y *M*, *N* es menor que *D*; en otro caso, el píxel actual es clasificado como foreground.

En [27] se usó una técnica en tiempo real que utilizó dos métodos para la actualización del background, la actualización basada en el píxel y la basada en el objeto. El método para la actualización basada en píxel reajusta el background periódicamente para adaptarlo a los cambios de iluminación, mientras que el basado en el objeto actualiza el background para adaptarlo a los cambios físicos y el fondo de la escena.

Este modelo de fondo estadístico puede adaptarse a los cambios de iluminación, debido al entrenamiento de la varianza histórica de cada píxel. Adicionalmente, la detección se puede realizar en tiempo real, gracias a la sencilla forma en que se modela y actualiza el fondo.

Horprasert *et al.*, [67] presentaron un nuevo algoritmo para la detección de movimiento en un fondo estático con sombras. Las sombras y las luces tienen una cromaticidad similar, sin embargo el brillo es diferente. Al tener en cuenta la anterior propiedad, el modelo de Horprasert *et al.*, mejoran el rendimiento de la sustracción de fondo frente a los cambios locales y globales en la iluminación. Sin embargo en [67] se asume que el modelo del background proviene de una escena estática, por lo cual esta técnica no es ideal para los cambios en los fondos dinámicos, como la entrada de nuevos objetos; por lo tanto, el problema de la actualización o adaptación del background todavía persiste [29].

### 2.4.2. Diferenciación temporal (temporal differencing)

La diferenciación temporal [4] es un método que extrae las regiones en movimiento, mediante el análisis de una secuencia de imágenes y el estudio de la evolución de los píxeles a lo largo del tiempo. La diferencia temporal se adapta a los entornos dinámicos y la computación para extraer los píxeles en movimiento es simple y rápida.

La aproximaciones basadas en la diferenciación temporal extraen las regiones en movimiento, mediante la diferencia pixel a pixel entre frames consecutivos en una secuencia de video [31]. Este tipo de método es muy adaptable a los cambios dinámicos de las escenas. Sin embargo, generalmente tiene un rendimiento pobre a la hora de extraer todos los pixeles relevantes de los objetos en movimiento, pueden haber huecos en los objetos, y es sensitivo en los valores que el umbral toma para determinar los cambios en las

diferencias de imágenes consecutivas [29]. Las aproximaciones basadas en este método normalmente incorporan técnicas adicionales con el objetivo de hacer frente a estos problemas.

Un esquema típico en la diferenciación temporal, consiste en la comparación del frame actual con el último frame, en donde los pixeles son considerados del foreground si la diferencia supera al valor del umbral:

$$|I_t - I_{t-1}| > \tau \qquad (2.16)$$

El resultado de la segmentación depende únicamente del método para la umbralización usado en la binarización. Para mejorar el esquema típico de la diferenciación temporal, algunas aproximaciones usan una diferenciación de tres-frames. Además para superar algunos defectos de la diferenciación, algunos enfoques usan algoritmos híbridos, los cuales combinan una diferenciación a tres-frames con un modelo adaptativo de sustracción de fondo [28].

El trabajo de Shen [38] es un ejemplo de un algoritmo híbrido, que hace uso de los espacios de colores RGB y HSI[1], la información difusa y la diferenciación temporal, con el objetivo de realizar la segmentación. La segmentación se ejecuta en dos pasos. En el primer paso, consiste en una clasificación difusa, que considera la movilidad del pixel, generada de la combinación de los resultados del umbral de diferenciación en cada canal RGB de la imagen. En el segundo paso, los pixeles falsos generados en el primer paso son eliminados usando el resultado de la segmentación anterior, y la información obtenida de los frames consecutivos. Finalmente, el espacio de color HSI es usado para eliminar las sombras.

Spagnolo *et al.,* [57] también combina la información temporal para obtener un mejor resultado en la segmentación. La aproximación combina el uso de la similitud radiometría entre regiones para comparar píxeles, tanto en el análisis de la imagen temporal como en la sustracción del fondo. Un ejemplo del frame se muestra en la figura 2.5.

### 2.4.3. Flujo óptico (optical flow)

Este método [31, 62] extrae las regiones en movimiento a partir de las características que ofrecen los vectores de movimiento de los objetos a lo largo del tiempo, para detectar cambios en regiones en una secuencia de imágenes. Los algoritmos de este tipo ofrecen como gran ventaja la detección de objetos incluso con cámaras en movimiento. Sin embargo, la mayoría de ellos son muy sensibles al ruido, computacionalmente muy complejos [72] y difícilmente pueden ser aplicados directamente sobre flujo de vídeo en tiempo real a no ser que se disponga de un hardware especializado.

[1] Modelo de color en términos de sus componentes constituyentes, matiz, saturación e intensidad

(a) Frame de ejemplo (b) Segmentación

**Figura 2.5. S**egmentación con algoritmo híbrido (Tomado de Spagnolo [57]).

## 2.5. Seguimiento de objetos

La finalidad del seguimiento de objetos es generar la trayectoria de un objeto a lo largo de un período de tiempo, mediante la localización de su posición en cada uno de los frames de un vídeo. Las tareas de detectar un objeto y establecer una correspondencia (localizar dicho objeto en otra imagen) a lo largo de una serie de frames de vídeo, puede realizarse de forma conjunta o separada.

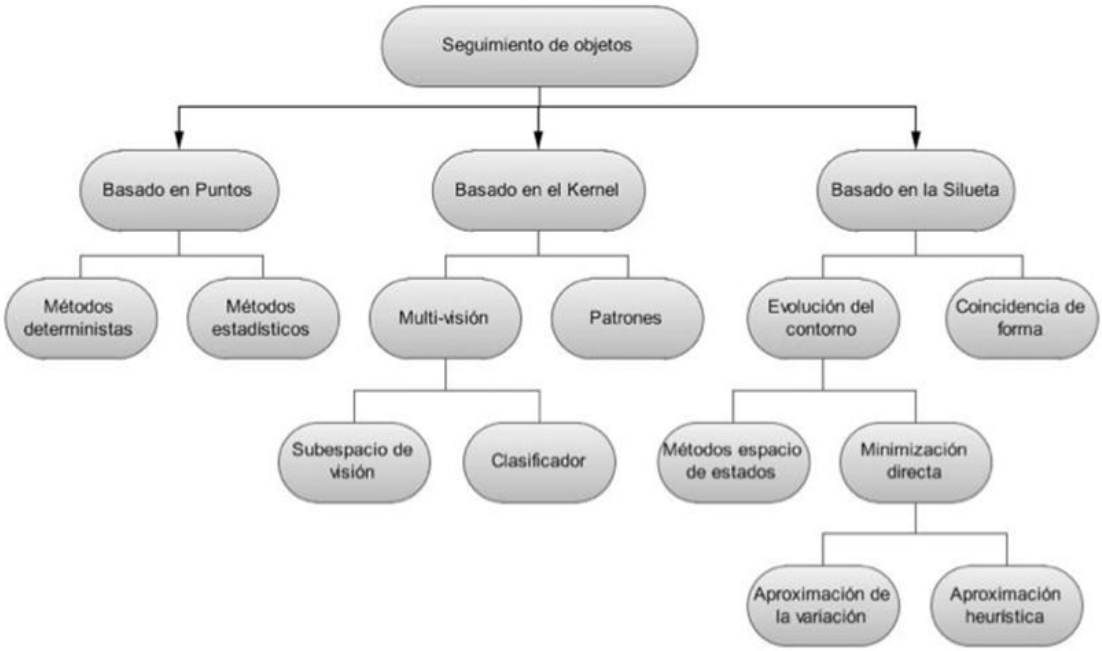



**Figura 2.6.** Clasificación de los métodos de seguimiento (Adaptado de Yilmaz [6]).

En el primer caso, las regiones que componen el objeto en cada frame se obtienen a través de un algoritmo de detección de objetos, y la función del seguimiento de objetos es establecer la relación entre los objetos en los diferentes frames. En el segundo caso, las regiones que constituyen el objeto y su correspondencia, se estiman de forma conjunta por medio de actualizaciones iterativas de la localización del objeto y la información de la región obtenida del análisis de los frames previos.

Yilmaz [6] clasificó el seguimiento de objetos en tres categorías diferentes, la figura 2.6 presenta la propuesta introducida por Yilmaz.

### 2.5.1. Seguimiento de objetos basado en puntos

Los objetos detectados en frames consecutivos son representados mediante puntos, y la asociación de los puntos se basa en el estado previo del objeto que puede incluir su posición y su movimiento. Esta aproximación requiere de un mecanismo externo que detecte los objetos en cada frame. Un ejemplo de la correspondencia entre objetos se muestra en la figura 2.7(a).

Los seguidores basados en puntos son aptos para rastrear objetos muy pequeños, que puedan ser representados por un único punto, para rastrear objetos más grandes múltiples puntos son necesarios. Para rastrear objetos grandes, la agrupación automática de puntos pertenecientes a un mismo objeto es un problema importante, esto se debe a la necesidad de distinguir entre múltiples objetos, entre objetos y el fondo. La agrupación de puntos basada en movimiento o en la segmentación usualmente supone que los puntos que se están siguiendo pertenecen a cuerpos rígidos con el fin de simplificar el problema de segmentación [6].

La correspondencia en los puntos es un problema complicado, especialmente en presencia de oclusiones, falsos positivos, entrada y salida de objetos. En general, los métodos para hallar la correspondencia en los puntos pueden dividirse en dos grandes categorías, deterministas y estadísticos. Los métodos deterministas usan el movimiento heurístico cualitativo para limitar el problema de la correspondencia. Por otro lado, los métodos probabilísticos tienen en cuenta el objeto en cuestión y la incertidumbre para establecer la correspondencia [6].

Un punto importante en los seguidores basados en puntos es el manejo de la desaparición de objetos y el ruido. Para hacer frente a estos problemas, los métodos deterministas a menudo utilizan una combinación de restricciones en el movimiento y en la proximidad. Los métodos estadísticos tratan con el ruido tomando en cuenta la incertidumbre en el modelo. Se supone, que generalmente, la incertidumbre en el ruido tiene una distribución normal, sin embargo, la suposición de que las mediciones estén normalmente distribuidas en torno a su posición no siempre es cierta, es más, en muchos casos, los parámetros del ruido ni siquiera se conocen. En el caso de que se asuma correctamente la distribución y el ruido, el filtro de Kalman y MHT dan resultados óptimos. Otra aproximación posible es

tratar con el ruido y los objetos no observables es forzar una restricción que defina la estructura 3D de los objetos [6].

### 2.5.2. Seguimiento de objetos basado en el kernel

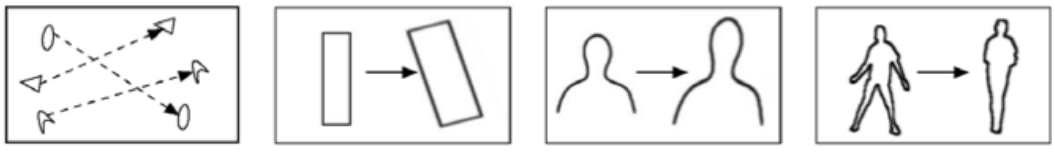
El kernel (núcleo) del objeto se refiere a la forma del objeto y su apariencia. Por ejemplo, el kernel puede ser un patrón rectangular o una forma elíptica con un histograma asociado. El seguimiento basado en el kernel generalmente se logra calculando de un frame a otro el movimiento del objeto, que es representado por una región primitiva del objeto (figura 2.6 (b)).

### 2.5.3. Seguimiento de objetos basado en la silueta

El seguimiento se realiza estimando la región del objeto en cada frame. Los métodos basados en la silueta utilizan información codificada dentro de la región del objeto. Esta información puede encontrarse en forma de modelos de densidad de aparición o forma que se utilizan para generar mapas de bordes. Dados unos modelos de un objeto, se realiza un seguimiento de las siluetas mediante la búsqueda de coincidencias o bien siguiendo la evolución del contorno (figura 2.7 (c) y (d)). Ambos métodos pueden considerarse en términos generales como una segmentación del objeto aplicada en el dominio temporal usando la información generada en los frames previos.

Los objetos que tienen formas complejas, como las manos, la cabeza y los hombros, no pueden describirse bien con simples formas geométricas. Los métodos basados en la silueta proporcionan una descripción más exacta para la forma de estos objetos. El objetivo de estos métodos es encontrar la región de los objetos en cada frame, por medio de un modelo del objeto que es generado usando los frames anteriores. Este modelo puede ser en la forma de histogramas, bordes de objetos o contorno.

Los seguidores basados en la silueta se dividen en dos categorías, evolución del contorno y coincidencia de la forma. Las aproximaciones basadas en la coincidencia de la forma buscan la silueta del objeto en el frame actual. Los métodos de rastreo basados en el contorno evolucionan de un contorno inicial a su nueva posición en el frame actual, bien sea usando modelos de espacio de estados o la minimización de alguna energía funcional [6].



**(a)** **(b)** **(c)** **(d)**

**Figura 2.7.** Diferentes aproximaciones del seguimiento de objetos. (a) Correspondencia multipunto, (b) Transformación paramétrica de un patrón rectangular, (c, d) Dos ejemplos de evolución del contorno (Tomado de Yilmaz [6]).

El seguimiento basado en la silueta se usa cuando se necesita hacer el seguimiento de toda la región de un objeto, y la ventaja más importante de estos tipos de seguidores son su flexibilidad para tratar con una gran variedad de objetos con diferentes formas. La forma más común para representar la silueta, es en la forma de una función de indicador binario, la cual marca con unos las regiones del objeto y con ceros las regiones que no pertenezcan al objeto [6].

## 2.6. Clasificación de objetos

Las regiones en movimiento detectadas en un video pueden corresponder a diferentes objetos del mundo real, tales como: peatones, vehículos, etc. Es muy importante reconocer el tipo de objeto detectado, con el objetivo de realizar un seguimiento fiable y poder analizar sus actividades correctamente [73]. Actualmente, hay dos grandes enfoques para clasificar objetos en movimiento, clasificación basada en formas y clasificación basada en movimiento [41]. Los métodos basados en formas hacen uso de la información del espacio 2D de los objetos, mientras que los métodos basados en movimiento usan características temporales del seguimiento de los objetos.

### 2.6.1. Clasificación basada en formas

Una de las posibles formas que existen para clasificar un objeto, es a partir de su silueta [37, 19, 65 y 70]. En esta categoría se establecen mecanismos de comparación entre los patrones de formas definidos previamente para cada una de las clases y la silueta actual del objeto. Estos mecanismos proporcionan un valor numérico que indica el grado de pertenencia de un objeto a una clase. Finalmente el sistema optará por aquella clase cuyo valor de pertenencia sea mayor.

Las características comunes usadas en los esquemas de clasificación basados en forma son, los límites del rectángulo, el área, la silueta y el gradiente de la región de los objetos detectados.

La aproximación presentada en [4] hace uso de la longitud del contorno de la silueta y de la información del área para clasificar los objetos detectados en tres grupos: humanos, vehículos y otros. El método presume que los humanos son, en general, más pequeños que los vehículos y con formas más complejas. El grado de dispersión es usado como una métrica para la clasificación y está definida en términos del área y la longitud del contorno (perímetro) del objeto de la siguiente manera:

$$Dispersión = \frac{Perimetro^2}{Area} \qquad (2.17)$$

La clasificación es realizada en cada frame y los resultados del seguimiento son usados para mejorar la consistencia de la clasificación temporal.

Es sistema VSAM, desarrollado por Collins et *al.*, [60], usa vistas dependientes de las características visuales de los objetos detectados, para entrenar una red neuronal capaz de reconocer cuatro clases: humanos, grupo de humanos, vehículos y aglomeración de formas no identificadas. Las entradas a la red neuronal son el grado de dispersión, el área y la relación del aspecto, de cada objeto. Al igual que el método previo, la clasificación es realizada en cada frame y los resultados son guardados en un histograma para mejorar la consistencia temporal de la clasificación.

Saptharishi *et al.,* [49] propusieron un esquema de clasificación, que usa hace uso de una red neuronal linear con aprendizaje diferencial, para reconocer dos clases, vehículos y personas. Papageorgiou *et al.,* [14] presentaron un método que uso de Support Vector Machine entrenado con la transformada wavelet, en imágenes de video, para una base de datos de peatones, capaz de reconocer objetos en movimiento que concuerden con humanos.

### 2.6.2. Clasificación basada en el movimiento

Una alternativa al método anterior es la clasificación de objetos en función de los movimientos que éstos realicen [59, 3]. Para distinguir entre personas y vehículos, la mayoría de autores parte de la premisa de que las personas varían con cierta facilidad su forma (objetos no rígidos) y realizan un movimiento periódico en su desplazamiento. En cambio, con los vehículos sucede todo lo contrario, no cambian de forma con frecuencia (a menos que realizan algún giro) y no tiene un movimiento cíclico.

El método propuesto en [59] se basa la auto-similaridad temporal del objeto en movimiento. Mientras que un objeto que realiza un movimiento periódico evoluciona, su medida de auto-similaridad también muestra un movimiento periódico. El método hace uso de esta particularidad para categorizar los objetos en movimiento.

El análisis de flujo óptico también es útil para distinguir los objetos rígidos de los no rígidos. Lipton [3] propuso un método que hace uso del análisis local del flujo óptico de las regiones detectadas del objeto. Se espera que los objetos que no son rígidos, como los humanos, presenten un alto flujo residual alto, mientras que los objetos rígidos como los vehículos presentan un bajo flujo residual; de igual manera, el flujo residual generado por los humanos es periódico. Con el uso de estas propiedades, es posible distinguir humanos de otros objetos como los vehículos.

## 2.7. Detección de objetos estáticos

Una tarea vital en los sistemas de videovigilancia orientados a la detención de objetos abandonados o robados, es la detención de objetos estáticos. La detección de objetos estáticos busca determinar cuáles objetos del foreground se han mantenido estáticos durante *t* cantidad de tiempo. La mayoría de aproximaciones de detección de objetos estáticos encontradas en la literatura se basan en el seguimiento de objetos, los cuales usan los mapas

del foreground calculados en dicha etapa. El seguimiento consiste en establecer una correspondencia entre los blobs[1] en los frames consecutivos. Mediante el acertado cálculo de la posición del objeto en secuencias de imágenes, se obtienen parámetros de movimiento como la velocidad y la trayectoria, es posible, por ejemplo, determinar cuáles regiones del foreground no se han movido mediante el análisis de su velocidad.

### 2.7.1. Clasificación de detección objetos estáticos

La clasificación de objetos estáticos se puede dividir en dos categorías: aproximaciones que usan un modelo de fondo y en aproximaciones que usan más de un modelo de fondo (figura 2.8)

#### 2.7.1.1. Aproximaciones que usan un modelo de fondo

Las aproximaciones basadas en un modelo de fondo y dependiendo de uso que hagan con las máscaras foreground se pueden clasificar en:

- Análisis imagen a imagen: Esta categoría describe los métodos que emplean modelos de segmentación frente-fondo bastante comunes, seguidos de otro tipo de análisis. En función de dicho tipo de análisis, pueden aparecer diferentes categorías, que son las siguientes:

    - Basados en el uso de técnicas sencillas de segmentación frente-fondo, un postprocesado de la máscara de foreground, seguido a su vez de otra etapa de análisis (por ejemplo, seguimiento de objetos). Esto se conoce como aproximaciones clásicas [8, 13, 43, 66, 34].

    - Basados en la acumulación de máscaras de foreground. Dicha acumulación se realiza frame a frame y con ella se puede modelar una máscara final de foreground, de donde se extraen las regiones estáticas [26, 11, 22].

    - Basados en algunas propiedades del modelo de fondo utilizado, como las transiciones entre los diferentes estados de un modelo de mezcla de gaussianas, u observando el valor de parámetros como, el peso de las gaussianas [46, 44, 23 y 9]

- Análisis de máscaras de foreground muestreadas**:** Estas aproximaciones intentan detectar regiones estáticas analizando la secuencia de vídeo a diferentes velocidades, aprovechándose de las ventajas espacio-temporales que ello conlleva [40, 54].

[1] Segmento o región de una imagen definida por el color blanco en una imagen binaria.

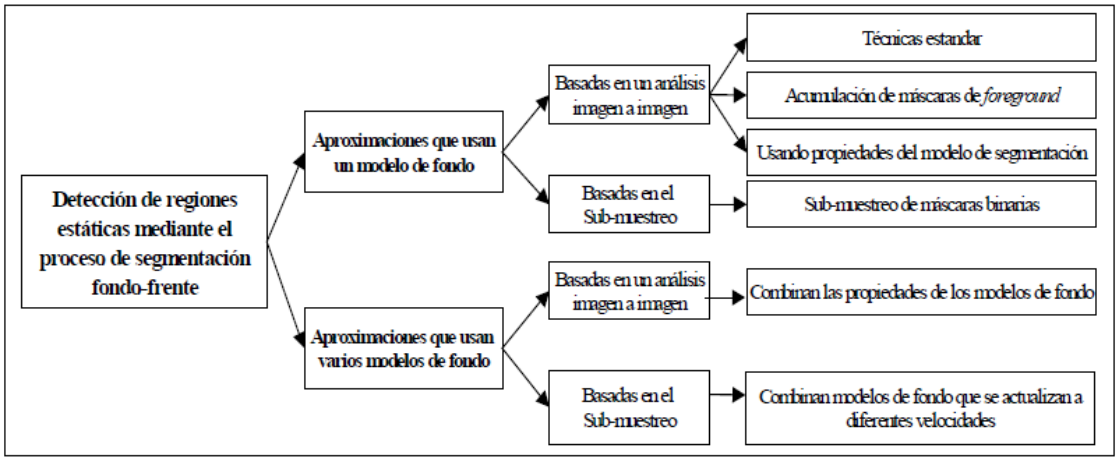



**Figura 2.8.** Clasificación de métodos basados en detectar regiones estáticas (Tomando de Bayona *et al.*, [7]).

#### 2.7.1.2. Aproximaciones que usan más de un modelo de fondo

Existen aproximaciones que combinan más de un modelo de fondo para cada píxel. Este tipo de aproximaciones han sido menos utilizadas para tratar de detectar regiones estáticas. Sin embargo, en función de la tasa binaria de procesamiento del vídeo, o del número de modelos de fondo utilizados para detectar regiones estáticas, se puede realizar la siguiente clasificación:

- Aproximaciones basadas en el análisis imagen a imagen**:** En esta categoría tenemos métodos que combinan las propiedades de los diferentes modelos de fondo que utilizan. [69].

- Aproximaciones basadas en el sub-muestreo**:** Estas aproximaciones detectan regiones estacionarias analizando la secuencia de vídeo a través de los diferentes modelos de fondo debido a que cada modelo de fondo se muestrea con una tasa binaria diferente [55].

## 2.8. Detección de objetos abandonados y robados

En esta sección, se estudia los diferentes enfoques encontrados en la literatura para distinguir entre objetos abandonados y objetos robados.
Las principales aproximaciones se basan en la detección de objetos estáticos. Algunos enfoques en la literatura simplifican el problema al suponer que sólo se permite inserciones de objetos en la escena [25, 36, 45]. Estas técnicas pueden funcionar en entornos controlados, como la detección de equipajes abandonados en aeropuertos, lo cual no toma en cuenta los posibles objetos del primer plano generados por la técnica de sustracción de fondo, o fantasmas producidos por el movimiento de las partes estáticas del escenario. Por lo tanto estos enfoques no están orientados para entornos complejos.

Existen varias técnicas propuestas para la detección de objetos abandonados y robados, entre las cuales están aquellas que clasifican de acorde con las características basadas en el color, el contorno o una combinación de las anteriores aproximaciones.

### 2.8.1. Aproximaciones basadas en el contorno

Las aproximaciones basadas en el contorno estudian la energía de los límites del objeto estático. Esta energía se supone que es alta cuando un objeto ha sido añadido a la escena y baja cuando un objeto ha sido removido. Por ejemplo, en [32], se analizó el cambio de la energía en el contorno. Para los objetos abandonados, se espera que la energía promedio del contorno sea más alta. Por el contrario, se espera que esta energía sea menor cuando el objeto haya sido retirado de la escena (objetos robados). Enfoques similares se describen en [53, 17]. Estos últimos proponen el uso del detector de bordes de *Canny*[1] dentro del rectángulo envolvente del objeto estático, tanto en el fondo como en frame actual. Si la presencia del contorno es más grande en el frame actual, entonces el objeto es clasificado como abandonado, de lo contrario, el objeto es clasificado como robado.

Un ejemplo de aproximaciones basadas en el contorno se muestra en la figura 2.9 donde en el frame actual, los contornos grandes indican que un objeto ha sido abandonado y los contornos débiles indican que el fondo ha sido descubierto debido a un objeto robado. Sin embargo estas aseveraciones solo son verdaderas para fondos simples.


**Figura 2.9.** Ejemplos de aproximaciones basadas en el contorno para objetos abandonados y robados (Adaptado de San Miguel *et al.*, [34]).

### 2.8.2. Aproximaciones basadas en el color

Las técnicas que utilizan el color para la detección de objetos abandonados o robados, analizan los colores de los objetos estáticos y lo comparan con el color del rectángulo que envuelve el objeto, asumiendo que, si un objeto se ha colocado en escena, las características de los colores del objeto difieren significativamente con las características del espacio que

[1] Es un operador desarrollado por John F. Canny en 1986 que utiliza un algoritmo de múltiples etapas para detectar una amplia gama de bordes en imágenes.

lo rodea. Por otro lado, cuando un objeto es extraído de la imagen, se espera que la fracción del fondo que queda al descubierto tenga propiedades de color similar al área que lo rodea.

En [63], se compara las distancias Bhattacharyya[1] de los histogramas del color del fondo y la imagen actual, si la diferencia supera un umbral determinado entonces se clasifica el objeto como robado o abandonado.

### 2.8.3. Aproximaciones basadas en el contorno y el color

Las aproximaciones hibridas combinan la información del contorno y el color. En [34], se usan dos detectores (de color y de contorno) para construir un modelo probabilístico de cada algoritmo en cada clase (abandonado y robado). La discriminación se realiza mediante el cálculo de la probabilidad media de cada clase, el objeto pertenece a la clase que obtenga la probabilidad media más alta.

[1] En estadística, la distancia Bhattacharyya mide la similitud de dos distribuciones de probabilidad discretas o continuas.

# *CAPÍTULO 3*

# *CONSIDERACIONES METODOLÓGICAS*

# 3. Consideraciones Metodológicas

Este capítulo presenta el desarrollo metodológico del sistema, en donde se presenta las diferentes tareas realizadas en cada fase de la metodología RUP[1], la cual fue usada para el desarrollo del sistema. Igualmente también se presenta algunas consideraciones metodologías para la selección de la arquitectura del sistema, donde se describe que necesidades debe atacar un sistema de vigilancia inteligente, con el fin de entender porque la arquitectura del sistema se diseñó como tal. Por último se presenta el flujo metodológico para la selección de los algoritmos de análisis.

## 3.1. Desarrollo metodológico para la construcción del sistema

La construcción del sistema se realizó usando la metodología RUP, la cual define un proceso de desarrollo iterativo y dicta que el ciclo de vida del desarrollo de software se descomponga en las fases de: inicio, elaboración, construcción y transición (figura 3.1).

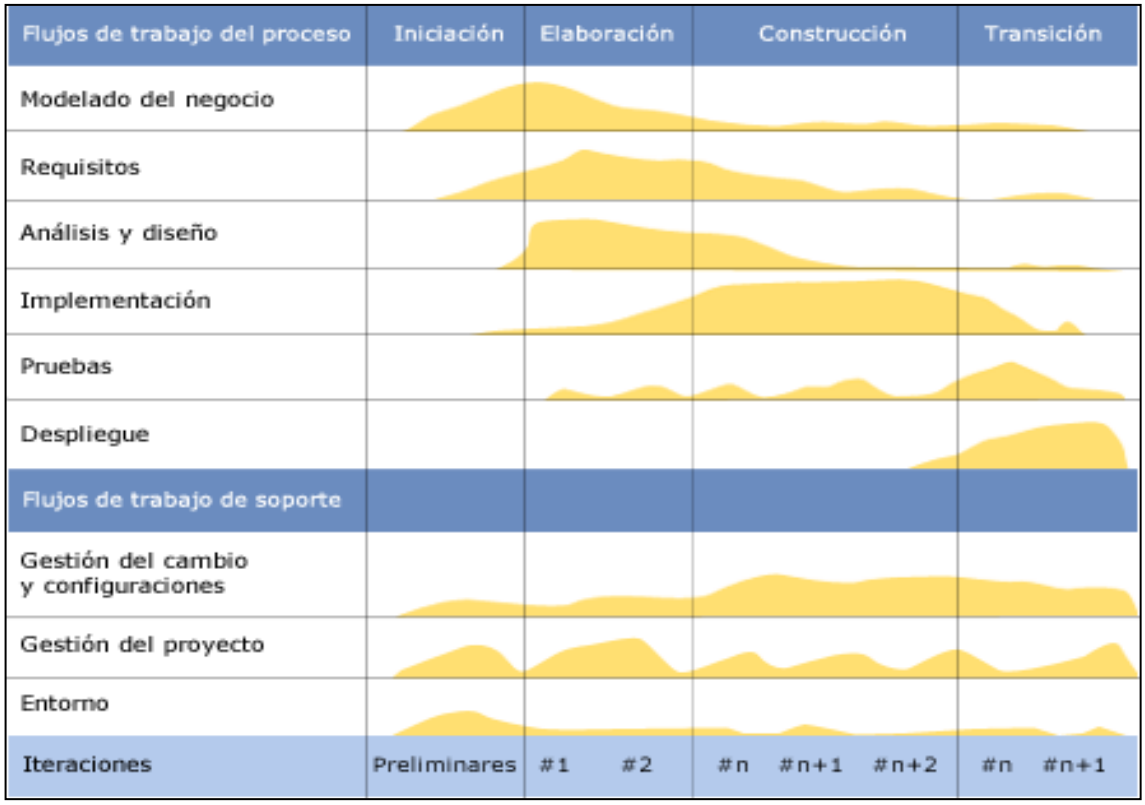



**Figura 3.1**. Iteraciones en las fases de la metodología RUP (Tomado de http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational).

[1] Proceso Unificado de Rational, del inglés Rational Unified Process, es un proceso de desarrollo de software que constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

### 3.1.1. Fases de inicio

Para la ejecución de esta etapa se generó el documento de visión (Anexo D), en donde se realizó un estudio inicial sobre la situación actual de los sistemas de videovigilancia, su problemática, desafíos e importancia; con la finalidad de:

- Establecer los objetivos y la justificación del proyecto.
- Determinar el alcance y las limitaciones del sistema.
- Identificar los riesgos del proyecto.
- Determinar una visión general de los requerimientos, mediante un diagrama inicial de casos de uso.
- Vislumbrar arquitecturas posibles.
- Plantear un cronograma tentativo.

### 3.1.2. Fases de elaboración

Durante esta fase se construyó el Documento de Especificación de Requisitos de Software (ERS), el cual se encuentra en el Anexo C, se construyó y valido la mayor parte de la arquitectura del sistema, se determinó como tratar con los riesgos identificados en la fase anterior y se elaboró el cronograma de acuerdo a los requerimientos planteados en el ERS.

Debido a que el sistema propuesto se planteó para ser escalable y multipropósito, es decir, no solo para tratar con la detección de objetos robados y abandonados, se hizo un estudio detallado de los sistemas de videovigilancia y las diferentes arquitecturas que se han construido; con el fin de determinar y elaborar la arquitectura del sistema. En la sección 3.2 se encuentran las consideraciones metodológicas para la construcción de la arquitectura y en el capítulo cuatro se detalla la arquitectura del sistema.

Para el sistema propuesto, se realizó una descripción detallada de los requerimientos funcionales y no funcionales del sistema. Los requerimientos funcionales se ampliaron mediante el diagrama y las especificaciones de casos de uso. La definición de los modelos del sistema se realizó mediante el diagrama entidad relación y los diagramas del Lenguaje de Modelado Unificado (UML) segunda versión, en los cuales se definen aspectos de las estructura, comportamiento e interacción del sistema.

Por último en esta fase se especificaron las tecnologías a usar para implementar el sistema, como: el lenguaje de programación, las librerías, ambientes de desarrollo, motor gestor de base de datos. También se desarrolló el documento de convenciones de código (Anexo E), el cual tiene los lineamientos en cuanto a implementación que debe seguir el equipo de desarrollo.

En esta fase se determinó que el sistema haría uso de las siguientes tecnologías para la construcción del sistema:

- La implementación de todos los algoritmos de análisis se hizo primero en Matlab®, debido a que este entorno de desarrollo permite agilizar drásticamente la implementación y evaluación de los algoritmos.

- El lenguaje de programación usado para la construcción del sistema fue C++, debido a que las aplicaciones de visión artificial tienen una alta demanda de recursos y se requiere de un lenguaje compilado para poder procesar la información en tiempo real.

- La librería multiplataforma *Qt* fue usada para la construcción de la interfaz gráfica de usuario. Además este framework permite abstraer la complejidad de C++, y proporciona una serie de API[1]s de propósito específico.

- Se usó el conjunto de librerías Boost, el cual extiende las capacidades del lenguaje de programación C++.

- TBB es una libreria basada en plantillas para C++, que se usó con el fin de facilitar la escritura de funciones orientadas al paralelismo de los procesadores con arquitectura multinúcleo.

- La librería OpenCV se usó para implementar los algoritmos de análisis en el sistema final, debido a que esta librería especializada en visión artificial y por lo tanto su rendimiento ha sido comprobado.

- Se escogió el sistema de gestión de base de datos relacional *MySQL* para gestionar la base de datos encargada de guardar la información generada por la aplicación y la configuración del sistema.

### 3.1.3. Fases de construcción

En esta fase se terminó el desarrollo de la arquitectura del sistema y se desarrollaron los componentes y módulos siguiendo los lineamientos planteados en la fase de elaboración.

[1] Interfaz de programación de aplicaciones, del inglés Application Programming Interface, es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Igualmente y como RUP define un proceso de desarrollo iterativo, se revisaron requerimientos funcionales y ajustaron algunas especificaciones de casos de uso, así como el diagrama entidad relación y algunos diagramas UML.

En esta etapa también se realizó un estudio cualitativo y cuantitativo de diferentes algoritmos de análisis, en donde se seleccionaron los algoritmos que se implementaron en el sistema final. En la sección 3.3 se detallan consideraciones metodológicas para la selección de los algoritmos de análisis.

### 3.1.4. Fases de transición

Esta etapa se destinó para la realizar las pruebas de software, con el propósito de detectar los errores que se hayan cometido en la implementación de las funcionalidades del sistema y de garantizar la integridad y calidad del producto.

Se realizaron tres tipos de pruebas: pruebas unitarias, pruebas de integridad y pruebas del sistema. Las pruebas unitarias y de integridad se desarrollaron e implementaron a medida que las clases o módulos que se iban desarrollando, las pruebas del sistema cada que había una liberación con la finalidad de determinar que los requerimientos planteados en el ERS eran correctamente desarrollados.

## 3.2. Consideraciones metodológicas para la construcción de la arquitectura

Normalmente, los sistemas expertos que interactúan con el mundo real se enfrentan a problemas de gran magnitud, cuyas soluciones se obtienen como consecuencia del trabajo llevado a cabo en diversas etapas o fases. La mayoría de los sistemas de vigilancia inteligentes no son escalables o la taza de frames por segundo es muy baja debido al uso de algoritmos muy costosos. Además usualmente los sistemas se diseñan para un contexto específico y no pueden adaptarse fácilmente a otro entorno, es de común acuerdo incorporar la información contextual en los sistemas, debido a que permiten mejorar los resultados de análisis en los procesos de análisis [58], tradicionalmente esta información se ha incorporado en los procesos a través de un procesos de parametrización manual.

La mayoría de las arquitecturas de los sistemas inteligentes se caracterizan por estar divididas jerárquicamente en diferentes capas, donde cada una de ellas cumple una función bien definida. Dicha división dota al sistema de una mayor flexibilidad y modularidad, facilitando las tareas de mantenimiento e inclusión de nuevas funcionalidades.

Los sistemas de vigilancia inteligentes, son un claro ejemplo de sistemas complejos que pueden beneficiarse de un diseño basado en capas. En este tipo de sistemas las capas inferiores se centran en el procesamiento de señales y la generación de información espacio-temporal relativa a cada uno de los elementos que interactúa en el entorno, para que las capas intermedias puedan interpretar los eventos y comportamientos que suceden

alrededor. Finalmente, las capas situadas en los niveles superiores suelen estar enfocadas a la monitorización de los resultados obtenidos en las capas inferiores y a la ayuda para la toma de decisiones en situaciones críticas.

Por otra parte, la complejidad de los sistemas de vigilancia inteligentes actuales lleva a buscar la reutilización de los diseños y desarrollos existentes a la hora de diseñar y desarrollar sistemas que vigilen nuevos entornos.

Se han propuesto múltiples arquitecturas y son de interés aquellas que pertenecen a la tercera generación de los sistemas de vigilancia, en donde los sistemas se conciben para tratar con un número elevado de cámaras, recursos geográficamente distribuidos, múltiples puntos de monitorización y tratan de imitar la jerárquica y distribuida naturaleza humana en cuanto a los procesos llevados a cabo en la vigilancia. Desde el punto de vista del procesamiento digital de imágenes, estos sistemas se basan en la distribución de procesos sobre una red.

Los entornos con múltiples sensores distribuidos presentan oportunidades y desafíos interesantes para los sistemas de vigilancia. La comunicación con diferentes partes del sistema juega un rol importante, con desafíos particulares, ya que se enfrentan a limitaciones en la banda ancha o a la naturaleza asimétrica de la comunicación. La comunicación entre los diferente módulos también tienen que tener en cuenta aspectos de seguridad, dado que para algunos sistemas de vigilancia la información debe trasmitirse a través de redes abiertas y es crítico mantener la privacidad y autenticación.

Existe una tendencia de que los sistemas de videovigilancia deben incluir capacidades de aprendizaje automático, con la finalidad de proporcionar la capacidad de que se caractericen modelos de escenas que se puedan catalogar como eventos potencialmente peligrosos.

### 3.2.1. Etapas de un sistema de videovigilancia

Generalmente, los sistemas de vigilancia de tercera generación disponen de una arquitectura multicapa. Cada capa de la arquitectura desempeña una función bien definida y genera una serie de resultados que sirven como flujo de entrada para las otras capas. En [52, 20, 72] se realizan tres propuestas interesantes sobre las etapas o fases de las que debería constar un sistema de vigilancia inteligente. La figura 3.2 unifica en un único esquema las fases propuestas por diferentes autores.

La primera fase, o fase de modelización [72], consiste en definir los elementos o actores que pueden participar en el entorno, así como su propiedades y las principales relaciones que existen entre éstos y el entorno. Las máquinas, al igual que las personas, necesitan adquirir conocimiento a partir de experiencias previas para poder interpretar la información procedente de los sentidos (red de sensores en el caso de las máquinas).

El modelado de entornos es una manera efectiva de interpretar y predecir los comportamientos en escenas estáticas. Sin embargo, en escenas donde los actores o las condiciones ambientales pueden variar con facilidad, el modelo diseñado inicialmente podría dejar de ofrecer buenos resultados ante tales cambios, por este motivo, las herramientas de adquisición de conocimiento que se utilizan en esta fase para modelar el entorno, deberían ser complementadas con algoritmos de aprendizaje cuando las escenas son complejas y dinámicas, de esta forma, el modelo inicial varía y se adapta a los cambios que se producen en el tiempo.

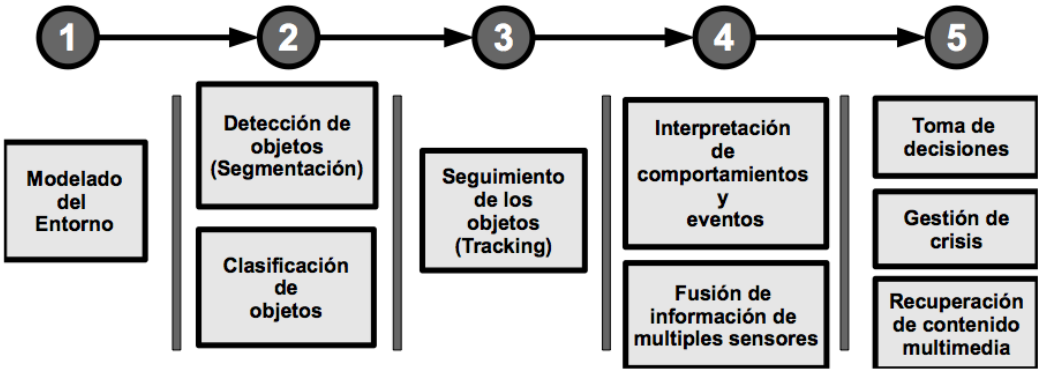



**Figura 3.2.** Etapas de un sistema de video vigilancia (Tomado de Albusac [30]).

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

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

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

A partir de la información obtenida en las etapas anteriores, el sistema de vigilancia debe intentar reconocer los comportamientos y eventos que suceden en el entorno observado. El comportamiento de un objeto viene dado por una simple acción o evento que se produce en un instante concreto, o bien por una secuencia de acciones simples a lo largo del tiempo. Si el comportamiento es complejo se representa mediante una secuencia de acciones, las cuales siguen un orden determinado y cumplen una serie de restricciones temporales. Por tanto, para que un sistema artificial pueda reconocer comportamientos complejos, es necesario que éste identifique las acciones simples cuando suceden y estudiar las relaciones temporales que existen entre ellas. La principal problemática de esta etapa es la fuerte dependencia que existe con las etapas anteriores. Una clasificación errónea de un objeto o una reproducción equivocada de su trayectoria derivaría, casi con toda seguridad, en una interpretación inapropiada de su conducta.

Opcionalmente, la capa de razonamiento e interpretación de comportamientos puede incluir la posibilidad de fusionar la información que proviene de múltiples sensores. Esta fusión puede proporcionar grandes beneficios como por ejemplo la eliminación de ruidos y distorsiones y el tratamiento del problema de la oclusión.

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

## 3.3. Consideraciones metodológicas para la selección de los algoritmos de análisis

Actualmente existen en la literatura muchos algoritmos de análisis en las áreas de preprocesado, modelado de fondo, eliminación de sombras, detección de objetos, seguimiento de objetos, detección de objetos estáticos, clasificación de objetos y detección

de objetos abandonados y robados. Sin embargo no todos los algoritmos propuestos son aptos para los sistemas orientados a la vigilancia.

Los sistemas videovigilancia requieren que los algoritmos de análisis se ejecuten en tiempo real, para lograr esto es necesario que los algoritmos tengan un bajo costo computacional, tanto en memoria como el procesamiento, lo cual limita drásticamente la complejidad que pueden tener los algoritmos de vigilancia.

Además de lo anteriormente descrito, los algoritmos no solo se clasifican por complejidad algorítmica, consumo de memoria o costo computacional, sino también en el entorno para el que están orientados. Por ejemplo, el algoritmo de sustracción de fondo *frame difference* (descrito en la sección 2.4.2) tiene un bajo coste computacional y es bueno para detectar objetos del foreground en fondos simples, sin embargo es ineficiente para trabajar con fondos multimodales, a diferencia de la *MoG* (descrito en la sección 2.2.2.1) que puede modelar fondo multimodales, pero requiere mayor potencia computacional y puede tener menor rendimiento en fondos simples.

Otro punto a tener en cuenta a la hora de elegir los diferentes algoritmos de análisis, es que el rendimiento de los algoritmos de las etapas superiores de análisis tienen una fuerte dependencia con las etapas inferiores, por lo tanto, cuando hay que evaluar el rendimiento de uno de estos algoritmos se debe tener en cuenta las anteriores fases.

Según los anteriores lineamientos y teniendo en cuenta el alcance y las limitaciones propias del proyecto se deben seleccionar los algoritmos de análisis para posteriormente implementarse en el sistema.

La selección de los algoritmos se realizó en las siguientes etapas:

Etapa 1: Discriminación de los algoritmos según el estado del arte. En esta etapa se descartó una serie de algoritmos, los cuales la literatura cataloga como de alto coste computacional y por ende no aptos para los sistemas de videovigilancia.

Etapa 2: Categorización de los algoritmos. Los algoritmos seleccionados en la anterior fase se categorizan según su posible aplicación en los diferentes entornos, la cual se ve limitada por el alcance del proyecto, especificado en el documento de visión.

Epata 3: Selección de los algoritmos más simples. Entre los algoritmos seleccionados en la etapa anterior, se descartaron algoritmos excesivamente complejos, con la finalidad de que la implementación de los mismos en el sistema final se realice de forma más ágil.

Etapa 4: Comparación cuantitativa de los algoritmos. Los algoritmos elegidos en las anteriores etapas se evalúan mediante una implementación inicial realizada en Matlab® y mediante el uso de datasets que permitan medir el comportamiento en cada etapa de análisis.

## 3.4. Datasets utilizados

Actualmente existen muchos datasets públicos orientados a la detección de objetos abandonados y robados en video. Adicionalmente, estos son ampliamente usados en el campo de la videovigilancia para medir el rendimiento de los diferentes módulos de análisis, como la detección o seguimiento de objetos.

Además de utilizar los datasets para evaluar el sistema objetivo, estos se usaron también para entrenar algunos algoritmos de análisis, puesto que se pueden usar para determinar los diferentes parámetros de los diferentes algoritmos.

Los datasets usados se listan a continuación, la figura 3.3 muestra frames de ejemplo de los diferentes datasets.

### 3.4.1. PETS2006[1]

Las secuencias de video de este dataset contienen diferentes ejemplos de abandono de equipaje en escenarios sencillos, cuya dificulta va aumentando según la distancia a la que se abandona el equipaje. El dataset se compone de 28 videos con seis eventos de abandono de equipaje en una estación de metros, las grabaciones se hicieron desde cuatro cámaras en ángulos diferentes. Cada video tiene una duración de uno a dos minutos y fueron grabadas con una resolución PAL estándar de 768 x 576 pixeles a 25 frames por segundo.

### 3.4.2. AVSS2007[2] (iLids dataset)

Este dataset se compone de tres secuencias de video, en donde cada video tiene un evento de abandono de video. El dataset tiene tres niveles de complejidad: fácil, medio y difícil; que se definen en términos de la densidad de población y la distancia de abandono del equipaje. Cada video tiene una duración de un poco más de tres minutos y fueron grabadas con una resolución PAL estándar de 768 x 576 pixeles a 25 frames por segundo.

### 3.4.3. CVPR2012[3]

Dataset orientado a la detección de movimiento, formado por 6 categorías diferentes, donde cada categoría contiene de cuatro a seis secuencias de video. Las secuencias de video están determinadas por las sucesiones de imágenes, las cuales varían en resolución.

[1] URL: http://www.cvg.rdg.ac.uk/PETS2006/data.html
[2] URL: http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html
[3] URL: http://wordpress-jodoin.dmi.usherb.ca/dataset

PETS2006 AVSS 2007 (i- LIDS) CVPR2012

**Figura 3.3.** Frames de ejemplo de los datasets seleccionados (Fuente: El autor).

# *CAPÍTULO 4*

# *CONSTRUCCIÓN DEL SISTEMA*

# 4. Construcción del sistema

En este capítulo se presenta un análisis del sistema objetivo, en donde se describen los usuarios del sistema y como éstos interactúan con el sistema. Después se presenta una descripción general del sistema, tanto a nivel físico como lógico, en donde se detallan los componentes de cada los mismos. Por último se hace una descripción del proceso implementado para la detección de objetos abandonados y robados.

## 4.1. Análisis del sistema

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

Debido a la alta complejidad que presenta el análisis de comportamiento y la interpretación de eventos, y a que los sistemas de videovigilancia que generalmente se diseñan están orientados a entornos específicos, las prestaciones de los sistemas de vigilancia no suelen ir más allá de la detección, clasificación y el seguimiento de objetos, por lo que el diseño de un sistema de propósito general debe ser lo suficientemente flexible para permitir cumplir con los requerimientos de los diferentes sectores sin estar ligado a ninguno en específico.

Por otra parte, la complejidad de los sistemas de vigilancia inteligentes actuales lleva a buscar la reutilización de los diseños y desarrollos existentes a la hora de diseñar y desarrollar sistemas que vigilen nuevos entornos. Un entorno vigilado $E$ consta de un conjunto de sensores heterogéneos $S_1$, $S_2$,…, $S_n$ distribuidos geográficamente en dicho entorno. Cada uno de estos sensores ofrece una visión particular de una región o zona del entorno $E$, el cual puede ser considerado a su vez como un subentorno $E_i$ de $E$. El problema de la vigilancia está enfocado principalmente a la interpretación de los sucesos en cada uno de estos subentornos $E_i$, con respecto a los aspectos que se quieran vigilar.

Teniendo en cuenta lo anterior se propone una arquitectura, basada en componentes de análisis que se pueden combinar de diferentes formas para formar motores de analítica, que al combinarse con los motores de reglas le dan la capacidad al sistema de trabajar con diferentes entornos.

### 4.1.1. Tipos de usuarios al que sistema está orientado

El sistema de videovigilancia inteligente para la detección de objetos abandonados y robados permitirá a las organizaciones apoyar las labores del personal de seguridad, igualmente permitirá a estudiantes o docentes, trabajar en la implementación de nuevas técnicas de análisis, gestión de contenido multimedia, técnicas de procesamiento digital de imágenes, etc. Lo que permitirá, además de brindar una mejor seguridad a las organizaciones que lo implemente, avanzar en las diferentes áreas de análisis y de esta manera disminuir la dependencia humana en las tareas de supervisión de seguridad.

### 4.1.2. Actores del sistema

Los actores que van a interactuar con el sistema videovigilancia inteligente propuesto son: el administrador, el vigilante y el vigilado. Cabe recordar que como indica el Lenguaje de Modelamiento Unificado (UML), los actores no representan siempre personas físicas, sino diferentes formas de interactuar con el sistema.

#### 4.1.2.1. Administrador

El administrador es el encargado de la instalación y configuración del sistema, igualmente y dado que el sistema está enfocado a la investigación, el administrador puede ser a su vez un investigador que desee probar nuevas configuraciones. Las principales funcionalidades del administrador son:

- Configurar las cámaras con las que el sistema interactúa, en esta configuración se definen parámetros como: la resolución de captura, FPS y formato de video de la cámara, entre otros.

- También es responsabilidad del administrador definir las políticas de análisis de cada cámara, esto es, con qué fin se van a utilizar, que algoritmos se van a usar, que situaciones van a lanzar una alarma, que zonas se van a vigilar, cuales se van a excluir, etc.

- El administrador debe configurar y gestionar la base de datos, definiendo las políticas de almacenamiento, en donde se debe parametrizar el espacio de almacenamiento y el tiempo máximo que van a durar los videos almacenados en la base de datos.

- El administrador será el encargado de cargar una imagen que represente el mapa del sitio vigilado, aquí deberá especificar la ubicación individual de las cámaras en el mapa.

- Es función del administrador gestionar las cuentas de los usuarios del sistema, en donde puede crear, modificar y eliminar usuarios.

- El administrador del sistema tiene también las funcionalidades del vigilante, sin embargo, no es su principal razón de ser, por lo que se entiende que es un actor secundario.

#### 4.1.2.2. Vigilante

El vigilante es todo aquel que controle el cliente, se supone que sea una persona con bajos conocimientos informáticos pues generalmente será un vigilante de la organización que use el sistema. Las principales actividades que debe realizar el vigilante son:

- Observar los videos de las diferentes cámaras, ya sea uno a la vez o varios al mismo tiempo, según la distribución de videos por pantalla que escoja.

- El vigilante puede mirar los mapas que el administrador haya creado, y puede elegir una de las cámaras de los mapas para observar el video de la misma.

- El vigilante puede mediante los controles PTZ, controlar una cámara PTZ para que se mueve horizontal o verticalmente, al igual que para alejar y acercar.

- El vigilante puede consultar los videos que hayan almacenados en la base de datos y puede buscar mediante filtros temporales o de eventos determinados.

- Cada alarma que se detecte por algún evento, como robo o abandono de objetos, puede ser consultada por el vigilante y ésta se notificará en tiempo real al vigilante.

- El vigilante podrá actualizar los datos de usuario, de la cuenta que el vigilante haya usado para iniciar sesión.

#### 4.1.2.3. Vigilado

El vigilante es todo objeto, o persona, al cual el algoritmo de seguimiento de objetos está rastreando y que finalmente puede o no disparar una alarma de un evento de robo, abandono o movimiento.

### 4.1.3. Modelo de casos de uso

Los casos de uso son una técnica para especificar el comportamiento de un sistema. De igual manera se puede decir que un caso de uso es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. El modelo de casos de uso describe el uso del sistema y cómo éste interactúa con el usuario. Al ser parte del análisis se describe qué es lo que es sistema debe hacer.

Los casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante la interacción con los usuarios y otros sistemas. Los casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona una respuesta a eventos que se producen en el mismo.

La figura 4.1 muestra el diagrama de casos de uso a alto nivel del sistema, donde se puede observar que el sistema interactúa con tres actores: Administrador, Vigilante y Vigilado.

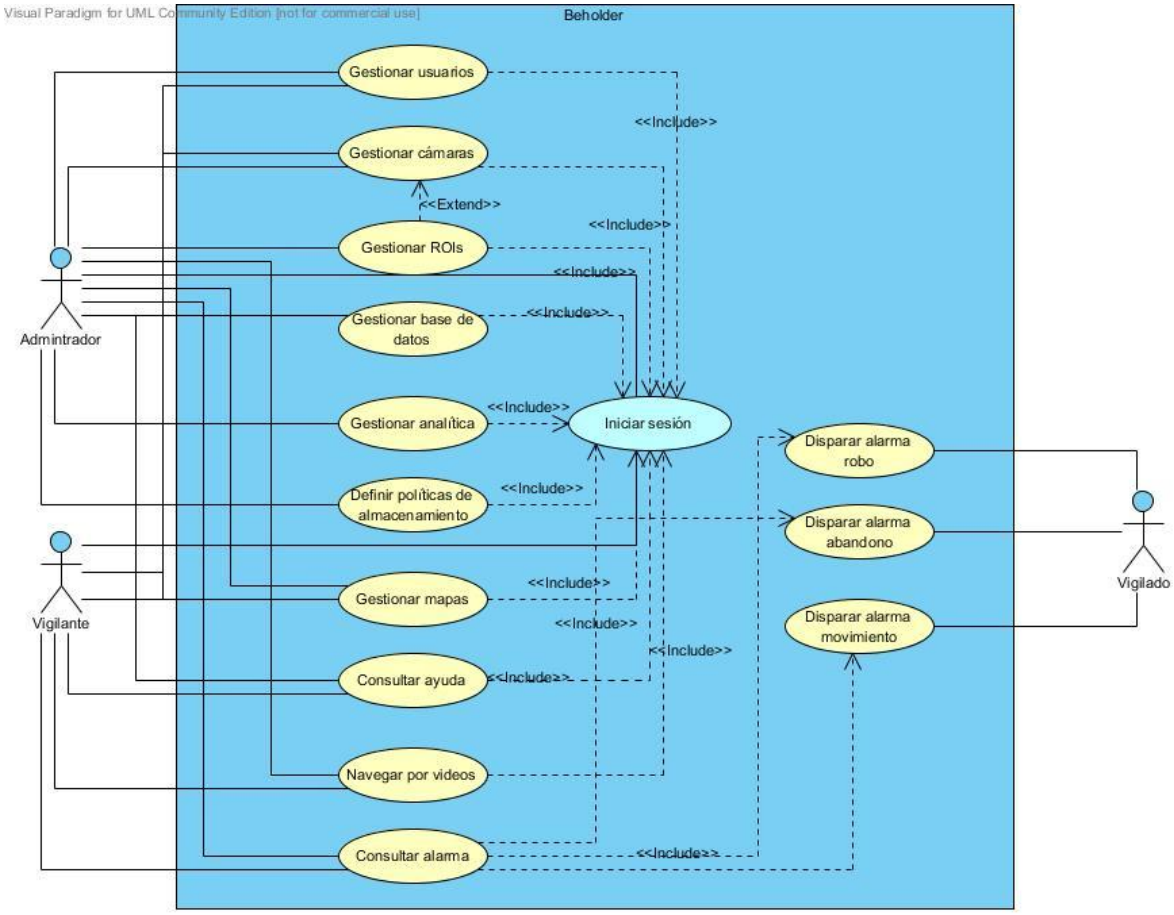



**Figura 4.1** Diagrama de caso de uso a alto nivel del sistema.

La tabla 4.1 lista los casos de uso del sistema, los diagramas de caso de uso a bajo nivel, así como las especificaciones de cada uno de ellos, se pueden consultar en el *Documento de Especificación de Requisitos de Software* (Anexo C).

**Tabla 4.1** Casos de uso del sistema propuesto

| Identificador | Nombre de caso de uso |
|---|---|
| **CU100** | **Gestionar usuarios** |
| CU101 | Iniciar sesión |
| CU102 | Cerrar sesión |

| | |
|---|---|
| CU103 | Registrar usuario |
| CU104 | Ver listado de usuarios |
| CU105 | Consultar usuario |
| CU106 | Actualizar usuario |
| CU107 | Eliminar usuario |
| CU108 | Recuperar contraseña |
| **CU200** | **Gestionar analítica** |
| CU201 | Listar motores de analítica |
| CU202 | Añadir motor de analítica |
| CU203 | Consultar motor de analítica |
| CU204 | Modificar motor de analítica |
| CU205 | Eliminar motor de analítica |
| CU206 | Exportar la configuración del motor de analítica a XML[1] |
| CU207 | Cargar la configuración del motor de analítica desde un XML |
| **CU300** | **Gestionar cámaras** |
| CU301 | Listar cámaras |
| CU302 | Consultar cámara |
| CU303 | Registrar cámara |
| CU304 | Modificar cámara |
| CU305 | Eliminar cámara |
| CU306 | Ver video de la cámara seleccionada |
| CU307 | Organizar distribución de video |
| CU308 | Controlar cámaras PTZ |
| **CU400** | **Gestionar regiones de interés** |
| CU401 | Mostrar ROI de actividad |
| CU402 | Añadir ROI de actividad |
| CU403 | Modificar ROI de actividad |
| CU404 | Quitar ROI de actividad |
| CU405 | Mostrar ROI de alarma |
| CU406 | Añadir ROI de alarma |
| CU407 | Modificar ROI de alarma |
| CU408 | Quitar ROI de alarma |
| **CU500** | **Gestionar base de datos** |
| CU501 | Consultar espacio de almacenamiento |
| CU502 | Consultar tiempo de almacenamiento |
| CU503 | Determinar espacio de almacenamiento |
| CU504 | Determinar tiempo de almacenamiento |

[1] Lenguaje de marcas extensible, del inglés eXtensible Markup Language, es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.

| | |
|---|---|
| **CU600** | **Definir políticas de almacenamiento** |
| CU601 | Modificar política de almacenamiento |
| CU602 | Modificar horarios de almacenamiento |
| **CU700** | **Gestionar mapas** |
| CU701 | Agregar mapa |
| CU702 | Listar mapas |
| CU703 | Consultar configuración de mapa |
| CU704 | Eliminar mapa |
| CU705 | Modificar mapa |
| CU706 | Agregar cámaras a mapa |
| CU707 | Quitar cámara de mapa |
| CU708 | Consultar mapa |
| **CU800** | **Navegar por videos** |
| CU801 | Navegar por el video de una cámara |
| CU802 | Consultar video según parámetros de búsqueda temporales |
| CU803 | Consultar video según parámetros especificados a descriptores |
| **CU900** | **Consultar ayuda** |
| **CU901** | **Disparar alarma de movimiento** |
| **CU902** | **Disparar alarma de robo** |
| **CU903** | **Disparar alarma de abandono** |
| **CU904** | **Consultar alarma** |

## 4.2. Descripción del sistema

El sistema propuesto se compone de diferentes elementos distribuidos y se basa en un modelo cliente/servidor, está divido en dos niveles de abstracción: fisco y lógico.

### 4.2.1. Modelo físico

La parte física (figura 4.2) se compone varios computadores, los cuales actúan como: servidor, base de datos o clientes; y que están interconectados a través de una Ethernet de alta velocidad. Diferentes tipos de cámaras se pueden conectar bien sea a la red de alta velocidad o directamente al servidor a través de una tarjeta de adquisición de video o de puertos USB. La arquitectura propuesta también permite que los clientes y las cámaras IP se conecten a través de Internet.

La comunicación entre los elementos del sistema se basa en un modelo cliente/servidor y se realiza mediante el protocolo TCP. Para iniciar la transferencia de información el cliente debe iniciar sesión en el servidor. Para tratar con los problemas de la red, el sistema soporta, en ambos lados de la comunicación, el almacenamiento de datos en buffers.

El sistema consta de dos aplicaciones diferentes, pero complementarias, la aplicación del cliente y la aplicación del servidor. El servidor puede gestionar tantos clientes como se desee y como las capacidades técnicas lo permitan.

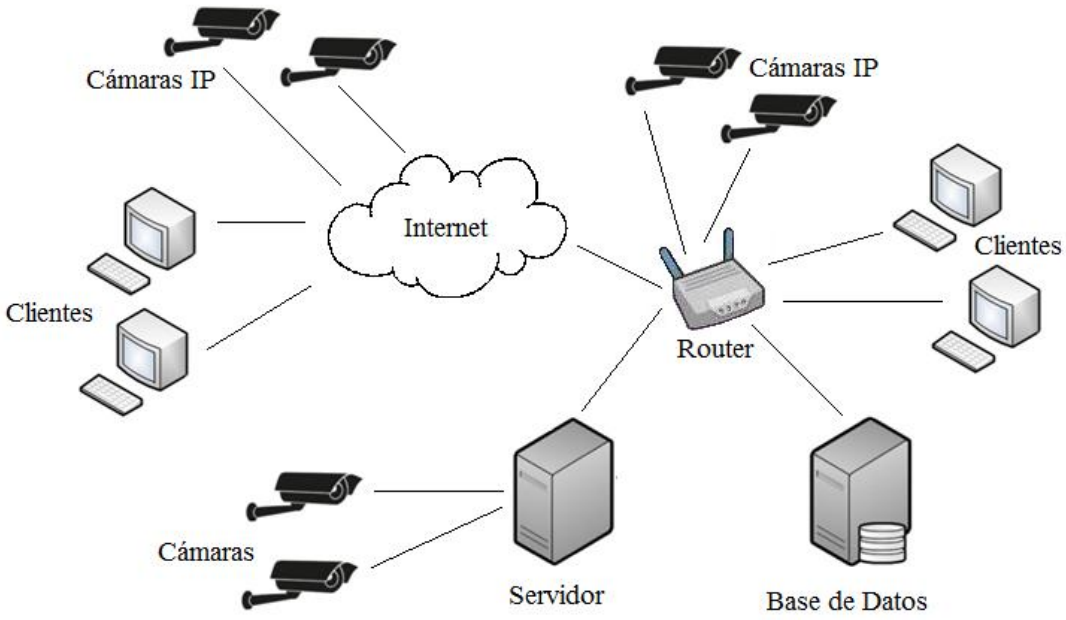



**Figura 4.2.** Descripción física del sistema (Fuente: El autor).

Es posible usar el sistema para organizaciones pequeñas o caseras sí: el servidor, la base de datos y el cliente, se encuentran en el mismo computador, sin embargo hay que considerar que el servidor va a tener que tratar con cargas adicionales y por lo tanto la cantidad de cámaras conectadas al sistema va hacer mucho menor.

#### 4.2.1.1. Cámara

La cámara es la principal fuente de entrada del sistema, se encarga de capturar las imágenes del área vigilada y enviar el stream[1] de video al servidor. Las cámaras que actualmente se usan como fuente de video son la cámara Logitech HD Pro Webcam C920 y PlayStation Eye (figura 4.3). Las características más importantes que presentan las cámaras se resumen en la siguiente tabla.

**Tabla 4.2.** Principales características de las cámaras usadas en el sistema.

| Cámara | Logitech HD Pro Webcam C920 | PlayStation Eye |
|---|---|---|
| **Tipo de conexión** | USB 2.0<br>USB 3.0 | USB 2.0 |

[1] Es la distribución multimedia a través de una red, de manera que el usuario consume el producto al mismo tiempo que se descarga.

| | | |
|---|---|---|
| **Formatos de compresión** | H.264, MPEG-2 | JPEG |
| **Resolución** | 1920 x 1080 a 30 fps | • 640 x 480 a 60 fps<br>• 320 x 240 a 120 fps |

**(a)** **(b)**

**Figura 4.3.** Cámaras usadas en el sistema. (a) Logitech HD Pro Webcam C920. (b) PlayStation Eye (Fuente: El autor).

#### 4.2.1.2. Servidor

El servidor es el principal componente del sistema, encargado de capturar el video de las cámaras, realizar la analítica de la secuencia de video, dar alarma, guardar y enviar el video procesado a los clientes y la base de datos.

Las características mínimas que debe cumplir el servidor son: CPU multinúcleo Intel o AMD con una velocidad de 2.5 Ghz y soporte SSE2, 500 megas libre de disco duro y una Gigabyte de RAM.

El servidor en sí mismo está compuesto por los siguientes módulos:

- Módulo de captura: Encargado de establecer la comunicación con las cámaras, capturar el stream de video de estás, descomprimirlo y enviar frame por frame al motor de analítica.

- Módulo analítica: Generalmente el servidor tiene más de un motor de analítica, orientados a entornos diferentes. El motor de analítica está compuesto por varios componentes: preprocesado, modelamiento de fondo, detección, clasificación y rastreo de objetos y reconocimiento de eventos.

- Módulo de comunicación: Este módulo es el encargado de gestionar las peticiones entrantes del cliente y enviar a éste el stream de video de las cámaras que solicite.

- Módulo de alarma: Éste módulo recibe las señales de alarma provenientes del componente de detección de eventos y envía las señales de alarma según la parametrización, donde las señales de alarma son correos a los destinatarios especificados, alarma visual y sonora al cliente y almacenamiento de la incidencia en la base de datos.

El servidor funciona como sistema de gestión inteligente, en el cual se ejecutan las tareas de análisis, los cuales son los procesos más complicados y complejos, por lo cual, y aunque la arquitectura permita tener el cliente en el mismo computador, se recomienda que el servidor sea una maquina dedicada única y exclusivamente para las tareas de análisis. Por lo anterior el servidor no tiene una interfaz gráfica de usuario, y su configuración se debe realizar desde el cliente.

#### 4.2.1.3. Cliente

El cliente es un computador estándar, en el cual corre la aplicación cliente, el cual es el programa encargado de mostrar por pantalla la salida de las cámaras que se soliciten, dar una alarma visual y sonora al usuario cada vez que ocurra un evento determinado, permitir configurar el sistema y ver las secuencias de video almacenadas.

Las características mínimas que debe cumplir el servidor son: CPU Intel o AMD 2.0 Ghz, 100 megas libre de disco duro y una Gigabyte de RAM.

El cliente consta de los siguientes módulos:

- Módulo de reproducción de video: Este módulo se encarga de recibir el video y los descriptores del cliente y/o base de datos, mostrar el video por pantalla, se encarga de las diferentes distribuciones de video, del control de las cámaras PTZ y de los controles que permiten navegar por el video.

- Módulo de gestión de alarmas: Es el encargado de recibir las señales de alarma del servidor, emitir una alarma visual y/o sonora y de navegar entre las diferentes alarmas.

- Módulo de Recuperación de datos multimedia: Éste módulo es el encargado de proporcionar mecanismos de búsqueda para poder recuperar los diferentes videos almacenados en la base de datos.

- Módulo de comunicación: Este módulo se encarga de solicitar el video de las diferentes cámaras al servidor y de gestionar la comunicación que se haga con éste.

- Módulo de Configuración: Este módulo es el encargado de la configuración del sistema.

#### 4.2.1.4. Base de datos

La base de datos se encarga de guardar los videos y los descriptores de los mismos, igualmente guarda la configuración del sistema Beholder. El motor gestor de base de datos que utiliza el sistema es *MySQL*.

Las características físicas que debe tener el computador que corra la base de datos dependen del número de cámaras que posea la organización, de las políticas de almacenamiento que se definan y del tiempo que se configure para que los videos permanezcan almacenados.

El sistema permite definir tres tipos de políticas de almacenamiento por cámara:

- Almacenamiento activo: Si una cámara se configura para trabajar con almacenamiento activo ésta siempre está grabando.

- Almacenamiento en movimiento: Graba siempre y cuando se detecte movimiento en la región de actividad.

- Almacenamiento en alarma: Una cámara configurada para trabajar con este tipo de política graba solo cuando se detecte una alarma.

- Almacenamiento manual: Aunque no es una política de almacenamiento, el sistema también permite grabar manualmente, es decir que el operador del cliente puede comenzar una grabación y detenerla cuando éste lo desee.

Además de lo descrito anteriormente, el sistema permite definir horarios de almacenamiento por cámaras, para que las cámaras con horarios definidos solo graben en las zonas horarias determinadas.

### 4.2.2. Modelo lógico

El modelo lógico del sistema (figura 4.4) se compone de cinco capas independientes, las cuales cumplen con un rol específico y están diseñadas de forma modular.

#### 4.2.2.1. Capa de adquisición

Está capa se encarga de conectarse a las cámaras instaladas en el sistema, adquirir las secuencias de video de las cámaras o archivos, para posteriormente almacenarlos en un buffer, con el fin de distribuirlo frame a frame a la capa de análisis. Actualmente el intercambio de frames se puede hacer sin compresión o mediante el formato JPEG (ISO/IEC 10918-1).

La figura 4.5 describe parte de las clases que intervienen en la capa de adquisición de video.

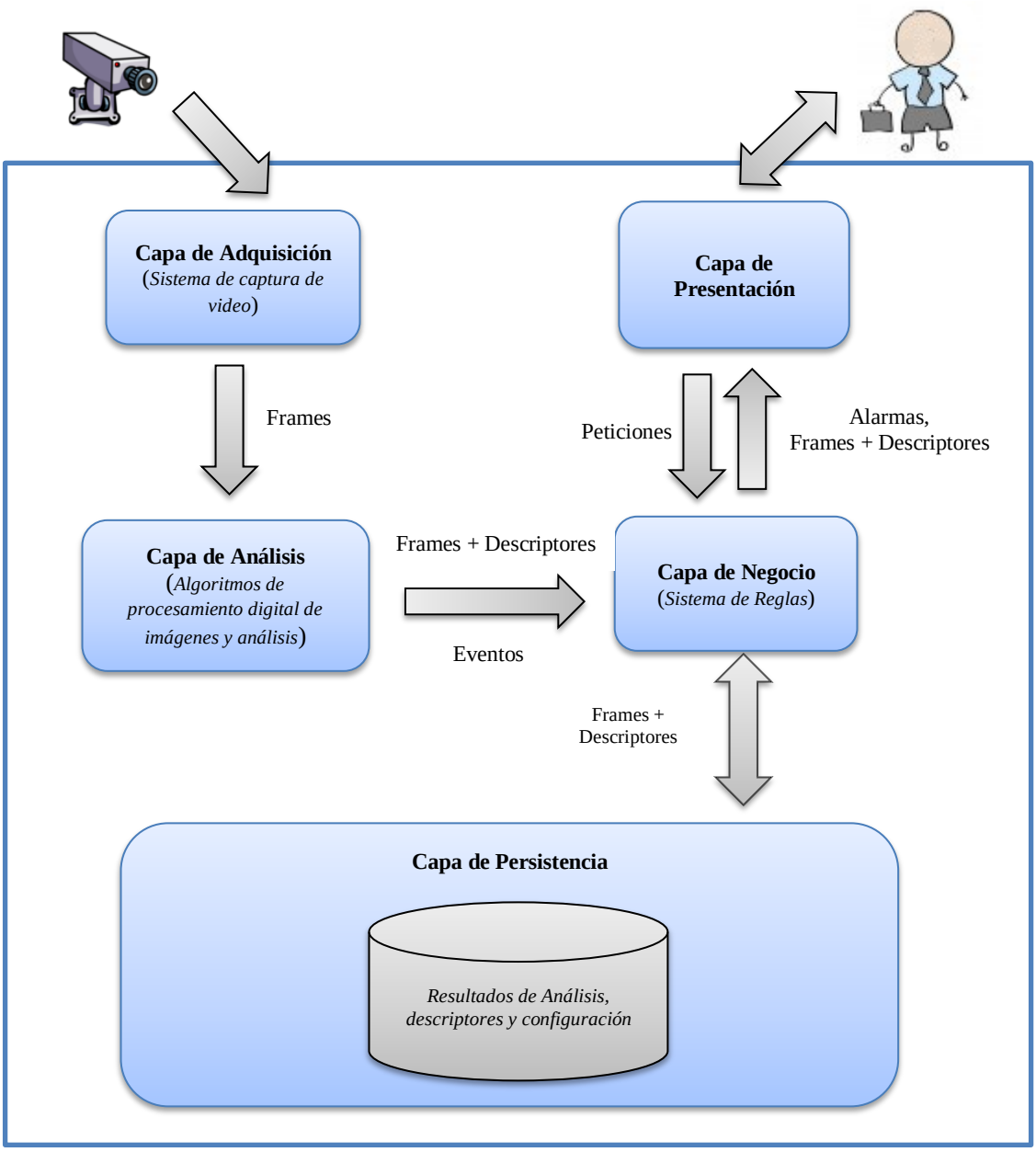



**Figura 4.4.** Modelo lógico del sistema (Fuente: El autor).

El sistema adquiere video de cámaras o archivos de video, esto lo hace mediante las clases *FileStream*, *BufferedFileStream*, *CameraStream* y *BufferedCameraStream*. Este diseño permite una fácil adopción de nuevos tipos de cámaras.

Todas las clases de captura derivan de la clase *VideoStream*, *que* sirve de interfaz única de acceso, proporciona unos métodos estándar para todas las clases de adquisición y hereda de

*QThread*, por lo que cada secuencia de video es un hilo. La clase *VideoStream* almacena en un buffer configurable los frames capturados, lo que permite evitar problemas en la comunicación.

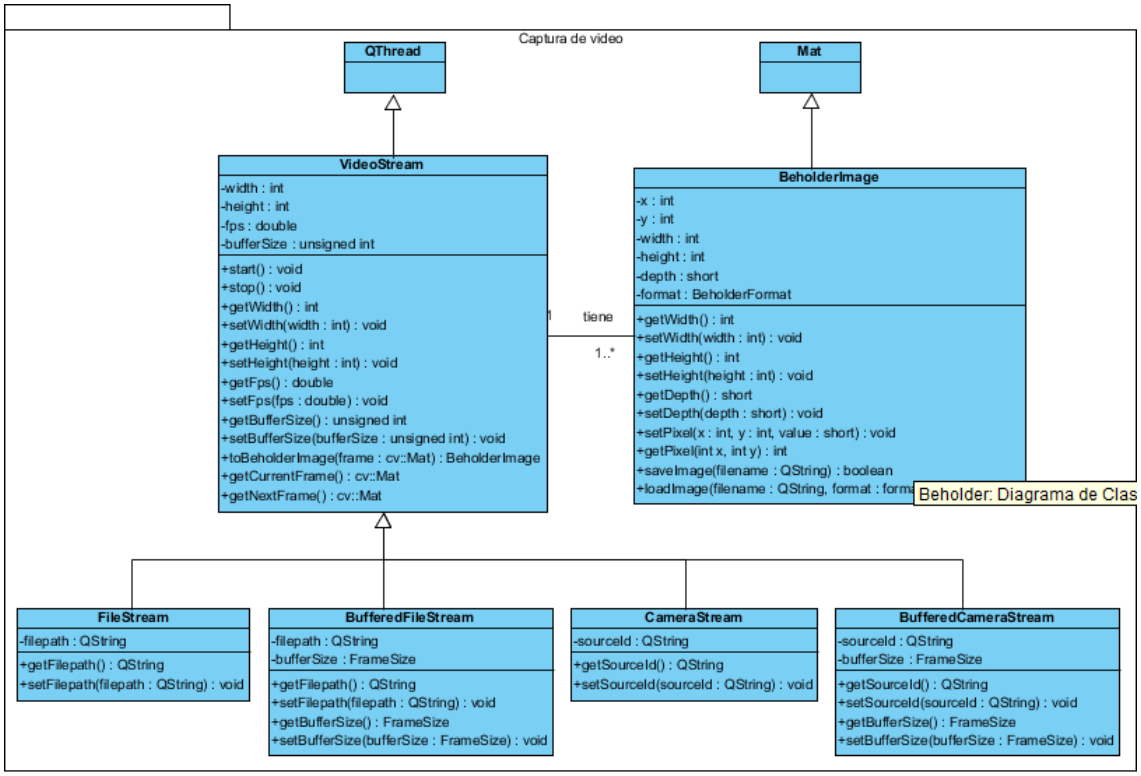



**Figura 4.5.** Diagrama de clases del módulo de adquisición (Fuente: El autor).

Las imágenes proporcionadas por el módulo de adquisición consisten en frames almacenados en el formato de imagen *BeholderImage*, el cual a su vez se basa en la clase *Mat* de librería *OpenCV*. *BeholderImage* es el formato de imagen usado en todo el sistema, por lo que es necesario transformar los frames capturados en otro formato a *BeholderImage.*

La secuencia de la capa de adquisición de video es la siguiente:

- El módulo de adquisición de video se inicializa con los parámetros que se cargan de la configuración del sistema, los cuales se cargan de la base de datos.

- Por cada cámara conectada al sistema se lanza un hilo para capturar los frames. Los funciones de este hilo son:

    - Inicializa la instancia que captura los frames de la cámara según los parámetros de configuración de cada cámara.

- Comienza la captura de frames, el cual se ejecuta en un bucle infinito. La única condición para detener el bucle es que el usuario decida detener la captura de frames.

- Los frames capturados son transformados al formato de imagen *BeholderImage*.
- Las imágenes son puestas en el buffer de video.

– Se lanza otro hilo encargado de distribuir las imágenes que se encuentran en el buffer a la capa de análisis.

#### 4.2.2.2. Capa de análisis

La capa de análisis captura los frames trasmitidos por la capa de adquisición, con el fin de detectar automáticamente eventos y actividades específicas. Cada frame que entra en esta capa sufre una serie de transformaciones, al finalizar el proceso cada frames estará acompañado de un descriptor XML, que contiene la información resultante de determinados módulos de análisis.

En la arquitectura propuesta, un evento o actividad específica está definido por un motor de analítica. Cada cámara conectada al sistema tiene asociada a ella uno o más motores de analítica, los cuales se parametrizan para ejecutarse en el entorno definido por cada cámara.

Cada motor de analítica transfiere los frames y los descriptores XML a la capa de negocio, al igual que los eventos detectados por el motor de analítica.

La secuencia que realiza la capa de análisis se puede describir en el siguiente proceso:

– El módulo de análisis se inicializa con la paremetrización definida por el administrador del sistema, está parametrización se carga desde la base de datos.

– Cada cámara puede ejecutar n motores de analítica, por tanto se lanzan tanto hilos como motores asociados existan. El proceso de inicialización de los motores de analítica son:

- Al momento de construir cada motor asociado a una cámara, se cargan los parámetros de configuración propios de esa asociación.

- Cada motor solicita a la capa de adquisición los frames de la cámara a la que pertenece.

- Cada motor comienza un ciclo infinito, con el fin de procesar los frames según la configuración de cada motor. La única forma de detener este ciclo es mediante la instrucción del usuario.

- Los motores envían los frames más los descriptores resultantes a la capa de negocio.

– Cada vez que se un motor de analítica detecte un evento, el motor envía a la capa de negocio la incidencia.

Aunque la capa de adquisición envíe toda el frame capturado por la cámara, el motor de analítica generalmente no procesa toda la imagen. Por cada motor de analítica configurado se parametriza una región de interés de actividad y una región de interés de alarma. En la región de interés de actividad, el motor va a correr la mayoría de sus módulos de análisis, a excepción de los módulos de análisis destinados a disparar una alarma. Las únicas alarmas que se ejecutan en las regiones de interés de actividad son las alarmas de movimiento. La figura 4.6 muestra el diagrama de bloques de los procesos de la capa de análisis

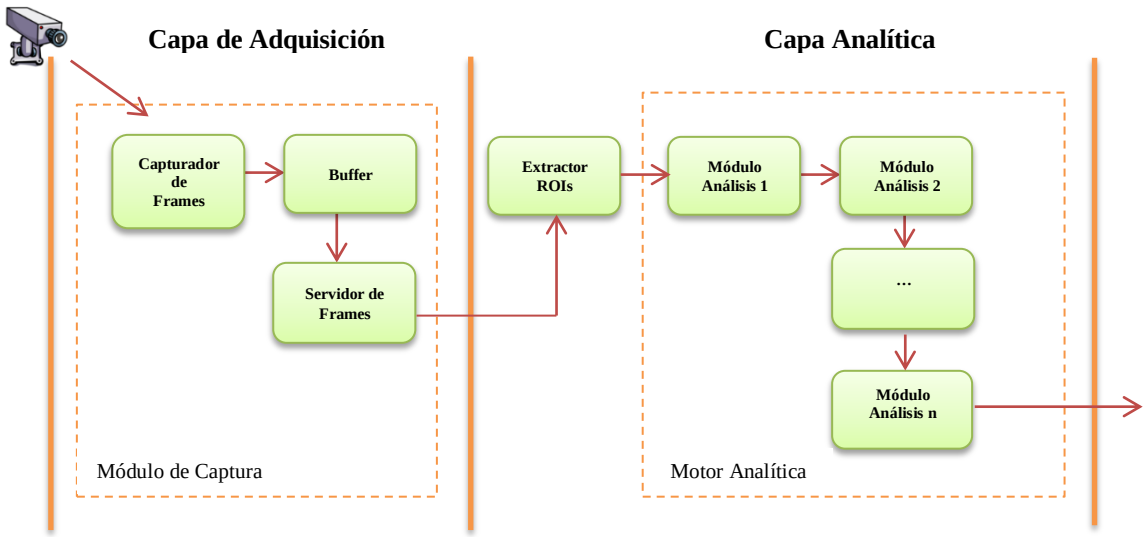



**Figura 4.6.** Diagrama de bloques de la capa de análisis (Fuente: El autor).

Un motor tiene asociado varios módulos de análisis, inicialmente se plantearon los módulos de: preprocesado, sustracción de fondo, postprocesado, extracción de sombras, seguimiento de objetos, clasificación de objetos, clasificación de regiones estáticas, detección de objetos abandonados y detección de objetos robados. En el seguimiento de objetos se determinaron interfaces para rastreadores basados en puntos y en siluetas. La figura 4.7 muestra las clases asociadas al motor de analítica.

Cada motor de analítica transfiere los frames y los descriptores XML a los clientes conectados a él. Los eventos detectados por cualquier motor de analítica son trasmitidos a la capa de negocio.

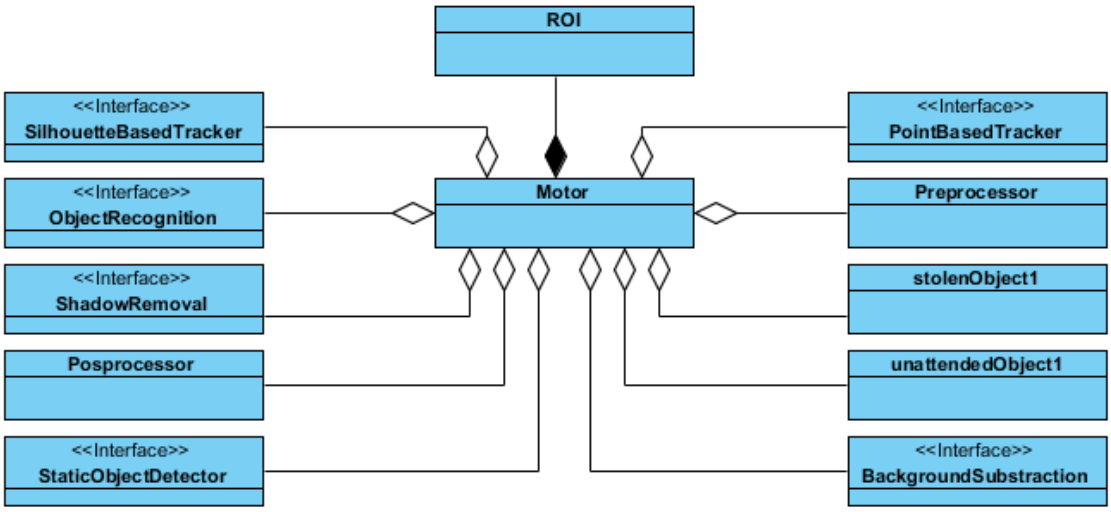



**Figura 4.7.** Diagrama de clases parcial de módulo de analítica (Fuente: El autor).

#### 4.2.2.2.1. Descriptor XML

Cada vez que se procesa que un módulo de análisis de un motor de analítica se ejecuta sobre una secuencia de video, éste transforma los frames para poder ser usados por el próximo módulo de análisis. Sin embargo, algunos módulos generan información contextual, la cual puede ser usada en la capa de negocio, para detectar eventos definidos o en la capa de presentación para mostrar información al usuario, como rectángulos o etiquetas haciendo referencia a un objeto (figura 4.8).

La información contextual se transfiere en formato XML, lo que garantiza que los otros módulos de análisis sepan cómo leer e interpretar los datos que allí aparezcan. El intercambio de información mediante XML, permite estandarizar convenciones de intercambio de información, lo que garantiza que los módulos o motores de análisis que estén por desarrollar se ajusten a la arquitectura propuesta.

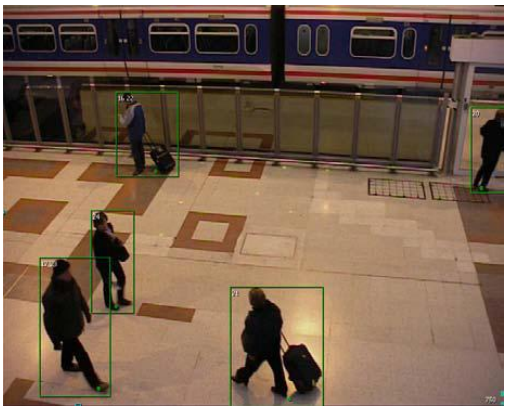


**Figura 4.8.** Ejemplo de rectángulos y etiquetas que describen objetos (Tomado de http://www.cvg.rdg.ac.uk/PETS2006/data.html).

El esquema XML del descriptor se puede observar en la siguiente figura.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="frame">
        <xsd:complexType>
            <xsd:attribute name="id" type="xsd:integer"/>
            <xsd:attribute name="frame" type="xsd:integer"/>
            <xsd:sequence>
                <xsd:element name="width" type="xsd:integer"/>
                <xsd:element name="height" type="xsd:integer"/>
                <xs:element ref="analityc-module" maxOccurs='unbounded'/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name="analityc-module">
        <xsd:complexType>
            <xsd:attribute name="id" type="xsd:integer"/>
            <xsd:attribute name="name" type="xsd:string"/>
            <xsd:attribute name="timestamp" type="xsd:string"/>
            <xsd:sequence>
                <xs:element ref="param" maxOccurs='unbounded'/>
                <xs:element ref="boundingbox" minOccurs='0' maxOccurs='1'/>
                <xs:element ref="label" minOccurs='0' maxOccurs='1'/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

   <xsd:element name="param">
        <xsd:complexType>
            <xsd:sequence>
                <xs:element name="param-name" minOccurs='1' maxOccurs='1'/>
                <xs:element name="param-value" minOccurs='1' maxOccurs='1'/>
            </xsd:sequence>
        </xsd:complexType>
   </xsd:element>

    <xsd:element name="boundingbox">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="top-left-corner" type="xsd:integer"/>
                <xsd:element name="bottom-right-corner" type="xsd:integer"/>
            </xsd:sequence>
        </xsd:complexType>
   </xsd:element>

   <xsd:element name="label">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="top-left-corner" type="xsd:integer"/>
                <xsd:element name="bottom-right-corner" type="xsd:integer"/>
                <xsd:element name="value" type="xsd:string"/>
            </xsd:sequence>
        </xsd:complexType>
   </xsd:element>

</xsd:schema>
```

**Figura 4.9.** Descriptor XML generado por los módulos de análisis (Fuente: El autor).

#### 4.2.2.3. Capa de negocio

La capa de negocio es el puente principal entre la capa de análisis, la capa de persistencia y la capa de presentación. Cualquier comunicación que se debe realizar entre éstas se debe realizar mediante la capa de negocio. La cual valida los permisos de la petición y ejecuta una serie de reglas definidas.

Esta capa se encarga de gestionar todas las reglas que se parametrizan, esta capa decide quienes pueden ver una cámara, quienes reciben alarmas, cuando se almacena, etc. Por lo tanto la capa de negocio tiene un flujo continuo de comunicación con las capas de: análisis, presentación y persistencia.

Como ya se dijo, cada motor de analítica que se define detecta un evento particular, el motor de objetos robados y abandonados, por ejemplo, detecta el abandono o robo de objetos; cada vez que un motor de analítica detecta un evento especifico, éste le envía una señal a la capa de negocio, que es la encargada de determinar de qué manera debe tratar este evento.

En la arquitectura propuesta se plantea el concepto de motor de reglas. Cada motor de analítica contiene un motor de reglas, el cual a través de información contextual puede definir nuevas reglas.

Las responsabilidades de la capa de negocio son:

- Procesar mediante los motores de reglas la información recibida de la capa de análisis, lo cual puede generar nuevas alarmas.

- Enviar un mensaje de alerta a los usuarios con los respectivos permisos, una vez que se detecte una alarma.

- Almacenar en la base de datos los videos cada vez que un motor de analítica emita un evento, aunque es posible que el comportamiento anterior cambie si las políticas de almacenamiento sean activas o por movimiento.

- Enviar un correo electrónico a usuarios que estén en la lista de correo de alarmas de las diferentes cámaras.

- Borrar los videos almacenados una vez superen el tiempo de almacenamiento especificado.

- Recibir las peticiones del usuario para ver el video de cámaras determinadas y enviarle el video de la cámara solicitada.

Además de los puntos anteriores esta capa se diseñó para tratar con una serie de reglas más complejas pero aún no disponibles en el software actual, tales como gestión automática de alarmas, en donde el sistema debe tomar decisiones basadas en las alarmas, como llamar a la policía, activar una alarma de incendio, etc. También existen una serie de reglas que se podrán definir los motores de reglas como, detectar personas caminando en una dirección determinada.

#### 4.2.2.4. Capa de presentación

La capa de presentación es la interfaz principal de interacción del usuario con la aplicación, la cual se diseñó usando conceptos de usabilidad y usando componentes de la librería gráfica de *Qt*.

La principal función de la capa de presentación es ver la secuencia de video de las cámaras, para esto el módulo de presentación de video recibe una serie de frames y descriptores de la capa de negocio. El módulo de presentación de video debe convertir los frames a una secuencia de video, estos se deben almacenar en un buffer el cual habilita que el usuario pueda pausar o navegar por el video. Igualmente la capa de presentación debe interpretar el descriptor XML y superponer la información planteada en él en las secuencias de video.

Como está capa recibe todas las entradas del administrar o vigilante, esta capa tiene varias responsabilidades, las cuales se pueden resumir en el siguiente listado:

- Permite ver a un usuario determinadas cámaras instaladas en el sistema, según los roles que éste posea.

- Permite ver la secuencia de video de una cámara que el usuario elija.

- Muestra al usuario, según los roles que posea, determinados mapas, en donde se ve la distribución de las cámaras instaladas y permite elegir una para ver la secuencia de video.

- Permite ver el video de varias cámaras al mismo tiempo, y da la capacidad de que le usuario elija una distribución de cámaras de un conjunto ya predeterminado.

- La capa de presentación permite al usuario buscar, bien sea por filtros temporales o espaciales, videos almacenados en la base de datos.

- Muestra las alarmas que se hayan trasmitido, al usuario que haya iniciado sesión, desde la capa de negocio y permite que el usuario vea la incidencia si da clic en la alarma.

- Muestra un log de alarmas, permite que el usuario pueda filtrar el log e ver el video de la incidencia si interactúa con alguna alarma.

- La capa de presentación da la posibilidad a un usuario de interactuar con los controles de navegación, los cuales permiten: pausar el video, reproducirlo a una velocidad lenta, atrasar o avanzar el video.

- La capa de presentación también tiene controles PTZ, destinados a tales cámaras, aunque estas aún no están soportadas por el módulo de captura.

- La capa de presentación permite que un administrador gestione: las cámaras, las regiones de interés, los usuarios, los motores de analítica y los mapas.

La capa de prestación tiene una continua comunicación con la capa de negocio. Cada vez que un usuario, selecciona una cámara para ver el video de la misma, envía una petición a la capa de negocio, la cual valida si el usuario puede o no ver el video de la cámara, de ser así, la capa de negocio, envía las secuencia de frames descriptores a la capa de presentación para que se visualicen.

La capa de presentación también puede buscar entre los videos almacenados en la base de datos, para ello solicita a la capa de negocios la lista de videos, dependiendo de los filtros de búsqueda. La capa de negocio debe validar cada petición antes de retornar la lista de videos, en donde se verifica si el usuario solicitante tiene permisos. De igual manera cada vez que un usuario, desde la capa de presentación solicita un video determinado, la capa de negocio valida la solicitud y recupera los datos almacenados para posteriormente enviarlos a la capa de presentación.

#### 4.2.2.5. Capa de persistencia

El propósito principal de la capa de persistencia es aislar el resto de las capas de la aplicación del motor o tecnología de base de datos que se esté usando. Esta capa se encarga de centralizar todos los accesos a la base de datos, a través de un framework, con la finalidad de leer, actualizar, modificar o borrar los objetos persistentes en el sistema de base de datos.

La comunicación entre la base de datos y el sistema Beholder se realiza a través *QtSql*, el cual usa el driver de *MySQL* para la conexión con el motor relacional. *QtSql* permite acceder de manera fácil a las base de datos, proporcionando un lenguaje de consulta embebido en el framework *Qt*.

La base de datos tiene dos fines en esta aplicación, almacenar los videos y descriptores resultantes de las cámaras y guardar la parametrización del sistema. El modelo entidad-relación se puede encontrar en el Anexo C.

## 4.3. Motor de detección de objetos abandonados y robados

Actualmente el sistema está compuesto por un motor de analítica orientado a la detección de objetos abandonados, es decir que dicho motor emite una señal de alarma si detecta un objeto que haya sido abandonado o robado.

Éste motor se compone los módulos de análisis de: preprocesado, modelado de fondo, sustracción de fondo, postprocesado, eliminación de sombras, extracción de blobs, seguimiento de objetos, clasificación de objetos, detección de objetos estáticos y detección de objetos abandonados y robados (figura 4.10).

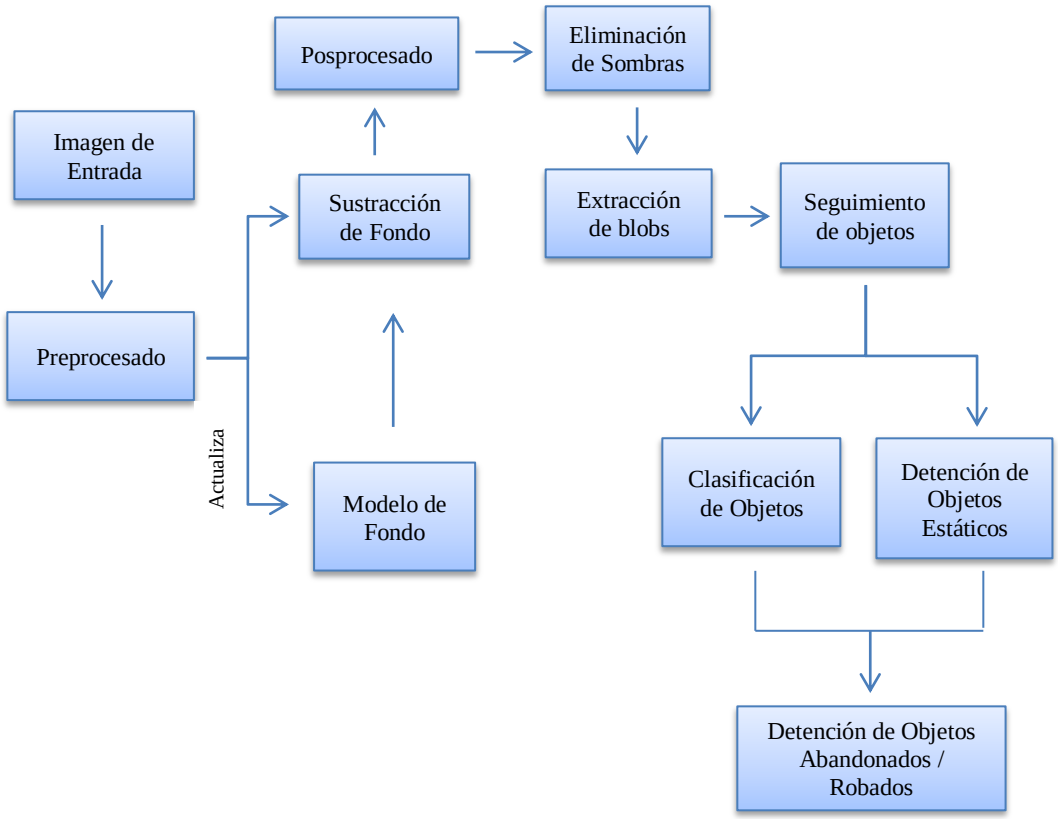



**Figura 4.10.** Diagrama de bloques del motor de detección de objetos abandonados y robados (Fuente: El autor).

### 4.3.1. Descripción de la actividad del motor

En primer lugar, la capa de adquisición captura la secuencia del video de la cámara deseada, la cual es enviada al motor para la detección de robo y abandono. Cada frame que recibe el motor actualiza el modelo de fondo y a su vez es usado por el módulo de sustracción del fondo, con el fin de detectar los píxeles que no pertenecen al fondo de la imagen, acto seguido se analizan estos pixeles mediante el módulo de postprocesado, el

cual elimina regiones muy pequeñas que se consideran como ruido. El módulo de postprocesado entonces transfiere el frame resultante al módulo de detección de sombras el cual se encarga de detectar y eliminar las sombras de la máscara binaria del foreground que determina los objetos. La máscara binaria final es entonces trasmitida al módulo de extracción de blobs, el cual analiza las regiones compactas de la máscara, con la finalidad de detectar los blobs.

Después, el módulo de seguimiento de objetos, genera la trayectoria de los objetos detectados entre imágenes consecutivas, con el fin de determinar parámetros de posición y velocidad.

El sistema analiza, mediante el módulo de detección de objetos estáticos, los datos obtenidos en el módulo anterior, con el fin de determinar cuáles regiones del foreground permanecen estáticas. Los objetos estáticos se analizan para determinar si se trata de personas u objetos.

En el módulo de detección de objetos abandonados o robados, los objetos estáticos, que no son personas, son analizados para determinar si son objetos abandonados o robados.

# *CAPÍTULO 5*

# *RESULTADOS*

# 5. Resultados

En el presente capítulo los resultados experimentales de la evaluación realizada sobre los diferentes algoritmos de sustracción de fondo y detección de objetos abandonados y robados, posteriormente, se realiza un análisis del comportamiento final del sistema, una vez implementado los módulos desarrollados en las distintas capas que componen el sistema.

Para la evaluación de los módulos de análisis, los algoritmos se implementaron en Matlab®. Las características más importantes del computador donde se ejecutaron los algoritmos son: procesador AMD Phenom II X4 945 3.4GHz, 8 GB de memoria Ram, 1 TeraByte de espacio en disco duro y una tarjeta de video Radeon 6850.

## 5.1. Detección de objetos

En esta sección se realiza una evaluación comparativa de diferentes algoritmos de sustracción de fondo, lo anterior por la inmensa importancia que tiene para las capas siguientes detectar correctamente los blobs.

En primer lugar se presentan los datasets que serán usados para la evaluación de las diferentes técnicas, a continuación se presentan las métricas que se usaron para evaluar los algoritmos, finalmente se presentan los resultados experimentales de la evaluación realizada sobre las diferentes aproximaciones.

### 5.1.1. Datasets

Para realizar la evaluación de los métodos de sustracción de fondo se han seleccionado un conjunto de diez secuencias de video extraídos del dataset CVPR2012. Los videos seleccionados se dividen en dos categorías: básicos y fondos dinámicos.

Cada frame de las secuencias de videos viene acompañado por el ground truth[1]. El ground truth es una imagen binaria donde se han marcado los objetos del frente con color blanco mientras que el color negro representa al fondo.

### 5.1.2. Métricas

Cada secuencia de video es procesada por un algoritmo de sustracción de fondo, al finalizar se genera una mascará binara, la cual es comparada con el ground truth con la finalidad determinar el rendimiento del algoritmo procesado. La exactitud del algoritmo evaluado se expresa en términos del recall, la precisión y mediante su media harmónica f-score. La figura 5.1 resume el framework usado para la evaluación.

[1] Es el proceso de marcar manualmente lo que se espera que un algoritmo muestre en la salida. Creadon anotaciones de referencia confiables y consistentes contra las cuales los algoritmos miden su rendimiento.

**Recall**: Se define como el número de pixeles del foreground correctamente identificados con respecto a los pixeles del ground truth. Se expresa en la siguiente ecuación:

$$\boldsymbol{RC} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FN}}$$

Donde *TP* indica el número de pixeles detectados correctamente y *FN* indica el número de pixeles que se clasificaron como background pero eran del foreground.

**Precisión**: Se define como el número total de pixeles del foreground correctamente identificados con respecto al total de pixeles clasificados como foreground. La ecuación que describe a la precisión se describe en la siguiente ecuación:

$$\boldsymbol{PR} = \frac{\boldsymbol{TP}}{\boldsymbol{TP} + \boldsymbol{FP}}$$

Donde *FP* son los pixeles que se clasificaron como foreground cuando en el ground truth están marcados como pixeles del background.

**$F_1$Score:** *$F_1$Score* es la medida de precisión que tiene una prueba. Se emplea en la determinación un valor único ponderado de la *precisión* y el *recall*. La fórmula que describe el *$F_1$Score* es:

$$\boldsymbol{FS} = 2\frac{\boldsymbol{Recall} * \boldsymbol{presicion}}{\boldsymbol{Recall} + \boldsymbol{presicion}}$$

La siguiente figura muestra el resumen de la metodología para evaluar la detección de objetos estáticos.

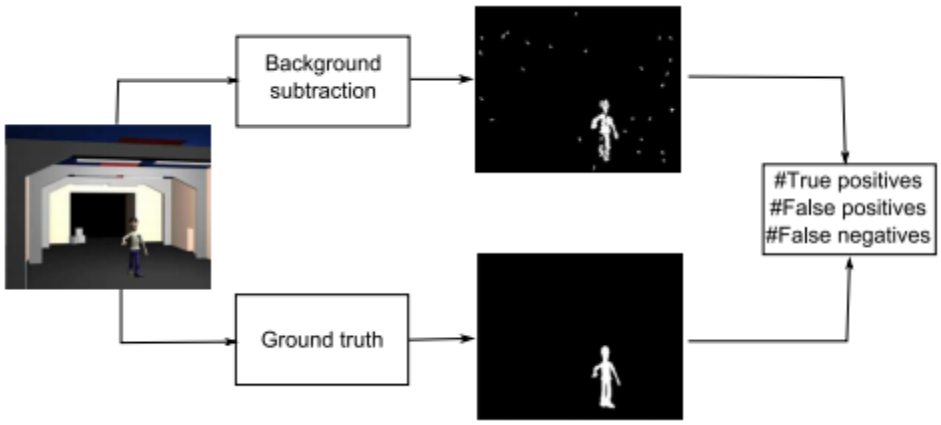



**Figura 5.1.** Metodología para determinar el resultado cuantitativo de la sustracción de fondo (Adaptado de S.C. Cheung y C. Kamath [61]).

### 5.1.3. Parametrización

Los algoritmos evaluados y los parámetros utilizados para realizar los estudios comparativos se muestran en la siguiente tabla:

**Tabla 5.1.** Parámetros utilizados para la evaluación de los algoritmos de sustracción de fondo.

| Algoritmo | Parámetros fijos | Parámetros de prueba |
|---|---|---|
| Frame Difference (FD) | | Umbral del foreground $T$ |
| Mean | Tamaño del buffer $L$ = 21<br>Tasa de muestreo $r$ = 5 fps | Umbral del foreground $T$ |
| Median | Tamaño del buffer $L$ = 21<br>Tasa de muestreo $r$ = 5 fps | Umbral del foreground $T$ |
| Mezcla de Gaussianas (MoG) | Número de componentes K = 3<br>Taza de adaptación $\alpha$ = 0.05<br>Peso del umbral T = 0.25<br>Variación inicial $\sigma^2$ = 35<br>Peso inicial $W_0$ = 0.1 | Umbral de desviación $D$ |
| KDE | Tasa de muestreo $r$ = 5 fps | Umbral del foreground $T$ |
| Eigenbackgrounds (Eigenbg) | Taza de aprendizaje $N_L$ = 100<br>Dimensión del eigenespacio $d$ = {10, 20, 30} | Umbral del foreground $T$ |

La selección de los parámetros adecuados es crítica para la evaluación de los métodos de sustracción de fondo. Existen múltiples métodos para la adaptación de parámetros, sin embargo el alcance de este trabajo no abarca la adaptación de parámetros. Por simplicidad, se buscaron los valores óptimos alrededor de los valores iniciales publicados por los autores de los métodos particulares, asumiendo que solo se debían hacer pequeñas adaptaciones con la finalidad de que los algoritmos se ajustaran a las pruebas. Además, el sistema se diseñó para que los parámetros de los algoritmos se pudieran calibrar en caliente y ajustar a los diferentes entornos.

### 5.1.4. Resultados

Las secuencias de videos que pertenecen a la categoría básica demuestran la capacidad que tienen los algoritmos para hacer frente a fondos que no tienen movimientos interesantes, la figura 5.2 muestra un ejemplo de aplicar los algoritmos de sustracción evaluados a una secuencia de video que pertenece a la categoría básica, en el Anexo B se pueden observar el resultado de la segmentación en las otras secuencias de video.

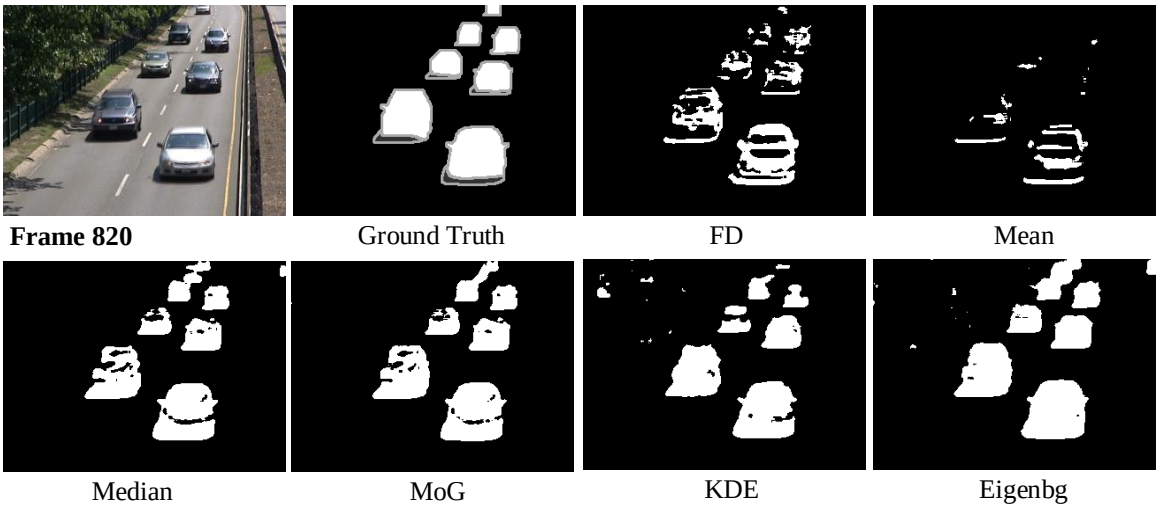



**Figura 5.2.** Resultado de los diferentes algoritmos de sustracción de fondo en la secuencia de video highway (Fuente: El autor).

Como se puede observar en la tabla 5.2 y la figura 5.4, las técnicas *Frame Difference* y *media* no tienen buen rendimiento, aunque tienen una buena precisión el bajo recall demuestra que estos métodos no detectan la mayoría de los pixeles que pertenecen al foreground. El rendimiento de *Eigenbackgrounds* tampoco es bueno, esta técnica presenta el mejor recall de todos los métodos, sin embargo aproximadamente el 45% de los pixeles que detecta como frente no pertenecen realmente al foreground. *Median*, *MoG* y *KDE* presentan un rendimiento similar, los tres algoritmos tienen una buen precisión y recall.

**Tabla 5.2.** Resultado promedio de la evaluación en la categoría básica.

| Método | TP | FP | FN | TN | RC | PR | FS |
|---|---|---|---|---|---|---|---|
| **FD** | 4513802 | 835175 | 15085333 | 636640391 | 0,32 | 0,82 | 0,41 |
| **Mean** | 1538177 | 390367 | 18060958 | 637085199 | 0,11 | 0,80 | 0,19 |
| **Median** | 15731805 | 2035027 | 3867330 | 635440539 | 0,84 | 0,91 | 0,87 |
| **MoG** | 16240855 | 1784379 | 3358280 | 635691187 | 0,87 | 0,92 | 0,89 |
| **KDE** | 17636479 | 5321154 | 1962656 | 632154412 | 0,91 | 0,81 | 0,86 |
| **Eigenbg** | 19093196 | 25651335 | 505939 | 611824231 | 0,98 | 0,55 | 0,68 |

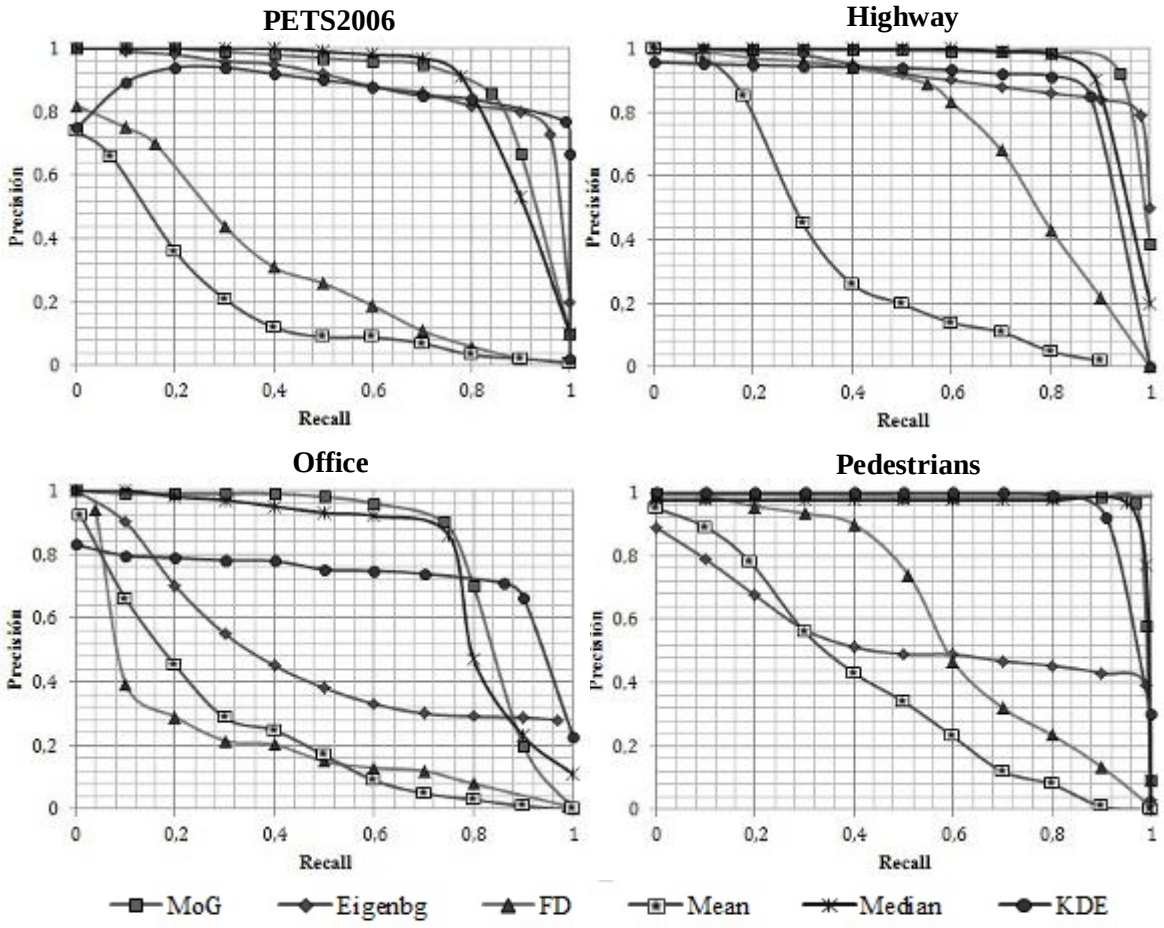



**Figura 5.3.** Gráficas de precisión y recall del rendimiento de los algoritmos de sustracción de fondo en la categoría básica (Fuente: El autor).

La categoría de fondos dinámicos permite ver como las diferentes técnicas evaluadas se comportan con elementos del fondo que se mueven significativamente, como el movimiento de los arboles causado por el viento o las olas del mar. La figura 5.4 muestra un ejemplo de aplicar los algoritmos de sustracción evaluados a una secuencia de video que pertenece a la categoría de fondos dinámicos, en el Anexo B se pueden observar el resultado de la segmentación en las otras secuencias de video.

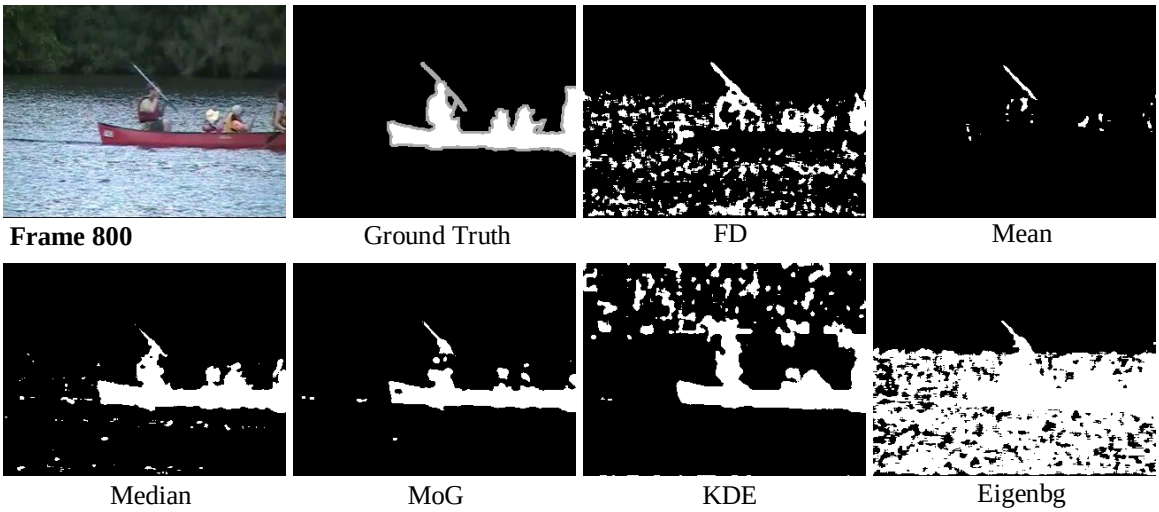



**Figura 5.4.** Resultado de los diferentes algoritmos de sustracción de fondo en la secuencia de video canoe (Fuente: El autor).

Como se aprecia en la tabla 5.3, el método *Frame Difference* presenta un mal comportamiento, esto se debe a que este algoritmo solo tiene en cuenta el último frame para modelar el fondo. Las aproximaciones *Median, MoG y Eigenbackgrounds* presentan la mejor representación de los fondos dinámicos, pero su rendimiento en la secuencias de video *fountain01* y *fall* dañan su promedio.

**Tabla 5.3.** Resultado promedio de la evaluación en la categoría de fondos dinámicos.

| Método | TP | FP | FN | TN | RC | PR | FS |
|---|---|---|---|---|---|---|---|
| **FD** | 10535616 | 51820154 | 12484154 | 1599727958 | 0,36 | 0,28 | 0,20 |
| **Mean** | 5658811 | 4547146 | 17596030 | 1649270248 | 0,11 | 0,70 | 0,15 |
| **Median** | 19710699 | 109944141 | 3544142 | 1543873253 | 0,81 | 0,51 | 0,57 |
| **MoG** | 19970192 | 43561481 | 3284649 | 1610255913 | 0,82 | 0,55 | 0,59 |
| **KDE** | 19993888 | 71015175 | 3260953 | 1582802219 | 0,82 | 0,37 | 0,49 |
| **Eigenbg** | 22516527 | 563982012 | 738314 | 1089835382 | 0,97 | 0,04 | 0,08 |

Como se ve en la figura 5.5, todos los algoritmos evaluados tienen problemas al modelar el movimiento de las ramas de los árboles y los chorros de agua de las fuentes, lo que demuestra que es bastante difícil modelar el movimiento aleatorio de estos elementos. En el Anexo A se pueden observar el comportamiento de cada algoritmo frente a cada secuencia de video particular.

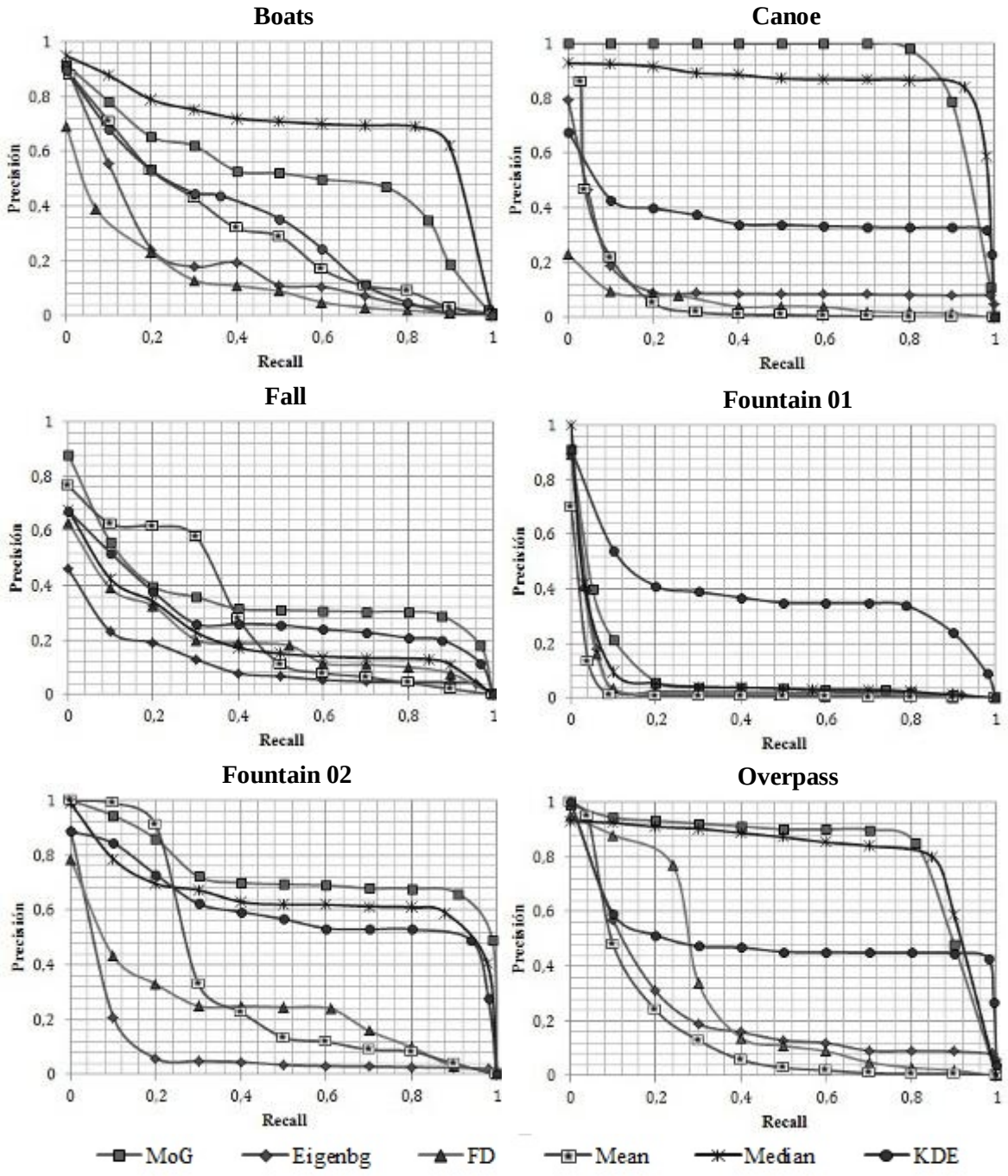



**Figura 5.5.** Gráficas de precisión y recall del rendimiento de los algoritmos de sustracción de fondo en la categoría de fondos dinámicos (Fuente: El autor).

En general y como se ve en la tabla 5.4 las técnicas *MoG*, *Median* y *KDE* presentan el mejor comportamiento, tanto en entorno básicos como con fondos dinámicos. *Mean*

seguido de *Frame Difference* y *Eigenbackgrounds* son las aproximaciones que presentan peor comportamiento a nivel general.

**Tabla 5.4.** Resultado general de la evaluación de los métodos.

| Método | Recall | Precisión | $F_1$Socre |
|---|---|---|---|
| **FD** | 0,340132933 | 0,547832681 | 0,30616412 |
| **Mean** | 0,11221558 | 0,750140588 | 0,168966161 |
| **Median** | 0,828672198 | 0,711343619 | 0,723113344 |
| **MoG** | 0,844023041 | 0,72977822 | 0,740879253 |
| **KDE** | 0,866058971 | 0,591159212 | 0,671229834 |
| **Eigenbg** | 0,975059311 | 0,296082942 | 0,378221457 |

## 5.2. Objetos abandonados y robados

En esta sección se presentan los resultados experimentales de dos algoritmos para la detección de objetos abandonados y robados, para esto en primer lugar se presentan los datasets que serán usados para la evaluación de las diferentes técnicas, después se presentan las métricas que se usaron para evaluar los algoritmos, finalmente se presentan los resultados experimentales de la evaluación realizada sobre las diferentes aproximaciones.

### 5.2.1. Datasets

Para la evaluación de los algoritmos de detección de abandono y robo de objetos se han seleccionado un conjunto de videos de los datasets PETS2006 y AVSS2007, los cuales se video se dividieron en las categorías simples y complejas. Las secuencias de video PETS2006 pertenecen a la categoría simples y los videos de AVSS2007 a la categoría compleja. En la siguiente tabla se muestran las características más relevantes de las categorías.

**Tabla 5.5.** Descripción de los escenarios usados para la evaluación de los algoritmos de detección de robo y abandono.

| Categoría | Simple | Compleja |
|---|---|---|
| Densidad de objetos | Medio | Alto |
| Cambios de iluminación | Bajo | Medio |
| Velocidad de los objetos | Bajo | Media – Alta |
| Oclusiones de objetos del foreground | No | Si |

El ground truth de los datasets contiene imágenes binarias marcando los diferentes objetos del foreground y dos archivo XML. El primer archivo XML contiene los parámetros de calibración de la cámara, el segundo archivo contiene información del ground truth.

### 5.2.2. Métricas

Para evaluar el rendimiento de los métodos implementados se han determinado las siguientes mediciones:

**Abandono del objeto:** Esta variable determina el momento en el cual una persona deja un objeto desatendido.

**Disparo de alarma**: Esta variable determina el momento en el cual el algoritmo dispara la alarma de abandono del objeto.

**Finalización de alarma:** Determina el momento cuando el objeto deja de estar abandonado.

### 5.2.3. Parametrización

Como se explicó en la sección 2.8, la detección de objetos abandonados y robados se aplica sobre las regiones que permanecen estáticas y que se hayan determinado como objetos.

La detección de regiones estáticas se realizó estudiando la velocidad de cada blob, en donde si un blob permanece quieto por más de diez segundos se considera estático. Una vez una región se haya determinado como estática, ésta se debe estudiar para clasificar si es una persona o un objeto.

La clasificación de objetos/personas se realizó mediante el estudio morfológico de la región estática, en donde se combinaron dos algoritmos: el primero calcula la relación entre el ancho y el alto del blob y el segundo el porcentaje de pixeles que pertenecen a la máscara binaria dentro del rectángulo que enmarca la detección del objeto, la figura 5.6 muestra un ejemplo de las diferencias entre una personas y un objeto.

El algoritmo que calcula la relación entre el ancho y la altura del blob, asume que generalmente una persona será más alta que ancha y un objeto será más ancho que alto. Las pruebas realizadas determinaron que dicha proporción se puede aproximar a una distribución Gaussiana con $\mu = 0,3$ y $\sigma^2 = 0,2$.

El algoritmo que calcula el porcentaje de pixeles que pertenecen a la máscara binaria dentro del rectángulo envolvente asume que un objeto "llenará" más el rectángulo envolvente, es decir que el porcentaje será mayor que el de una persona. Se determinó que si el porcentaje es menor que 75% entonces la región estática es una persona.

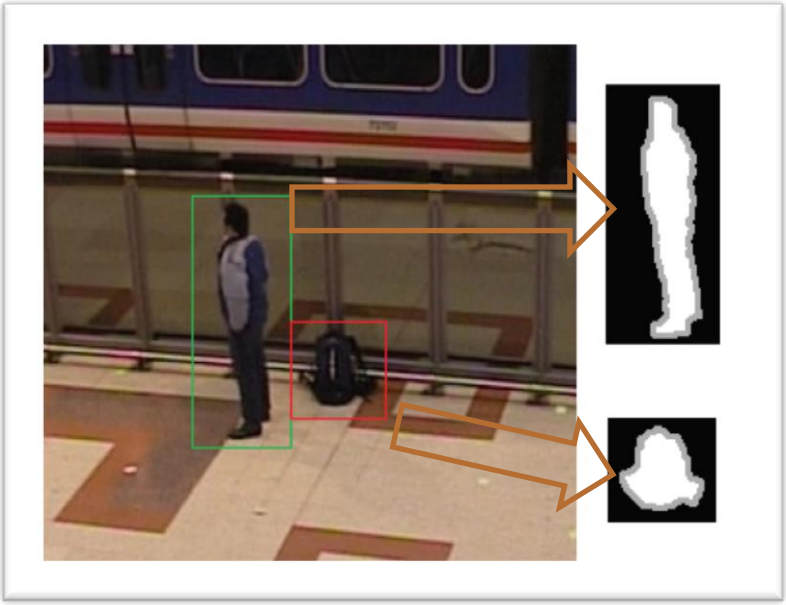


**Figura 5.6.** Diferencias morfológicas entre una persona y un objeto (Fuente: El autor).

Los métodos que se implementaron para la detección de objetos abandonas y robados fueron las aproximaciones basadas en la energía del contorno (sección 2.8.1) y el de histograma del color (sesión 2.8.2).

Para determinar los parámetros óptimos del algoritmo de contorno activo se intentaron diferentes combinaciones de los parámetros $\alpha$, $\beta$ y $\lambda$ en el intervalo de 0 a 5. Los valores óptimos que se determinaron fueron $\alpha$ = 0,83, $\beta$ =1,5 y $\lambda$ = 0.65.

Para determinar la media y la varianza del algoritmo que detecta objetos abandonados y robados mediante el histograma de color, se capturaron los frames correspondientes al evento y se determinaron los parámetros, la tabla 5.6 muestra los parámetros determinados por secuencia de video.

**Tabla 5.6.** Valores de $\mu$ y $\sigma^2$ del detector basado en el histograma de color.

| Secuencia | Media ($\mu$) | Varianza ($\sigma^2$) |
|---|---|---|
| PETS2006_S2-T3-C | 0,42 | 0,15 |
| PETS2006_S3-T7-A | 0.39 | 0,09 |
| PETS2006_S4-T5-A | 0.45 | 0,08 |
| PETS2006_S5-T1-G | 0.53 | 0,17 |
| PETS2006_S6-T3-H | 0,46 | 0,16 |
| PETS2006_S7-T6-B | 0,68 | 0,04 |
| AVSS2007_EASY | 0,72 | 0,54 |
| AVSS2007_MEDIUM | 0,81 | 0,32 |
| AVSS2007_HARD | 0,80 | 0,98 |

### 5.2.4. Resultados

Como se puede observar en la tabla 5.7 los algoritmos que trabajan con contornos activos y con histogramas de color tienen resultados prácticamente idénticos. Lo anterior se debe a que, es el algoritmo de detección de regiones estáticas quien determina el tiempo inicial. Hay ciertas circunstancias donde el sistema falla, esto se debe a fallas en la detección de regiones estáticas, clasificación de personas/objetos y problemas con fondos complejos.

Como se puede observar en la secuencia PETS2006_S3-T7-A, el ground truth nunca arroja una alarma de abandono, en la secuencia de video la persona que lleva el objeto nunca se aleja a más de un metro del mismo, por lo tanto no se considera abandono, sin embargo los algoritmos detectan el evento, esto se debe a que los algoritmos evaluados no tienen en cuenta la distancia del sujeto que deja el objeto desatendido.

En la secuencia PETS2006_S6-T3-H, ninguno de los algoritmos detectan en el evento de abandono o robo, esto se debe a que las características morfológicas evaluados del objeto abandonado son similares a las de una persona, por lo tanto el clasificador determina que ese objeto es una persona, por lo cual no se puede determinar como objeto abandonado.

**Tabla 5.7.** Resultados experimentales de los detectores de abandono y robo de objetos.

| **Secuencia** | **Abandono** | | | **Lanzamiento** | | | **Finalización** | | |
|---|---|---|---|---|---|---|---|---|---|
| | GT | 2.8.1 | 2.8.2 | GT | 2.8.1 | 2.8.2 | GT | 2.8.1 | 2.8.2 |
| PETS2006_S2-T3-C | 00:51 | 00:53 | 00:53 | 01:21 | 01:23 | 01:23 | 01:32 | 01:32 | 01:32 |
| PETS2006_S3-T7-A | - | 00:36 | 00:36 | - | 00:46 | 00:46 | - | 00:53 | 00:53 |
| PETS2006_S4-T5-A | 01:01 | 01:03 | 01:03 | 01:31 | 01:33 | 01:33 | 02:02 | 02:02 | 02:02 |
| PETS2006_S5-T1-G | 01:06 | 01:07 | 01:07 | 01:36 | 01:37 | 01:37 | 02:16 | 02:16 | 02:16 |
| PETS2006_S6-T3-H | 00:27 | - | - | 00:57 | - | - | 02:16 | - | - |
| PETS2006_S7-T6-B | 00:26 | 00:26 | 00:26 | 00:56 | 00:56 | 00:56 | 01:52 | 01:52 | 01:52 |
| AVSS2007_EASY | 01:54 | 01:49 | 01:49 | 02:24 | 02:19 | 02:19 | 03:12 | 03:17 | 03:12 |
| AVSS2007_MED | 01:40 | 01:47 | 01:47 | 02:10 | 02:17 | 02:17 | 03:00 | 03:02 | 03:01 |
| AVSS2007_HARD | 01:40 | 01:48 | 01:48 | 02:10 | 02:18 | 02:18 | 03:08 | 03:11 | - |

## 5.3. Análisis del comportamiento final del sistema

En este apartado se realiza un análisis del comportamiento final del sistema, en donde tanto el cliente, como el servidor y la base de datos se ejecutaron en el mismo ordenador, cuyas características se describieron al principio de este capítulo.

Para validar el correcto funcionamiento del sistema se han evaluado los casos de uso en el sistema final. Los resultados de la evaluación fueron satisfactorios, permitiendo que el sistema gestione usuarios, cámaras, ROIs, mapas y políticas de almacenamiento. Debido a que solo existe un motor de analítica la gestión de los mismos no se pudo validar más allá de agregar, modificar o remover el mismo motor de analítica.

### 5.3.1. Capa de adquisición

Como se ven en la tabla 5.8 el módulo de adquisición puede capturar a los máximos frames por segundo admitidos por las cámaras cuando solo se está viendo el video de una cámara. Lo anterior aplica tanto para los formatos de compresión H.264 y JPEG.

**Tabla 5.8.** Taza de cuadros por segundo con una cámara conectada.

| | **320x240** | **640x480** | **1280x720** |
|---|---|---|---|
| **Logitech C920** | 30 fps | 30 fps | 30 fps |
| **PlayStation Toy** | 120 fps | 90 fps | - |

Si se conectan las dos cámaras al tiempo la velocidad de captura de las imágenes se ve reducida, esto se debe a que el sistema debe procesar otra cámara conectada y como se observa en la siguiente tabla los frames por segundo reducen significativamente si la cámara C920 se configura para capturar a 1280x720, debido a que el sistema debe trabajar con más datos entre mayor sea la resolución.

**Tabla 5.9.** Taza de cuadros por segundo con dos cámaras conectadas.

| | | **PlayStation Toy** | | | |
|---|---|---|---|---|---|
| | | **320x240** | | **640x480** | |
| | | PS Toy | C920 | PS Toy | C920 |
| **Logitech C920** | 320x240 | 120 fps | 30 fps | 90 fps | 30 fps |
| | 640x280 | 75 fps | 30 fps | 45 fps | 30 fps |
| | 1280x720 | 50 fps | 25 fps | 28 fps | 18 fps |

En definitiva el sistema tiene la capacidad de trabajar con dos cámaras sin problemas, ofreciendo un comportamiento correcto y permitiendo trabajar en tiempo real, que es el requisito para los sistemas de videovigilancia.

### 5.3.2. Capa de adquisición y negocio

El motor de analítica parametrizado para trabajar con el sistema fue el descrito en la sección 4.3, el cual se probó con las secuencias de video de PETS2006 y AVSS2007, ofreciendo el mismo resultado, lo cual es lógico debido a que la implementación de los algoritmos se hizo bajo los mismo parámetros configurados en Matlab®. Sin embargo con secuencias de video provenientes de las cámaras los algoritmos no funcionan tan bien, esto se debe a que la parametrización de los algoritmos es complicada y por ende el proceso de segmentación no es fácil de realizar, lo cual afecta la detección de objetos estáticos y por ende la detección de objetos abandonados y robados.

Por otro lado el procesamiento de las secuencias de video se da en tiempo real, con una frecuencia de imágenes por segundo de hasta 25 fps a una resolución de 640x480 píxeles y 30 fps a una resolución de 320x240 píxeles.

### 5.3.3. Capa de persistencia

La evaluación de este componente ha sido favorable, se ha logrado recuperar la información de almacenada en la base de datos de forma fluida. Sin embargo, y como se comentó al inicio de esta sección, la base de datos corre en la misma máquina en que se ejecutan el servidor y el cliente, por lo que para validar correctamente el funcionamiento del sistema se debe probar bajo una topología distribuida.

# *CAPÍTULO 6*

# *CONCLUSIONES Y TRABAJO FUTURO*

# 6. Conclusiones y Trabajo Futuro

El objetivo de este trabajo fue definir una arquitectura flexible y escalable para sistemas de videovigilancia inteligentes, el cual permitiera, como su primer detector de eventos, la detección de objetos abandonados y robados. Este capítulo presenta las conclusiones generadas después del desarrollo de este trabajo y los trabajos futuros que se pueden generar a partir del mismo.

## 6.1. Conclusiones

Desde los primeros desarrollos, los sistemas de videovigilancia tradicionales han sido diseñados para vigilar entornos determinados. Sin embargo, estos sistemas tienen muchas limitaciones para satisfacer las demandas de seguridad requeridas por la sociedad. Estas necesidades, junto con la llegada de las nuevas tecnologías y la reducción del precio del hardware dedicado a la seguridad, representan algunas de las principales razones para que la videovigilancia inteligente haya cobrado gran interés como tema de investigación.

Uno de los principales retos en este campo es proporcionar sistemas expertos de seguridad con la autonomía y la capacidad necesaria para comprender de forma automática los eventos, con el fin de mejorar la productividad y la eficacia en las tareas de vigilancia. En entornos complejos donde múltiples situaciones tienen lugar al mismo tiempo, los agentes humanos tienen que tratar con todos ellos, y se ven afectados por factores negativos tales como la fatiga generada después de un período prolongado de observación. Sin embargo, los sistemas expertos artificiales no tienen estas limitaciones debido a su propia naturaleza.

En la última década, diferentes sistemas de vigilancia, de segunda y tercera generación, se han desarrollado tanto en el campo comercial como el académico. En este último campo, se han propuesto varios enfoques novedosos y se han desarrollo prototipos que han servido de puntos de partida para desarrollos comerciales. La mayoría de los sistemas propuestos han sido diseñados para resolver problemas específicos en situaciones específicas. Estos sistemas pueden ser entendidos como un conjunto de piezas que actúan por separado para detectar eventos específicos en escenarios particulares, sin establecer relaciones entre ellos para obtener un análisis global.

A pesar de ello, estos sistemas proporcionan importantes ventajas sobre los sistemas tradicionales, pero dos objetivos deben ser alcanzados con el fin de mejorar la videovigilancia En primer lugar, el diseño de sistemas de vigilancia escalables que permiten la inclusión de nuevos módulos para aumentar la capacidad de análisis. En segundo lugar, la necesidad de que los sistemas permitan variar el comportamiento del sistema artificial en función de los requisitos del entorno vigilado y de los eventos que se desean supervisar.

Estos problemas fueron parte de las principales motivaciones para proponer una arquitectura flexible y escalable para sistemas de videovigilancia inteligente. En la sección

3.2 se analizó a profundidad los problemas que debe solventar un sistema de vigilancia inteligente. Para dar solución a estos problemas, se diseñó una arquitectura cliente-servidor basada en componentes reutilizables e independientes, los cuales se pueden combinar para formar estructuras destinadas a trabajar con eventos específicos: estos componentes, bautizados como *módulos de análisis*, representan a un algoritmo o técnica perteneciente a alguna de las etapas de análisis de video, desde el preprocesado hasta la interpretación de eventos. Cada módulo de análisis puede tener *n* instancias diferentes, las cuales se parametrizan para trabajar con entornos determinados. Para permitir el fácil desarrollo de nuevas técnicas de análisis se proporcionaron interfaces que definan el comportamiento que deben tener los nuevos algoritmos para que puedan trabajar con el sistema, por lo tanto la integración de módulos de análisis en el sistema de videovigilancia no implica la modificación de los módulos ya implementados, lo cual incrementa la capacidad de análisis del sistema artificial.

Los módulos de análisis se combinan en unas estructuras denominas *motores de analítica*, cada motor de analítica está destinado a tratar con un evento determinado, el cual se define por el módulo de análisis implementado en la detección de eventos; según se implemente un módulo u otro, se podría ejecutar otro proceso de análisis o se podría trabajar mejor en entornos especificados. La capacidad de poder adaptar los motores a diferentes entornos o ampliar su procesamiento mediante la inclusión de nuevos módulos de analítica incrementa la flexibilidad del sistema.

Cada motor de analítica genera información contextual en un archivo XML, el cual y junto con los frames provenientes de la capa de análisis se procesa por un *motor de reglas*. Los motores de reglas proporcionan mayor versatilidad al sistema, dado que permiten definir nuevos eventos, como por ejemplo analizar la dirección de movimiento de un objeto y disparar una alarma si ésta va en una dirección especificada por el motor de reglas.

Mediante los módulos de análisis, motores de analítica y motores de reglas la arquitectura propuesta proporciona una alta flexibilidad lo que permite escalar el sistema para vigilar nuevos entornos y variar el comportamiento del sistema en función de las áreas vigiladas. Además, al diseñarse la arquitectura bajo un modelo cliente-servidor posibilita que el sistema pueda ser usado por grandes y pequeñas organizaciones.

Paralelo al diseño de la arquitectura se realizó un exhaustivo estudio del estado del arte, para realizar una clasificación y selección de las técnicas más destacadas en la literatura de las diferentes etapas de análisis. Lo anterior con la finalidad de implementar los métodos seleccionados en el sistema final.

Se determinó que las etapas más críticas eran las de modelado de fondo y detección de objetos, por lo que se implementaron los métodos más relevantes y se realizó una comparativa entre estos. Esta evaluación se hizo mediante el dataset CVPR2012, en donde las secuencias de video se dividieron en los escenarios básicos y fondos dinámicos, en donde se determinó que los algoritmos *Median* y *MoG* presentaron los mejores resultados

tanto en la categoría básica como de fondos dinámicos, igualmente se pudo observar que todos los algoritmos tuvieron problemas al modelar la ondulación de las hojas de los árboles el movimiento aleatorio de los chorros de agua de las fuentes.

Se evaluó las dos aproximaciones más destacadas en la detección de objetos abandonados y robados, se escogieron unos algoritmos sencillos para la detección de objetos estáticos y clasificación de personas que según la literatura han tenido buen comportamiento. Se utilizaron los datasets PETS2006 y AVSS2007, en donde las secuencias de video de PETS2006 se categorizaron como simples mientras que los videos de AVSS2007 como complejos, los resultados experimentales de la evaluación arrojaron resultados muy similares, se determinó que los errores generalmente provenían de una mala clasificación entre personas y objetos, una mala detección de regiones estáticas o que los algoritmos evaluados no contemplaban la información sobre el responsable del objeto abandonado.

## 6.2. Trabajo futuro

Este trabajo abre la posibilidad para múltiples trabajos futuros en diferentes áreas, a corto plazo se espera el desarrollo de nuevos módulos de análisis que incrementen la capacidad de detectar nuevos eventos por parte del sistema, algunos ejemplos son:

- Detección de personas merodeando alrededor de un área determinada.
- Conteo de peatones.
- Detección de multitudes.
- Análisis de trayectorias.
- Detección de caídas o desmayos.
- Reconocimiento de rostro.
- Reconocimiento de placas.
- Detección de humo o fuego.
- Clasificación y reconocimiento de objetos.
- Seguimiento de objetos con cámaras PTZ.

Aparte de los módulos de análisis se pueden incluir otras funcionalidades como:

- Incluir módulos de gestión automática de alarmas en la capa de negocio, que permitan acciones como llamar a la policía, enviar mensajes SMS, activar alarmas, etc.

- Generar descriptores semánticos MPEG-4 para habilitar búsquedas inteligentes de los videos.

- Uso de la GPU para el procesar los algoritmos de análisis.

- Habilitar una interfaz WEB para acceder remotamente al sistema.

- Desarrollar un sistema de entrenamiento en caliente que permita al sistema parametrizar de manera fácil y en caliente los diferentes algoritmos.

La detección de objetos abandonados y robados se puede mejorar incluyendo información concerniente a las personas que abandonan o retiran un objeto.

# REFERENCIAS

[1] A.B. Gómez, "Detección de objetos abandonados/robados en secuencias de video-seguridad", Tesis de pregrado, Universidad Autónoma de Madrid, 2009.

[2] A. Elgammal, D. Harwood, y L. Davis, "Non-parametric model for background subtraction", In Proceedings of the 6th European Conference on Computer Vision-Part II, pp. 751-767, 2000.

[3] A.J. Lipton,"Local application of optic flow to analyse rigid versus non-rigid motion", Proc. Int. Conf. Computer Vision Workshop Frame-Rate Vision, Corfu, Grecia, 1999.

[4] A.J. Lipton, H. Fujiyoshi, y R.S. Patil, "Moving target classification and tracking from real-time video", Proc. of the IEEE Workshop Applications of Computer Vision, pp. 8-14, 1998.

[5] A.R. Dick y M.J. Brooks, "Issues in automated visual surveillance", Proc. VIIth Digital Image, pp. 195-204, 2003.

[6] A. Yilmaz, O. Javed y M. Shah, "Object tracking: A survey", ACM Computing Surveys (CSUR), vol .38, no. 4, pp. 13.1-31.45, 2006.

[7] Bayona A., San Miguel J., Martínez J., Comparative evaluation of stationary foreground object detection algorithms based on background subtraction techniques, Proc. of AVSS, pp. 1-2, 2007.

[8] Beynon, M. "Detecting abandoned packages in a multi-camera vídeo surveillance system", Proc. of AVSS, pp. 221–228, 2003.

[9] Bhargava, M.; Chen, C-C. Ryoo, M. S.; Aggarwal, J. K. "Detection of abandoned objects in crowded environments", Proc. of AVSS, pp. 271-276, 2007.

[10] Blauensteiner P, Kampel M. Visual Surveillance of an Airports Apron-An Overview of the AVITRACK Project. Digital Imaging in Media and Education, Annual Workshop of AAPR, 2004.

[11] Bloisi D, Iocchi L, Remagnino P, Monekosso N.D., ARGOS–A Video Surveillance System for Boat Traffic Monitoring In Venice. To appear in International Journal of Pattern, Recognition and Artificial Intelligence (IJPRAI), 2009.

[12] Brodsky, T., Cohen, R., Cohen-Solal, E., Gutta, S., Lyons, D., Philomin, V., and rajkovic, M.: 'Visual surveillance in retail stores and in the home'. In: 'Advanced Video-based Surveillance Systems' (Kluwer Academic Publishers, Boston, Chapter 4, pp. 50–61, 2001.

[13] Cheng, S.; Xingzhi Luo; Bhandarkar, S.M. “A Multiscale parametric background Model for Stationary Foreground Object Detection”, Proc. of Motion and Video Computing, pp.8, 2007.

[14] C. Papageorgiou, T. Evgeniou, y T. Poggio, “A trainable pedestrian detection system”, Proc. of IEEE Int. Conf. on Intelligent Vehicles, pp. 241-246, Germany, 1998.

[15] C. Stauffer y E. Grimson, “Learning patterns of activity using real time tracking”, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 747-757, 2000.

[16] Cucchiara, R., Grana, C., Piccardi, M., and Prati, A.: 'Detecting objects, shadows and ghosts in video streams by exploiting color and motion information', in Proceedings of the IEEE Int'l Conference on Image Analysis and Processing, to appear, 2001.

[17] Deng-Yuan Huang Wu-Chih Hu and Wei-Hao Chen. Adaptive wide eld-of-view surveillance based on an ip camera on a rotational platform for automatic detection of abandoned and removed objects. In ICIC Express Letters, volume 1, pp. 45-50, 2010.

[18] F. Nilsson, "Intelligent Network Video, Understanding Modern Video Surveillance Systems", CRC Press, Taylor & Francis Group, ISBN: 13: 978-1-4200-6156-7, 2009.

[19] F. Fusier, V. Valentin, F. Brémond, M. Thonnat, M. Borg, D. Thirde, y J. Ferryman, “Video understanding for complex activity recognition”, Machine Vision and Applications, vol. 18, no. 3, pp. 167-188, 2007.

[20] F. Bremond, “Scene Understanding: perception, multi-sensor fusion, spatio-temporal reasoning and activity recognition”, Tesis doctoral, University of Nice Sophia Antipolis, 2007.

[21] Gevers, T., and Stokman, H.: 'Classifying color edges in video into shadow-geometry, highlight, or material transitions', IEEE Trans on multimedia 5, pp., 237-243, 2007.

[22] Guler, S.; Farrow, K. “Abandoned Object detection in crowded places”, Proc. Of PETS, pp. 18-23, 2006.

[23] Guler, S.; Silverstein, J.A.; Pushee, I.H. “Stationary objects in multiple object tracking”, Proc. of AVSS, pp. 248-253, 2007.

[24] Herodotou, N., Plataniotis, K.N., and Venetsanopoulos, A.N.: “A color segmentation scheme for object-based video coding”, en Proceedings of the IEEE Symposium on Advances in Digital Filtering and Signal Processing, pp. 25-29, 1998.

[25] Hui Kong, J.-Y. Audibert, and J. Ponce, “Detecting abandoned objects with a moving camera. Image Processing”, IEEE Transactions, vol. 19, pp. 2201-2210, 2010.
[26] Huwer, S.; Niemann, H. “Adaptive Change detection for Real-Time Surveillance Applications”, Proc. of Visual Surveillance, pp. 37-46, 2000.

[27] I. Haritaoglu, D. Harwood, y L.S. Davis, “W4: real-time surveillance of people and their activities”, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 809-830, 2000.

[28] I. Huerta, “Foreground Object Segmentation and Shadow Detection for Video Sequences in Uncontrolled Environments”, Tesis PhD, Universidad Autónoma de Barcelona, 2010.

[29] I.S. Kim, H.S. Choi, K.M. Yi, J.Y. Choi, y S.G. Kong, “Intelligent Visual Surveillance - A Survey”, Kong International Journal of Control, Automation, and Systems, vol. 8, no. 5, pp. 926-939, 2010.

[30] J.A. Albusac, “Vigilancia Inteligente: Modelado de Entornos Reales e Interpretación de Conductas para la Seguridad”, Tesis de Máster, Universidad de Castilla-La Mancha, 2008.

[31] J. Barron, D. Fleet, y S. Beauchemin, “Performance of optical flow techniques”, International Journal of Computer Vision, vol. 12, no. 1, pp. 42-77, 1994.

[32] J. Connell, A. W. Senior, A. Hampapur, Y.-L. Tian, L. Brown, and S. Pankanti, “Detection and Tracking in the IBM PeopleVision System,” in Proceedings of Int’l Conference on Multimedia and Expo, 2004, vol. 2, pp. 1403–1406.

[33] J.C. San Miguel, J. Bescos, J.M Martinez, y A. Garcia, “DiVA: a Distributed Video Analysis framework applied to video-surveillance systems”, Proceedings of the International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS’2008, pp. 207-210, 2008.

[34] J.C. San Miguel y J.M. Martínez, “Robust unattended and stolen object detection by fusing simple algorithms”, AVSS '08 Proceedings of the 2008 IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance, 2008.

[35] J. Hogan, “Your every move will be analysed”, New Scientist, 2003

[36] Jing Ying Chang, Huei-Hung Liao y Liang-Gee Chen. “Localized detection of abandoned luggage”. EURASIP Journal on Advances in Signal Processing, 2010.

[37] J.M. Ferryman, S.J. Maybank, y A.D. Worrall, “Visual Surveillance for Moving Vehicles”, International Journal of Computer Vision, vol .37, no. 2, pp. 187-197, 2000.

[38] J. Shen, “Motion detection in color image sequence and shadow elimination”, Visual Communications and Image Processing, vol. 5308, pp. 731-740, 2004.

[39] Khoudour D, Deparis J.P, Bruyelle J.L, Cabestaing F, Aubert D, Bouchafa S, Velastin S.A, Vincencio-Silva M.A y Wherett M, “Project Cromatica, Lecture Notes in Computer Science”, Springer, pp. 157–764, 1997.

[40] Liao,H. H., Chang,J. Y. y Chen, L. G., “A localized Approach to abandoned luggage detection with Foreground Mask sampling”, Proc. of AVSS, pp. 132-139, 2008.

[41] L. Wang, W. Hu, y T. Tan,“Recent developments in human motion analysis”, Pattern Recognition, vol. 36 no. 3, pp. 585-601, 2003.

[42] M.A. Patricio, J. Carbó, O. Pérez, J. García, y J.M. Molina, “Multi-Agent Framework in Visual Sensor Networks”, Multi-Agent Framework in Visual Sensor Networks, vol. 2007, no. 1, 2006.

[43] Martínez, J.; Herrero, J.; Orrite, C. “Automatic Left luggage Detection and Tracking using a Multi-camera UKF”, Proc. of PETS, pp 59-66, 2006.

[44] Mathew, R.; Yu, Z.; Zhang, J. “Detecting new stable objects in surveillance video”, Proc. of Multimedia Signal Processing, pp. 1-4, 2005.

[45] Medha Bhargava, Chia-Chih Chen, M. Ryoo, y J. Aggarwal., “Detection of object abandonment using temporal logic”, Machine Vision and Applications, pp. 271-281, 2009.

[46] Miezianko, R.; Pokrajac, D. “Detecting and Recognizing Abandoned Objects in Crowded Environments”, Proc. of Computer Vision System, pp. 241-250, 2008.

[47] Mikic, I., Cosman, P., Kogut, G., y Trivedi, M.: “Moving shadow and object detection in traffic scenes”, Proceedings of Int'l Conference on Pattern Recognition, 2000.

[48] M. Piccardi, “Background subtraction techniques: a review”, In Proc. of IEEE SMC 2004 International Conference on Systems, Man and Cybernetics, pp. 3099-3104, 2004.

[49] M. Saptharishi, J.B. Hampshire II, y P. Khosla, “Agent-based moving object correspondence using differential discriminative diagnosis”, Proc. of Computer Vision and Pattern Recognition, pp. 652-658, 2000.

[50] M. Valera, y S.A. Velastin, “Real-time architecture for a large distributed surveillance system”, IEE Intelligent Distributed Surveillance Systems, pp. 41-45, 2005.

[51] M. Valera, “An approach for designing a real-time intelligent distributed surveillance system”, Tesis de PhD, Kingston University, 2006.

[52] M. Valera, y S.A. Velastin, "Intelligent distributed surveillance systems: a review", IEE Proc. of Vision, Image, and Signal Processing, vol. 152, no. 2, pp. 192-204, 2005.

[53] P.L. Venetianer, Z. Zhang, W. Yin, y A.J. Lipton, "Stationary target detection using the object video surveillance system", Advanced Video and Signal Based Surveillance, pp. 242-247, 2007.

[54] Porikli, F. "Detection of temporarily static regions by processing video at different frame rates", Proc. of AVSS 2007, pp. 236-241, 2007.

[55] Porikli, F.; Ivanov, Y.; Haga, T. "Robust Abandoned Object Detection Using Dual Foregrounds", Journal on Advances in Signal Processing, art. 30, pp. 11, 2008.

[56] Prati, A., Mikic, I., Cucchiara, R., Trivedi, M.: "Comparative Evaluation of Moving Shadow Detection Algorithms", 2001.

[57] P. Spagnolo, T.D Orazio, M. Leo, y A. Distante, "Moving object segmentation by background subtraction and temporal analysis", Image and Vision Computing, vol. 24, no. 5, pp. 411-423, 2006.

[58] R. Alan Matchett. "CCTV for security professionals", Butterworth-Heinemann, ISBN 13: 9780750673037, Estados Unidos, 2003.

[59] R. Cutler y L.S. Davis, "Robust Real-Time Periodic Motion Detection, Analysis, and Applications", IEEE Computer Society, vol. 22, no. 8, pp. 781-796, 2000.

[60] R.T. Collins, A.J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, y L. Wixson, "A system for video surveillance and monitoring", Technical report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, 2000.

[61] S.C. Cheung y C. Kamath, "Robust techniques for background subtraction in urban traffic video", Video Communications and Image Processing, SPIE Electronic Imaging, pp. 881-892, 2004.

[62] S. Fejes y L.S. Davis, "Detection of independent motion using directional motion estimation", Computer Vision and Image Understanding, vol. 74 no.2 pp. 101-120, 1999.

[63] S. Ferrando, G. Gera, y C. Regazzoni, "Classification of Unattended and Stolen Objects in Video Surveillance System", IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS), 2006.

[64] Siebel N. T, y Maybank S., "The advisor visual surveillance system", Proceedings of the ECCV workshop Applications of Computer Vision, pp. 103–111, 2004.

[65] S.J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, y H. Wechsler. “Tracking groups of people”, Computer Vision and Image Understanding, vol. 80 no. 1, pp. 42-56, 2000.

[66] Stauffer C. y Grimson W. E. L., “Adaptive background mixture models for real-time tracking,” Proc. of CVPR 1999, vol. 2, pp. 2246-2252, 1999.

[67] T. Horprasert, D. Harwood, y L.S. Davies, “A robust background subtraction and shadow detection”, Proc. of Asian Conf. Computer Vision, pp. 8-11, 2000.

[68] T. Horprasert, D. Harwood, y L.S. Davis: “A statistical approach for real-time robust background substraction and shadow detection”, Proceedings of IEEE ICCV'99 FRAME-RATE Workshop, 1999.

[69] Tian, Y. “Robust and efficient foreground analysis for realtime video surveillance”, Proc. of CVPR 2005, pp. 1182-1187, 2005.

[70] T. Zhao y R. Ñevatia, “Tracking multiple humans in crowded environment”, Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, vol. 2, no. 27, pp. 406-413, 2004.

[71] Velastin S.A, Khoudour L, Lo B. y Sun J, Vicencio-Silva M. “PRISMATICA: a multisensor surveillance system for public transport networks”, 12th IEE International Conference on Road Transport Information and Control, pp. 19–25, 2004.

[72] W. Hu, T. Tan, L. Wang, y S. Maybank, “A survey on visual surveillance of object motion and behaviors”, IEEE Transactions on Systems, Man and Cybernetics, vol. 34, no. 3, pp. 334-352, 2004.

[73] Y. Dedeoglu, “Moving Object Detection, Tracking and Classification for Smart Video Surveillance”, Tesis de maestría, Bilkent University, 2004.

[74] Y. Tian, L. Brown, A. Hampapur, M. Lu, A. Senior, y C. Shu , “IBM Smart Surveillance System (S3): Event Based Video Surveillance System with an Open and Extensible Framework”, Machine Vision and Applications, Vol 19, no. 5-6, pp. 315– 17, 2006.

# ANEXOS

## Anexo A. Detalles de resultados experimentales

**Tabla A.1.** Resultado de Frame Difference en la categoría básica.

| Dataset | TP | FP | FN | TN | RC | PR | FS |
|---|---|---|---|---|---|---|---|
| PETS2006 | 765309 | 334113 | 4062881 | 366379191 | 0,16 | 0,70 | 0,26 |
| highway | 3021431 | 356139 | 2436558 | 86298330 | 0,55 | 0,89 | 0,68 |
| office | 385918 | 24592 | 8256176 | 116514094 | 0,04 | 0,94 | 0,09 |
| pedestrians | 341144 | 120331 | 329718 | 67448776 | 0,51 | 0,74 | 0,60 |

**Tabla A.2.** Resultado de Frame Difference en la categoría fondos dinámicos.

| Dataset | TP | FP | FN | TN | RC | PR | FS |
|---|---|---|---|---|---|---|---|
| boats | 89870 | 139297 | 1225910 | 243311040 | 0,07 | 0,39 | 0,12 |
| canoe | 270605 | 3222247 | 772764 | 25197474 | 0,26 | 0,08 | 0,12 |
| Fall | 9508505 | 42732081 | 8829492 | 974057826 | 0,52 | 0,18 | 0,27 |
| fountain01 | 38240 | 5081638 | 39263 | 88201417 | 0,49 | 0,01 | 0,01 |
| fountain02 | 162651 | 501924 | 104578 | 123460407 | 0,61 | 0,24 | 0,35 |
| overpass | 465745 | 142967 | 1512147 | 145499794 | 0,24 | 0,77 | 0,36 |

**Tabla A.3.** Resultado de Mean en la categoría básica.

| Dataset | TP | FP | FN | TN | RC | PR | FS |
|---|---|---|---|---|---|---|---|
| PETS2006 | 333312 | 169901 | 4494878 | 366543403 | 0,07 | 0,66 | 0,13 |
| highway | 1005417 | 177428 | 4452572 | 86477041 | 0,18 | 0,85 | 0,30 |
| office | 69731 | 5786 | 8572363 | 116532900 | 0,01 | 0,92 | 0,02 |
| pedestrians | 129717 | 37252 | 541145 | 67531855 | 0,19 | 0,78 | 0,31 |

**Tabla A.4.** Resultado de Mean en la categoría fondos dinámicos.

| Dataset | TP | FP | FN | TN | RC | PR | FS |
|---|---|---|---|---|---|---|---|
| boats | 14015 | 1965 | 1536836 | 245717654 | 0,01 | 0,88 | 0,02 |
| canoe | 33022 | 5553 | 1010347 | 28414168 | 0,03 | 0,86 | 0,06 |
| fall | 5473876 | 4033302 | 12864121 | 1012756605 | 0,30 | 0,58 | 0,39 |
| fountain01 | 6766 | 496960 | 70737 | 92786095 | 0,09 | 0,01 | 0,02 |
| fountain02 | 53080 | 5116 | 214149 | 123957215 | 0,20 | 0,91 | 0,33 |
| overpass | 78052 | 4250 | 1899840 | 145638511 | 0,04 | 0,95 | 0,08 |

**Tabla A.5.** Resultado de Median en la categoría básica.

| Dataset | TP | FP | FN | TN | RC | PR | FS |
|---|---|---|---|---|---|---|---|
| PETS2006 | 3758649 | 366365 | 1069541 | 366346939 | 0,78 | 0,91 | 0,84 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| highway | 4846401 | 559097 | 611588 | 86095372 | 0,89 | 0,90 | 0,89 |
| office | 6486913 | 1090988 | 2155181 | 115447698 | 0,75 | 0,86 | 0,80 |
| pedestrians | 639842 | 18577 | 31020 | 67550530 | 0,95 | 0,97 | 0,96 |

**Tabla A.6.** Resultado de Median en la categoría fondos dinámicos.

| Dataset | TP | FP | FN | TN | RC | PR | FS |
|---|---|---|---|---|---|---|---|
| boats | 1266236 | 565248 | 284615 | 245154371 | 0,82 | 0,69 | 0,75 |
| canoe | 966838 | 185190 | 76531 | 28234531 | 0,93 | 0,84 | 0,88 |
| fall | 15525822 | 106999810 | 2812175 | 909790097 | 0,85 | 0,13 | 0,22 |
| fountain01 | 44090 | 1625945 | 33413 | 91657110 | 0,57 | 0,03 | 0,05 |
| fountain02 | 236267 | 160908 | 30962 | 123801423 | 0,88 | 0,59 | 0,71 |
| overpass | 1671446 | 407040 | 306446 | 145235721 | 0,85 | 0,80 | 0,82 |

**Tabla A.7.** Resultado de MoG en la categoría básica.

| Dataset | TP | FP | FN | TN | RC | PR | FS |
|---|---|---|---|---|---|---|---|
| PETS2006 | 4048966 | 645342 | 779224 | 366067962 | 0,84 | 0,86 | 0,85 |
| highway | 5113989 | 435814 | 344000 | 86218655 | 0,94 | 0,92 | 0,93 |
| office | 6426429 | 681493 | 2215665 | 115857193 | 0,74 | 0,90 | 0,82 |
| pedestrians | 651471 | 21730 | 19391 | 67547377 | 0,97 | 0,97 | 0,97 |

**Tabla A.8.** Resultado de MoG en la categoría fondos dinámicos.

| Dataset | TP | FP | FN | TN | RC | PR | FS |
|---|---|---|---|---|---|---|---|
| boats | 1160429 | 1292867 | 390422 | 244426752 | 0,75 | 0,47 | 0,58 |
| canoe | 835319 | 20960 | 208050 | 28398761 | 0,80 | 0,98 | 0,88 |
| fall | 16065163 | 39898765 | 2272834 | 976891142 | 0,88 | 0,29 | 0,43 |
| fountain01 | 57709 | 1937812 | 19794 | 91345243 | 0,74 | 0,03 | 0,06 |
| fountain02 | 243219 | 125339 | 24010 | 123836992 | 0,91 | 0,66 | 0,77 |
| overpass | 1608353 | 285738 | 369539 | 145357023 | 0,81 | 0,85 | 0,83 |

**Tabla A.9.** Resultado de KDE en la categoría básica.

| Dataset | TP | FP | FN | TN | RC | PR | FS |
|---|---|---|---|---|---|---|---|
| PETS2006 | 4789613 | 1407307 | 38577 | 365305997 | 0,99 | 0,77 | 0,87 |
| highway | 4775833 | 823723 | 682156 | 85830746 | 0,88 | 0,85 | 0,86 |
| office | 7462695 | 3037150 | 1179399 | 113501536 | 0,86 | 0,71 | 0,78 |
| pedestrians | 608338 | 52974 | 62524 | 67516133 | 0,91 | 0,92 | 0,91 |

**Tabla A.10.** Resultado de KDE en la categoría fondos dinámicos.

| Dataset | TP | FP | FN | TN | RC | PR | FS |
|---|---|---|---|---|---|---|---|
| boats | 561944 | 727571 | 988907 | 244992048 | 0,36 | 0,44 | 0,40 |
| canoe | 1020207 | 2140809 | 23162 | 26278912 | 0,98 | 0,32 | 0,49 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| fall | 16153154 | 65131355 | 2184843 | 951658552 | 0,88 | 0,20 | 0,32 |
| fountain01 | 61610 | 118729 | 15893 | 93164326 | 0,79 | 0,34 | 0,48 |
| fountain02 | 250259 | 265608 | 16970 | 123696723 | 0,94 | 0,49 | 0,64 |
| overpass | 1946714 | 2631103 | 31178 | 143011658 | 0,98 | 0,43 | 0,59 |

**Tabla A.11.** Resultado de Eigenbackgrounds en la categoría básica.

| Dataset | TP | FP | FN | TN | RC | PR | FS |
|---|---|---|---|---|---|---|---|
| PETS2006 | 4627572 | 1695860 | 200618 | 365017444 | 0,96 | 0,73 | 0,83 |
| highway | 5375244 | 1390895 | 82745 | 85263574 | 0,98 | 0,79 | 0,88 |
| office | 8425781 | 21531583 | 216313 | 95007103 | 0,97 | 0,28 | 0,44 |
| pedestrians | 664599 | 1032997 | 6263 | 66536110 | 0,99 | 0,39 | 0,56 |
| **Total** | 19093196 | 25651335 | 505939 | 611824231 | 0,98 | 0,55 | 0,68 |

**Tabla A.12.** Resultado de Eigenbackgrounds en la categoría fondos dinámicos.

| Dataset | TP | FP | FN | TN | RC | PR | FS |
|---|---|---|---|---|---|---|---|
| boats | 1160429 | 1292867 | 390422 | 244426752 | 0,75 | 0,47 | 0,58 |
| canoe | 835319 | 20960 | 208050 | 28398761 | 0,80 | 0,98 | 0,88 |
| fall | 16065163 | 39898765 | 2272834 | 976891142 | 0,88 | 0,29 | 0,43 |
| fountain01 | 57709 | 1937812 | 19794 | 91345243 | 0,74 | 0,03 | 0,06 |
| fountain02 | 243219 | 125339 | 24010 | 123836992 | 0,91 | 0,66 | 0,77 |
| overpass | 1608353 | 285738 | 369539 | 145357023 | 0,81 | 0,85 | 0,83 |

## Anexo B. Frames de ejemplos de resultados de sustracción de fondo

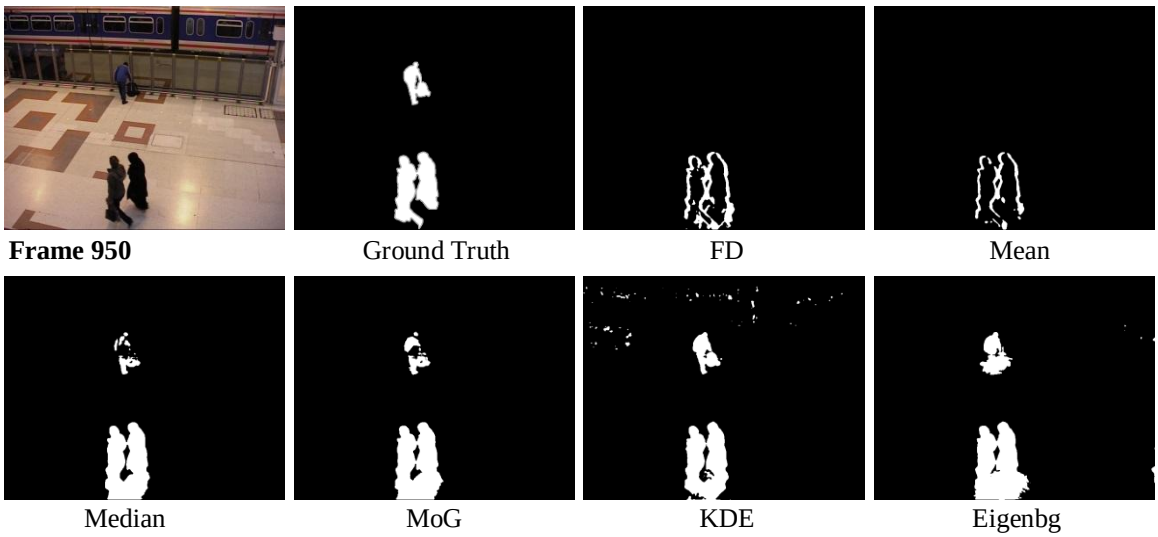



**Figura A.1.** Resultado de los diferentes algoritmos de sustracción de fondo en la secuencia de video *PETS2006* (Fuente: El autor).

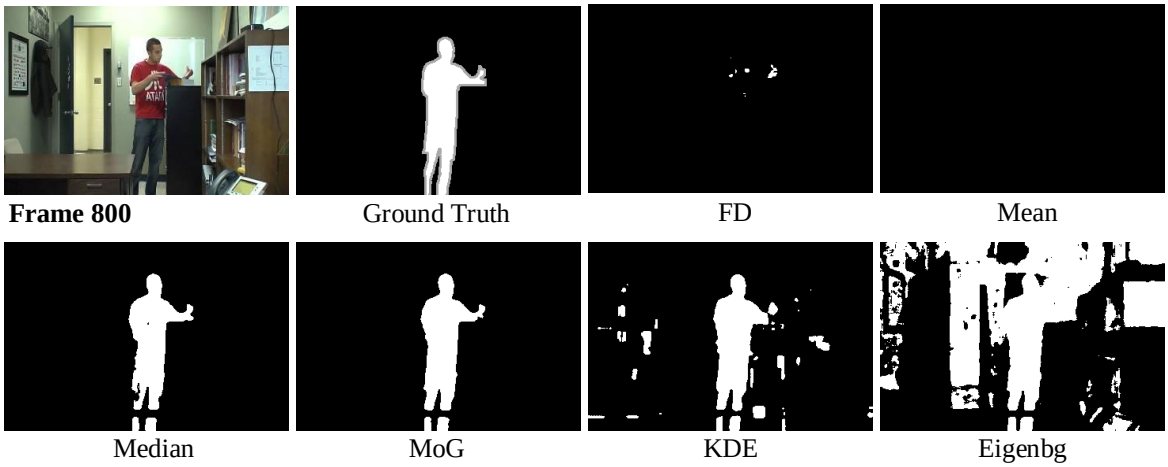



**Figura A.2.** Resultado de los diferentes algoritmos de sustracción de fondo en la secuencia de video *office* (Fuente: El autor).

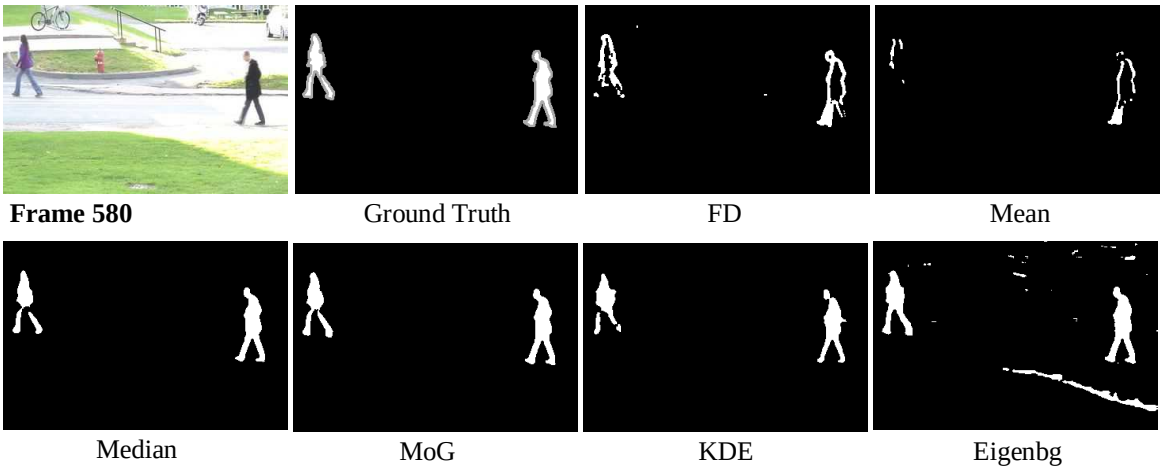



**Figura A.3.** Resultado de los diferentes algoritmos de sustracción de fondo en la secuencia de video *pedestrian* (Fuente: El autor).

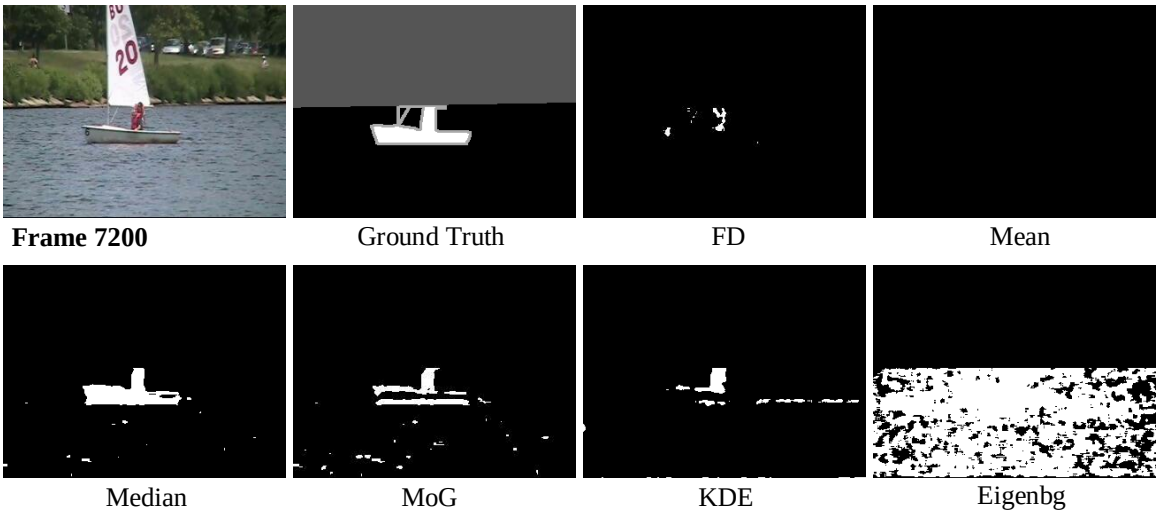



**Figura A.4.** Resultado de los diferentes algoritmos de sustracción de fondo en la secuencia de video *boats* (Fuente: El autor).

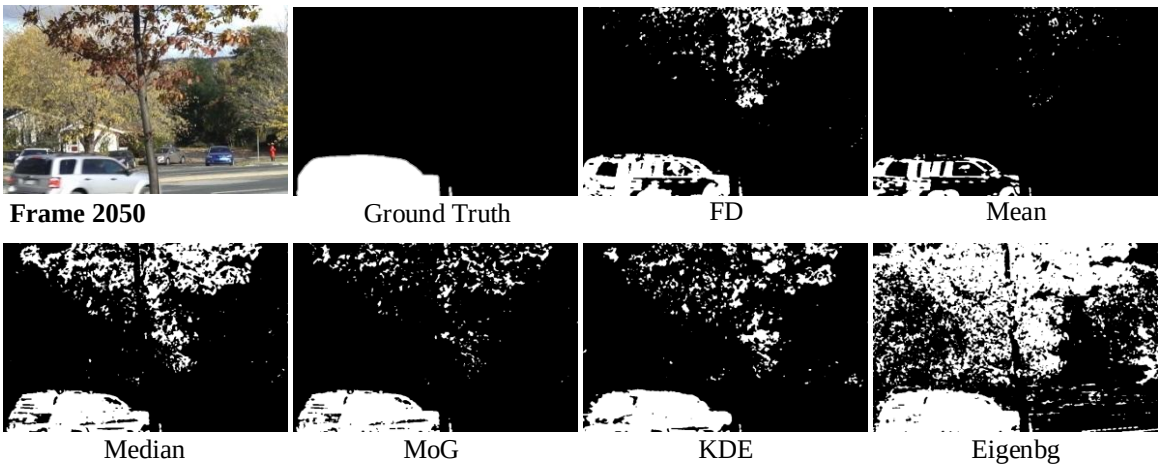



**Figura A.5.** Resultado de los diferentes algoritmos de sustracción de fondo en la secuencia de video *fall* (Fuente: El autor).

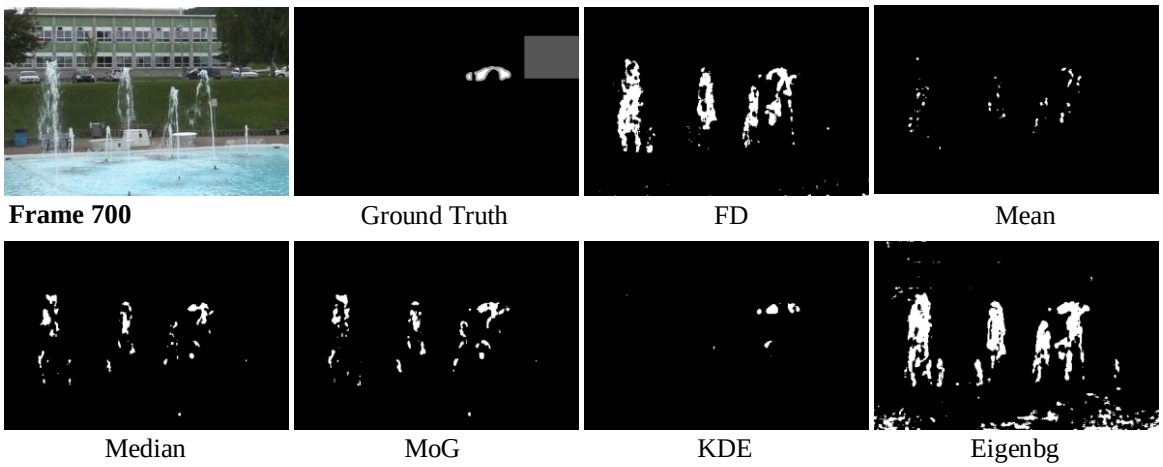



**Figura A.6.** Resultado de los diferentes algoritmos de sustracción de fondo en la secuencia de video *fountain01* (Fuente: El autor).

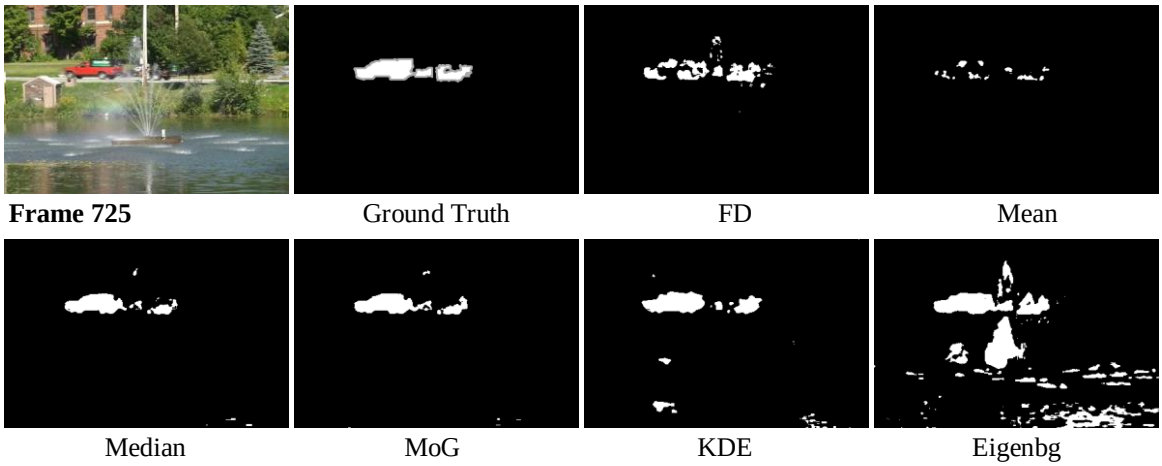



**Figura A.7.** Resultado de los diferentes algoritmos de sustracción de fondo en la secuencia de video *fountain02* (Fuente: El autor).

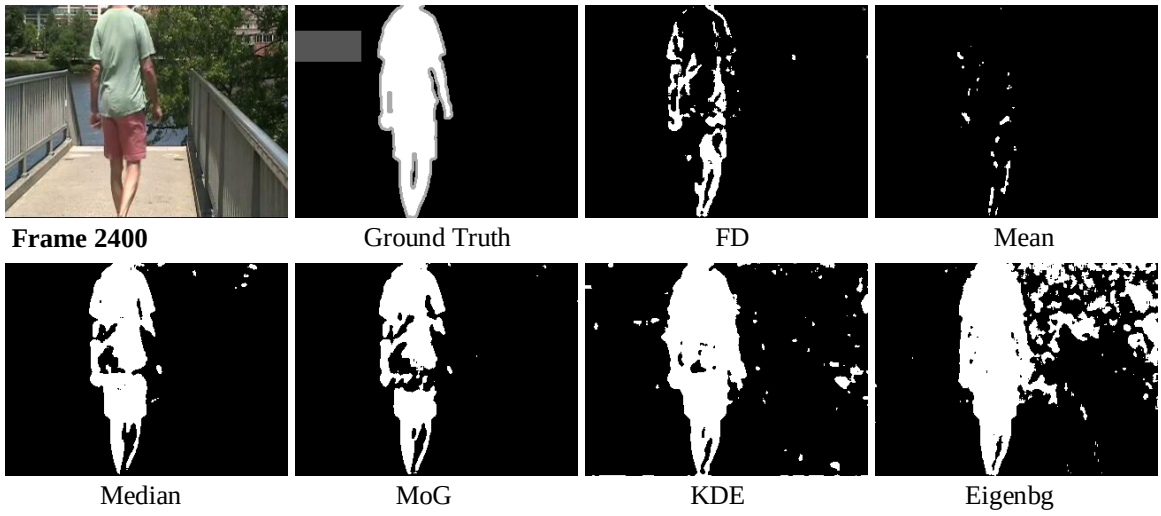



**Figura A.8.** Resultado de los diferentes algoritmos de sustracción de fondo en la secuencia de video *overpass* (Fuente: El autor).