UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

TRABAJO FIN DE GRADO

DETECCIÓN DE INTRUSIÓN EN EXTERIORES EN TIEMPO REAL

Alejandro Blanco Carrasco

Julio 2014

Detección de intrusión en exteriores en tiempo real

AUTOR: Alejandro Blanco Carrasco

TUTOR: Marcos Escudero Viñolo

PONENTE: Jesús Bescós Cano

Video Processing and UnderstandingLab

Dpto. de Tecnología Electrónica y de las Comunicaciones

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Julio de 2014

Trabajo parcialmente financiado por el gobierno español bajo el proyecto TEC2011-25995 (EventVideo)

PALABRAS CLAVE

Segmentación frente-fondo, fondo dinámico (multimodal), multicapa, componente conexa, modelado de fondo, modelado de frente, confianza, vecindario, sombras, sustracción de fondo.

RESUMEN

Este trabajo presenta un sistema de detección de intrusión en exteriores en tiempo real utilizando la técnica de substracción de fondo, basada en análisis a nivel de píxel.

Los objetivos marcados parten de un análisis del estado del arte actual para posteriormente ofrecer una combinación flexible de dichos métodos que integre diversas funcionalidades en un esquema escalable que permita la sustitución de unos módulos por otros sin cambiar la filosofía del algoritmo.

Podemos distinguir cuatro fases en los procesos de modelado de fondo: definición de la naturaleza del modelo, inicialización del modelo, actualización del modelo y detección de frente por comparación con el modelo.

Respecto a los modelos se propone el uso de un esquema no paramétrico que almacene las muestras de fondo y de frente mediante una aproximación multicapa.

El esquema de inicialización propuesto permite la inicialización local de áreas en los modelos en función de una preclasificación en clases de píxel, un proceso que puede entenderse como inicialización selectiva.

Para el proceso de actualización, el sistema utiliza un proceso guiado por las confianzas, o frecuencias de aparición de modos, que controlan la actualización selectiva mediante el uso de etiquetas o clases. Los modelos se adaptan temporalmente utilizando un proceso de actualización selectiva paramétrico que controla la evolución de los modelos. En cuanto a la fase de comparación con los modelos, se propone un método de comparación con el entorno espacial o vecindario que da robustez al ruido introducido por las vibraciones de captura. El sistema integra un proceso comparativo que aporta robustez frente a sombras y reflejos moderados.

El sistema ha sido evaluado mediante una base de datos pública, lo que permite una comparativa fiable con el estado del arte actual. Los resultados obtenidos no han sido sometidos a ninguna técnica de post-procesado que mejoren los estadísticos de clasificación.

El sistema ha sido desarrollado e implementado usando el lenguaje de programación orientada a objetos *C++* y la librería de análisis de vídeo *OpenCV*. Finalmente, se incluye una interfaz de usuario implementada mediante el programa *Qt Designer* que permita la interacción instantánea con el sistema, alterando sus parámetros si se considera necesario. El algoritmo se integra en un entorno de procesado común que permite su utilización con la salida directa de una cámara.

KEY WORDS

Background-foreground segmentation, dynamic background (multimode), multilayer, connect component, background modelling, foreground modelling, confidence, neighbourhood, shadows, background subtraction.

ABSTRACT

This Project presents an outdoor intrusion detection system in real time using a background subtraction technique based on a pixel-driven analysis.

The objectives are based on an analysis of the current state of the art to later provide a flexible combination of such methods to integrate several functionalities in a scalable scheme that allows the substitution of some of the modules without changing the philosophy of the algorithm.

The use of a non-parametric scheme that stores samples of the background and foreground using a multilayer approach is used for the models.

The proposed initialization scheme relies in a pixel classification stage to provide local-updating mechanisms under a selective spirit.

For the updating process, the system uses a process that is guided by the confidence matrix or frequency of occurrence of the modes, which control a selective updating process by using pixel's classes. The models are temporarily adapted using a parametric selective updating process that controls the evolution of the models.

In the stage of comparison against the models, we propose a method of comparison with the pixel spatial environment or pixel's neighborhood. Such method provides robustness to noise due to the vibrations of the capturing system. The system also integrates a comparative process that gives robustness to shadows and moderate reflections.

The system has been evaluated using a public database which allows a reliable comparative with the current state of the art. The results have not undergone any post-processing technique that improves the statistical classification.

The system has been developed and implemented using the object-oriented programming language *C++* and the video analysis library *OpenCV*. Finally, it is enhanced by a user interface implemented by *Qt Designer* program that allows instantaneous interaction with the system, altering its parameters if necessary. The algorithm is integrated into a common processing environment that allows its use with the direct output of a camera.

AGRADECIMIENTOS

Con este trabajo finaliza un capítulo del libro que es mi vida. Un capítulo muy duro y a la vez muy bonito, en el que aparecen grandes personas a las que tengo mucho que agradecer.

Quiero dar las gracias a mi tutor Marcos Escudero, por su apoyo, compromiso, paciencia, dedicación y confianza a lo largo de estos meses, gracias por enseñarme tanto, por tu compromiso y por tratarme casi como a un compañero, tengo que agradecer que tu labor como tutor ha sido excelente. Gracias.

También quiero expresar mi gratitud a José María Martínez y a Jesús Bescós por haberme dado la oportunidad de realizar este trabajo, el cual ha sido muy satisfactorio para mí. Gracias.

Quiero expresar mi agradecimiento en particular a los tres profesores mencionados anteriormente, junto a Luis Salgado, Daniel Ramos,… por haberme dedicado su tiempo personalmente y de buen grado para reforzar algunos conocimientos cuando lo he necesitado, y en general a todos los profesores que me han impartido clase durante la carrera, de los que tanto he aprendido. Gracias.

Agradecer al grupo de trabajo VPU Lab, en el que he realizado este trabajo, su amabilidad, el compañerismo y el buen clima de trabajo, y en especial a mi compañero Carlos Sánchez por haber estado ayudándome con las camaritas tantas mañanas y con tan buena cara. Gracias a todos.

Me gustaría también dar las gracias a todos esas personas, que primero fueron compañeros y acabaron siendo amigos, que he conocido a lo largo de estos años, sois tantos que merecéis un Anexo. Todos esos buenos momentos juntos nunca los olvidaré. En especial quiero agradecer a mi compañero Alberto Palero el haber estado conmigo estos años y haberte conocido.

Quiero agradecer también a mis amigos de Tres Cantos; Manuel, Adrián, Javier, Alejandro, Luis, Daniel, Óscar, Jorge, Pablo, Jesús, Mario, Fernando, Sergio, Esther, Silvia, Alba, Carlos, Gonzalo, Eloy, Juan Pablo, Cecilia,… Todos esos amigos que se han interesado por mis estudios y me han animado o felicitado a lo largo de la carrera. Gracias.

Tengo mucho que agradecer a mi novia Celia Pérez, que ha sido mi gran apoyo durante todos estos años. Gracias por estar siempre en los peores momentos y también en los buenos, gracias por tu paciencia, comprensión y por estar siempre ahí.

Por último quiero agradecer a mi familia, en especial a mis padres Jesús y Ángela y a mi hermana Cristina, por confiar siempre en mí y haber hecho de mí la persona que soy. Gracias por apoyarme en todas mis decisiones, y aconsejarme cuando pensáis que no son las más acertadas.

Este trabajo os lo dedico a todos vosotros, ya que todos y cada uno habéis aportado algo de vuestra parte a él, y sin vosotros no habría sido posible realizarlo. Estaré siempre agradecido.

Alejandro Blanco Carrasco

Julio 2014

I. GLOSARIO

Azimut	Ángulo que con el meridiano forma el círculo vertical que pasa por un punto de la esfera.
Frame	Imagen contenida en la secuencia de vídeo, también llamado cuadro de vídeo.
Hardware	Partes tangibles del sistema.
Matching	Correspondencia entre unidades de análisis en un mismo rango de valores determinado.
OpenCv	Librería con funciones de programación para tratamiento de vídeo.
Píxel	Menor unidad de superficie homogénea en que se divide una imagen.
Cámara PTZ	Cámara de uso en sistemas de video-vigilancia.
Qt Designer	Herramienta utilizada para el diseño de la Interfaz Gráfica.
Shifting	Desplazamiento de la imagen para comparar cada *píxel* con sus vecinos.

II. ACRÓNIMOS

AVI	Formato contenedor de audio y vídeo (Audio Video Interleave).
DiVA	Análisis de vídeo distribuido (Distributed Video Analysis).
EPS	Escuela Politécnica Superior (de la Universidad Autónoma de Madrid).
GUI	Interfaz gráfica (Graphical User Interface).
IP	Dirección para identificar un dispositivo conectado a la red (Internet Protocol).
PFC	Proyecto de Fin de Carrera.
RGB	Canales en una imagen en dicho formato: rojo, verde y azul (Red, Green and Blue).
RAM	Memoria de acceso aleatoria (Random Access Memory).
ROI	Regiones de interés (Regions of interest).
VPU	Grupo de investigación en el que se desarrolla el TFG (Video Processing Understanding Lab).

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ECUACIONES

CAPÍTULO 1: INTRODUCCIÓN

1.1 MOTIVACIÓN

La mayor parte de sistemas de detección de intrusión se basan en la substracción de fondo (*background substraction*), que se encarga de la discriminación entre fondo (*background*) y frente (*foreground*) siendo una etapa clave en sistemas de análisis de vídeo.

En el pasado el procesado de aplicaciones en tiempo real estaba muy limitado por las barreras tecnológicas. Los recientes avances en el desarrollo tecnológico han conseguido suprimir dichas limitaciones y este campo ha evolucionado a grandes pasos en los últimos años. Se han hecho muchos estudios en análisis de vídeo y en el capítulo dedicado al estado del arte en esta memoria se hará un seguimiento de alguno de estos métodos, analizándolos para obtener los más ventajosos de cara al desarrollo del algoritmo.

La motivación de este trabajo busca la creación de un sistema diseñado modularmente, de forma que cada etapa del sistema se encuentre en un módulo. De esta forma, en caso de querer sustituir un método por otro este paso sea sencillo generando un sistema adaptable. El procesado de cada módulo debe ser ligero para permitir que el sistema completo opere en tiempo real.

En entornos exteriores, los píxeles cambian de apariencia constantemente debido a cambios de iluminación y debido a la fuerza de la naturaleza (por ejemplo, el movimiento de las hojas en un árbol o del agua en un lago por el viento). Estas variaciones que se suceden en los píxeles se conocen como multimodalidad. El sistema que va a desarrollarse en este trabajo tiene que ser robusto frente a estos cambios debido a que una de sus aplicaciones es su uso en exteriores.

El diseño del sistema debe implementar métodos que consideren estas limitaciones pero que afecten lo menos posible al tiempo de proceso global del sistema.

1.2 OBJETIVOS

El objetivo global de este *TFG* es el diseño y desarrollo de un sistema de detección de objetos en movimiento (intrusos) en exteriores que opere en tiempo real.

Los objetivos específicos de este trabajo son suministrar herramientas para que el sistema sea robusto ante las limitaciones que se dan en entornos exteriores mientras se trabaja en tiempo real. Estas limitaciones son:

- *Cambios de iluminación producidos en la escena.*
- *Aparición de sombras y reflejos en la escena.*
- *Cambios en el fondo de la escena.*
- *Aparición de ruido introducido en la captura.*
- *Aparición de fondos multimodales en exteriores.*

Para lograr los retos que se plantean en este trabajo, se llevarán a cabo las siguientes tareas:

i. **Estudio del arte actual**: Se realizará un análisis del estado del arte de sistemas que utilizan substracción de fondo analizando algunos de los principales métodos existentes y considerando sus ventajas y limitaciones. Este análisis resulta en una comparativa que nos permite seleccionar los métodos que mejor se adaptan a los objetivos propuestos.

ii. **Diseño del sistema: selección e implementación de técnicas seleccionadas en el ámbito de sustracción de fondo**: Tras realizar la comparación entre métodos y conocer sus limitaciones, se seleccionarán algunas de las aproximaciones que, en nuestra opinión, resultan adecuadas para cumplir los objetivos que propone este trabajo.

iii. **Desarrollo e implementación del sistema**: Se desarrollará un sistema modular que contemple los métodos seleccionados y la forma de implementarlos creando el sistema.

iv. **Interacción con el sistema**: Se consideraran los parámetros configurables que afectan al sistema de forma que puedan modificarse para dar más funcionabilidad al sistema.

v. **Análisis de resultados y conclusiones**: Tras cotejarse los resultados del sistema con los de la base de datos *ChangeDetection* [37] se procede a un análisis de resultados para medir la bondad del sistema.

1.3 ESTRUCTURA DE LA MEMORIA

La memoria está dividida en los siguientes seis capítulos:

Capítulo 1: Introducción. Motivación, objetivos y estructura de la memoria.

Capítulo 2: Estado del arte. Segmentación, métodos, plataforma *DiVA y* diseñador *Qt*.

Capítulo 3: Diseño.

Capítulo 4: Desarrollo e implementación.

Capítulo 5: Pruebas y resultados.

Capítulo 6: Conclusiones, trabajo futuro y fases de desarrollo del *TFG*.

Capítulo 7: Referencias.

A lo largo de la memoria aparecerán una serie de palabras de uso común al ámbito del análisis de vídeo. Dichas palabras aparecerán en letra cursiva y además, tienen una breve descripción en el apartado Glosario o en el apartado Acrónimos, ambos anteriores al Índice.

CAPÍTULO 2: ESTADO DEL ARTE

2.1 SEGMENTACIÓN FONDO / FRENTE

La segmentación fondo/frente o *Background subtraction* (*BS*) es una técnica cuyo objetivo es detectar objetos en movimiento en secuencias de vídeo. *BS* ha sido utilizada en aplicaciones tales como sistemas de seguridad y video-vigilancia [1][2], indexación de contenidos [3] o compresión de vídeo [4]. Su objetivo es el de extraer las zonas relevantes (regiones de interés, *ROI*) de los cuadros de vídeo en cada instante temporal.

En más detalle, este proceso busca la discriminación entre los objetos (frente o *foreground*) en movimiento del cuadro de vídeo y lo que permanece estable en dicho cuadro (fondo o *background*) [5].

Para secuencias con cámara fija el método más común es el de substracción del modelo de fondo (*background model*, *BM*) que detecta objetos en movimiento mediante la diferencia entre el cuadro de vídeo y un cuadro de referencia (*BM).* Este cuadro debe ser una representación de la escena sin movimiento y debe actualizarse regularmente debido a los cambios de iluminación, al ruido introducido en el proceso de captación del vídeo y/o a cambios en la naturaleza del modelo de fondo.

Un buen sistema de *BS* debe tener un coste computacional eficiente y preferentemente debe operar en tiempo real por las razones expuestas en Toyama en [6], Elgammal en [7] o Harville en [8], y además debe proponer soluciones los siguientes problemas:

- *Adaptación a cambios de iluminación en la escena.*
- *Exclusión de sombras y reflejos.*
- *Obtención y actualización del fondo de la escena.*
- *Determinación de los parámetros de funcionamiento del algoritmo.*
- *Reducción del ruido introducido en la secuencia.*
- *Obtención de fondos multimodales.*
- *Solución a problemas causados por el camuflaje.*

Se definen fondos multimodales a los fondos que suelen darse en entornos exteriores debido a condiciones climáticas. Son leves movimientos que se deben evitar clasificar como frente. Por ejemplo, el movimiento de las hojas de los árboles o el agua brotando de una fuente.

Se define camuflaje al efecto que se da cuando un objeto de frente posee el mismo color que el fondo sobre el que se expone.

Con el fin de solucionar estos retos existen diversas aproximaciones de *BS* y la mayoría siguen un diagrama de flujo similar (ver Figura 2.1-1), al expuesto por Piccardi en [5] o Cheung en [11].

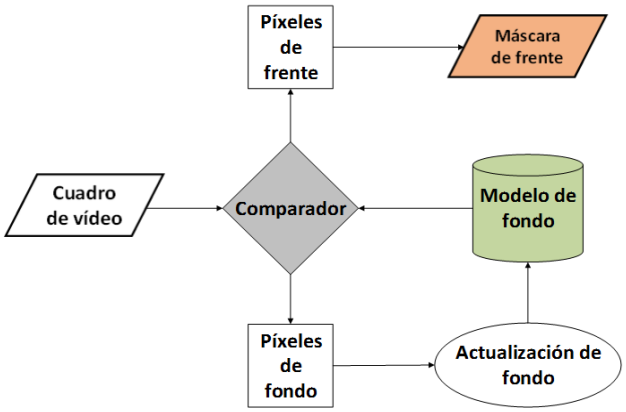


Figura 2.1-1: Esquema de substracción de fondo (BS)

Así pues, existen tres procesos que definen un sistema de *BS*: modelado, comparación y actualización. En los siguientes apartados se abordarán algunas de las técnicas más relevantes utilizadas en estos procesos.

Se define un modelo como la caracterización de una imagen basándose en la variación de los valores y la frecuencia de variación que van tomando los píxeles de la imagen con el tiempo.

En particular, Cristiani expone en [13] que existen 3 características para definir un modelo de fondo:

- **Modelo de representación**: Modelo matemático utilizado para representar el fondo (ver Sección 2.3).
- **Modelo de inicialización**: Forma de obtener el primer estado del modelo de fondo (ver Sección 2.4).
- **Modelo de adaptación**: Mecanismo para adaptarse a variaciones como cambios de iluminación (ver Sección 2.5).

2.2 INFLUENCIA DE LA MULTIMODALIDAD EN EL MODELADO

La función del modelado de fondo es la inicialización, actualización y representación de un modelo de fondo robusto de la secuencia de vídeo analizada [12].

Se define un píxel como $\vec{x} = (x, y)$ donde x es la posición del píxel en el eje horizontal del cuadro e y la posición vertical.

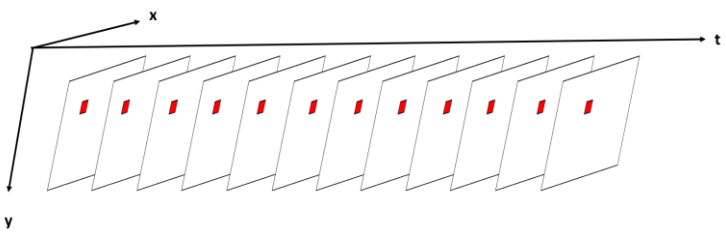


Figura 2.2-1: Evolución de un píxel en el tiempo

La primera apreciación al modelar un fondo es la necesidad de distinguir entre fondos controlados donde los píxeles de fondo que no cambian de valor (unimodales) y entre fondos en entornos variables (multimodales). Se define modo o moda el pequeño rango de valores que toma un píxel que no cambia de apariencia.

En secuencias a color, los cuadros de vídeo están representados por tres canales, cada uno de los cuales lleva una información determinada del valor de cada píxel en la imagen. Generalmente, el modelo de color más utilizado es el que descompone el color en sus componentes roja, verde y azul: *RGB*.

En la Figura 2.2-2 se muestra la evolución temporal de un píxel en los 400 primeros cuadros de vídeo de dos secuencias con canales rojo, verde y azul extraídas de la base de datos *ChangeDetection* [37]. Se ha elegido el píxel (19,142) de la secuencia '*office.avi*' como ejemplo de un píxel unimodal y el píxel (200,143) de la secuencia '*fountain01.avi*' como píxel multimodal.

Cuadro de vídeo (píxel marcado en rojo) **Evolución temporal**

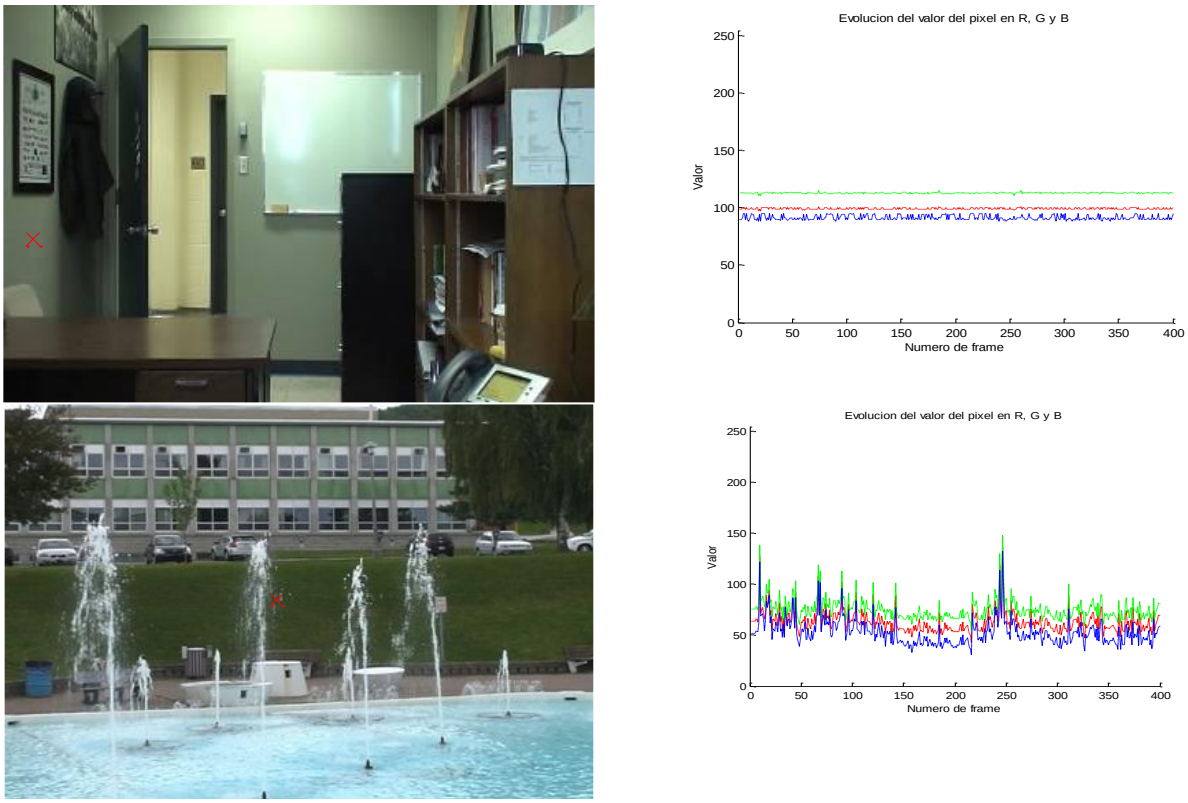


Figura 2.2-2: Comparación píxel unimodal (arriba) vs. multimodal (abajo)

En la gráfica de evolución se muestra la evolución de los valores que toman los canales de los píxeles: rojo, verde y azul, cada una representada por su color. En la gráfica superior se observa como apenas existe variación en los valores del píxel, ya que éste es estático. Sufre pequeñas variaciones en un rango pequeño causado por el ruido introducido en la secuencia. En la gráfica inferior, sin embargo, se observa la gran variación temporal de los valores debido al dinamismo de ese píxel producido por una escena multimodal.

Como se observa en la Figura 2.2-3 los valores de un modelo unimodal (gráficas superiores) presentan únicamente un pico o modo, que se mueve en un rango muy limitado de valores.

Por otro lado, un píxel de naturaleza multimodal (gráficas inferiores) presenta varios picos o modos, ya que sus píxeles sufren variaciones de valor en el tiempo, por lo que es importante utilizar un sistema de modelado robusto que contemple estas variaciones.

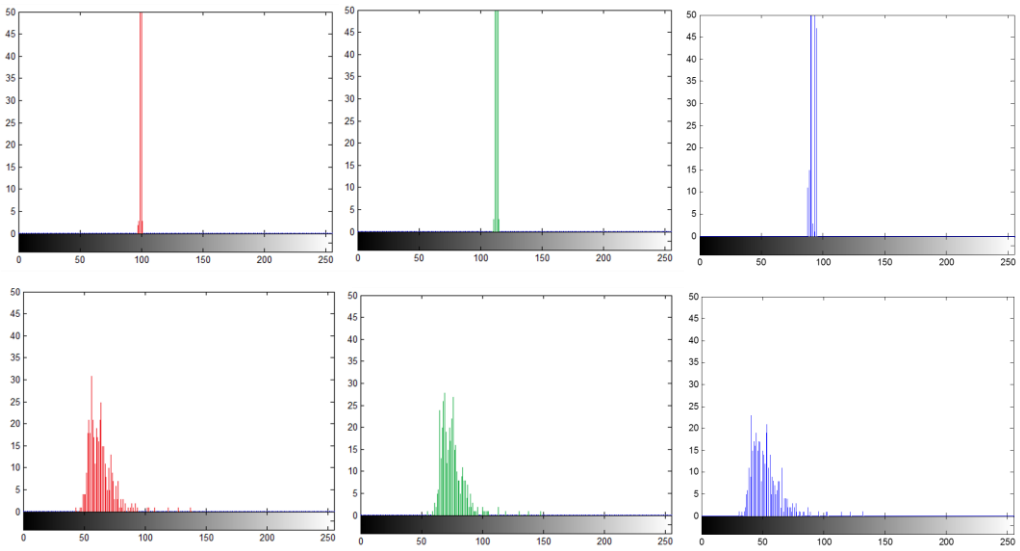


Figura 2.2-3: Histograma de canales RGB para un píxel en el tiempo: unimodal vs. multimodal

El principal problema en el modelado de los píxeles de naturaleza multimodal es que sobre las mismas posiciones espaciales que representan al píxel pueden aparecer modos de frente. Esta situación se ejemplifica a continuación. Para ello, se ha tomado como ejemplo el vídeo *'canoe.avi'* y se ha marcado el píxel (162,151) en rojo para seguir su evolución en los valores cuando $t \in (800,1199)$ (ver Figura 2.2-4). En la figura se muestran algunos de dichos cuadros (arriba) y los valores que toma cada canal *RGB*, de nuevo cada uno representado por su color. Es interesante ver como los valores que predominan mientras el píxel se encuentra sobre el agua son los del canal *B*, y en los instantes en que aparece el barco es el canal *R* el predominante. Al ser un píxel multimodal, puede apreciarse como la variación de cada canal es elevada debido al constante movimiento del agua.

Debido a las variaciones que se dan en algunos píxeles temporalmente, es necesaria una actualización de los modelos para que éstos puedan adaptarse a esos cambios y no disminuya la calidad del sistema.

Siguiendo la propuesta de Cristiani en [13] anteriormente mencionada y, en base a estas características, se van a estudiar distintas posibilidades para las diferentes etapas de diseño de los modelos:

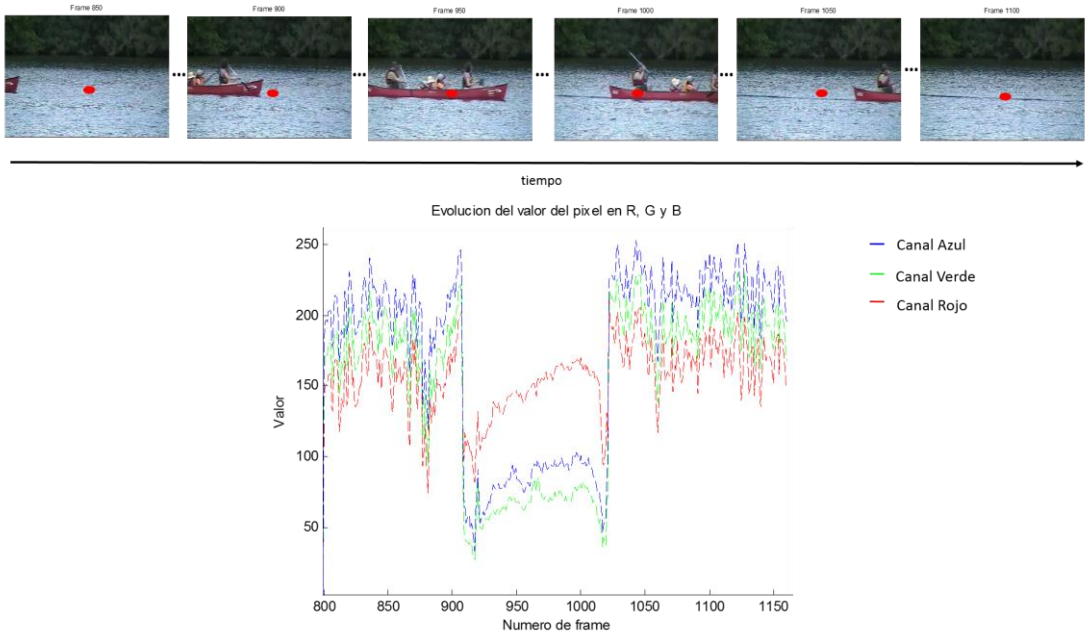


Figura 2.2-4: Evolución de un píxel en una secuencia

2.3 MODELO DE REPRESENTACIÓN

Aunque algunos métodos sencillos utilizan técnicas básicas como la diferencia de imágenes [14] o filtros promediados en el tiempo o de mediana [15], la mayoría de los métodos son algo más complejos y pueden dividirse en: modelos paramétricos y modelos no paramétricos. Estos diferentes métodos de representación están recogidos y analizados por Piccardi en [5].

2.3.1 Modelos paramétricos

Un modelo paramétrico es el modelo que define que el fondo se caracteriza por un conjunto de parámetros que describen la distribución de cada píxel. El tipo de distribución se asume como parte del modelo, aunque los parámetros sean variables.

Estos modelos permiten cierta tolerancia al ruido y a pequeñas variaciones debidas, por ejemplo, a la climatología. Históricamente, los modelos más utilizados utilizan distribuciones paramétricas basadas en Gaussianas simples o Mezclas de Gaussianas:

- **Gaussiana simple (‘*SG*’):**

Representa cada píxel modelando los cambios que ocurren en el modelo de fondo $BG(\vec{x};t)$ con una distribución unimodal Gaussiana definida por dos parámetros: media $\mu(\vec{x};t)$ y varianza $\sigma^2(\vec{x};t)$.

$$\mu(\vec{x};\mathrm{t}) = \sum_{i=1}^{i=t} \frac{BG(\vec{\mathrm{x}};\mathrm{i})}{t} \quad \sigma^2(\vec{x};\mathrm{t}) = \sum_{i=1}^{i=t} \frac{BG^2(\vec{\mathrm{x}};\mathrm{i})}{t} - \mu^2(\vec{x};\mathrm{t})$$

Ecuación 1: Media y varianza en Gaussiana simple

Se determina para cada instante t si un píxel pertenece al fondo $BG(\vec{\mathrm{x}};\mathrm{t})$ si el valor de dicho píxel se encuentra dentro de la Gaussiana definida para ese píxel, es decir, si la diferencia entre el píxel y la media $\mu(\vec{\mathrm{x}};\mathrm{t})$ es inferior a la desviación $\sigma(\vec{\mathrm{x}};\mathrm{t})$.

Para imágenes de 3 canales puede utilizarse un modelo de Gaussiana Simple para cada canal como el propuesto por Gordon en [16], que genera para cada canal de un píxel una distribución Gaussiana de matriz de covarianza diagonal.

- **Mezcla de Gaussianas (‘*MoG’*):**

Existen escenarios multimodales donde los valores de los píxeles varían en torno a un rango finito de valores característicos. Esto sucede en entornos en los que, por ejemplo, aparecen fuentes, hojas de los árboles en movimiento por el viento, etc.

Por este motivo, un píxel no puede modelarse a partir de un valor (una media $\mu(\vec{\mathrm{x}};\mathrm{t})$) y un conjunto en torno a este valor (una desviación $\sigma(\vec{\mathrm{x}};\mathrm{t})$) utilizando *SG*. Stauffer en [9] y [17] introduce un método conocido como Mezcla de Gaussianas (*MoG)* que consiste en modelar la intensidad de los píxeles con una mezcla de k distribuciones Gaussianas, siendo k el valor que determina la multimodalidad del escenario. Los parámetros que definen la *MoG* son tres: media $\mu_k(\vec{\mathrm{x}};\mathrm{t})$, varianza $\sigma_k^2(\vec{\mathrm{x}};\mathrm{t})$ y peso $w_k(\vec{\mathrm{x}};\mathrm{t})$.

$$P(BG(\vec{x};t)) = \sum_{j=1}^{j=k} w_j(\vec{\mathrm{x}};\mathrm{t}) N(\mu_j(\vec{\mathrm{x}};\mathrm{t}), \sigma_j^2(\vec{\mathrm{x}};\mathrm{t}))$$

Ecuación 2: Función de distribución de probabilidad del modelo por MoG

MoG evalúa la pertenencia de los nuevos píxeles con todo el modelo. Si se encuentra pertenencia en el modelo para alguna distribución k, se actualizan los parámetros del modelo para ese píxel y se caracteriza como fondo. En caso de no encontrar pertenencia, la distribución de menor peso se sustituye por una nueva Gaussiana de media el valor del píxel y una desviación de poco valor.

En el modelo de *MoG*, la combinación de k distribuciones Gaussianas cuya probabilidad de ocurrencia (suma de pesos $w_k(\mathrm{x},\mathrm{y};\mathrm{t})$) supere un determinado umbral, denominado umbral de frente, permitirá modelar cada píxel del fondo en cada instante.

Las <u>limitaciones</u> de este método son:

- Gran carga computacional debido a k.
- Poca robustez frente a cambios de iluminación.
- Necesidad de un método de actualización de medias y varianzas adecuado para adaptarse a cambios en el fondo.

Se han propuesto multitud de variantes al modelo clásico de Stauffer en [9] o combinaciones, por ejemplo, Harville en [8] combina la *MoG* con la información de luminancia y profundidad para modelar el fondo. Otro ejemplo es la implementación de Javed en [18] que utiliza *MoG* para llevar a cabo la substracción de fondo *BS* de cada canal de color relacionando cada canal mediante una matriz de covarianza.

2.3.2 Modelos no paramétricos

Un modelo no paramétrico asume que la representación del fondo se realiza por medio de funciones de densidad de probabilidad u otras funciones matemáticas, sin asumir distribuciones definidas mediante parámetros:

- **Densidad de núcleo ('Kernel Density Estimatie'):**

El método de Densidad de núcleo *KDE* calcula la función de densidad de probabilidad de cada píxel del modelo de fondo en cada instante de tiempo t como indica Elgammal en [7].

Esta operación se realiza gracias a la información de la historia reciente de dicho píxel que se haya almacenada en un buffer. El objetivo es obtener mayor sensibilidad de detección que utilizando un método de representación de fondo con una distribución de probabilidad fija. La pertenencia al modelo de fondo $BG(\vec{x};t)$ se estima mediante el promedio de funciones de núcleo o *kernel* (por ejemplo de tipo Gaussiano) y evaluados en el píxel de valor actual $\vec{I}(\vec{x};\mathrm{t})$, es decir, se calcula la probabilidad de parecido entre el píxel actual y los valores que ese mismo píxel ha tomado en las η imágenes anteriores y se discrimina entre fondo o frente a partir de un umbral, donde si la probabilidad supera el umbral el píxel es fondo y, en caso contrario, frente.

Si se elige, por ejemplo, un *KDE* de tipo Gaussiano, cada una de las η muestras almacenadas se considera que siguen una distribución normal de media $\mu_k(\vec{\mathrm{x}};\mathrm{t})$ y varianza $\sigma_k^2(\vec{x};\mathrm{t})$.

$$\Pr(\vec{I}(\vec{x};\mathrm{t})) = \frac{1}{\eta}\sum_{i=t-1-\eta}^{i=t-1} \frac{1}{\sqrt{2\pi\sigma_k^2(\vec{x};\mathrm{t})}} e^{-\frac{(\vec{I}(\vec{x};i)-\vec{\mu}(\vec{x};\mathrm{t}))^2}{2\sigma_k^2(\vec{x};\mathrm{t})}}$$

Ecuación 3: Probabilidad de un píxel por KDE

Las ventajas de este método son la robustez ante el parpadeo de fondo y ruido en la imagen y que es capaz de adaptarse a los cambios rápidos y progresivos de fondo.

Como limitación nos encontramos con el incremento en el coste computacional causado por el almacenamiento y acceso a la información en el buffer.

- **Modelos ocultos de Markov ('*HMM*'):**

Los métodos analizados hasta este punto se adaptan a cambios de iluminación si el cambio es gradual, sin embargo, les cuesta adaptarse ante cambios bruscos.

Otro método para modelar las variaciones de intensidad de los píxeles que aparecen en el fondo es utilizando *HMMs* que representan estas variaciones como un conjunto de estados

discretos que corresponden a los distintos modos de iluminación que pueden presentarse en la escena: luces encendidas, apagadas, día soleado, día nublado, etc… Por ejemplo, en seguimiento de tráfico [19], Stenger propone 3 estados de distribución Gaussiana: fondo, frente y sombra.

El modelo de fondo $BG(\vec{x};t)$ queda definido por un conjunto de estados b_j, donde j representa el número de estados, caracterizados por una función de densidad de probabilidad para cada píxel:

$$BG(\vec{x};t) = b_j(\vec{x})$$

Ecuación 4: Caracterización de fondo en HMMs

Los modelos de representación del fondo basado en *HMMs* requieren un tiempo de cálculo elevado ya que suponen la valuación de un conjunto de estados, y la topología del sistema puede ser muy compleja según las características de la escena.

- **Métodos basados en códigos ('codebooks'):**

Este método codifica cada píxel del modelo de fondo $BG(\vec{x};t)$ con un conjunto de valores llamados *codewords* c_m que constituyen un descriptor o *Codebook* ζ .

Al procesar una nueva imagen de la secuencia, los valores de intensidad de los píxeles se comparan con los que forman su *Codebook* mediante diferencias de color y luminancia, de tal forma que si existe parecido, se estiman y actualizan los códigos y, si no hay coincidencia, se inserta el nuevo código.

$$BG(\vec{x};t) = \{c / c_m\} \in \zeta$$

Ecuación 5: Caracterización del modelo por Codebook

Una limitación de este método es el coste computacional que implica el hecho de mantener un conjunto de códigos para cada píxel que actualizar en cada instante. No obstante, se han desarrollado técnicas que resuelvan este problema, por ejemplo, Butler [20] clasifica los píxeles dotándoles de un peso que indica la cantidad de píxeles del mismo valor; la suma de los pesos es una estimación de la probabilidad de píxeles de fondo y puede utilizarse como umbral de decisión para determinar los píxeles pertenecientes al fondo y al frente.

2.4 MODELO DE INICIALIZACIÓN

Otra forma de clasificación del modelo es analizar la forma en que se inicializa el modelo. La mayoría de los modelos de fondo se basan en la generación inicial de un fondo mediante un conjunto de parámetros obtenidos a través de un número pequeño de imágenes de la secuencia en las que no están presentes los objetos en movimiento [28]. En algunas secuencias es muy difícil ejercer este proceso por ejemplo, en zonas de alta población caracterizadas por una presencia continua de objetos en movimiento u otros efectos perturbadores.

Existen varios métodos para solventar este inconveniente. Una opción es inicializar el fondo mediante el uso de la media de un conjunto de imágenes, como en sistemas de monitorización de tráfico [29]. Otra opción es utilizar la mediana en lugar de la media, ya que la media mezcla los valores de un píxel y puede no dar un valor de píxel real en la imagen, mientras que utilizando la mediana el valor obtenido se encuentra en un píxel de la imagen.

En relación al número de modos que pueden inicializarse sin tener que sustituir modos anteriores, hay que distinguir también el método de inicialización entre modelos monocapa y multicapa.

2.4.1.1 Monocapa

Se utiliza un sistema monocapa para entornos en los que el fondo se modela en un único modo o una combinación pesada de modos y no se contemplan cambios bruscos respecto a éstos, o al menos, no se contempla que, tras el cambio, vuelvan a aparecer. La inicialización en este tipo de modelos suele basarse en los modos obtenidos a partir de los primeros cuadros de la secuencia. Sistemas como el *MoG* de Stauffer [9] o el modelo de Morde [30] utilizan este método de inicialización.

El principal inconveniente de estos modelos es que su capacidad de representación es limitada. Además, la presencia de modos muy diferentes que actualizan una única representación puede influir en el correcto modelado de los modos dominantes e incluso crear modos inexistentes como combinación de los anteriores.

2.4.1.2 Multicapa

Los sistemas multicapa independizan el aprendizaje de cada modo, evitando la creación de modos combinados. Por esto están especialmente diseñados para entornos multimodales, que generalmente se dan en exteriores, donde el fondo se modela con varios modos, en este caso independientes, dependiendo de la escena. En estos casos se utilizan modelos no paramétricos (ver Sección 2.3.2) o paramétricos, como el que se propone en [21] que utiliza modelos Bayesianos multicapas.

La inicialización es este tipo de modelos suele basarse en parámetros obtenidos a partir de los primeros cuadros de la secuencia, al igual que los sistemas monocapa, con la particularidad de que esos cuadros inicializan la primera capa. El resto de capas se inicializa con cuadros en los que aparezcan nuevos modos, y exclusivamente se inicializan esas zonas, y no todo el modelo como ocurre en la primera capa (ver Figura 2.4-1). Además estas capas pueden eliminar esas zonas y volver a inicializarlas con posterioridad.

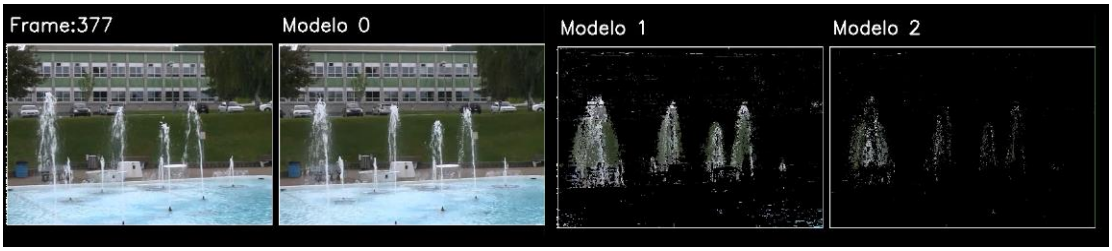


Figura 2.4-1: Cuadro actual y modas inicializadas del modelo de fondo

En la Figura 2.4-1 se muestra el cuadro 377 de la secuencia *'fountain01.avi'* de [37], junto con los modos de las capas inicializadas. Se observa como la primera capa se inicializa completamente, y el resto únicamente en las regiones de la secuencia que presentan modalidad.

2.5 MODELO DE ADAPTACIÓN

Este modelo es necesario para que el modelo se adapte convenientemente a los cambios que se introducen en el modelo. Es cierto, que es un modelo prescindible para algunas secuencias: por ejemplo, existen secuencias de programas de televisión donde el fondo es estático y digital (*chroma-keys*), por lo que no se espera obtener ningún cambio en el modelo. Si bien estas secuencias son la minoría, y en general nos encontraremos ante una secuencia en la que se producirán cambios en el fondo, y será necesaria una adaptación constante ante esos cambios.

2.5.1 Mecanismos de actualización

El proceso de actualización es muy importante en cualquier sistema de *BS*, ya que de este proceso depende la rapidez y la calidad de la adaptación del sistema a cambios en el fondo de la secuencia. Se han propuesto multitud de métodos de actualización, y en este apartado se van a analizar los más interesantes considerando los objetivos propuestos al comienzo de la memoria.

Antes de analizar las diferentes técnicas de actualización, todos los sistemas se pueden clasificar en dos mecanismos:

- **Mecanismo de adaptación selectiva**: Selecciona y actualiza, utilizando cualquier técnica de actualización, sólo los píxeles clasificados como fondo permitiendo incorporar los objetos que aparezcan en la escena y que pasado un determinado tiempo corresponden al modelo de fondo.

- **Mecanismo de adaptación a ciegas**: Actualiza, utilizando cualquier técnica de actualización, todos los píxeles del modelo de fondo sin tener en cuenta ningún tipo de decisión previa. La ventaja es que es un mecanismo más sencillo, pero puede presentar actualizaciones con objetos no deseados al no seleccionar considerar la clase del píxel en el proceso.

Una vez conocido el mecanismo de adaptación del sistema, éste puede adaptarse utilizando las diferentes técnicas que se analizan a continuación:

2.5.1.1 Media móvil

Este sistema (también conocido como *running average*) utiliza un parámetro de adaptación α que determina la velocidad de adaptación del modelo.

$$w(\vec{x};t) = \alpha w'(\vec{x};t) + (1-\alpha) w(\vec{x};t-1)$$

Ecuación 6: Ecuación de factor de actualización por media móvil

En esta ecuación $w(\vec{x};t)$ es la característica o parámetro que se desea actualizar, y $w'(\vec{x};t)$ un nuevo valor para dicho parámetro. Un ejemplo puede ser el de actualización de las modas del modelo en función de la muestra procesada:

$$\vec{\mu}(\vec{x};t) = \alpha \vec{I}(\vec{x};t) + (1-\alpha)\vec{\mu}(\vec{x};t-1)$$

Ecuación 7: Ejemplo de utilización del factor de actualización por media móvil

Mediante esta operación, el modelo de fondo se adapta a las variaciones progresivas que puedan producirse en la escena, como por ejemplo, variaciones en la iluminación del día, ya que la actualización del fondo se realiza con parte de la imagen actual. Este tipo de adaptación del modelo de fondo tiene el inconveniente que contempla la detección como objetos en movimiento a sombras y reflejos.

2.5.1.2 Actualizaciones probabilísticas

Estos sistemas se basan en modelos probabilísticos para actualizar los modelos. Utiliza funciones de distribución de probabilidad, que calculan la probabilidad de pertenencia a cada capa.

Un método es, por ejemplo, definir varias clases cada una modelada en una capa diferente y se estima la probabilidad de pertenecer a cada una de las clases. Benedek en [42] utiliza tres clases: fondo, frente y sombra. En Polikri [24] solo se modela el fondo y se crea una capa distinta para las distintas apariencias que puede tomar un píxel (fondo multimodal). Se modela mediante *SG* y se actualiza con el sistema probabilístico de aprendizaje Bayesiano.

Estos esquemas de actualización son capaces de adaptarse rápidamente a apariencias marginales que adquieren rápidamente más presencia, al mismo tiempo que mantienen adecuadamente las apariencias predominantes no observadas durante un tiempo.

2.5.1.3 Sistemas acumulativos

Estos sistemas utilizan un buffer que almacena los últimos modos observados para cada píxel. En el caso de que sea necesario actualizarse, sustituye el valor actual por uno de los valores almacenados en el buffer. Este tipo de sistema es el utilizado por Elgammal en [7] y por Morde en [30].

2.5.1.4 Sistemas basados en confianzas

En este sistema cada moda tiene asociado un valor de confianza. La confianza aumenta o disminuye en función de la frecuencia con que aparezca la moda en la secuencia. Cuando la confianza disminuye hasta alcanzar un valor bajo, el sistema tiende a actualizarse en las zonas de baja confianza. En [36] se utiliza un sistema de confianza que nos indica la fiabilidad de cada capa almacenada en el modelo generado.

2.5.1.5 Sistemas multiclase

El sistema clasifica los píxeles de cada cuadro asociándole unas etiquetas con la función que se ha decidido que cumpla en el algoritmo. Basándose en esa etiqueta, es posible actualizar o no el modelo mediante procesos selectivos. Por ejemplo, en [21] se utilizan etiquetas de tipo fondo actualizable para clasificar determinados píxeles, cuyos modos en el modelo serán actualizados de acuerdo a esta clasificación.

2.6 CARACTERIZACIÓN / COMPARACIÓN

Con el objetivo de detectar el frente como cambio respecto al fondo almacenado, son necesarios los procesos de caracterización: cómo se describe un píxel, y comparación: cómo se detectan cambios respecto al modelo.

2.6.1.1 Caracterización

Cada píxel, y cada modo en el modelo, pueden representarse mediante cualquier característica que se pueda modelar. Algunos ejemplos de características utilizadas por el modelado son la luminancia Y [8], el color [24] (generalmente organizado siguiendo el esquema rojo-verde-azul: *RGB*), la textura [25] o la distribución del color etc.

En el caso del color *RGB*, un píxel se representaría mediante un vector de características tridimensional. Por ejemplo, para una imagen RGB de entrada al sistema: $\vec{I}(\vec{x};t)$, la caracterización del píxel $\vec{x}$ en el instante *t* sería: $\{I_R(\vec{x};t), I_G(\vec{x};t), I_B(\vec{x};t)\}$, donde $I_R(\vec{x};t)$ representa el canal rojo de la imagen.

2.6.1.2 Métrica

Se define métrica o distancia cualquier función matemática $d(\vec{a},\vec{b})$ que verifique las siguientes condiciones:

- No negatividad. $d(\vec{a},\vec{b}) \geq 0$
- Simetría. $d(\vec{a},\vec{b}) = d(\vec{b},\vec{a})$
- Desigualdad triangular. $d(\vec{a},\vec{b}) \leq \mathrm{d}(\vec{a},\vec{c}) + \mathrm{d}(\vec{c},\vec{b})$
- Identidad. $d(\vec{a},\vec{b}) = 0$ si y sólo si $\vec{a} = \vec{b}$

Para poder decidir si un píxel pertenece al frente o al fondo hay que establecer unas reglas que permitan medir el grado de similitud o la probabilidad de pertenencia a una u otra clase. Por este motivo, los cuadros de vídeo se comparan con el modelo de fondo mediante la métrica para obtener una medida objetiva que indique el parecido al modelo.

2.6.1.3 *Métricas específicas*

Para obtener esa media se pueden utilizar diferentes métodos como pueden ser la distancia Euclídea, la distancia de Mahalanobis, o la CityBlock [23], que no será analizada en este documento.

Estas distancias miden el parecido del valor modelado a la nueva muestra de entrada en función de la caracterización seleccionada.

- **Distancia Euclídea o normal L2:**

La distancia Euclídea es la distancia entre dos puntos de un espacio euclídeo. En ella se obtiene la diferencia por separado de cada una de las componentes que forman las muestras a comparar y se hace la raíz cuadrada de todas ellas (Teorema de Pitágoras).

Estructuralmente un espacio euclídeo es un espacio vectorial normado sobre los números reales de dimensión finita.

Dado un píxel en el instante t del cuadro de entrada descrito por el vector de características en el espacio RGB $f(\vec{x},t)$ (vector tridimensional). La distancia Euclidea al modelo $\vec{\mu}(\vec{x};t)=\{\mu_R(\vec{x};t),\mu_G(\vec{x};t),\mu_B(\vec{x};t)\}$ se define:

$$\left\|\vec{I}(\vec{x},t),\overrightarrow{\mu_\lambda}(\vec{x},t))\right\|_2=\sqrt{(I_R(\vec{x},t)-\mu_R(\vec{x},t))^2+(I_G(\vec{x},t)-\mu_G(\vec{x},t))^2+(I_B(\vec{x},t)-\mu_B(\vec{x},t))^2}$$

Ecuación 8: Distancia Euclídea

- **Distancia de Mahalanobis:**

La distancia de Mahalanobis contempla la distribución de los datos en espacios de características multidimensionales, considerando la correlación entre las características utilizadas para el modelado, la distancia de Mahalanobis: $d^2(\vec{I}(\vec{x},t),\vec{\mu}(\vec{x},t))$ entre el píxel y su modelo dado que la matriz de covarianza entre las componentes RGB observadas para el píxel $\vec{x}$ desde el instante 0 al t se conoce: $\sum(\vec{x};t)$ viene definida por la Ecuación 9.

$$d^2(\vec{I}(\vec{x},t),\vec{\mu}(\vec{x},t))=(\vec{I}(\vec{x},t)-\vec{\mu}(\vec{x},t))^T\times\sum{}^{-1}(\vec{x},t)\times(\vec{I}(\vec{x},t)-\vec{\mu}(\vec{x},t))$$

Ecuación 9: Distancia de Mahalanobis

Este cálculo se utiliza en sistemas como el propuesto por Colmenarejo en [21].

2.6.1.4 *Mejoras al cálculo de distancia*

- **Distancia del cono:**

La distancia del cono es un método propuesto por Evangelio en [22] para dar robustez al sistema ante cambios de iluminación. La idea parte del modelo de cilindro propuesto por Horprasert en [27], que define un rango de valores en el dominio *RGB* que se encuentran dentro de un volumen de forma cilíndrica:

- El centro del cilindro es la representación de un píxel del cuadro de vídeo.

- El radio del cilindro define la permisividad del sistema, es decir, cuanto mayor sea el radio del cilindro, más permisivo será el sistema con la diferencia entre el cuadro y el modelo.
- La altura del cilindro dota al sistema de robustez frente a cambios de iluminación ya que al aumentar o disminuir la luminancia el valor de los píxeles aumenta o disminuye en una proporción similar para cada canal.

Evangelio propone modificar el cilindro a forma cónica basándose en que los valores más oscuros tienen menos resolución frente a los valores más claros. Por eso el radio en la base inferior del cilindro se ve reducido y el de la superior aumentado. Esta aproximación está detallada en la Sección 3.5.1.2.2.

2.6.1.5 Unidades de comparación

Existen varias unidades con las que se puede obtener la medida de parecido entre la imagen y el modelo.

- **Píxel:** Es la menor unidad en la que puede dividirse un cuadro de vídeo. El método más común es comparar entre píxeles que se encuentran en las mismas coordenadas del mismo cuadro en diferentes instantes de tiempo o entre un cuadro y el modelo.
- **Blob:** Es un conjunto de píxeles conexos espacialmente.
- **Región:** Una región es un conjunto de píxeles conexos espacialmente y que además tienen un rango de valores de intensidad, textura o color en común. La agrupación de estos píxeles en regiones evita la aparición de píxeles independientes en las zonas de fondo o frente.
- **Entorno espacial:** Realiza comparación entre un píxel y los píxeles de otra imagen con coordenadas vecinas a su posición, tantas posiciones en cada dirección como el parámetro *shift* indique. Este método es útil para eliminar el ruido procedente de las vibraciones sometidas a la cámara. Pero es un método que aumenta el coste computacional exponencialmente (ver Figura 2.6-1) en función del valor de *shift*.

Valor de shift	Matriz de comparación	Píxeles a comparar
0	1x1	1
1	3x3	9
2	5x5	25
3	7x7	49
4	9x9	81

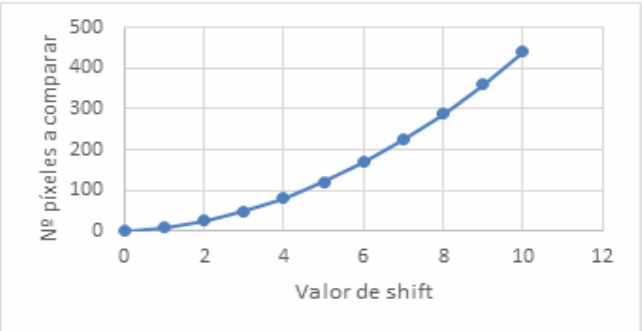


Figura 2.6-1: Evolución del parámetro shift

La mayor parte de sistemas utilizan el píxel como medida de análisis, si bien, algunos sistemas como el propuesto por Escudero en [36] se basan en regiones o el propuesto por Hofmann en [32] que se basa en el entorno espacial.

2.7 COMPARATIVA DE LAS DIFERENTES TÉCNICAS

Algunos de los sistemas predecesores analizados en el *SoA* utilizan diferentes técnicas para los tres modelos estudiados (representación, inicialización, y adaptación). En la Tabla 2.7-1, se analiza que métodos se utilizan y junto a que métodos de los otros modelos suelen aparecer.

Algoritmo	Representación	Inicialización	Adaptación
Stauffer [9]	Paramétrico: MoG	Monocapa: Con primeros cuadros	Media móvil
Elgammal [7]	No paramétrico: KDE	Multicapa	Método acumulativo
Evangelio [22]	Paramétrico: GMM	Monocapa: Mediana	Media móvil
Evangelio [26]	Paramétrico: 2 GMM	Monocapa: Mediana	Media móvil
Morde [30]	Paramétrico: MoG	Monocapa: Con primeros cuadros	Selectivo: Método acumulativo
Colmenarejo [21]	Paramétrico: MoG	Multicapa: Con primeros cuadros	Sistema multiclase, media móvil, probabilístico y sistema de confianzas
Hofmann [32]	Paramétrico: GMM	Monocapa: Con primeros cuadros	Selectivo: Vecindario
Schick [33]	Paramétrico: SG o MoG	Monocapa: Media primeros cuadros	Modelos probabilísticos
Maddalena [34]	No Paramétrico: modelo neuronal	Monocapa: Con primeros cuadros	Media móvil
Van Droogenbroeck [35]	No paramétrico: modos independientes	Monocapa: 20 cuadros	Sistema multiclase y acumulativo
Escudero [36]	Mediante sistema de capas	Multicapa: Con primeros cuadros	Media móvil, Confianza

Tabla 2.7-1: Comparativa de sistemas de BS

2.8 PLATAFORMA *DIVA*

El sistema *DiVA* [39] es una plataforma implementada por el grupo de investigación *VPU* de la *EPS* en la Universidad Autónoma de Madrid. La función de esta plataforma es servir cuadros de vídeo a diferentes algoritmos en tiempo real, desde las diferentes cámaras accesibles por el sistema. A continuación se va a realizar un breve resumen de cómo funciona esta plataforma (ver Figura 2.8-1).

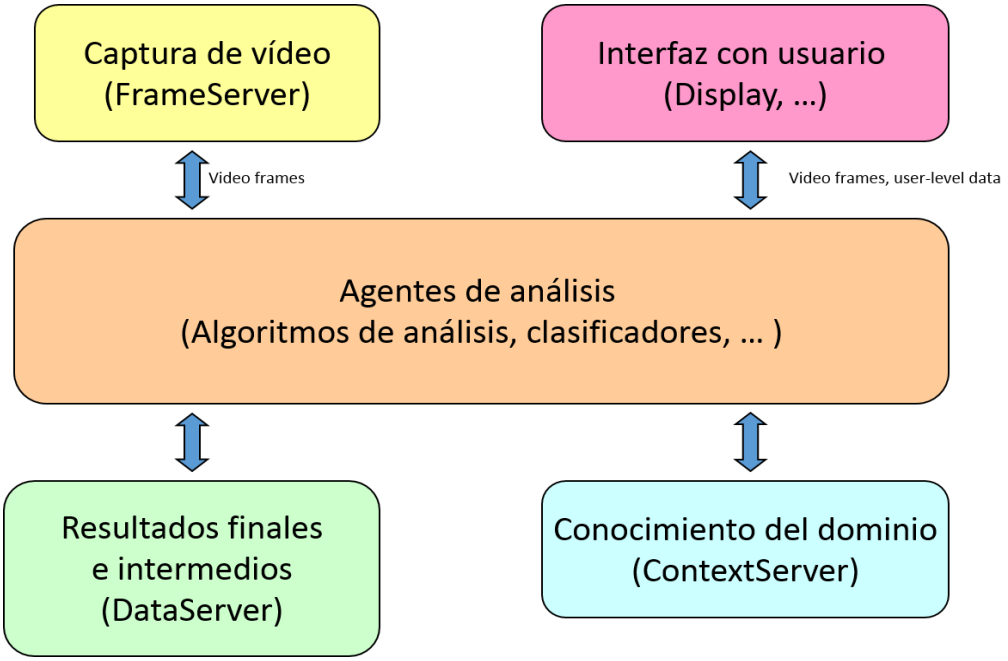


Figura 2.8-1: Arquitectura DiVA

El sistema está formado por un grupo de cámaras distribuidas por los edificios de la *EPS* que se encarga de la grabación de secuencias de vídeo. Estas cámaras pueden ser tanto cámaras fijas como cámaras *PTZ* (estás pueden realizar movimiento en todas las direcciones y hacer uso de zoom). Este módulo de captura de vídeo recibe el nombre de *Frame Server* y se encarga de capturar la secuencia, transcodificarla con los parámetros elegidos (tamaño del cuadro, velocidad de captura, tipo de codificación, etc.) y de enviar los cuadros al algoritmo.

El sistema es robusto a errores, es un sistema eficiente, permite distintos protocolos y no tiene un gran coste computacional. Otra ventaja es que es un sistema distribuido y puede enviar cuadros a varios algoritmos distintos para que se ejecuten simultáneamente, y permite modificar los parámetros del algoritmo mientras éste se ejecuta. A partir de un puerto y una dirección *IP* el algoritmo es capaz de recibir los cuadros con la configuración deseada.

Aunque el sistema esté diseñado para las cámaras de la *EPS* es capaz de recibir cuadros de otros dispositivos como una webcam, de modo que es posible probar el algoritmo desde un mismo ordenador, si éste actúa de servidor de cuadros.

Además, recientemente en [41] se han implementado novedades sobre el sistema tales como poder recibir cuadros de varias cámaras a la vez en un mismo instante.

Otra novedad, es que propone un método para que todos los algoritmos tengan una misma estructura y así se puedan implementar fácilmente en la plataforma. Además, permite integrar varios algoritmos que estén implementados en *DiVA*.

Esta estructura consiste en 3 niveles:

- **Primer nivel:** El algoritmo se encargar de procesar los cuadros de secuencias de vídeo y además tiene dos funciones, una para visualizar los resultados y otra para guardarlos en un archivo.

- **Segundo nivel:** El algoritmo se introduce en la plataforma *DiVA* utilizando una serie de librerías creadas que permiten obtener cuadros del servidor de cuadros utilizando la dirección IP de dicho servidor y un puerto de escucha.

- **Tercer nivel:** Consiste en la creación de una *GUI* que permite visualizar los resultados y modificar los parámetros mientras se ejecuta el algoritmo. Existen dos casos para este nivel, una versión para usuarios expertos que permite visualizar los resultados intermedios y modificar los parámetros desde la propia interfaz y una versión para clientes que solo muestra los resultados finales y permite la modificación de parámetros desde una pestaña de opciones. Existen unas *GUI* creadas a partir de las cuales se pueden diseñar la *GUI* más eficaz para cada algoritmo modificándolas.

2.9 BIBLIOTECA QT

Qt es una biblioteca multiplataforma [40] utilizada principalmente para el desarrollo de interfaces gráficas *GUI* para aplicaciones aunque también es utilizada como herramienta para servidores de consola o para línea de comandos.

Algunas aplicaciones conocidas que han utilizado *Qt* son *Google Earth*, *Skype* o *VirtualBox*.

El lenguaje que utiliza esta biblioteca es lenguaje de programación *C++* aunque el entorno de trabajo es gráfico por lo que puede construirse la interfaz a partir de herramientas disponibles mientras el programa crea el código de programación.

2.10 CONSIDERACIONES

Una vez analizadas las distintas posibilidades para realizar el sistema, se seleccionan los métodos que, en nuestra opinión, son los más adecuados para cumplir los objetivos que se proponen en este *TFG*.

Teniendo en consideración que el sistema que se propone debe operar en espacios exteriores, es intuitivo el desarrollo de un sistema multicapa, debido a la multimodalidad que suele aparecer en este tipo de entornos. Este tipo de sistemas tiene un mayor coste computacional que sistemas monocapa, que además aumenta progresivamente cuanto mayor sea el número de capas utilizado por el modelo. Sin embargo, es un coste computacional asumible, considerando la necesidad de almacenar las distintas apariencias que toman los píxeles en la secuencia a través de capas.

Para la adaptación del modelo, en la parte de comparativa se ha analizado el *BS* a nivel de regiones (ver Sección 2.6.1.5). Este proceso necesita un segmentador que clasifique los píxeles en distintas regiones, un proceso que exige mucho coste computacional. Las ventajas son la exclusión de píxeles independientes (ruido) cuya probabilidad de ser frente es nula y sin embargo pueden introducirse como frente. El gran coste computacional exigido hace que este proceso sea prescindible.

Además, está la técnica de distancia del cono para eliminar sombras, cuya inclusión en un sistema que trabaja en exteriores es importante, y la técnica de comparación con el vecindario, que elimina los ruidos procedentes de vibraciones. Esta técnica aumenta el coste exponencialmente con el aumento de su valor (ver Figura 2.6-1), pero para valores de *shift* bajos se obtienen mejoras considerables ya que las capturas de vídeo casi siempre introducen un ruido de vibración aunque sea muy pequeño.

La parte de actualización contempla diferentes técnicas. La primera decisión es la de actualización selectiva vs actualización a ciegas. La actualización selectiva es algo más compleja pero evita que se actualice con el frente lo cual bajaría sistemáticamente los resultados del sistema.

Debido a que la parte de actualización es determinante en sistemas multimodales, es posible realizar un sistema que utilice varias técnicas (como media móvil, sistema acumulativo y sistema de confianzas). El mayor número de técnicas aporta al sistema más coste computacional pero a la vez mayor robustez permitiendo obtener una buena actualización.

Por otro lado, la plataforma *DiVA* permite obtener capturas de vídeo en tiempo real sin un gran coste computacional. Dicha plataforma aporta al sistema independencia para obtener pruebas y resultados ya que se pueden hacer pruebas sin necesidad de recurrir a vídeos ya editados y por tanto, la funcionalidad del sistema es mayor.

Por último, la inclusión de una interfaz de usuario *GUI* es muy beneficiosa debido a las diferentes situaciones que se encuentran en entornos exteriores. Mediante la interfaz es posible modificar los parámetros configurables simultáneamente con la ejecución del sistema, ahorrando en coste computacional y mejorando los resultados. Por ejemplo, se puede variar el número de capas que forman el modelo en función del número de apariencias que tomen los píxeles en la secuencia: si estamos ante un entorno con muchas modas se aumentará el valor frente a pocas modas donde fijará un valor pequeño, e incluso, es posible utilizar un sistema de una sola capa (monomodal) si la naturaleza de la secuencia lo permite.

CAPÍTULO 3: DISEÑO

A partir del estudio realizado en el Estado del Arte *SoA* se ha decidido diseñar un modelo con las características que más favorecen al sistema en base a los objetivos planteados.

En la Figura 3-1 se muestran las técnicas analizadas en el *SoA* clasificadas según el esquema propuesto por Cristani en [13], para posteriormente decidir las técnicas que utilizará el sistema propuesto en la Figura 3-2.

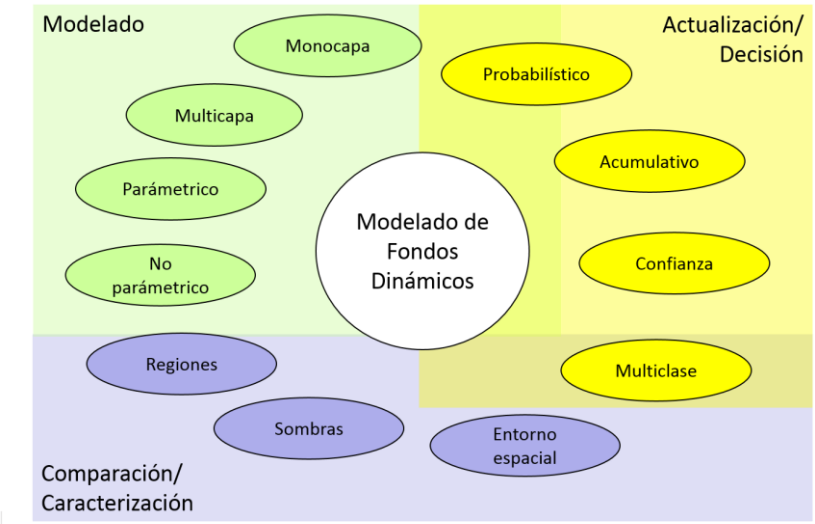


Figura 3-1: Métodos del Estado del Arte

Se ha diseñado un sistema que realiza un modelado de fondo multicapa, ya que se enfrentará a secuencias multimodales. Además modela tanto el fondo, como el frente. El sistema depende de un factor para caracterizar el modelado por lo que puede considerarse como paramétrico aunque no en su forma estricta (ya que no utiliza modelado por funciones de densidad de probabilidad *fdp*), pero lo que mejor define la caracterización es la aproximación multicapa. El método de inicialización es multicapa, inicializando la primera capa con los primeros cuadros, y el resto de capa cuando aparecen otras modas diferentes a las del modelo. El sistema utiliza como métodos de comparación un análisis a nivel de píxel, como se utiliza generalmente, y además, un sistema de eliminación de sombras y una comparación con el entorno espacial o vecindario para eliminar el ruido procedente de la captura. El sistema utiliza una actualización selectiva, utilizando técnicas de actualización basadas en confianzas, sistema acumulativo y multiclase (ver Figura 3-2).

Las confianzas dan robustez a las modas almacenadas en las capas del modelo permitiendo distinguir las apariencias de los píxeles que se dan a menudo a lo largo de la secuencia y las que aparecen en pocas ocasiones, las cuales pueden ser eliminadas.

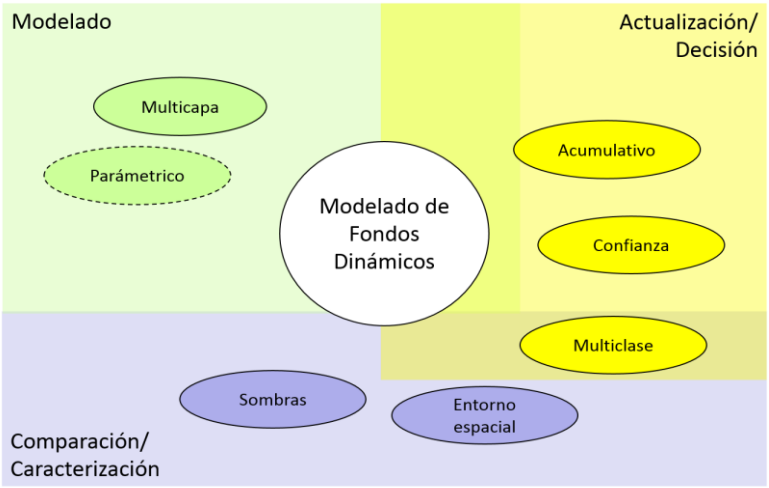


Figura 3-2: Sistema propuesto

El sistema multiclase permite clasificar entre fondo y frente por una parte, y estático y dinámico por otra, permitiendo mejoras en la actualización. Por ejemplo, cuando un objeto en movimiento se mantiene parado durante un largo periodo de tiempo, pasa a ser considerado frente estático, en lugar de incorporarse directamente al fondo. Al tener la clase frente estático, somos conscientes de que esos píxeles pueden clasificarse como fondo estático, si su confianza se ve aumentada. Por tanto, la inclusión de estas técnicas en el sistema aporta robustez y dinamismo a la actualización del modelo.

3.1 PRESENTACIÓN DEL SISTEMA

El sistema propuesto es un sistema diseñado en forma modular. La técnica utilizada es el uso de módulos independientes para cada etapa del sistema de forma que sea muy sencillo modificar, añadir o eliminar una fase sin afectar al resto del sistema. Los distintos módulos cumplen funciones diferenciadas dentro del sistema. El sistema propuesto también es ligero para poder ejecutarse en aplicaciones que trabajan en tiempo real.

Como se indicó en la Sección 1.2, los objetivos que se han de cumplir son la robustez frente al ruido, robustez frente a las vibraciones de la cámara, consideración de sombras y cambios de iluminación, consideración de fondos multimodales y solución a problemas de camuflaje.

Para conseguir alcanzar estos objetivos se proponen diferentes algoritmos como el de comparación con el vecindario, que reduce el ruido introducido por las vibraciones de la cámara, la implementación de la distancia del cono para conseguir robustez frente a sombras, el uso de modelos multicapas para entornos multimodales y el sistema de organización en clases.

Además, es muy importante el uso de la confianza asociada a cada píxel (x, y) para los métodos de comparación y actualización utilizados por el algoritmo.

En particular, es importante señalar que la mayor innovación del sistema propuesto radica en que el proceso de actualización de la confianza es el encargado de definir la fase de comparación.

En este apartado se representa el diagrama de flujo del sistema (Ver Figura 3.1-1) y posteriormente se explica cada paso de dicho diagrama.

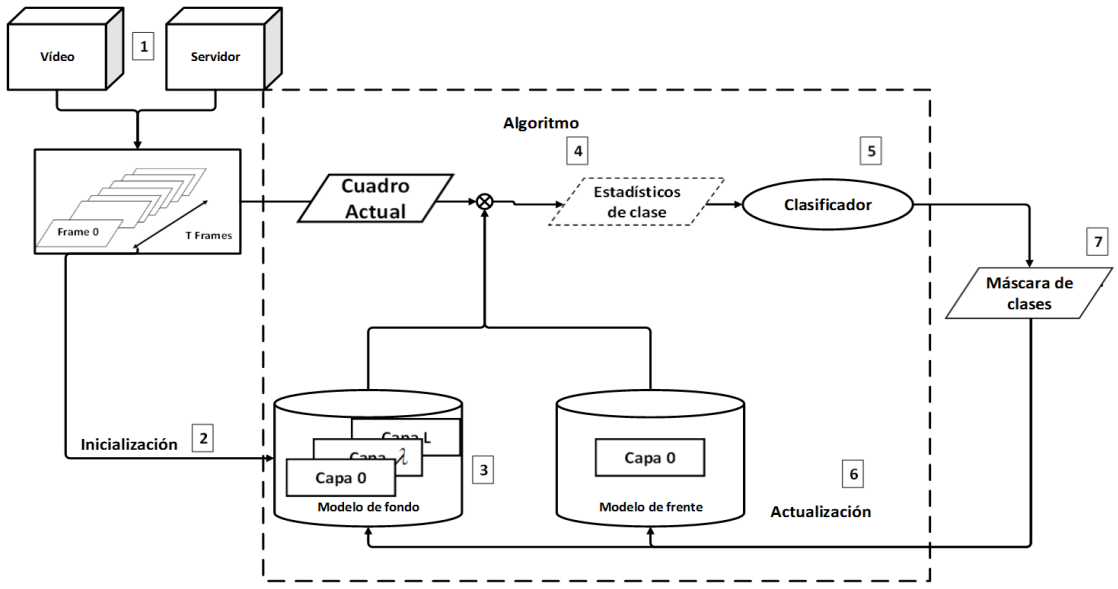


Figura 3.1-1: Diagrama de flujo del sistema

A continuación se realiza una breve explicación de cada paso, que se explicará más detalladamente en cada apartado del capítulo, clasificándolos en la estructura de modelos analizada en el *SoA*, cada etapa está asociada con la etiqueta correspondiente en la Figura 3.1-1:

- **Modelo de inicialización:**

1. **Entrada:** El sistema puede recibir cuadros de vídeo, tanto de un archivo de vídeo en formato *AVI*, como del servidor de cuadros de la plataforma *DiVA*, si se desea recibir la captura de vídeo en tiempo real de cualquier cámara (ver Sección 3.3).

2. **Inicialización:** El proceso de inicialización comienza con el primer cuadro que inicializa la primera capa del modelo de fondo. Más adelante se inicializarán otras capas utilizando los modos de otros cuadros cuando sea necesario (ver Sección 3.3).

- **Modelo de representación:**

3. **Modelado:** El sistema se compone de dos modelos: el modelo de fondo, que ha sido activado en el paso 2, y el modelo de frente que se activa cuando se detecte frente (ver Sección 3.4). El modelo de fondo puede ir aumentando el número de capas hasta alcanzar el máximo L mientras que el de frente consta sólo de una capa.

- **Modelo de adaptación:**

4. **Comparación:** el sistema recibe el cuadro en el instante t para ser procesado, y éste es comparado con el modelo de fondo para obtener los datos estadísticos (ver Sección 3.5.1). Si el modelo de frente está activado, este módulo calculará también los datos estadísticos entre el cuadro actual (o cuadro en el instante t) y el modelo de frente. Al final se obtienen los mejores datos estadísticos (estadísticos de la mejor correspondencia) para todas las capas inicializadas del fondo y para la del frente, si ésta está también inicializada.

5. **Clasificación:** el clasificador se basa en etiquetar cada píxel a partir del valor de la distancia recibida del módulo de comparación y otros factores (ver Sección 3.6). Este algoritmo discrimina cuatro clases: fondo estático, fondo dinámico, frente estático y frente dinámico.

6. **Actualización:** utilizando un nuevo sistema de clasificación previo al módulo de actualización, a cada píxel se le asocia una etiqueta de actualización (ver Sección 3.7). En función de esta etiqueta y de la clase, el píxel actualiza el modelo, inicializa un espacio libre en el modelo o bien crea una nueva capa.

7. **Salida:** El algoritmo devuelve un cuadro de salida con una máscara de clases por cada cuadro procesado (ver Sección 3.6.4).

3.2 NOMENCLATURA

Se define N como el número total de cuadros del vídeo, λ como cada capa del modelo con $\lambda \in [0, L-1]$, donde L es el número máximo de capas del modelo.

Por otro lado, como se comentó en la Sección 2.2, se define $\vec{I}(\vec{x};t)$ al cuadro de entrada en el instante t siendo t un valor desde *0* hasta N ambos incluidos.

$\vec{I}(\vec{x};t)$ es una representación vectorial de cada imagen en la cual cada píxel (x, y) en el instante t se describe por su vector *RGB*: $\vec{I}(\vec{x};t) = \{I_R(\vec{x};t), I_G(\vec{x};t), I_B(\vec{x};t)\}$.

3.3 ENTRADA E INICIALIZACIÓN DEL SISTEMA

El sistema tiene dos técnicas para obtener cuadros de análisis (ver Figura 3.3-1):

- **Vídeo:** Mediante un fichero de vídeo en formato *AVI* cargado se obtiene una captura de la que se extraen secuencialmente los cuadros $\vec{I}(\vec{x};t)$, $t \in (0, N)$.
- **Servidor:** Un servidor envía cuadros de vídeo al sistema que los recibe y los procesa de la misma forma que lo hace con un vídeo.

Cuadro de vídeo 1471 de 'highway.avi'

Cuadro de vídeo del servidor de cuadros

Figura 3.3-1 : Procedencia de los cuadros

El primer cuadro del sistema $\vec{I}(\vec{x};0)$ inicializa la primera capa del modelo de fondo. Como se ha visto en el *SoA*, es posible que en dicho cuadro aparezca un intruso, lo que se conoce como "*inicio en caliente*", pero con el paso del tiempo el modelo de fondo se actualizará y el intruso se eliminará del modelo.

Cuando aparecen nuevas modas en cuadros $\vec{I}(\vec{x};t)$, éstos inicializan nuevas capas en las zonas donde aparecen las nuevas modas. Ocurre lo mismo en el caso de aparecer frente, que inicializa las zonas del modelo de frente en las que se haya detectado el objeto.

Nótese que la inicialización solo es completa para el primer cuadro en el modelo de fondo, en los sucesivos modos observados para el fondo o para el frente sólo se iniciarán aquellas partes de las nuevas capas que sean necesarias para su almacenamiento (ver Figura 2.4-1). Este proceso está descrito con detalle en el capítulo 3.7.

3.4 MODELADO DE FONDO Y FRENTE

El método de substracción de fondo (*BS*) diseñado se basa en modelado de fondo y modelado de frente. Como se ha explicado en la sección anterior, el modelo de fondo se inicializa con el primer cuadro recibido por el algoritmo y se actualiza con cuadros posteriores. Cada modelo tiene L capas donde se crean localmente y dependiendo de la modalidad de la secuencia nuevos modos (ver Figura 3.4-1). En dicha figura, extraída del sistema procesando la secuencia '*fountain01.avi*' de [37], se muestra un modelo con 3 capas de fondo inicializadas y una de frente.

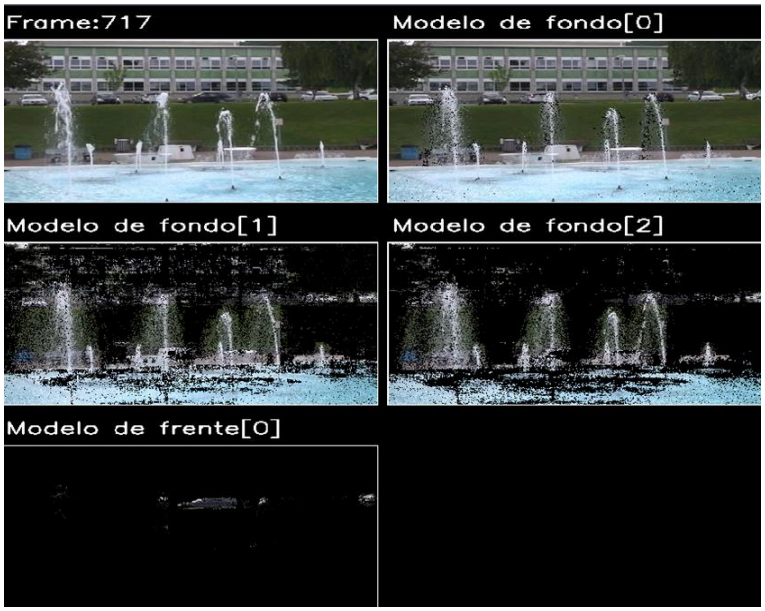


Figura 3.4-1: Representación de modelos de fondo y frente multicapa.

Los parámetros que se encuentran tanto en el modelo de fondo como en el de frente dependen de tres variables por capa λ. Estas variables, en realidad imágenes o matrices almacenan estadísticos para cada píxel en cada instante de tiempo.

El modelo se compone de (ver Figura 3.4-2):

- $\vec{\mu}_\lambda(\vec{x};t)$: Esta matriz de vectores o de modos es la representación mediante caracterización por color RGB del modelo en la capa λ. El valor puede ser diferente para cada posición del cuadro, cada instante temporal y cada capa del modelo.

- $C_\lambda(\vec{x};t)$: Cada modo tiene asociado un valor de confianza en cada instante de tiempo: $C_\lambda(\vec{x};t) \in (-1, C_{\max})$, donde -1 representa las zonas de los modelos que aún no han sido inicializadas y $C_{\max}$ es la confianza máxima.
- $K_\lambda(\vec{x};t)$*:* Cada modo tiene también asociado un valor $K_\lambda(\vec{x};t)$. Este valor determina la permisividad del sistema al realizar la comparación y la actualización por lo que es un factor determinante para el sistema. El valor de la permisividad será más alto cuanto mayor sea la variabilidad asociada al modo al que esté asociado.

Cada capa se compone de estas tres matrices que se actualizan en el tiempo. El modelo de fondo y el modelo de frente pueden tener un distinto número de capas. Generalmente, el modelo de frente se compone de una única capa mientras que el modelo de fondo dispondrá de varias capas en función de la multimodalidad de la secuencia.

Figura 3.4-2: Composición del modelo

En la Figura 3.4-2 puede observarse como la capa de frente no está inicializada por ausencia de objetos en movimiento, mientras la multimodalidad que el agua aporta a la escena ha inicializado las capas de fondo. Los valores de la confianza y la permisividad se representan en escala de grises, siendo el color blanco la representación del máximo valor.

En los siguientes apartados se explicarán los procesos de comparación y cálculo de distancias, las ecuaciones que definen $K_\lambda(x, y; t)$ y el proceso de actualización del modelo.

3.5 CORRESPONDENCIAS CON EL MODELO

La principal diferencia del sistema propuesto con el estado del arte es que liga la fase de comparación con la actualización de la confianza esperada. Es decir, fija los umbrales de correspondencia en función del incremento/decremento de la confianza que dicha correspondencia acarrea.

Por ello, esta sección detalla primero los procesos comparativos existentes en el sistema para a continuación describir la evolución de las confianzas en base a estos procesos. Finalmente, se detalla como ambas: la comparación y la evolución de la confianza definen el proceso de correspondencia.

3.5.1 Comparación con el modelo

La métrica del sistema se basa en la comparación entre el cuadro actual $\vec{I}(\vec{x};t)$ y los modos $\vec{\mu}_\lambda(\vec{x};t)$ cuando la capa λ ha sido previamente inicializada. El resultado es una media que indica el grado de diferencia entre dos píxeles (x, y), para diferentes instantes.

Como se ha visto en el Capítulo 2: Estado del Arte, existen diferentes métodos para calcular estas diferencias en función de la aplicación deseada. A continuación se explicarán los métodos utilizados para el desarrollo de este sistema.

El apartado de comparación recibe la matriz $\vec{\mu}_\lambda(\vec{x};t)$ de cada capa del modelo de fondo activa (y la del modelo de frente si éste está activado) para compararlo con el cuadro actual $\vec{I}(\vec{x};t)$. Dentro de este módulo, pueden configurarse distintas etapas (Ver Figura 3.5-1):

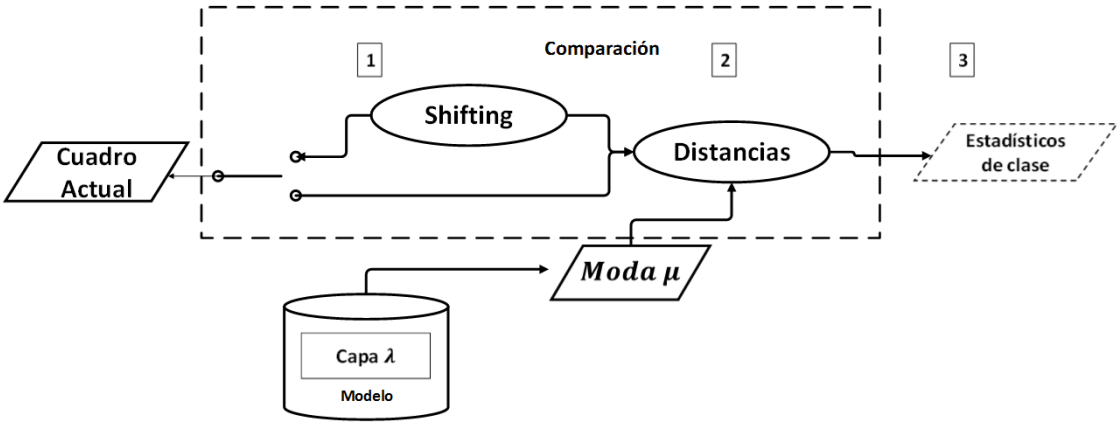


Figura 3.5-1: Diagrama de flujo del módulo de comparación

En la Figura 3.5-1 se observa, como para cada cuadro del vídeo, el módulo de comparación recibe $\vec{I}(\vec{x};t)$ junto con la matriz $\vec{\mu}_\lambda(\vec{x};t)$ de las capas de los modelos y que hayan sido inicializadas (total o parcialmente).

A continuación se describen las etapas del módulo de comparación con el modelo por las que pueden pasar los cuadros (dependiendo de la configuración del sistema pasan por determinadas etapas):

3.5.1.1 Entorno espacial

Esta etapa se encarga de realizar la comparación entre cada píxel (x, y) de cada modelo con los píxeles $(\mathrm{x} \pm shift, \mathrm{y} \pm shift)$ de la muestra, donde $shift$ es un valor que representa el número de píxeles que se desplaza la muestra para comparar (ver Sección 2.6.1.5 y Figura 3.5-3).

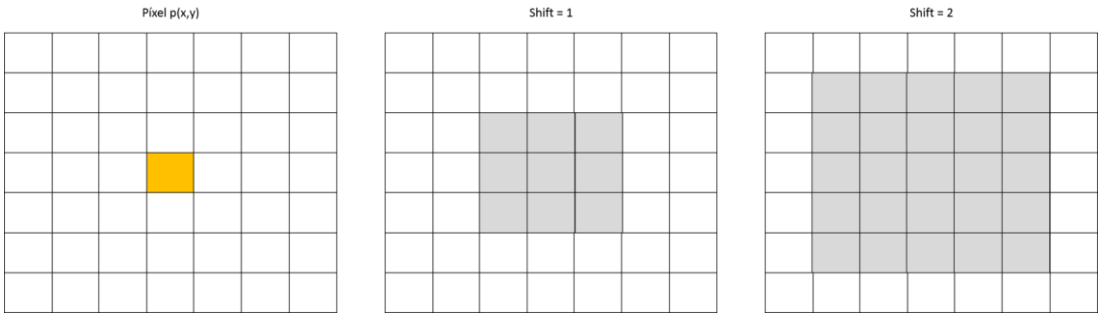


Figura 3.5-2 : Entorno espacial analizado para diferentes valores shift

El proceso de entorno espacial realiza una comparación entre un píxel del cuadro actual $\vec{I}(\vec{x};t)$ con el vecindario que determina el parámetro $shift$ de ese píxel en el modelo $\vec{\mu}_\lambda(\vec{x} \pm shift;t)$. Por ejemplo, si el valor de $shift = 1$, se compara en una matriz 3x3 alrededor del píxel, en caso de $shift = 2$ la matriz será 5x5, y así sucesivamente (Ver Figura 3.5-2).

Podemos ver el uso del entorno espacial en la secuencia de vídeo '*boulevard.avi*' de la base de datos de *changedetection* [37] (ver Figura 3.5-3).

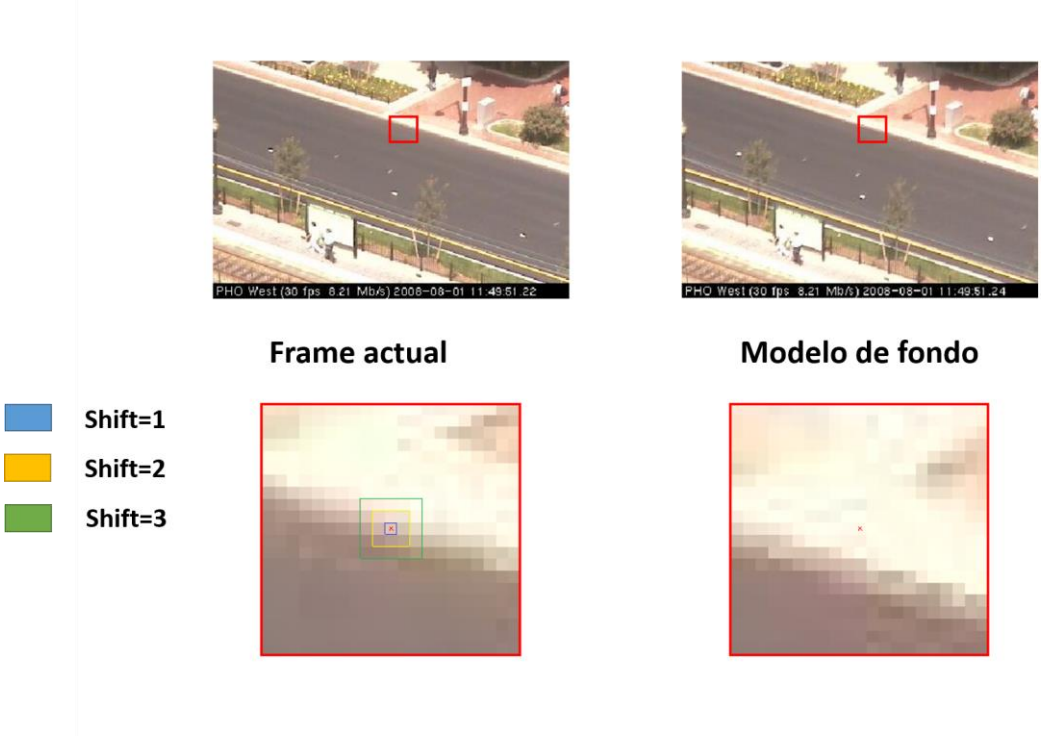
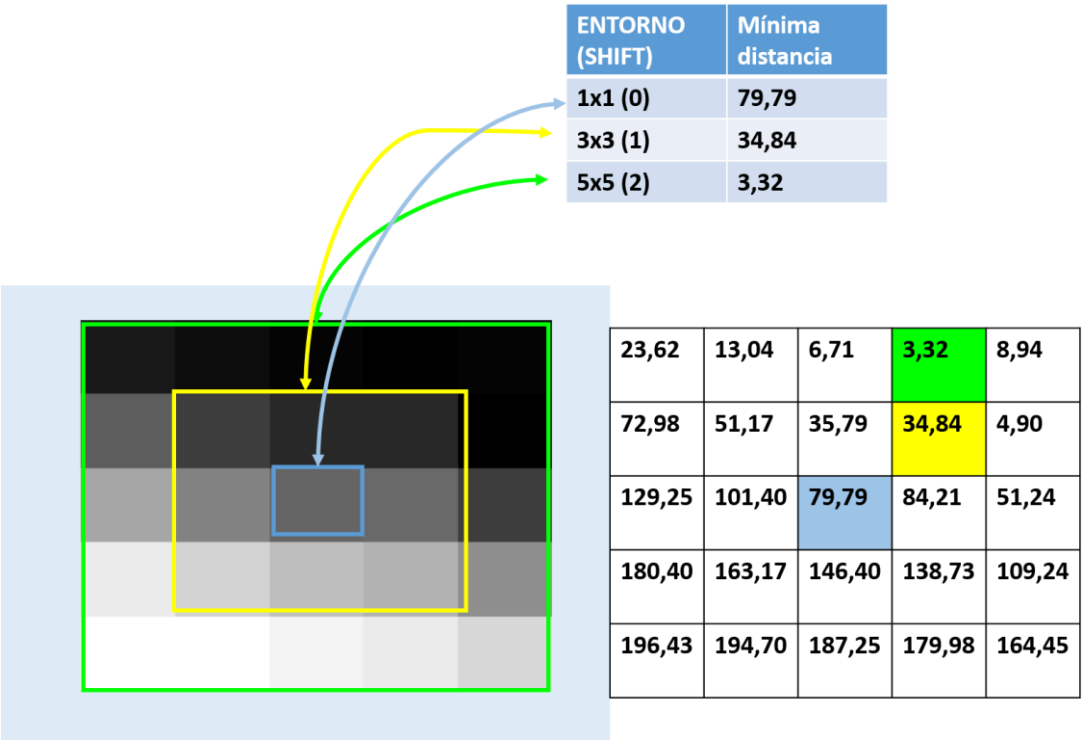


Figura 3.5-3: Proceso de análisis del entorno espacial en una secuencia de vídeo

Tras realizar la comparación con el vecindario, la distancia final $d_\lambda(\vec{x};t)$ será la que resulte en la menor distancia obtenida entre todo el vecindario (ver Figura 3.5-4).



ENTORNO (SHIFT)	Mínima distancia
1x1 (0)	79,79
3x3 (1)	34,84
5x5 (2)	3,32

23,62	13,04	6,71	3,32	8,94
72,98	51,17	35,79	34,84	4,90
129,25	101,40	79,79	84,21	51,24
180,40	163,17	146,40	138,73	109,24
196,43	194,70	187,25	179,98	164,45

Figura 3.5-4: Resultados de comparación con el entorno espacial para la secuencia

En la Figura 3.5-4, en la parte inferior izquierda se observa la representación de las distancias en escala de grises y a la derecha la representación numérica de la distancia, para el caso analizado en la Figura 3.5-3.

Observando la dirección hacia donde las distancias se van haciendo más pequeñas podemos estimar la dirección en la que está desplazada la imagen con respecto al modelo.

Dependiendo de la configuración del sistema, esta etapa del módulo de comparación puede inhibirse, en tal caso el valor del parámetro $shift$ es cero.

3.5.1.2 Distancias

En el sistema propuesto existen dos alternativas para medir la similitud entre cada nueva muestra y los valores almacenados en el modelo. El primer método compara, mediante una variante de la norma L2, el vector *RGB* que representa el color de un píxel en el cuadro actual con los vectores *RGB* que representan cada una de las modas almacenadas en los modelos.

Por otro lado, con el objetivo de suministrar al proceso de comparación de robustez a sombras y reflejos, se propone el uso del método de comparación descrito en [22], que define un rango o vecindario alrededor de cada moda almacenada en forma de tronco de cono.

3.5.1.2.1 **Adaptación de la Norma L2**

La norma L2 o distancia Euclidea se basa en obtener el módulo de la magnitud de la diferencia de las dos imágenes a comparar (ver Sección 2.6.1.3). Esta implementación realiza

una normalización para obtener un valor máximo similar al valor máximo posible obtenible en un canal. Se define $d_\lambda(\mathrm{x},\mathrm{y})$ como la media de la normal L2 entre el vector *RGB* del píxel (x,y) en el cuadro actual, $\vec{I}(\vec{x};t)=\{I_R(\vec{\mathrm{x}};t),I_G(\vec{\mathrm{x}};t),I_B(\vec{\mathrm{x}};t)\}$ y el vector RGB que describe la moda: $\vec{\mu}_\lambda(\vec{x};\mathrm{t})=\{\mu_R(\vec{\mathrm{x}};t),\mu_G(\vec{\mathrm{x}};t),\mu_B(\vec{\mathrm{x}};t)\}$ almacenada para ese píxel en uno de los modelos.

$$d_\lambda(\vec{x};t)=\frac{\left\|\vec{I}(\vec{x},t),\overrightarrow{\mu_\lambda}(\vec{x},t))\right\|_2}{3}$$

Ecuación 10 : Media de la Norma L2

Nótese que dado que el rango de representación de cada canal en el espacio *RGB* es de 256 valores: $[0,255]$, la distancia por capa estará comprendida en un rango: $[0,d_{\max}]$, con $d_{\max}=255$.

3.5.1.2.2 **Distancia del cono**

Este método introducido en el *SoA* (ver Sección 2.6.1.4) está propuesto por Evangelio en [22] y su función es considerar la posibilidad de la existencia de sombras y cambios de iluminación previamente a la comparación entre el cuadro de vídeo y el modelo de fondo.

Sitúese el valor *RGB* de un píxel (x,y) en el espacio cartesiano $\mathbb{R}^3$ definido por sus canales *R*, *G* y *B* (ver 1 en Figura 3.5-5).

Para contemplar pequeñas variaciones en el valor del píxel en el tiempo, introducidas por ejemplo en el proceso de captación, se suele definir un rango de valores para los cuales se considera que el valor de píxel coincide. Este rango define una esfera alrededor del píxel (ver 2 en Figura 3.5-5).

Si asumimos que un cambio de iluminación local, tal como una sombra o un reflejo produce un decremento/incremento cuasi-proporcional en los tres canales, el vector *RGB* del píxel sometido al cambio de iluminación mantendrá una dirección similar, es decir, en coordenadas polares variará sustancialmente en magnitud pero levemente en elevación y azimut. Por ello, si utilizamos un cilindro en lugar de una esfera, se ampliaría el rango de valores en los márgenes inferior y superior, es decir, se tendrían en consideración píxeles con los mismos valores que en la esfera pero con menor o mayor iluminación (ver 3 en Figura 3.5-5).

Si en vez de una esfera, se utilizase un cilindro, se ampliaría el rango de valores en los márgenes inferior y superior, es decir, se tendrían en consideración píxeles con los mismos valores cromáticos que en la esfera pero sometidos a menor o mayor iluminación.

Teniendo en cuenta que la representación de los valores más oscuros tienen menos resolución que los claros, el cilindro no es el cuerpo volumétrico más indicado, y es donde toma fuerza el uso de un cono, que crece en anchura con la iluminación (magnitud) y decrece hacia el vértice del cono, en este caso emplazado en el origen de coordenadas (ver 4 en Figura 3.5-5).

Por último, queda definir los valores para determinar el tamaño del tronco de cono, es decir, la permisividad en magnitud o cambios de iluminación dentro del cono, mediante dos factores, *lower* (ver 5 en Figura 3.5-5) y *upper* (ver 6 en Figura 3.5-5), que determinan la longitud superior e inferior del tronco del cono respectivamente.

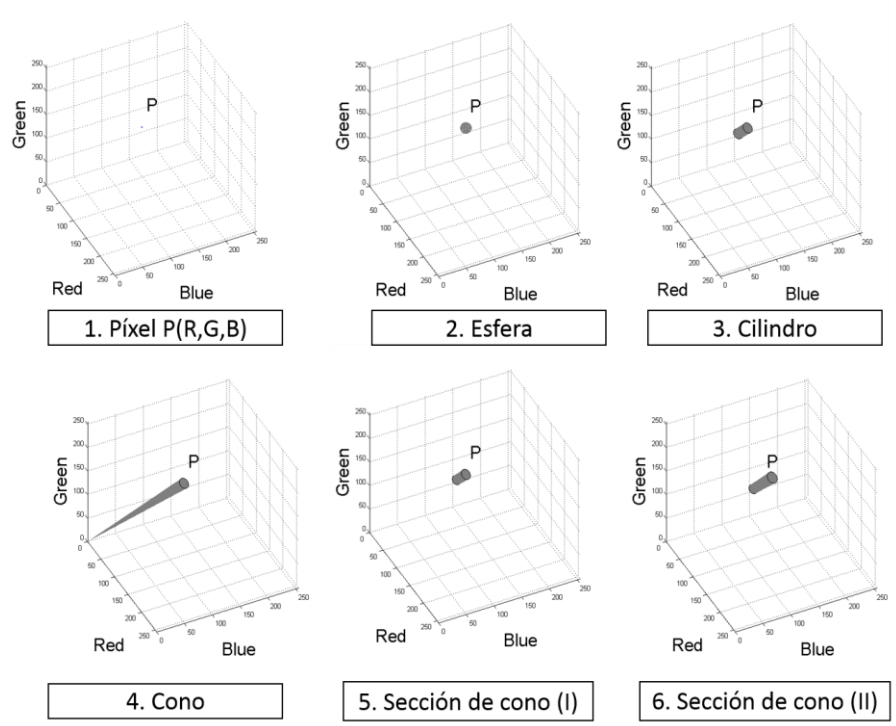


Figura 3.5-5 : Proceso de evolución del sistema métrico de distancia del cono alrededor de un píxel P

Si los valores del modelo y del cuadro se encuentran dentro del tronco de cono (o sección del cono) la distancia $d_\lambda(\vec{x};t)$ se considera nula, si por el contrario, la distancia del cuadro está fuera se calcula la distancia Euclídea media entre ambas, según se indica en Ecuación 10 (ver Sección 3.5.1.2.1).

3.5.2 Evolución de la confianza

A partir de las distancias calculadas para cada capa de cada modelo $d_\lambda(\vec{x};t)$ obtenidas en el módulo de comparación (ver Sección 3.5.1) se estudia la influencia de todas estas distancias en la evolución de la confianza.

Para ello se obtiene un factor de actualización de la confianza esperado en cada capa $\Theta_\lambda(\vec{x};t)$, siguiendo una aproximación exponencial:

$$\Theta_\lambda(\vec{x};t) = e^{\tilde{d}_\lambda(\vec{x};t)^{K_\lambda(\vec{x};t)}} - \delta, \quad \tilde{d}_\lambda(\vec{x};t) = 1 - \frac{d_\lambda(\vec{x};t)}{d_{\max}}$$

Ecuación 11: Cálculo del factor de actualización

, donde $\tilde{d}_\lambda(\vec{x};t)$ es la inversa de la distancia normalizada por el máximo valor alcanzable $d_{\max}$ y se define la constante de normalización $\delta = e^1 - 1$.

El factor de actualización tiene un rango de valores de entre -0,7 y 1, por lo que en un cuadro, la confianza puede aumentar de magnitud en 1 o reducirse 0,7 (ver Figura 3.5-6). La velocidad de crecimiento o decrecimiento con la distancia viene determinada por el factor de permisividad de cada capa $K_\lambda(\vec{x};t)$, como puede observarse en la Figura 3.5-6 .

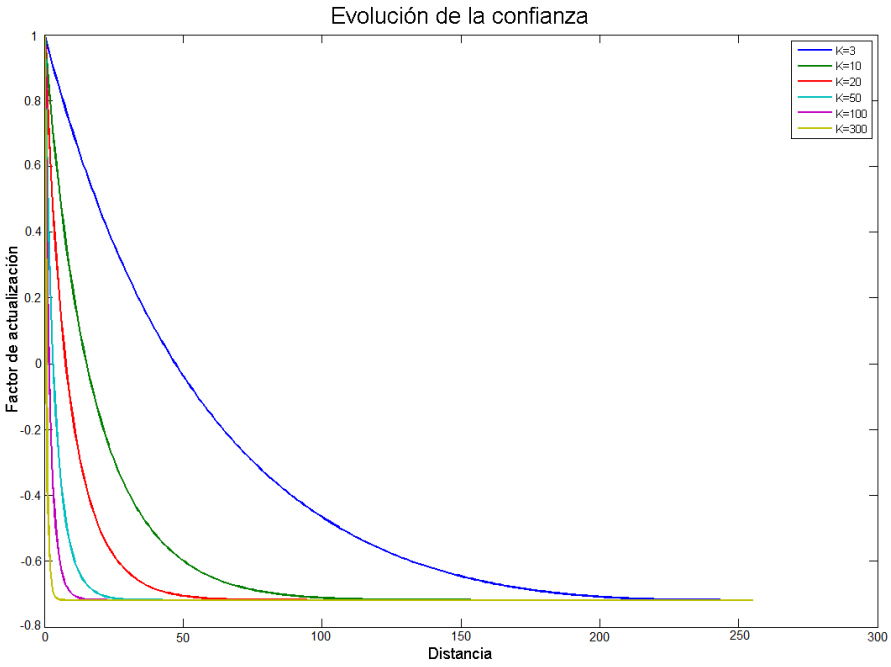


Figura 3.5-6: Factor de actualización en función del factor de permisividad K

El sistema diseñado utiliza valores de $K_\lambda(\vec{x};t)$ entre 3 y 300, como los representados en la Figura 3.5-6, en la que se observa como valores pequeños son muy permisivos (permiten más correspondencia entre el cuadro $\vec{I}(\vec{x};t)$ y la moda $\vec{\mu}_\lambda(\vec{x};t)$) mientras valores altos son muy restrictivos (permiten distancias muy pequeñas entre $\vec{I}(\vec{x};t)$ y $\vec{\mu}_\lambda(\vec{x};t)$).

3.5.3 Determinación de la correspondencia en función de la permisividad

La permisividad del sistema se define como la flexibilidad del sistema para fijar una correspondencia entre un píxel de la muestra $\vec{I}(\vec{x};t)$ y el píxel de la misma posición en las modas de los modelos $\vec{\mu}_\lambda(\vec{x};t)$. Conforme mayor sea el rango de valores para que ambos píxeles se correspondan, mayor será la permisividad del sistema.

El factor $K_\lambda(\vec{x};t)$ se denomina factor de permisividad puesto que define la laxitud del sistema; cuanto menor sea este factor mayor será el radio (ver Figura 3.5-7) y por ende más laxo será el criterio de correspondencia y más permisivo el sistema.

$K_\lambda(\vec{x};t)$ determina el radio $r_\lambda(\vec{x};t)$ de una esfera emplazada alrededor del píxel del modelo con el que se compara. En particular el radio $r_\lambda(\vec{x};t)$ se relaciona con la permisividad $K_\lambda(\vec{x};t)$ mediante la ecuación:

$$r_\lambda(\vec{x};t) = d_{\max}\left(\sqrt[K_\lambda(\vec{x};t)]{\ln(\Theta_{deseado}+\delta)} - 1\right)$$

Ecuación 12: Ecuación del radio en función de la permisividad

Esta ecuación, aparece como la solución óptima para la cual la confianza crece un valor esperado $\Theta_{deseado}$, aplicando $\Theta_\lambda(\vec{x};t) = \Theta_{deseado}$ en la Ecuación 11. Es decir, el radio es el valor de la distancia para la cual la confianza evoluciona $\Theta_{deseado}$, distancias inferiores al radio implicarán incrementos mayores de la confianza. Fijamos $\Theta_{deseado} = 0.5$.

La relación del radio con el factor de permisividad $K_\lambda(\vec{x};t)$ puede observarse en la Figura 3.5-7.

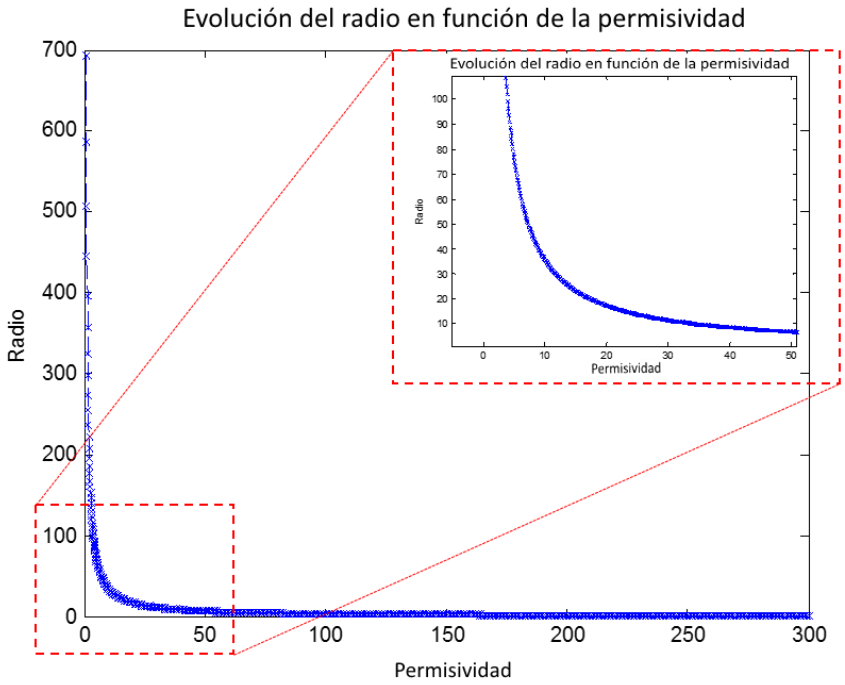


Figura 3.5-7: Evolución del radio en función de la permisividad

Como se observa en la Figura, cuando $K_\lambda(\vec{x};t)$ es pequeño el sistema es muy permisivo, y se vuelve más restrictivo, menor radio $r_\lambda(\vec{x};t)$, a medida que aumenta la permisividad $K_\lambda(\vec{x};t)$.

Si se utiliza la norma *L2* (ver Ecuación 10) para la comparación el radio $r_\lambda(\vec{x};t)$ definirá un entorno alrededor del modo del modelo para el cual se encontrará una correspondencia cuya bondad vendrá determinada por el valor de la distancia en sí.

En el caso de la distancia del cono define el radio de la base del cono, y por lo tanto define (junto con los parámetros *upper* y *lower*) el volumen para el cual la distancia es cero (ver Figura 3.5-8).

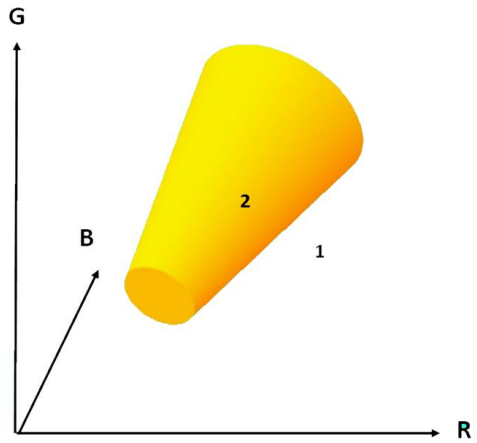


Figura 3.5-8: Determinación de la correspondencia utilizando la distancia del cono

En la Figura 3.5-8 se señalizan las dos zonas en las que se aplican las métricas, en la zona 1 se utiliza la norma L2, mientras que en la zona 2 la distancia es cero (se asume similitud entre $\vec{I}(\vec{x};t)$ y $\vec{\mu}_\lambda(\vec{x};t)$).

3.5.4 Estadísticos de correspondencia

La salida del proceso de comparación resulta en un conjunto de estadísticos de clase, así llamados puesto que se utilizan para definir tanto la clase del píxel como el proceso de actualización a realizar sobre el modelo.

Estos estadísticos son:

- **Capa para la que resulta la menor distancia a los modelos $\lambda^*(\vec{x};t)$.** Nótese que cada uno de los píxeles de la imagen de entrada puede resultar en una distancia mínima asociada a diferentes capas, es decir:

$$\lambda^*(\vec{x};t) = \arg\min_{\lambda}(d_\lambda(\vec{x};t)) \qquad \lambda = 0...L$$

Ecuación 13: Ecuación de capa en la que se encuentra la menor distancia

  Este factor $\lambda^*(\vec{x};t)$ tiene el valor de la capa en el que se haya encontrado la menor distancia de entre todas las activadas, tanto si la capa una de las del modelo de fondo como si es la única capa del modelo de frente. Para distinguir entre las capas 0 del modelo de fondo y de frente, el valor asignado para la capa del modelo de frente es L.
- **Distancia mínima al modelo $d_{\lambda^*}(\vec{x};t)$ para cada píxel.** Cada píxel de la imagen de entrada se compara con todas las capas del modelo para la cuales la posición de este píxel haya sido inicializada. Se calcula la menor distancia entre todas las disponibles.

- **Confianza asociada a la distancia mínima** $C_{\lambda^*(\vec{x};t)}(\vec{x};t)$ **para cada píxel**. La matriz $C_{\lambda^*(\vec{x};t)}(\vec{x};t)$ almacena para cada píxel la confianza de las capas obtenidas $\lambda^*(\vec{x};t)$**.**
- **Factor actualización asociado a cada capa** $\Theta_{\lambda}(\vec{x};t)$**.** Es el factor desarrollado en la Sección 3.5.2 y calculado como se indica en la Ecuación 11.
- **Factor asociado a la menor distancia** $\Theta_{\lambda^*(\vec{x};t)}(\vec{x};t)$**.** La matriz $\Theta_{\lambda^*(\vec{x};t)}(\vec{x};t)$ almacena para cada píxel el factor de actualización asociado a la menor distancia**.**

Estos estadísticos se utilizarán en los procesos de clasificación del píxel y actualización del modelo.

3.6 CLASIFICACIÓN DEL PÍXEL

Se propone clasificar el píxel en base a su evolución temporal respecto al modelo. Estas etiquetas guiarán los procesos de discriminación frente-fondo y de actualización selectiva.

Por un lado, las clases definidas buscan diferenciar entre píxeles en movimiento (dinámicos) y píxeles cuya representación no varía (estáticos). Para discriminar entre unos y otros se utilizará la confianza asociada a las modas almacenadas.

Por otro lado, se realiza una diferenciación entre los píxeles para los que no se ha observado apariencias similares con el modelo de fondo (frente) y píxeles para los que si se han encontrado similitudes con dicho modelo (fondo). Esta decisión se toma basándose en la correspondencia de una nueva muestra con los modelos (ver Sección 3.4) utilizando para obtener esta correspondencia uno de los esquemas descritos en la sección anterior (ver Sección 3.5.1.2).

Siguiendo estas premisas se definen las siguientes cuatro clases:

- Fondo estático.
- Fondo dinámico.
- Frente estático.
- Frente dinámico.

El proceso utiliza tres decisores para llevar a cabo esta clasificación. En el diagrama de flujo (ver Figura 3.6-1) se detalla la secuencia de aplicación de los tres decisores.

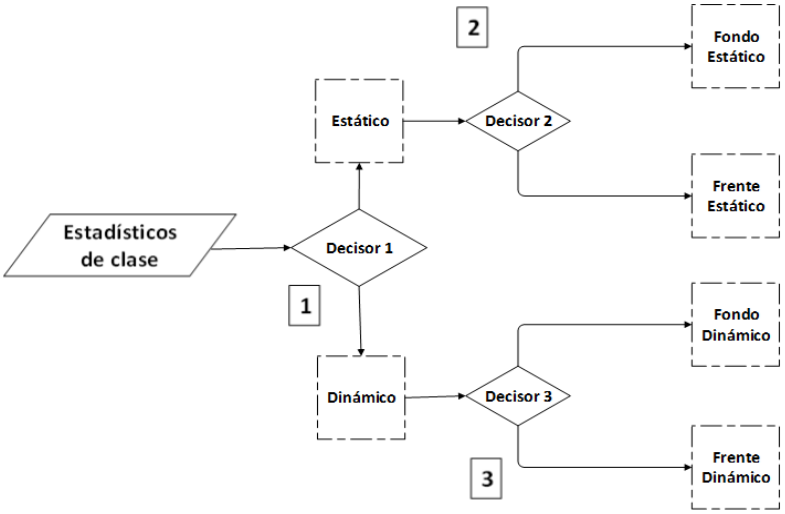


Figura 3.6-1: Diagrama de flujo del clasificador de clases

3.6.1 Decisor 1: estático / dinámico

El decisor 1 es función de los estadísticos: matriz de confianzas asociadas a la menor distancia entre todas las capas $C_{\lambda^*(\vec{x};t)}(\vec{x};t)$ y factor asociado a la menor distancia $\Theta_{\lambda^*(\vec{x};t)}(\vec{x};t)$. El decisor 1 es función también del umbral de decisión $C_{estatico}$.

La primera condición comprueba que la menor distancia obtenida se asocie a una verdadera correspondencia real, es decir, que la distancia sea menor que el radio asociado a ese píxel en la capa $\lambda^*(\vec{x};t)$ o de manera equivalente comprobar que el factor de actualización $\Theta_{\lambda^*(\vec{x};t)}(\vec{x};t)$ sea mayor o igual a cero.

Para los píxeles que superen esta primera condición se define un umbral de decisión basado en las confianzas para discriminar entre estático y dinámico. Como se ha definido anteriormente, un píxel estático es un píxel que no varía con el tiempo. Al no variar con el tiempo, la confianza asociada a la moda del píxel va aumentando por lo que para discriminar entre estático y dinámico se definen tomar esta decisión se basa en la confianza. En particular, para considerar un píxel como estático ha de superar un umbral $C_{estatico}$. Los píxeles excluidos de cualquiera de estas dos condiciones se consideran dinámicos:

$$\text{Clase del píxel} \begin{cases} \text{Estático} & \text{si } \Theta_{\lambda^*(\vec{x};t)}(\vec{x};t) \geq 0 \ \ y \ \ C_{\lambda^*(\vec{x};t)}(\vec{x};t) \geq C_{estático} \\ \text{Dinámico} & resto \end{cases}$$

Ecuación 14: Decisor 1 (estático o dinámico)

Una vez definidos los píxeles como estáticos o dinámicos, se clasificarán como fondo o frente. Los procesos son diferentes para dinámico y para estático.

3.6.2 Decisor 2: fondo estático / frente estático

Para discriminar entre fondo estático y frente estático se observa la correspondencia del píxel con el modelo, es decir, si la capa $\lambda^{*}(\vec{x};t)$ es parte del modelo de fondo se considera el píxel como fondo estático si, por el contrario, es la capa del modelo de frente se considera frente estático.

3.6.3 Decisor 3: fondo dinámico / frente dinámico

Por otra parte, para clasificar los píxeles denominados como dinámicos entre frente o fondo se diseña un clasificador que atiende a la distribución de color dentro de las componentes conexas que forman los píxeles clasificados como dinámicos. Para ello se construye el mejor modelo de fondo en cada instante temporal. Este modelo se construye con los modos que tienen en ese instante temporal la mayor confianza $\vec{\mu}_{\lambda^{**}(\vec{x};t)}(\vec{x};t)$, donde $\lambda^{**}(\vec{x};t)$ se obtiene como:

$$\lambda^{**}(\vec{x};t) = \arg\max_{\lambda}(C_{\lambda}(\vec{x};t))$$

Ecuación 15: Capa asociada a las confianzas más altas en el modelo de fondo

, solo para las capas del modelo de fondo.

Para llevar a cabo la clasificación, primero se extraen los blobs o agrupaciones de píxeles mediante un análisis de componentes conexas (ver Figura 3.6-2). Del conjunto de componentes conexas obtenidas se filtran aquellas para las cuales el rectángulo menor que las englobe sea inferior, en alguna de sus dimensiones a 10 píxeles.

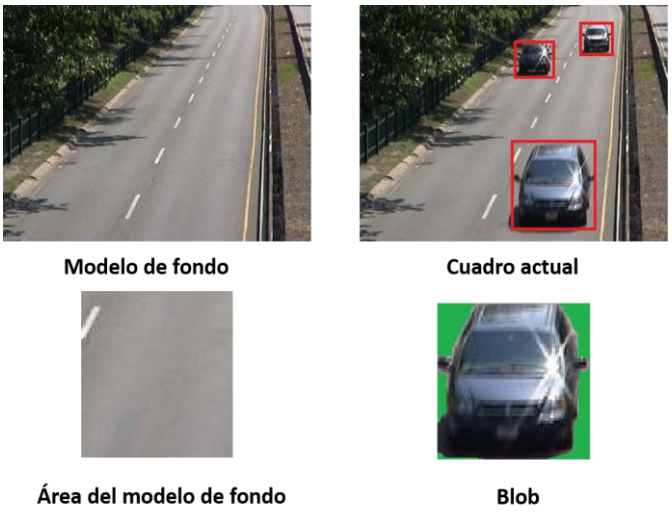


Figura 3.6-2: Extracción de áreas del modelo y del cuadro actual a comparar

Cada uno de los blobs restantes define un área tanto sobre el mejor modelo de fondo $\vec{\mu}_{\lambda^{**}(\vec{x};t)}(\vec{x};t)$ como sobre el cuadro actual $\vec{I}(\vec{x};t)$.

Para cada uno de estos blobs se estudian diferentes métricas:

- ***Structural SIMilarity (SSIM)*** [23]: Este método obtiene el grado de diferencia entre dos imágenes a partir de una combinación de comparaciones entre la luminancia, el contraste y la estructura de los píxeles de cada imagen.
- **Comparación de histograma a nivel de componente conexa:** Este método se basa en la comparativa de histogramas del área definida por los blobs. A partir de la correlación cruzada entre histogramas se obtiene un valor que determina el parecido de ambas regiones.

El método *SSIM* tuvo que ser descartado tras varias pruebas. A pesar de obtener unos resultados muy buenos, el gran coste computacional era una limitación para un sistema cuyo objetivo es trabajar en tiempo real.

El método de comparación de histogramas (ver Figura 3.6-3), calcula la diferencia entre los histogramas calculados usando como soporte cada *blob* sobre $\vec{\mu}_{\lambda^{**}(\vec{x};t)}(\vec{x};t)$ y sobre el cuadro actual $\vec{I}(\vec{x};t)$.

La siguiente figura muestra los histogramas de los áreas extraídos en la Figura 3.6-2, el área del modelo era carretera, y su uniformidad se muestra en el histograma izquierdo, mientras el blob del coche tiene más variedad de colores (los histogramas del Valor se muestran en un rango de 0 a 1 donde 0 es negro y 1 es blanco).

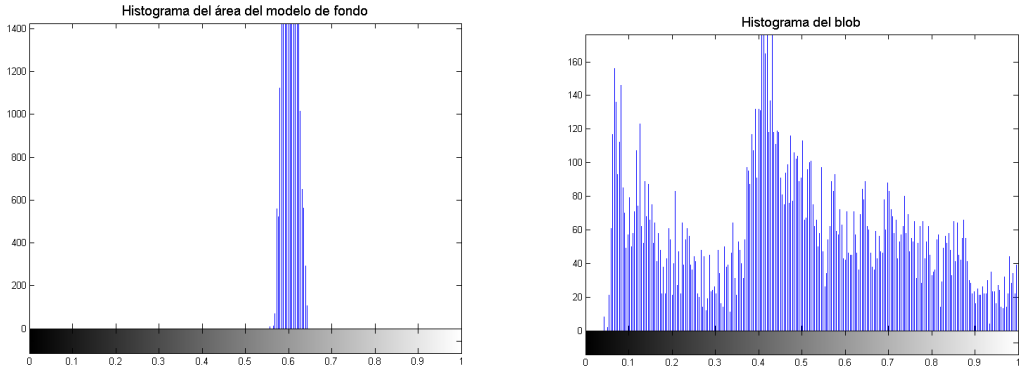


Figura 3.6-3: Representación de histogramas a comparar

La correlación cruzada devuelve un ratio entre 0 y 1 que determina el parecido entre los histogramas y por ende entre las regiones.

Utilizamos un nuevo umbral de decisión: Λ_3 que es un parámetro configurable. Cada blob se clasifica como fondo dinámico si la correlación cruzada es alta, mayor o igual que Λ_3, es decir si los histogramas son similares. En cambio, si la correlación es baja, menor que Λ_3, todos los píxeles que, conectados, forman el blob se clasificarán como frente dinámico.

3.6.4 Máscaras de salida

Como resultado de este módulo, se obtienen las máscaras de salida del sistema. Por un lado, una máscara de 4 valores en función de la clase determinada por el clasificador, llamada imagen de clases.

Por otro lado, también se crea una máscara binaria de color negro para el fondo y blanco para el frente como es normal en cualquier segmentador. La imagen de clases es una mejora de esta máscara binaria para observar mejor la clasificación en entornos multimodales.

En Figura 3.6-4 se puede observar la clasificación obtenida para el cuadro 794 del vídeo '*highway.avi*' de [37]. En la clasificación se aprecia como los coches son frente dinámico, el resto fondo estático y en algunas zonas como en los árboles se observa fondo dinámico. Finalmente, se agrupan fondo dinámico y estático para construir la máscara binaria donde el frente se representa de color gris y el fondo de color negro.

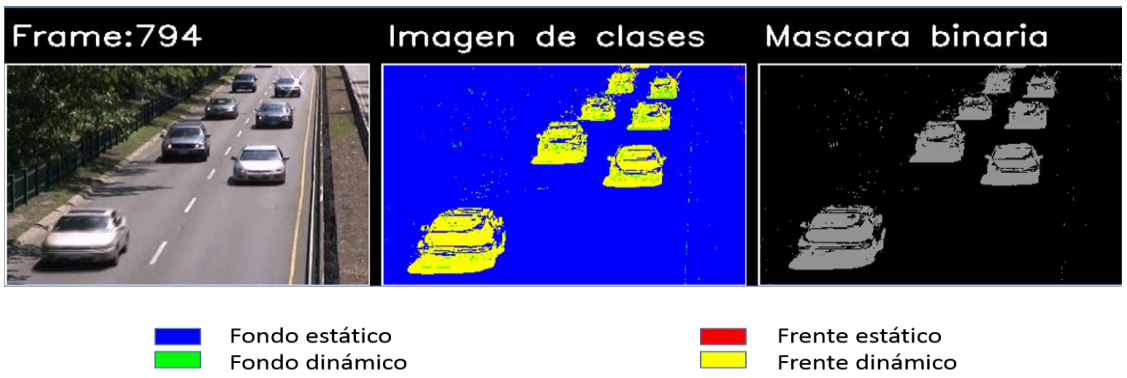


Figura 3.6-4: Imagen de clases y máscara binaria

3.7 PROCESO DE ACTUALIZACIÓN

En este apartado se define el proceso de actualización más indicado para cada una de las clases obtenidas y de sus estadísticos asociados. Como se ha definido en 3.4 cada el modelo de fondo tiene L capas. El estado inicial de las capas es inactivo (su confianza es -1) hasta que se inicializan localmente.

Se definen tres procesos de actualización: Parcial, Completa e Inicialización:

- **Actualización Parcial**: Actualiza únicamente las confianzas $C_\lambda(\vec{x};t)$ y el factor de permisividad $K_\lambda(\vec{x};t)$.
- **Actualización Completa**: Actualiza las modas $\mu_\lambda(\vec{x};t)$ así como las confianzas $C_\lambda(\vec{x};t)$ y el factor de permisividad $K_\lambda(\vec{x};t)$.
- **Inicialización**: Inicializa un nuevo área en una capa si existe una capa sin activar, o bien una nueva área en una capa activa si están todas activas.

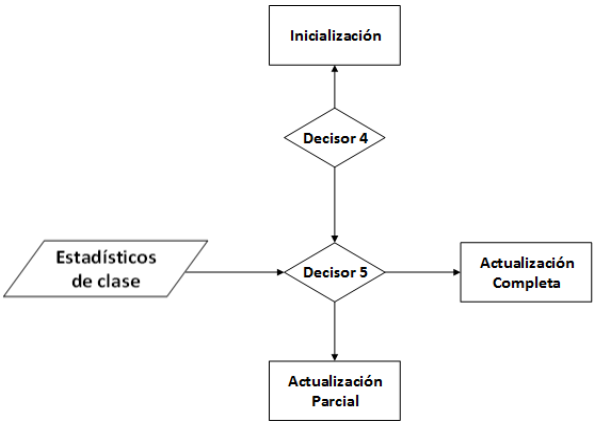


Figura 3.7-1: Diagrama de flujo del módulo de actualización

3.7.1 Decisor 4: Inicialización / Actualización

Este decisor utiliza el factor de actualización $\Theta_{\lambda^*(\vec{x};t)}(\vec{x};t)$ para clasificar la etiqueta de clasificación del píxel. Como se ha visto en la Sección 3.5.2 el valor de este factor varía entre -0.7 y 1. Se ha decidido un umbral fijado en $\Lambda_4 = 0.5$. Los valores inferiores a este valor, es decir, los que píxeles que resultan en altas distancias al modelo, se usarán para inicializar nuevas áreas (ver Sección 3.7.5).

Los píxeles superiores a este umbral serán píxeles actualizables y entrarán al decisor 5.

3.7.2 Decisor 5: Actualización parcial / Actualización completa

El objetivo es ir reemplazando los valores o los modos en el modelo, hasta que estos aumenten hasta una confianza mínima $C_{\min}$. Una vez llegados a este valor, las modas obtenidas representaran valores fiables para el modelo. Pero se mantendrá la actualización de confianza para que la confianza asociada a estos valores pueda disminuir y se repita el proceso de búsqueda de modos estables. Además, se contempla la bondad de la correspondencia en el proceso de actualización, para ello se introduce en el decisor el factor de actualización, es decir, correspondencias que resulten en factores de actualización altos (distancias pequeñas, modos similares) favorecerán que se realicen actualizaciones parciales.

Se propone el siguiente esquema para decidir si se realiza una actualización parcial:

$$\frac{C_{\lambda^*(\vec{x};t)}(\vec{x};t)}{\Theta_{\lambda^*(\vec{x};t)}(\vec{x};t) - 0.5} \leq C_{\min}$$

Ecuación 16 : Clasificador de actualizaciones

, es decir, se realiza una actualización parcial si el ratio entre la confianza asociada a la capa que resulta en la menor distancia entre el factor de actualización asociado a esa distancia normalizado al mínimo que se analiza en este decisor es menor que la confianza mínima. En caso contrario se realiza una actualización completa.

3.7.3 Inhibición de la actualización.

Nótese que es posible que píxeles clasificados como frente dinámico por el decisor 3 pueden en cambio generar sus mejores correspondencias con el modelo de fondo, por ello se inhibirá la actualización de estos pieles tanto en el modelo de fondo como en el modelo de frente, es decir, píxeles de fondo no actualizan el modelo de frente y píxeles de frente no actualizan el modelo de fondo.

3.7.4 Actualización del modelo

3.7.4.1 Actualización de la confianza

La actualización de la confianza consiste en sumar al valor de la confianza el factor de actualización (puede decrecer una magnitud de hasta 0.7 o aumentar hasta 1) (ver Figura 3.5-6), y comprobar que la confianza nunca sea negativa ni supere la confianza máxima.

$$C_{\lambda^*(\vec{x};t)}(\vec{x};t) = C_{\lambda^*(\vec{x};t-1)}(\vec{x};t-1) + \Theta_{\lambda^*(\vec{x};t)}(\vec{x};t)$$

Ecuación 17: Ecuación de actualización de la confianza

3.7.4.2 Actualización de los modos.

Los modos se actualizan sólo en los procesos de actualización completa mediante sustitución del modo en el modelo por el valor del píxel en el cuadro actual.

3.7.4.3 Actualización del factor de permisividad

La variación temporal que puede sufrir píxel estático no es la misma para píxeles emplazados en zonas planas de la escena que para aquellos emplazados en transiciones entre objetos. Con el objetivo de permitir que la permisividad sea mayor para las zonas que más varían y evitar así la clasificación incorrecta de estas zonas utilizamos un sistema de media móvil similar al descrito en la Sección 2.5.1.1.

$$K_\lambda(\vec{x};t) = (\alpha)K_\lambda(\vec{x};t) + (1-\alpha)K'_\lambda(\vec{x};t)$$

Ecuación 18 : Actualización del factor de permisividad

, donde $K'_\lambda(\vec{x};t)$ resulta de despejar $K_\lambda(\vec{x};t)$ de la ecuación del factor de actualización (ver Ecuación 11) para un factor de actualización deseado $\Theta_{deseado}$:

$$\Theta_{deseado} = e^{\tilde{d}_\lambda(\vec{x};t)^{K'\lambda(\vec{x};t)}} - \delta \quad \rightarrow K'_\lambda(\vec{x};t) = \frac{\ln(\ln(\delta + \Theta_{deseado}))}{\ln(\tilde{d}_\lambda(\vec{x};t))},$$

Ecuación 19: Ecuación del factor que actualiza a K

, es decir, desplazamos ligeramente el factor de permisividad de manera que distancias similares a la observada resulten poco a poco en incrementos de la confianza.

3.7.5 Inicialización del modelo

La inicialización de las modas se establece con el valor del cuadro actual, la confianza asociada se inicializa con valor 0, el factor de permisividad adopta el valor $K_{\min}$ establecido y el factor de actualización a 1.

Las capas solo se activan en las zonas que hayan sido inicializadas. En el caso en el que no existan capas inactivas disponibles para un píxel a inicializar, se remplaza el modo con la menor confianza asociada de entre todas las capas.

3.7.6 Consideraciones adicionales a la actualización

El módulo de actualización se encarga de renovar los modelos con el paso del tiempo. En este módulo se solucionan problemas como el "inicio en caliente" donde un objeto de frente se inicializa como modelo de fondo. Con el paso del tiempo, al desaparecer y no reaparecer los modos que describían al objeto, el modelo de fondo se actualiza junto con su imagen de confianzas asociada.

Por otro lado, se propone un esquema para la reclasificación de objetos abandonados, es decir, aquellos objetos de frente que permanecen estáticos tanto tiempo que pueden ser considerados como fondo. El proceso de actualización bebe del módulo de comparación y del módulo de clasificación de clases. Se ha definido que esto suceda si la confianza de un píxel de frente estático aumenta hasta alcanzar la confianza máxima $C_{\max}$.

Por ejemplo, si un coche se para en un semáforo, su confianza aumentará hasta que vuelva a moverse. Esa confianza no debe alcanzar el valor de $C_{\max}$ porque los píxeles con las modas del coche se clasificarían como fondo. Sin embargo, un coche que aparca irá aumentando su confianza hasta alcanzar $C_{\max}$ e incluirse en el modelo de fondo, de ahí la importancia de este factor.

CAPÍTULO 4: IMPLEMENTACIÓN Y DESARROLLO

El trabajo ha sido desarrollado utilizando el lenguaje de programación orientada a objetos *C++* para la realización del algoritmo utilizando la librería *OpenCv* diseñada para el tratamiento de imágenes. Además, para la implementación de la *GUI* se ha utilizado la herramienta *QT Designer* (ver Sección 2.9*).*

Por otra parte, la estructura del sistema consta de tres niveles de jerarquía (ver Figura 4-1):

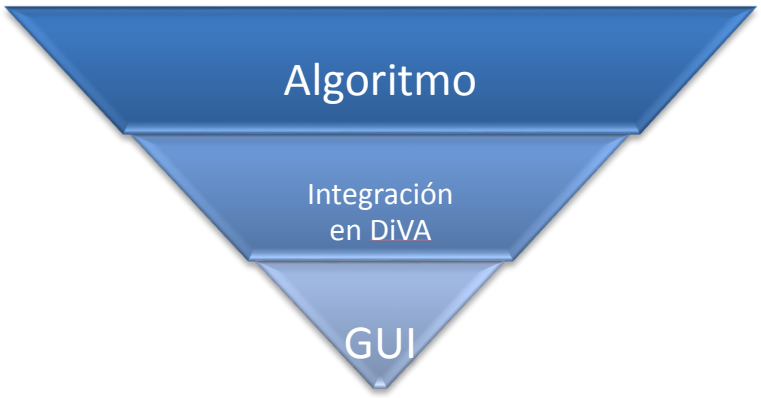


Figura 4-1: Pirámide de niveles de implementación

- **Nivel 1: Algoritmo**.

  Es el nivel más bajo de la pirámide. Se compone de las funciones en *C++* que forman el algoritmo. Lee las capturas de vídeo directamente de un archivo en un directorio. Los parámetros se establecen en el propio código antes de ejecutar la aplicación.

- **Nivel 2: Integración en *DiVA.***

  Tras la finalización del algoritmo, éste se integra con las funciones de la librería *DiVA* facilitadas en el laboratorio *VPU*. En este nivel las capturas se obtienen a partir de un servidor de cuadros, tras haber establecido la dirección IP del servidor y su puerto. Los parámetros se establecen en el código antes de la ejecución como ya ocurría en el nivel anterior.

- **Nivel 3: GUI.**

  Por último, el nivel 3 introduce una interfaz gráfica para poder interactuar con el algoritmo mientras éste se ejecuta. En este caso, la captura de vídeo puede obtenerse tanto del servidor de cuadros de *DiVA*, como de un fichero de extensión *AVI*. Además, es posible cambiar la configuración de los parámetros durante la ejecución, ya que el algoritmo y la secuencia se ejecutan en hilos distintos.

Por otro lado, cabe destacar que el sistema está construido en forma de módulos (Ver Figura 4-2). De esta manera, si es necesario modificar un método de los utilizados por el algoritmo, únicamente será necesario cambiar ese módulo dentro del código sin afectar al resto.

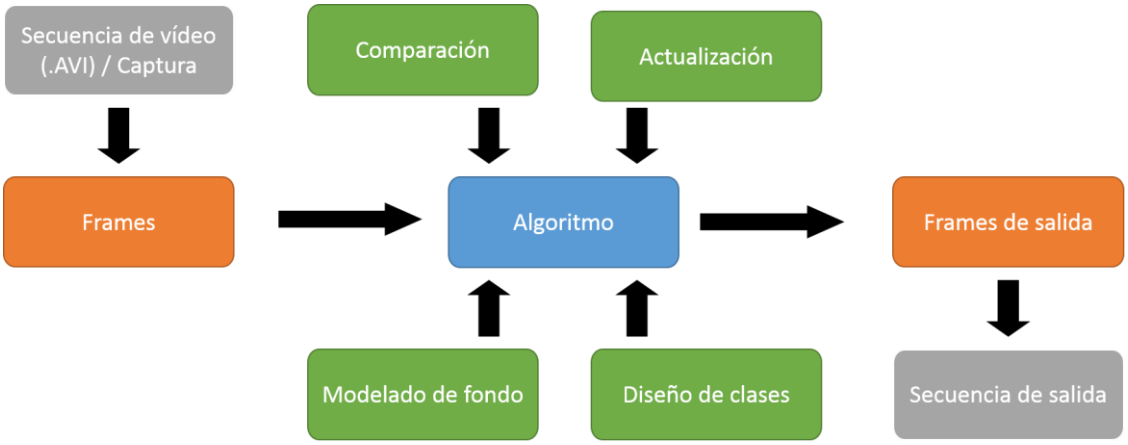


Figura 4-2: Diagrama de módulos del sistema

Como se observa en la Figura 4-2, se pueden cambiar los métodos se actualización, comparación, modelado de fondo y diseño de clases. También es posible cambiar el método de obtención de cuadros y las imágenes de salida que se deseen.

4.1 INTERACCIÓN CON EL SISTEMA

Como se observará en la sección de resultados, no todas las secuencias requieren los mismos parámetros de configuración. El sistema diseñado permite cambiar una serie de parámetros que conllevan cambios en la operación del algoritmo. Para poder interactuar en tiempo real con el sistema, se ha diseñado una interfaz de usuario.

4.1.1 Parámetros configurables

En el algoritmo existen unos parámetros predeterminados fijos que son invariantes. Sin embargo, otros son configurables para dar mayor consistencia al sistema. A continuación, se van a describir estos parámetros para comprender mejor su funcionalidad:

- **Confianza mínima:** Se utiliza para decidir si el píxel es dinámico o estático y para la actualización de modelos. El modelo necesita este valor de confianza mínimo $C_{\min}$ para poder actualizarse (ver Sección 3.4). Además se fija $C_{estatico} = 2C_{\min}$.
- **Omega (Ω):** Este factor multiplica la confianza mínima para obtener la máxima confianza $C_{\max} = \Omega C_{\min}$ que puede alcanzar un modelo en el sistema.
- **Número de modelos:** Es el número máximo de capas disponibles en modelo de fondo L. Debe ser mayor cuanto más multimodalidad presente el vídeo (ver Sección 3.4).

- ***Shift*:** Parámetro que define la matriz de comparación con píxeles vecinos (ver Sección 3.5.1.1).
- ***Lower*:** Parámetro de rango entre 0.5 y 1 que representa la extensión del tronco de cono inferior siendo 0.5 la máxima extensión y 1 sin extensión (ver Sección 3.5.1.2.2).
- ***Upper*:** Parámetro de rango entre 1 y 1.5 que representa la extensión del tronco de cono superior siendo 1.5 la máxima extensión y 1 sin extensión (ver Sección 3.5.1.2.2).
- **Kmin, Kmax:** Parámetros que determinan el rango de permisividad del sistema (ver Sección 3.5.1.2). Es decir valores del factor de permisividad por encima de **Kmax** saturan.
- Λ_3 : Parámetro que define donde se encuentra el umbral del clasificador entre fondo dinámico o frente dinámico (ver Sección 3.6.3).
- $\Theta_{deseado}$ **:** Factor de actualización deseado para la actualización del parámetro de permisividad del sistema (ver Sección 3.7.4.3).

4.1.2 Interfaz gráfica

Se ha desarrollado una interfaz gráfica (ver Figura 4.1-1) a partir del algoritmo y de la implementación en *DiVA*. La funcionalidad de esta *GUI* permite al usuario interaccionar con el sistema dando la oportunidad de configurar cada parámetro para el uso deseado o seleccionar el tipo de entrada al algoritmo: una cámara en particular de la red disponible o un vídeo almacenado en el equipo de procesado.

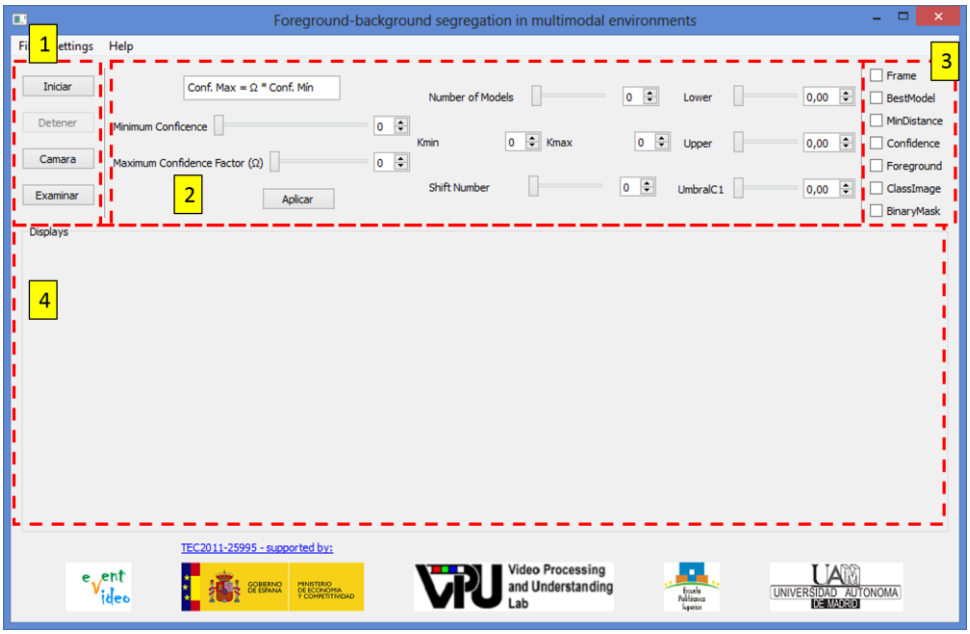


Figura 4.1-1: Interfaz gráfica

El primer módulo de la *GUI* lo forman cuatro botones: Iniciar, Detener, Cámara y Examinar (ver Figura 4.1-2). Esta parte tiene el control sobre la secuencia de vídeo y la descripción de cada botón es la siguiente:

- **Iniciar:** Tras haber cargado una secuencia de vídeo o recibido cuadros del servidor, al pulsar este botón se inicia el algoritmo.

- **Detener:** Al pulsar este botón se paraliza el algoritmo, dando la posibilidad de cambiar de secuencia o servidor de cuadros. Mientras el algoritmo está ejecutándose este botón se encuentra deshabilitado.

- **Cámara:** Si se desean recibir las imágenes de una cámara local (webcam, *cámaras PTZ* de la EPS-UAM,…) es necesario pulsar este botón. Una vez pulsado nos permite guardar una configuración de cámara introduciendo: el nombre de la cámara, la dirección *IP* del servidor de cuadros y el puerto por el que se envían.

- **Examinar:** Este botón permite al usuario seleccionar un vídeo de extensión *AVI* desde la carpeta en el que se encuentre, cargando el vídeo para poder ser ejecutado por el algoritmo posteriormente.

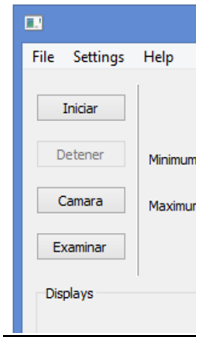


Figura 4.1-2 : Iniciar/detener secuencia en la GUI

El segundo módulo (ver Figura 4.1-3) permite modificar los parámetros (ver Sección 4.1.1) mientras se ejecuta el algoritmo. Cuando se ejecuta el algoritmo, aparecen unos parámetros predeterminados que funcionan bien en cualquier vídeo. A partir de estos parámetros predeterminados, pueden modificarse como se desee desplazando la barra o bien introduciendo el valor desde el teclado. Existe un botón llamado Aplicar. Los cambios realizados en los parámetros no tendrán ningún efecto hasta que no se pulse dicho botón.

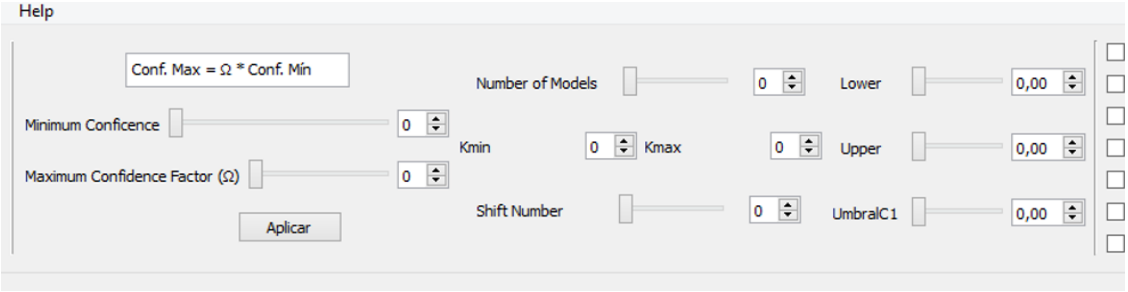


Figura 4.1-3: Configuración de parámetros en la GUI

Por otra parte, el tercer módulo (ver Figura 4.1-4) permite la visualización de las imágenes que se deseen. Estas son las imágenes que la GUI permite representar:

- **Cuadro Actual** $\vec{I}(\vec{x};t)$: El cuadro procesado en ese instante de tiempo.
- **Mejor modelo de fondo:** Imagen compuesta con los píxeles de mayor confianza entre los modelos de fondo C_{λ^*}.
- **Distancia mínima** $d_{\lambda}(\vec{x};t)$: Imagen de diferencias entre el frame actual y el modelo. Es mínima debido a que se compone de las menores distancias del vecindario.
- **Mejor confianza** $C_{\lambda^*(\vec{x};t)}(\vec{x};t)$: Representación de la confianza asociada a los píxeles del mejor modelo. Cuanto más claro es el píxel mayor confianza tiene, siendo blanco cuando alcanza la confianza máxima.
- **Modelo de frente** $\mu_L(\vec{x};t)$: Muestra el modelo de frente.
- **Imagen de clases:** Representación de las clases de la imagen: fondo estático (azul), fondo dinámico (verde), frente estático (rojo) y frente dinámico (amarillo) (ver Tabla 4.1-1).
- **Máscara Binaria:** Representación discriminante entre frente (blanco) y fondo (negro).

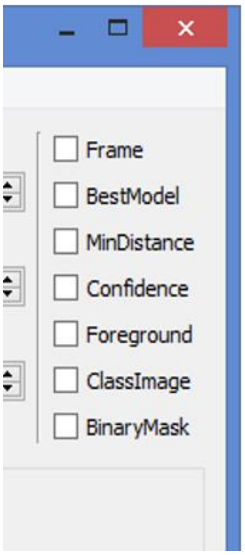


Figura 4.1-4: Selección de imágenes en la GUI

Por último, el cuarto módulo (ver Figura 4.1-5) es un espacio reservado para la visualización de las imágenes deseadas. Las imágenes pueden redimensionarse disminuyéndose en la interfaz o ampliarse pero su tamaño máximo será el tamaño máximo del vídeo para evitar que al interpolar para visualizarse aparezcan resultados ficticios.

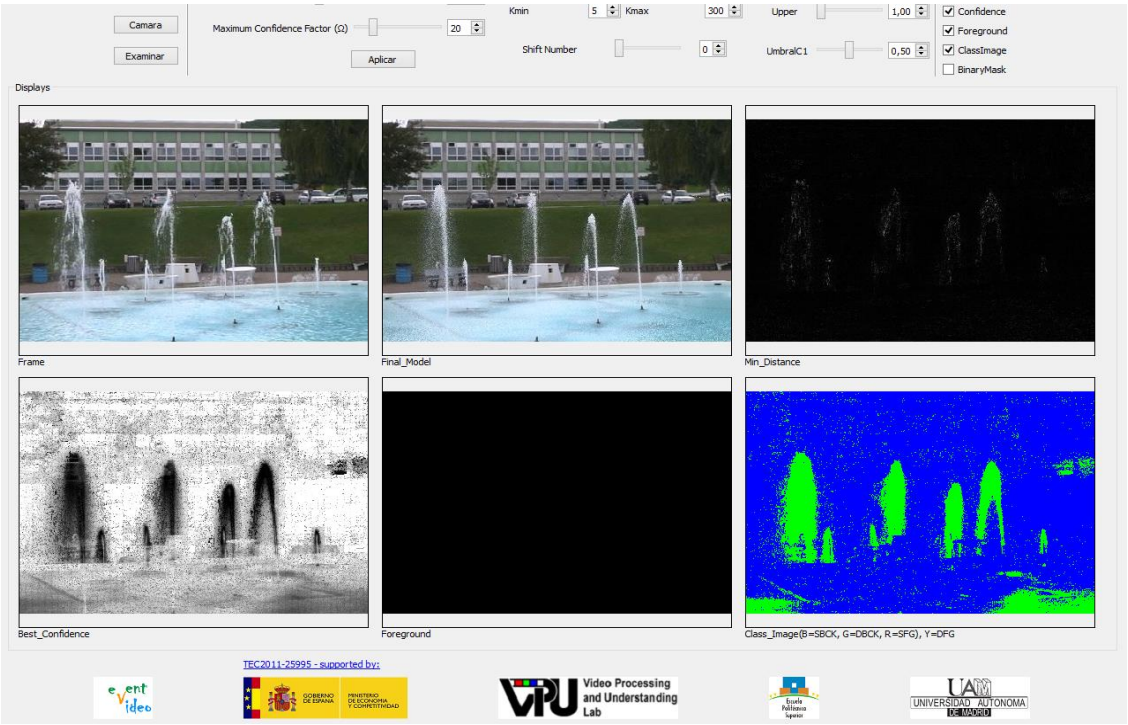


Figura 4.1-5: Visualización de resultados en la GUI

Fondo estático	Fondo dinámico	Frente estático	Frente dinámico
Azul	Verde	Rojo	Amarillo
RGB (0,0,255)	RGB (0,255,0)	RGB (255,0,0)	RGB (255,255,0)

Tabla 4.1-1: Coloreado de la imagen de clases

CAPÍTULO 5: PRUEBAS Y RESULTADOS

Para obtener los resultados del sistema se han realizado unas pruebas cualitativas y cuantitativas de evaluación para determinar la calidad real de la segmentación obtenida.

Para ello se han utilizado las máscaras de clase que devuelve el sistema ante diferentes vídeos existentes en una base de datos de uso público (ver Sección 5.1). Para cada secuencia de vídeo, se utiliza una máscara de segmentación de referencia, llamada *ground-truth* para compararla con la máscara del sistema, tras agrupación de clases ignorando el movimiento (frente: frente estático + frente dinámico, fondo: fondo estático + fondo dinámico). De esta comparación se obtienen una serie de estadísticos comúnmente utilizados en los esquemas de clasificación automático y mediante ellos mediadas cuantitativas de calidad que nos permiten comparar el sistema diseñado con otros sistemas del estado del arte.

5.1 DESCRIPCIÓN DE LAS SECUENCIAS DE PRUEBA

5.1.1 Base de datos

Para analizar el sistema se ha utilizado la base de datos *ChangeDetection* [37][23]. Esta base de datos permite evaluar la calidad del algoritmo y clasificarla en comparación con otros métodos analizados en el *SOA* (ver Capítulo 2: Estado del Arte).

Las secuencias utilizadas para analizar la calidad del sistema se clasifican en diferentes categorías (ver Tabla 5.1-1), que atienden a los factores críticos o limitaciones que suelen darse en este tipo de sistemas. Cada categoría está compuesta por diferentes vídeos, de forma que se estudia la robustez del sistema en diferentes escenarios.

ID	Categoría	Descripción
1	Baseline	Presenta una mezcla de las 4 categorías siguientes.
2	CameraJitter	Presenta vídeos en espacios interiores y exteriores con cámaras inestables (sufren vibración).
3	DynamicBackground	Las escenas de vídeo tienen fondo en movimiento.
4	IntermittentObjectMotion	Estos vídeos tienen objetos que se mueven o se detienen durante largos periodos de tiempo.
5	Shadow	Presenta vídeos en interiores y exteriores con presencia de sombras y/o cambios de iluminación.
6	Thermal	Vídeos capturados con cámaras infrarrojas.

Tabla 5.1-1: Descripción de las categorías de la base de datos

En algunas categorías, el sistema compara únicamente en un área de interés ROI, donde es interesante comprobar el comportamiento del sistema. Por ejemplo, la categoría 4 (*IntermittentObjectMotion*) analiza únicamente las regiones donde se encuentran los objetos que permanecen abandonados como por ejemplo en las secuencias '*abandonedBox.avi*' o '*parking.avi*' (ver columna izquierda y derecha de la Figura 5.1-1 respectivamente).

Figura 5.1-1: ROI de secuencias de la base de datos

5.1.2 Mejor algoritmo existente en cada categoría

Los resultados de varios algoritmos están disponibles en la página web de *ChangeDetection* [46]. En la tabla se incluyen las mejores aproximaciones para cada categoría:

ID	Categoría	Aproximación
1	Baseline	SC-SOBS [34]
2	CameraJitter	GPRMF [47]
3	DynamicBackground	STLBP [48]
4	IntermittentObjectMotion	SGMM-SOD [49]
5	Shadow	GPRMF [47]
6	Thermal	STLBP [48]
7	Overall	SuBSENSE [50]

Tabla 5.1-2: Aproximaciones con mejores resultados para cada categoría

La categoría 7 (Overall) tiene la media de todos los resultados para conocer la aproximación más robusta a todos los factores críticos en general.

5.2 PRUEBAS

5.2.1 Configuración de los parámetros del sistema

Para analizar la calidad del sistema y compararlo con otros sistemas que utilizan *BS* se ha optado por ejecutar el sistema con diferentes configuraciones para comprobar la influencia que tienen los parámetros en el sistema.

- **Configuración 1:**

Es la configuración con el valor de los parámetros definidos por defecto. Todas las secuencias de las categorías se han ejecutado utilizando los mismos valores de parámetros (ver Tabla 5.2-1).

Parámetro \ ID	1	2	3	4	5	6
Nº de Capas	3	3	3	3	3	3
Confianza mínima	20	20	20	20	20	20
Omega	10	10	10	10	10	10
Th	0.5	0.5	0.5	0.5	0.5	0.5
Upper	1	1	1	1	1	1
Lower	1	1	1	1	1	1
α	0.99	0.99	0.99	0.99	0.99	0.99
K mínima	5	5	5	5	5	5
K máxima	300	300	300	300	300	300
Shift	0	0	0	0	0	0
Λ_3	0.25	0.25	0.25	0.25	0.25	0.25

Tabla 5.2-1: Configuración de los parámetros para evaluación de resultados

- **Configuración 2:**

Esta configuración introduce modifica algunos parámetros para comprobar de qué forma afectan al sistema. En la categoría 2 (*cameraJitter*) se utiliza la comparación de píxeles con el vecindario debido al movimiento de la cámara por las vibraciones, en la categoría 3 (*dynamicBackground*) que utiliza un mayor número de capas debido a la multimodalidad que se encuentra en los escenarios de dicha categoría, en la categoría 4 (*intermittentObjectMotion*) se modifica el factor que hace que el frente se incorpore al modelo de fondo y la categoría 5 (*shadows*), en la que se utiliza la distancia del cono en lugar de la distancia Euclídea (ver Tabla 5.2-2).

Parámetro \ ID	1	2	3	4	5	6
Nº de Capas	3	3	5	3	3	3
Confianza mínima	20	20	20	20	20	20
Omega	10	10	10	20	10	10
Th	0.5	0.5	0.5	0.5	0.5	0.5
Upper	1	1	1	1	1.1	1
Lower	1	1	1	1	0.7	1
α	0.99	0.99	0.99	0.99	0.99	0.99
K mínima	5	5	5	5	5	5
K máxima	300	300	300	300	300	300
Shift	0	3	0	0	0	0
Λ_3	0.25	0.25	0.25	0.25	0.25	0.25

Tabla 5.2-2: Configuración 2

5.2.2 Estadísticos

Los estadísticos utilizados para evaluar el sistema se basan en la comparación de máscaras de frente generadas por el algoritmo y el *ground-truth* para cada cuadro de cada secuencia. La comparación se realiza a nivel de píxel utilizando las siguientes medidas:

- **Verdaderos positivos (*TP*):** El mismo píxel (x, y) de los cuadros a comparar tiene el valor de frente *FG*.
- **Verdaderos negativos (*TN*):** El mismo píxel (x, y) de los cuadros a comparar tiene el valor de fondo *BCK*.
- **Falsos positivos (*FP*):** El píxel (x, y) del sistema tiene valor *FG* mientras el *ground-truth* vale *BCK*.
- **Falsos negativos (*FN*):** El píxel (x, y) del sistema tiene valor *BCK* mientras el *ground-truth* vale *FG*.

En la Tabla 5.2-3 podemos visualizar las medidas descritas y sus relaciones con ambas máscaras:

		Ground - Truth		
		Positivos reales	Negativos reales	
Sistema de segmentación	Positivos detectados	TP	FP	TP + FP
	Negativos detectados	FN	TN	FN+TN
		TP+FN	FP+TN	

Tabla 5.2-3: Parámetros estadísticos

A partir de estas medidas, se pueden obtener otras medidas utilizadas generalmente en el SOA de segmentación de objetos:

- **Recall (*RC*):** Define el número de píxeles correctos detectados de un tipo con respecto al total real de ese tipo dado por el *ground-truth*.

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

Ecuación 20 : Recall

- **Especificidad (*SP*):** Es la capacidad del sistema para excluir una condición, es decir, la proporción de negativos que se han identificado correctamente.

$$SP = \frac{TN}{TN + FP}$$

Ecuación 21: Especificidad

- **Tasa de falsos positivos (*FPR*):** Es la posibilidad de obtener un resultado erróneo si el resultado es negativo.

$$FPR = \frac{FP}{FP + TN}$$

Ecuación 22: Tasa de falsos positivos

- **Tasa de falsos negativos (*FNR*):** Es la probabilidad de obtener un resultado acertado si el resultado es positivo.

$$FNR = \frac{FN}{TP + FN}$$

Ecuación 23: Tasa de falsos negativos

- **Porcentaje de malas clasificaciones (*PWC*):** Es la probabilidad de obtener un resultado positivo si el resultado es negativo o un resultado negativo si el resultado es positivo.

$$PWC = \frac{FN + FP}{TP + FN + FP + TN} \times 100$$

Ecuación 24: Porcentaje de malas clasificaciones

- **Precisión (*PRC*):** Se define como el número total de píxeles correctos de un tipo con respecto al total de los píxeles de ese tipo detectados por el sistema.

$$PRC = \frac{TP}{TP + FP}$$

Ecuación 25: Precisión

- **F-Score (*FSC*) o F-measure:** Combina la correcta detección del algoritmo respecto a la detecciones realizadas por él mismo (Precisión) con la correcta detección del algoritmo y la realidad (Recall) dándoles a ambas el mismo peso.

$$FSC = \frac{2 \times PRC \times RC}{PRC + RC}$$

Ecuación 26: F-Score

5.3 RESULTADOS

En esta Sección se va a realizar la comparativa entre los resultados del sistema frente a los resultados de las mejores aproximaciones para cada categoría (ver Sección 5.1.2).

5.3.1 Resultados cuantitativos

Las tablas Tabla 5.3-1 y Tabla 5.3-2 contienen los resultados numéricos obtenidos aplicando los estadísticos descritos en la Sección 5.2.2.

- **Configuración 1:**

	RC	SP	FPR	FNR	PWC	PRC	FSC
1	0.8348	0.9803	0.0197	0.1651	2.4674	0.7062	0.7515
2	0.6003	0.9590	0.0409	0.3996	5.3851	0.4911	0.5028
3	0.5278	0.9953	0.0046	0.4721	0.9330	0.4282	0.4289
4	0.4470	0.9764	0.0235	0.5529	6.2388	0.5630	0.4620
5	0.7029	0.9703	0.0296	0.2970	3.8670	0.5658	0.6081
6	0.5077	0.9862	0.0137	0.4922	3.4315	0.6861	0.5400
7	0.6034	0.9779	0.0220	0.3965	3.720	0.5734	0.5489

Tabla 5.3-1: Parámetros de configuración 1

- **Configuración 2:**

	RC	SP	FPR	FNR	PWC	PRC	FSC
1	0.8348	0.9803	0.0197	0.1651	2.4674	0.7062	0.7515
2	0.4093	0.9920	0.0079	0.590	3.069	0.7880	0.5086
3	0.5363	0.9961	0.0038	0.4636	0.8579	0.5207	0.4791
4	0.3404	0.9347	0.0652	0.6595	11.1064	0.3356	0.3095
5	0.6863	0.9769	0.0230	0.3136	3.4212	0.5817	0.6049
6	0.5077	0.9862	0.0137	0.4922	3.4315	0.6861	0.5400
7	0.5525	0.9777	0.0222	0.4474	4.0589	0.6031	0.5323

Tabla 5.3-2: Parámetros de configuración 2

Por otra parte, en la Figura 5.3-1, Figura 5.3-2 y Figura 5.3-3 se muestran los resultados del Recall, de la Precisión y de la F-Score descritos en la Sección 5.2.2, que son los estadísticos más representativos en general.

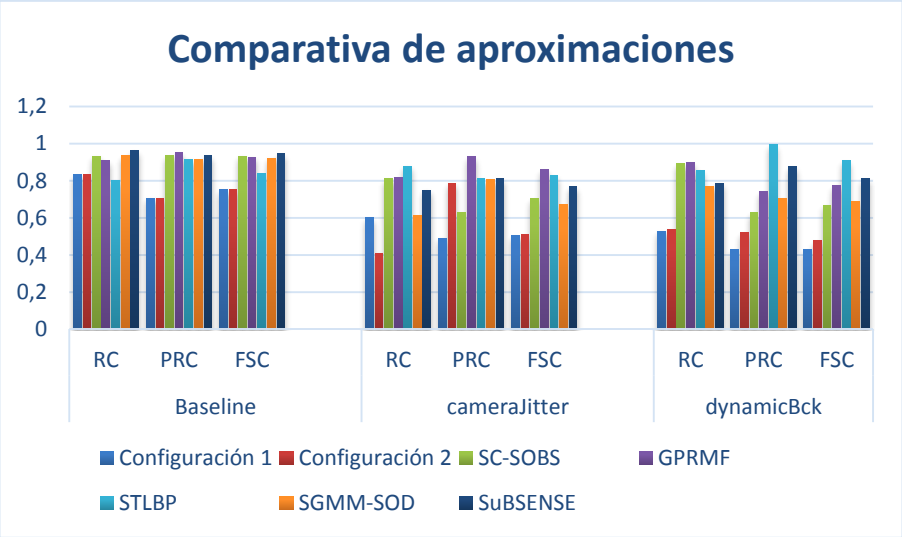


Figura 5.3-1: Gráfica comparativa entre categorías 1,2 y 3

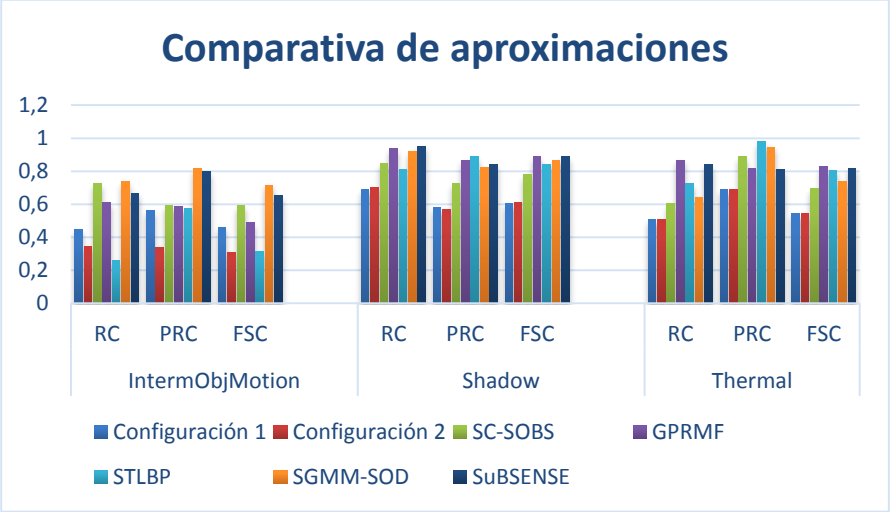


Figura 5.3-2 Gráfica comparativa entre categorías 4,5 y 6

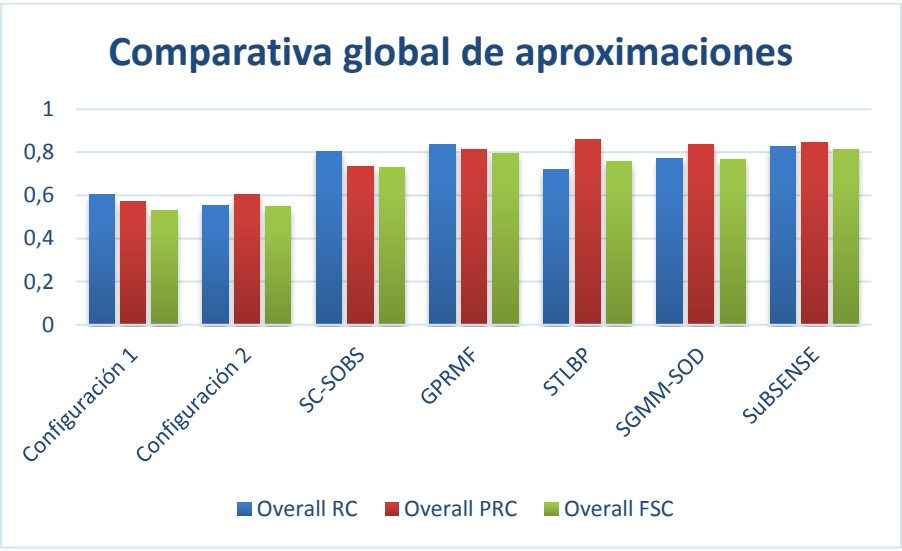


Figura 5.3-3: Gráfica comparativa global

5.3.2 Resultados cualitativos

A continuación se muestra un cuadro de un vídeo de cada categoría junto a su *ground-truth* para obtener una visión más intuitiva de los resultados obtenidos en las secuencias.

En la Tabla 5.3-3 se menciona el frame utilizado para la representación de los resultados cualitativos de la Tabla 5.3-4.

ID	Nº de frame	Nombre de secuencia
1	1679	highway.avi
2	1454	traffic.avi
3	906	canoe.avi
4	3765	abandonedBox.avi
5	1734	copyMachine.avi
6	2095	library.avi

Tabla 5.3-3: Cuadros utilizados para representar resultados cualitativos

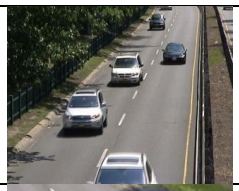
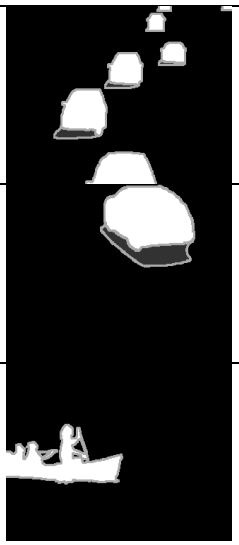
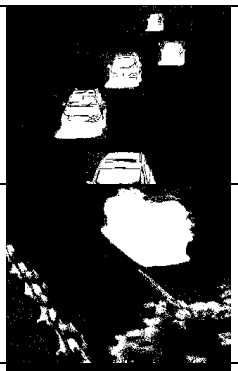
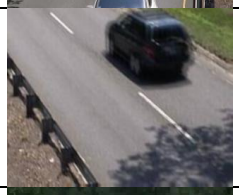
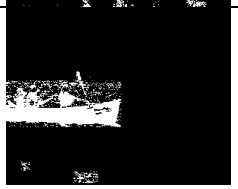
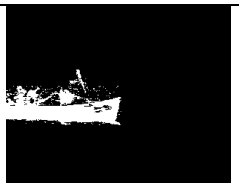
ID	Frame	*ground-truth*	Configuración 1	Configuración 2
1				
2				
3				
4				
5				
6				

Tabla 5.3-4: Resultados ground-truth vs. sistema

En la Tabla 5.3-4 se observan las diferencias entre el *ground-truth* y las dos configuraciones del sistema. Se van a valorar las mejoras cualitativas introducidas en la configuración 2:

- **Categorías 1 y 6**: son idénticas ya que no sufren ninguna modificación entre configuraciones.
- **Categoría 2**: se ha modificado el valor *shift* de 0 a 3, eliminando el ruido introducido por la vibración de la cámara.
- **Categoría 3**: se ha aumentado de 3 a 5 el número de capas eliminando píxeles multimodales que aparecían en el frente.
- **Categoría 4**: se ha aumentado el valor de Ω, por lo que los objetos que han pasado de dinámico a estático tardan más en introducirse en el fondo. En este caso empeora porque introduce frente que debería haberse introducido en fondo por el excesivo valor de Ω.
- **Categoría 5**: se ha pasado de distancia Euclídea a distancia del cono consiguiendo eliminar sombras que aparecían en el frente.

5.3.3 Eficiencia computacional

A continuación se muestran los tiempo estimados de procesado del algoritmo, considerando que se ha ejecutado en un equipo con procesador I*ntel Core i5-330* @ 3 GHz con memoria *RAM* de 8 GHz y sistema operativo de 64 bits.

Se han realizado una serie de configuraciones, para ver cómo afectan los métodos a los tiempos de procesado, que se definen en la Tabla 5.3-5:

Configuración	Nº de capas	Cono	Vecindario (shift)
C1: Predeterminado	3	No	0
C2: Sombras y reflejos	3	Sí	0
C3: Multimodal	5	No	0
C4: Vecindario 1	3	No	1
C5: Vecindario 2	3	No	2
C6: Vecindario 3	3	No	3

Tabla 5.3-5: Configuraciones de tiempos de procesado

La Tabla 5.3-6 muestra los tiempos (en segundos) de procesado obtenidos para estas configuraciones:

	C1	C2	C3	C4	C5	C6
Tiempo mínimo	0.059	0.045	0.032	0.055	0.086	0.128
Tiempo medio	0.124	0.193	0.134	0.178	0.256	0.423
Tiempo máximo	0.259	0.285	0.333	0.336	0.0409	0.583
Cuadros por segundo (F.P.S)	8.06	5.19	7.47	5.6	3.9	2.36

Tabla 5.3-6: Tiempos de procesado (en segundos)

En la Figura 5.3-4 se muestra una comparativa de los tiempos con las diferentes configuraciones establecidas:

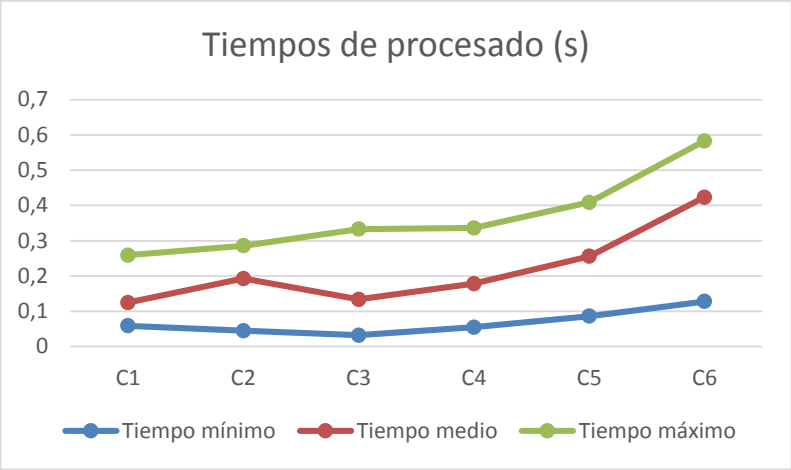


Figura 5.3-4: Tiempos de procesado

5.4 DISCUSIÓN

Como se puede comprobar en las Figuras de comparativa, el sistema se encuentra por debajo de los resultados obtenidos por las mejores aproximaciones con las que se han comparado.

Existen varios factores que repercuten directamente en los resultados obtenidos y por los que el sistema reduce su calidad:

- **Camuflaje:** Este es sin duda el factor más determinante en la obtención de estos resultados. El sistema desarrollado en este *TFG* no considera el camuflaje, considerarlo se ha decidido como trabajo futuro (ver Sección 6.2). El camuflaje afecta a todas las categorías, lo que hace que los resultados globales sean aún más bajos.
- **Parámetros no ajustados:** Para determinar los parámetros predeterminados (Configuración 1) no se ha realizado un barrido con el que obtener las mejores. En las posteriores configuraciones se han modificado algunos parámetros para ver cómo afectan en los resultados del sistema, pero sin ser los mejores. Por lo tanto, analizando y utilizando los parámetros ajustados, se obtendrían mejores resultados.
- **Resultados sin *GUI*:** Estos resultados no incluyen el uso de la interfaz gráfica, sin embargo, ésta permite la modificación de los parámetros con el sistema en ejecución de forma que eligiendo los ideales en distintos instantes temporales, los resultados mejorarían.

Por otra parte, se aprecian mejoras en el sistema siguiendo estos pasos:

- Aumento del número de capas.
- Uso de distancia del cono.
- Aumento de comparación con entorno espacial.

Los peores resultados se dan en la categoría 'intermittentObjectMotion' debido a que se ha escogido un factor Ω inadecuado, de forma que la confianza máxima, que es función de Ω,

tiene un valor inadecuado y los objetos que se vuelven estáticos durante unos instantes se introducen en el modelo de fondo, haciendo que los resultados disminuyan.

Hay que añadir que la mejora de resultados global entre configuraciones 1 y 2 no ha sido amplia, debido a que los parámetros modificados se han realizado solo en una categoría, mientras que utilizándose en todas estos resultados habrían mejorado.

En 'dynamicBackground', categoría de escenas multimodales, se consigue el objetivo de discriminar fondos multimodales (ver Figura 5.4-1) y no considerarlos frente (ver Sección 5.3.2) aunque por los factores comentados anteriormente los resultados están por debajo de las mejores aproximaciones.

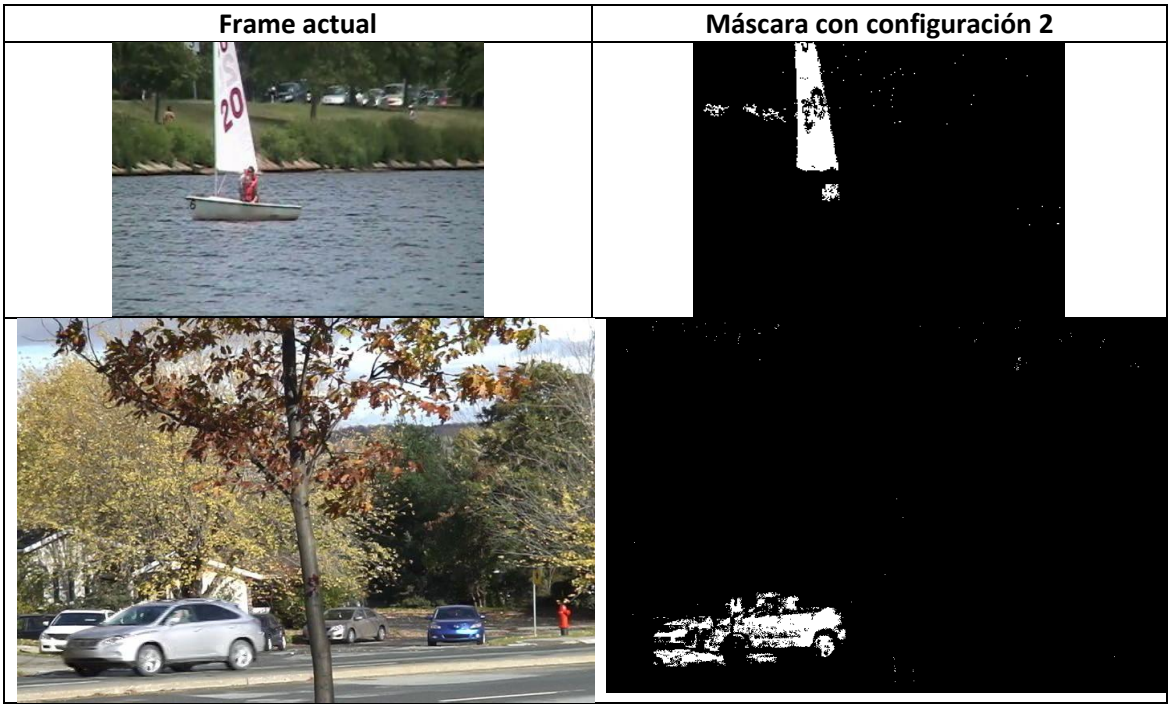


Figura 5.4-1: Resultados en escenarios multimodales

En la Figura 5.4-1 aparecen los frames 7243 y 3570 de las secuencias 'boats.avi' y 'fall.avi' respectivamente, junto con las máscaras obtenidas con la configuración 2. Como se observa, la multimodal que existe en el agua o en las hojas de los árboles no se introduce en el frente, pero el camuflaje hace que los resultados disminuyan.

En cuanto a procesado de tiempos, se observa como la distancia del cono tiene un mayor coste computacional que la distancia Euclídea y que la comparación con el vecindario tiene un coste computacional que crece exponencialmente con su aumento, ya que crece exponencialmente el número de píxeles a comparar (ver Figura 3.5-4). Se observa en los resultados como el sistema no procesa a tiempo real, pero podría mejorarse utilizando un sistema multihilo o con un equipo de hardware más potente.

CAPÍTULO 6: CONCLUSIONES Y TRABAJO FUTURO

6.1 CONCLUSIONES

El objetivo de este trabajo era el diseño de un sistema que permitiese la detección de intrusión en exteriores en tiempo real, utilizando una interfaz gráfica para poder adaptarse a las diferentes escenas.

Haciendo un balance de los objetivos del *TFG*, se cumple la adaptación del sistema a cambios de iluminación gracias a los métodos de actualización del modelo de fondo, se consiguen reducir las sombras y reflejos aparecidos, se ha reducido el ruido introducido por vibraciones y se obtienen fondos multimodales. La determinación de parámetros del sistema también ha sido un objetivo cumplido, ya que a partir de estos parámetros se puede controlar el proceso de sustracción de fondo, si bien no se han obtenido los mejores parámetros.

Para poder cumplir con estos objetivos se realizó un estudio del estado del arte actual sobre los sistemas existentes que se utilizan en este tipo de situaciones para analizar las diferentes formas de solventar estas limitaciones que proponen. En dicho capítulo se analizaron gran variedad de métodos, conociendo sus ventajas y limitaciones, algo de vital importancia para el diseño del sistema.

Se ha propuesto un sistema que utiliza un esquema no paramétrico con respecto al modelado de fondo y frente, que además almacena los modos de ambos a partir de una aproximación multicapa.

Para la inicialización del sistema se ha propuesto una inicialización total de la primera capa del modelo de fondo, e inicialización local para el resto de capas de los modelos en función de la clasificación de los píxeles, de forma que el sistema propone una inicialización selectiva.

Se ha propuesto también un sistema basado en confianzas para la actualización del sistema, junto a un sistema multiclase para determinar que modelos y áreas deben actualizarse y además se propone una actualización selectiva paramétrica para controlar la adaptación de los modelos.

En la etapa de comparación se ha propuesto un método de comparación con el entorno, para reducir el ruido, y además un proceso comparativo que da robustez al sistema frente a reflejos y sombras en la secuencia.

Se ha propuesto un sistema capaz de añadir, eliminar o modificar módulos sin cambiar la filosofía propuesta, y tras los cambios puede compararse con la base de datos utilizada [37] para comprobar si el sistema ha mejorado los resultados.

Tras el desarrollo del sistema, se diseñó una implementación para recibir cuadros de cámaras de vídeo, de forma que el sistema pueda operar en tiempo real y no trabajar sobre secuencias de vídeo exclusivamente. Esta integración en *DiVA* permite además integrar el sistema con otros algoritmos que necesiten sustracción de fondo para su aplicación.

También se diseñó una interfaz gráfica para poder controlar los parámetros del sistema en tiempo real, mientras éste está ejecutándose, teniendo así un sistema que puede ser utilizado en tareas de video-vigilancia, y en espacios exteriores como se pretendía.

Para finalizar se realizó un análisis y comparación con los mejores sistemas en sus categorías para determinar la calidad del sistema y conocer las limitaciones que posee de forma que puedan mejorarse con posterioridad.

6.2 TRABAJO FUTURO

El crear un sistema de cero y la limitación de la extensión del TFG ha supuesto una limitación a la hora del desarrollo completo del sistema y, por ello, se proponen varias sugerencias para el desarrollo en un futuro:

Tras el análisis de los resultados se ha llegado a la conclusión de que un objetivo futuro debe ser la robustez del sistema ante el camuflaje del frente, que está presente en la mayoría de secuencias analizadas.

Se propone mejorar la eficiencia del sistema para trabajar en tiempo real utilizando técnicas de procesado multihilo.

Se propone realizar una mejora en el proceso de regresión del frente al fondo, por un método más complejo que mejore la precisión en esta situación.

Otra propuesta es la de integrar nuevos módulos como un comparador que utilice regiones como unidad de análisis, que mejoraría los resultados del sistema.

La *GUI* desarrollada es una versión útil para ser utilizado por expertos en el ámbito del análisis de vídeo, que visualicen las imágenes intermedias del proceso y puedan configurar los parámetros desde la pantalla de interfaz. Sin embargo, se propone como trabajo futuro desarrollar otra *GUI* más sencilla para nivel de usuario en la que se visualice la secuencia de entrada y la de salida, y que sea necesario abrir una pestaña de opciones para configurar los parámetros.

Una limitación en la funcionalidad de la GUI es la necesidad de ser un sistema supervisado, y esto es de poca utilidad para la mayoría de aplicaciones. Se propone como trabajo futuro, desarrollar un método que en función de la escena, varíe los parámetros a sus valores más eficientes y se adapte a cualquier tipo de escenario, convirtiendo al sistema en un sistema automático.

Por último, se propone introducir en la *GUI* la opción de almacenar las máscaras binarias de salida en una carpeta para poder compararla posteriormente con el *ground-truth* de una base de datos.

6.3 FASES DE DESARROLLO DEL TFG

Se distinguen varias etapas en el desarrollo de este trabajo:

i. **Conocimientos previos:** La primera etapa consistió en un repaso de lenguaje de programación *C* y una lectura de los libros [43] y [44] sobre lenguaje de programación orientada a objetos en *C++* y la librería *OpenCv* para adquirir conocimientos sobre la programación desarrollada en este trabajo. Esta etapa tuvo una duración aproximada de un mes durante el verano de 2013.
ii. **Análisis del Estado del Arte actual:** Se ha realizado un estudio previo al desarrollo del algoritmo de los diferentes sistemas similares al diseñado. La duración aproximada del estudio del *SoA* es de 2 semanas.
iii. **Desarrollo del algoritmo:** El desarrollo del algoritmo es la parte que ha requerido más tiempo ya que ha sido desarrollado desde cero y ha sufrido varias modificaciones a partir de la idea original por las limitaciones surgidas. La duración de esta etapa es casi la misma del propio trabajo.
iv. **Implementación en DiVA:** La implementación en DiVA tuvo una duración aproximada de 3 semanas en la que fue necesario un análisis de otros sistemas implementados en esta plataforma para implementar el sistema de forma análoga.
v. **Desarrollo de la interfaz gráfica:** El desarrollo de la interfaz ha tenido una duración estimada de 1 mes y medio y ha sufrido modificaciones a la vez que el algoritmo era modificado por su total dependencia de este.
vi. **Obtención y análisis de resultados:** La obtención de resultados y análisis ha tenido una duración aproximada de 2 semanas debido al tamaño de la base de datos utilizada y las diferentes pruebas realizadas sobre el algoritmo.
vii. **Redacción de la memoria:** La redacción de la memoria se ha realizado simultáneamente conforme el desarrollo del trabajo, documentando los procesos que éste ha experimentado.

La duración completa del trabajo ha sido aproximadamente de 6 meses con una media de unas 25 horas de trabajo semanales aunque estos valores son algo difíciles de estimar ya que el tiempo trabajado cada mes no ha sido uniforme, pero dan una idea aproximada del tiempo requerido para la realización de este *TFG*.

Existen algunas propuestas que no han podido desarrollarse debido a limitaciones como:

- **Análisis a nivel de regiones**: tras implementar este método en el sistema, se observó como el coste computacional era demasiado elevado para implementar en una aplicación en tiempo real y se desestimó.
- **Comparación dentro del blob entre píxeles**: El problema introducido era que apareciesen huecos en el frente dependiendo del clasificador. Mediante el nivel de regiones, a través de la componente conexa se elimina este error asumiendo que si el clasificador se equivoca, el error afectará a toda la región y no solo a un píxel. Este riesgo puede asumirse ya que el clasificador se equivoque en varios píxeles por frame a lo largo de una secuencia da peores resultados a que una región se equivoque en un determinado frame de la secuencia.

- **Funciones**: A lo largo del desarrollo del algoritmo se han implementado diferentes funciones para procesos de actualización, que han ido desestimándose o adaptándose hasta conseguir las funciones utilizadas por el sistema, que aportan mejores resultados que las anteriores.

Cabe destacar la participación de este trabajo en el *Workshop: "Strategies for object Segmentation, Detection and Tracking in Complex Enviroments for Event Detection in Video Surviellance and Monitoring", TEC2011-25995 EventVideo (2012-2014)* organizado durante el mes de Mayo de 2014 por el grupo de investigación *VPU Lab* con el título 'Segregación frente-fondo en entornos multimodales' [45] .

CAPÍTULO 7: REFERENCIAS

[1] H. Dias, J. Rocha, P. Silva, C. Leao, “Distributed Surveillance System", Conference on Artificial Intelligence, 2005. EPIA 2005, pp 257-261.

[2] M. Valera and S.A. Velastin, “Intelligent Distributed Surveillance Systems”, IEEE Proc.-Vis. Image Signal Process., Vol. 152, No. 2, April 2005.

[3] M. M. Chang, M. I. Sezan, and A. M. Tekalp, “An algorithm for simultaneous motion estimation and scene segmentation,” in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, vol. V, Adelaide, Australia,Apr. 1994, pp. 221–224.

[4] O. Sukmarg, K.R. Rao, Fast object detection and segmentation in MPEG compressed domain, in: Proc. IEEE TENCON, Kuala Lumpur, Malaysia, 2000, vol. 3, pp. 364–368.

[5] Piccardi, M. Background subtraction techniques: a review. Systems, Man and Cybernetics, 2004 IEEE International Conference on, 2004, 4, 3099 – 3104.

[6] Kentaro T, John K, Barry B, Brian M. Wallflower: Principles and Practice of Background Maintenance, ICCV, Seventh Int. Conf. On Com Vision (ICCV'99) - 1999; 1: 255-261.

[7] Elgammal A, Duraiswami R, Harwood D, Davis LS. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proc. IEEE Jul. 2002; 90(7): 1151-1163.

[8] Harville M., Gordon G., Woodfill J. Foreground Segmentation Using Adaptive Mixture

[9] Stauffer, C. & Grimson, W. Adaptive Background Mixture Models for Real-Time Tracking Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, IEEE Computer Society, 1999, 2, 2246.

[10] Models in Color and Depth. event, IEEE Workshop on Detection and Recog of Events in Video (EVENT'01),2001; 3-12.

[11] S.C. Cheung, C. Kamath. “Robust background subtraction with foreground validation for urban traffic video” EURASIP J. Appl. Signal Process, vol 2005(1) pp 2330-2340, 2005.

[12] Herrero, S., & Bescós, J. (2009, January). Background subtraction techniques: systematic evaluation and comparative analysis. In Advanced Concepts for Intelligent Vision Systems (pp. 33-42). Springer Berlin Heidelberg.

[13] Cristani M., Bicego M., Murino V. Multi-level background initialization using Hidden Markov Models. In First ACM SIGMM Int. workshop on Video surveillance 2003; 11-20.

[14] Ralph Ewerth, Bernd Freisleben: Frame difference normalization: an approach to reduce error rates of cut detection algorithms for MPEG videos. ICIP (2) 2003: 1009-1012.

[15] Koller D., Weber J., Huang T., Malik J., Ogasawara G., Rao B. Russell S. Toward robust automatic traffic scene analysis in realtime. in Proc. Int. Conf. Patt Recog, 1994; 126-131.

[16] Gordon G., Darrell T., Harville M., Woodfill J. Background Estimationand Removal Based on Range and Color. CVPR, 1999IEEE Comp Society Conf. on Comp Vision and Patt Recog (CVPR'99) 1999; 2: 2459.

[17] C. Stauffer, W. Grimson. “Learning patterns of activity using real time tracking”. In IEEE Transactions. on Pattern Analysis and Machine Intelligence, vol 22(8) pp 747-757, 2000.

[18] O. Javed, K. Shafique, M. Shah. “A hierarchical approach to robust background subtraction using color and gradient information”. Motion and Video Computing, 2002. Procceedings. Workshop on, vol 1(1) pp 22-27, 2002.

[19] Stenger B., Ramesh V., Paragios N., Coetzee F., Buhmann J. M.Topology Free Hidden Markov Models: Application to Background Modeling. ICCV, Eighth Int. Conf. on Computer Vision (ICCV'01) 2001; 1: 294-301.

[20] Butler D., Sridharan S., Bove VMJr. Real-time Adaptive Background Segmentation. Acoustics, Speech, and Signal Processing.2003. Proceedings. (ICASSP '03). 2003 IEEE Int. Conf. on April 2003: 3: 349-52.
[21] Colmenarejo, A., Escudero-Viñolo, M., & Bescós, J. (2011). Class-driven Bayesian background modelling for video object segmentation. Electronics letters, 47(18), 1023-1024.
[22] Evangelio, R.; Patzold, M. & Sikora, T. Splitting Gaussians in Mixture Models Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on, 2012, 300 -305.
[23] R. Boesch, Z. Wang. "Segmentation Optimization for aerial images with spatial constraints" The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol 37(B4). 2008.
[24] F. Polikri, O. Tuzel. "Bayesian Background Modeling for Foreground Detection" In: Proceedings of the ACM Visual Surveillance and Sensor Network vol 1(1) pp 55-58, 2005.
[25] J. Chen, T.N Pappas "Adaptive image segmentation based on color and texture" Conference on Image Processing. 2002. Proceedings, vol 3(1) 777-780, 2002
[26] Evangelio, R. H. Background Subtraction for the Detection of Moving and Static Objects in Video Surveillance.
[27] Horprasert, T., Harwood, D., and Davis, L. S. (1999). A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection. In Proceedings of the IEEE Conference ICCV, volume 99, pages 1–19.
[28] Long W, Yang YH. Stationary background generation: An alternative to the difference of two images, Patt Recog 1990; 23(12): 1351-1359.
[29] Gloyer B., Aghajan HK, Siu KY, Kailath T. Video-based freeway monitoring system using recursive vehicle tracking. In Proc. Of IS&T-SPIE Symposium on Electronic Imaging: Image and Video Processing 1995.
[30] Morde, A.; Ma, X. & Guler, S. Learning a background model for change detection Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, 2012, 15 -20.
[31] Cavallaro, A. & Ebrahimi, T. Video object extraction based on adaptive background and statistical change detection 2000, 465-475.
[32] Hofmann, M.; Tiefenbacher, P. & Rigoll, G. Background segmentation with feedback: The Pixel-Based Adaptive Segmenter Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, 2012, 38 -43.
[33] Schick, A.; Bauml, M. & Stiefelhagen, R. Improving foreground segmentations with probabilistic superpixel Markov random fields Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, 2012, 27 -31.
[34] Maddalena, L. & Petrosino, A. The SOBS algorithm: What are the limits? Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, 2012, 21 -26.
[35] Van Droogenbroeck, M. & Paquot, O. Background subtraction: Experiments and improvements for ViBe Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, 2012, 32 -37.
[36] M Escudero-Viñolo, J Bescós. A robust framework for region based video object segmentation. Image Processing (ICIP 2010), 17th IEEE International Conference on, 2010. 3461-3464
[37] Goyette, N., Jodoin, P. M., Porikli, F., Konrad, J., & Ishwar, P. (2012, June). Changedetection. net: A new change detection benchmark dataset. InComputer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on (pp. 1-8). IEEE.
[38] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. Image Processing, IEEE Transactions on, 13(4), 600-612.

[39] San Miguel, J. C., Bescós, J., Martínez, J. M., & García, Á. (2008, May). Diva: a distributed video analysis framework applied to video-surveillance systems. InImage Analysis for Multimedia Interactive Services, 2008. WIAMIS'08. Ninth International Workshop on (pp. 207-210). IEEE.
[40] Qt projet. http://qt-projet.org/
[41] http://arantxa.ii.uam.es/~jms/pfcsteleco/lecturas/20140618CarlosSanchezBueno.pdf
[42] C. Benedek, T. Sziranyi. “Bayesian Foreground and Shadow Detection in Uncertain Frame Rate Surveillance Videos” Transactions on Image Processing, IEEE, vol 17(4) pp 608-621, 2008.
[43] Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer vision with the OpenCV library. " O'Reilly Media, Inc.".
[44] Oualline, S. (2003). Practical C++ programming. " O'Reilly Media, Inc.".
[45] VPU-Lab: http://www-vpu.eps.uam.es/eventvideo/TechnicalWS.html
[46] Change Detection: http://changedetection.net/
[47] Anónimo
[48] Zhang, S., Yao, H., & Liu, S. (2008, October). Dynamic background modeling and subtraction using spatio-temporal local binary patterns. In Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on (pp. 1556-1559). IEEE.
[49] Evangelio, R. H., & Sikora, T. (2011, August). Complementary background models for the detection of static and moving objects in crowded environments. In Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on (pp. 71-76). IEEE.
[50] St-Charles, P. L., Bilodeau, G. A., & Bergevin, R. Flexible Background Subtraction With Self-Balanced Local Sensitivity.