

# Trabajo Fin de Grado

## Detección y seguimiento de elementos móviles para videovigilancia

Autor

David Bernal Calmarza

Director

Alejandro Rituerto Sin

Ponente

José Jesús Guerrero Campo

Escuela de Ingeniería y Arquitectura

2015



*A Alejandro...*



## **Agradecimientos**

En primer lugar debo agradecer el apoyo recibido en la realización de este trabajo a mi tutor, Alejandro Rituerto, así como a José Jesús Guerrero por haberme permitido abordar este tema.

También al resto de profesores que, a lo largo de mi vida, han trabajado contribuyendo a mi formación tanto académica como personal y me han transmitido los conocimientos necesarios para poder hoy finalizar con éxito este grado.

Grado, que en gran medida he de agradecer a mis padres, Pilar y Eulalio, ya que han sido ellos quienes me han permitido cursarlo, y también les he de dar las gracias por su apoyo durante todos estos años, apoyo que también agradezco a mi hermana Raquel, al resto de mi familia, mi novia Elena y a todos mis amigos.

Al finalizar, hoy, una etapa de mi vida, creo que también es justo recordar cómo empezó y con quienes lo hizo, y un papel importante en ello lo tiene un buen amigo, Alejandro, al que siempre recordaré y al que hace años tenía prometido dedicar este trabajo que implica el fin de estos estudios que juntos hace unos años comenzamos.

También he de recordar a otro Alejandro... Otro proyecto de ingeniero con el que compartía veranos de estudio y descanso, y por qué no, a un tocayo.

*Ayer, hoy y siempre...*

David Bernal Calmarza



# Detección y seguimiento de elementos móviles para videovigilancia

## **Resumen**

En los últimos años se ha producido un gran avance en el mundo de la videovigilancia gracias a la aplicación de técnicas de visión artificial. Estas técnicas permiten que un ordenador analice las imágenes recibidas por las cámaras de seguridad detectando, por ejemplo, objetos perdidos o abandonados, así como personas que entran en zonas restringidas o peligrosas.

La mayoría de estas aplicaciones de vigilancia basan su funcionamiento en la detección de elementos móviles y su seguimiento. Este trabajo estudia dichas tareas, analizando los distintos algoritmos disponibles en distintos entornos que presentan dificultades.

Como resultado de dicho análisis se ha implementado un sistema de videovigilancia completo, combinado ambas etapas que puede servir de base para sistemas de vigilancia más complejos.





## DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D<sup>a</sup>. David Bernal Calmarza,

con nº de DNI 71302646-R en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo

de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la

Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)

Grado \_\_\_\_\_, (Título del Trabajo)

Detección y seguimiento de elementos móviles para videovigilancia

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, Septiembre de 2015

Fdo: David Bernal Calmarza



# Índice general

Agradecimientos .....	iv
Resumen .....	vi
Declaración de autoría y originalidad .....	viii
Índice de figuras .....	xi
Índice de tablas .....	xii
1. Introducción	
1.1 Motivación .....	1
1.2 Estado del arte .....	1
1.3 Objetivos del trabajo .....	2
1.4 Material y métodos .....	2
1.5 Organización del trabajo .....	4
2. Detección de elementos móviles	
2.1 Introducción. Objetivos .....	5
2.2 Estado del arte .....	5
2.3 Procedimiento de evaluación .....	8
2.4 Métodos de evaluación .....	9
2.5 Resultados experimentales .....	12
2.6 Conclusiones .....	17
3. Seguimiento de elementos móviles	
3.1 Introducción. Objetivos .....	19
3.2 Estado del arte .....	20
3.3 Algoritmo de Lucas y Kanade .....	21
3.4 Procedimiento de evaluación .....	22
3.5 Método de evaluación .....	23
3.6 Resultados experimentales .....	23
3.7 Conclusiones .....	25
4. Implementación del sistema completo	
4.1 Introducción. Objetivos .....	26
4.2 Implementación .....	26
4.3 Resultados y conclusiones .....	27
5. Resultados finales y conclusiones .....	29
ANEXO 1. Código C++ .....	30
ANEXO 2. Resultados completos .....	35

# Índice de figuras

<i>Figura 1: Ejemplo de secuencia perteneciente al data set</i>	2
<i>Figura 2: Ejemplo de sustracción de fondo y detección de elementos móviles</i>	5
<i>Figura 3: Esquema del funcionamiento de la sustracción de fondo</i>	5
<i>Figura 4: Esquema de funcionamiento del algoritmo de evaluación.</i>	9
<i>Figura 5: Esquema de la clasificación de resultados</i>	10
<i>Figura 6: Ejemplo de imagen a evaluar</i>	10
<i>Figura 7: Ejemplo de imagen evaluada</i>	10
<i>Figura 8: Ejemplos de distintos puntos en el espacio ROC.</i>	12
<i>Figura 9: Puntos en el espacio ROC para los métodos con mejores resultados,</i>	14
<i>Figura 10: Ejemplo de problemas generados en fondos dinámicos</i>	18
<i>Figura 11: Ejemplo de extracción de puntos característicos</i>	19
<i>Figura 12: Ejemplo de flujo óptico para una esfera que gira en sentido horario</i>	21
<i>Figura 13: Ejemplo de problema de apertura,</i>	21
<i>Figura 14: Esquema del funcionamiento piramidal del método de Lucas y Kanade</i>	22
<i>Figura 15: Esquema del procedimiento para el seguimiento.</i>	23
<i>Figura 16: Ejemplo de error en el seguimiento al cruzarse dos personas</i>	24
<i>Figura 17: Ejemplo de error de seguimiento al pasar por detrás de un árbol</i>	24
<i>Figura 18: Ejemplo de error en el seguimiento de un giro</i>	25
<i>Figura 19: Esquema de la implementación de un sistema completo de vigilancia</i>	27
<i>Figura 20: Ejemplo de implantación del algoritmo completo en la secuencia del sofá</i>	28
<i>Figura 21: Ejemplo de implantación del sistema completo en secuencia de carretera (highway).</i>	28
<i>Figura 22: Puntos en el espacio ROC para todos los métodos analizados</i>	49

# Índice de tablas

<i>Tabla 1: Métodos a analizar de la librería BGS.</i>	8
<i>Tabla 2: Clasificación de los métodos en función del tiempo medio de ejecución.</i>	13
<i>Tabla 3: Resultados de los métodos más significativos</i>	15
<i>Tabla 4: Resultados de los distintos grupos de modelos</i>	16
<i>Tabla 5: F-measure de los mejores métodos para los distintos videos</i>	17
<i>Tabla 6: Clasificación en función del tiempo medio de ejecución para los videos de 76800 pixeles.</i>	35
<i>Tabla 7: F-Measure en los distintos videos (promedio de todos los métodos)</i>	35
<i>Tabla 8: Resultados completos para cada una de las secuencias.</i>	37
<i>Tabla 9: Análisis del método de Lucas y Kanade</i>	50



# **1. Introducción**

## **1.1 Motivación**

El uso de técnicas de visión artificial en aplicaciones de videovigilancia es un tema de gran importancia en el sector desde hace muchos años, ya que permite la automatización de las tareas de vigilancia. Los sistemas tradicionales requieren un trabajo intenso por parte de una persona, ya que se limitan a guardar imágenes provenientes de las distintas cámaras. La utilización de sistemas de videovigilancia basados en visión artificial facilita la labor del personal humano al permitir la detección automática de eventos que puedan ser de interés.

Estos sistemas automáticos buscan detectar, reconocer y seguir objetos a partir de secuencias de imágenes obtenidas por una cámara. Se basan en la aplicación secuencial de dos tareas: la detección de elementos de interés y el seguimiento de los mismos. Estas tareas son la base sobre la que se desarrollan tareas más complicadas como la detección de objetos abandonados, accesos a zonas restringidas o actos de vandalismo.

Para la detección de elementos móviles a lo largo de las imágenes se realiza una segmentación de las mismas, conocida como *Background Subtraction* o sustracción de fondo, distinguiendo entre el primer plano, formado por los elementos móviles, y el fondo (*foreground* y *background* respectivamente). Se considera fondo aquello que permanece constante a lo largo del tiempo mientras las personas u objetos que cambian de lugar a lo largo del video se considerarán primer plano o frente, detectando así el movimiento de estos. Es importante matizar que un fondo dinámico, con árboles o sombras que varíen sus posiciones ha de ser considerado fondo.

La segunda tarea consiste en la realización de un seguimiento (o *tracking*) de los elementos en primer plano [1], detectando sus puntos característicos y calculando su posición a lo largo de las consecutivas imágenes.

A partir de estos sistemas se podrán crear aplicaciones capaces de detectar y enviar alertas frente a distintos fenómenos como robos o entrada de intrusos en zonas restringidas, creando aplicaciones de videovigilancia más complejas y con mayor nivel de automatización.

Para que cualquiera de estas aplicaciones funcione correctamente es fundamental que las tareas de detección y seguimiento lo hagan de manera robusta. A lo largo de este trabajo se estudiarán ambas tareas y se desarrollará una base sobre la que se podrán implementar aplicaciones más complejas.

## **1.2 Estado del arte**

Existen gran cantidad de aplicaciones de vigilancia basadas en visión artificial que, por ejemplo, permiten detectar y localizar objetos abandonados o detectar objetos en un paso a nivel [2]. Otras aplicaciones se dedican a realizar medidas en entornos de tráfico, como la monitorización de calles o el estudio de la ocupación de un parking y su seguridad a lo largo de un intervalo de tiempo prolongado (a lo largo del día, de la semana, de los meses, etc.). La Dirección General de Tráfico ha implando un sistema similar para proteger a los peatones y mejorar la fluidez del tráfico urbano de grandes ciudades [3]. También se ha implantado un sistema de videovigilancia para controlar el transporte urbano en Londres [4].

Los trabajos publicados en [5, 6, 7] abordan análisis de los distintos métodos de sustracción de fondo existentes en la literatura.

En cuanto a las técnicas de seguimiento, el trabajo presentado por Lucas y Kanade en 1981 [56] es uno de los más populares., también ha sido utilizado como base para la creación de

métodos similares [8, 9], por lo que es importante conocer en profundidad sus características y sus puntos fuertes y débiles.

### **1.3 Objetivos del trabajo**

Este trabajo pretende realizar un análisis y comparación de las distintas técnicas de sustracción de fondo en distintos entornos, buscando las ventajas e inconvenientes de cada una de ellas.

Del mismo modo se va a estudiar el seguimiento de elementos basado en el trabajo propuesto por Lucas y Kanade, se analizará su funcionamiento en distintas imágenes reales tomadas en entornos complicados, así como movimientos que hacen que se pierda el seguimiento del individuo o el objeto.

Por último se implementará un sistema completo de detección y seguimiento, es decir, un sistema que en primer lugar detecte el movimiento y en base a este se realice un seguimiento de lo detectado. Este sistema será la base de futuras aplicaciones de videovigilancia más complejas.

### **1.4 Material y métodos**

Todos los desarrollos realizados a lo largo de este trabajo se basan en el uso de software gratuito. Los modelos de sustracción de fondo (*Background Subtraction o BS*) estudiados forman parte de la librería *bgslibrary* [10], una librería que recopila gran número de métodos para realizar la segmentación detectando los elementos móviles, incluyendo muchos de los más novedosos y punteros. Muchos de estos algoritmos han sido utilizados con éxito para distintas aplicaciones [11, 12].

Esta librería se ha de utilizar junto con *OpenCv*, otra librería open source, que contiene numerosos métodos de visión artificial. Desde el año 1999 en que fue publicada ha sido utilizada en numerosas aplicaciones, muchas de ellas relacionadas con la seguridad y detección de movimiento [13].

Las distintas implementaciones en C++ se han desarrollado usando el entorno de desarrollo *Eclipse*, un software libre desarrollado por la Fundación Eclipse, en su versión para implementación de código en C/C++ [14].

Para realizar los experimentos se han utilizado las imágenes del *data set change detection dataset 2014*, alojada en el servidor *ChangeDetection.net* [15]. Dichos *data set* incluyen secuencias de diferentes tipos y la solución de la segmentación entre fondo y elementos móviles de las mismas, en la Figura 1 se puede ver un ejemplo de ello:



**Figura 1:** Ejemplo de secuencia perteneciente al *data set*, a la izquierda la imagen de entrada y a la derecha la solución.

Los videos se clasifican en distintas categorías: simples, de fondo dinámico, cámara *jitter* (incluye gran cantidad de ruido), objetos abandonados, tonos intermitentes, cámara térmica, mal clima, bajo índice de fps, nocturnos, cámara PTZ (se mueve horizontalmente, verticalmente o aumentando/disminuyendo zoom) y turbulencias.

Para la realización de los experimentos se han seleccionado una serie de videos que representan un amplio rango de situaciones con distintas dificultades para la segmentación de las imágenes. Los videos seleccionados son los siguientes:

- “Highway”: Muestra una carretera con algunos árboles alrededor y distintos coches avanzando por ella. Este es una de los ejemplos considerados como sencillos.
- “Fountain02”: El fondo es dinámico ya que está filmado frente a una fuente donde el movimiento de agua es constante. En la secuencia se pueden ver varios coches circular por la carretera que se encuentra al fondo de la imagen.
- “Traffic”: La cámara se encuentra en una pequeña carretera con un tráfico denso, la diferencia radica en el uso de una cámara con *jitter* (fluctuación), es decir, con un ruido no deseado, que hace que las imágenes entre frames se desvíen.
- “Sofa”: Este video muestra un sofá donde distintas personas van entrando en escena y dejando distintos objetos en él que serán recogidos por otros individuos más adelante.
- “People in shade”: Secuencia en la que se recoge un tramo de calle con un cajero, por el cual pasan varias personas parándose algunas de ellas en él.
- “Dinning Room”: Video tomado con una cámara térmica donde se observan a varias personas transitando el salón de una casa.
- “Skaitting”: Esta secuencia pertenece a la categoría de imágenes tomadas con mal tiempo meteorológico, muestra un paisaje nevado por el que avanzan distintos esquiadores.
- “Turnpike 0.5fps”: La cámara en este video filma una carretera de dos carriles, la dificultad de esta secuencia se encuentra en el bajo índice de frames o fotogramas por segundo, que en este caso será de 0.5 frames/segundo frente a los 25-60 con los que se suele grabar.
- “Street corner at night”: Secuencia tomada en un cruce de calles durante la noche donde se pueden observar distintos coches atravesándola además de farolas y semáforos que hacen variar la iluminación en las distintas zonas.
- “Zoom In Zoom Out”: Tomado por una cámara PTZ, aumentando y disminuyendo el zoom, lo que provoca que el fondo cambie constantemente, se debe apreciar el movimiento de un joven.
- “Turbulence 03”: En esta secuencia el viento es muy intenso, lo que provoca movimientos en el paisaje, al fondo un coche circula por la carretera.

La longitud de cada secuencia varía entre 1000 y algo más de 4000 frames. Para el análisis de los métodos de segmentación de fondo se ha realizado una selección de fragmentos de video dentro de las secuencias anteriores, estos fragmentos tienen una longitud de 300 frames durante los que se concentran eventos característicos para el estudio.

Para evaluar el seguimiento de puntos se han seleccionado distintos eventos como puede ser una persona pasando por la escena o el cruce de dos coches. La duración de estos segmentos

varía entre los 40 y los 150 frames. En los resultados de dicho experimento se explican los eventos seleccionados.

### **1.5 Organización del trabajo**

El trabajo se divide en dos grandes bloques, en el primer bloque se realiza el análisis detallado de los métodos de detección de fondo mientras que en el segundo se analiza el seguimiento de objetos móviles. Finalmente se describe la implementación de un sistema de videovigilancia completo uniéndolo estudiando en ambos bloques.

Los capítulos en los que se divide el trabajo realizado son los siguientes:

1. Introducción.
2. Detección de elementos móviles.
3. Seguimiento de elementos móviles.
4. Implantación del sistema completo.
5. Conclusiones finales y trabajo futuro.

## 2. Detección de elementos móviles

### 2.1 Introducción. Objetivos

La primera tarea que se estudiará es la detección de elementos móviles en el video mediante la sustracción de fondo. Esta tarea segmenta la imagen entre fondo (*background*) y elementos móviles (*foreground*). Existen distintos métodos para realizar dicha sustracción, la mayoría de ellos se basan en la creación de un patrón de fondo a partir del cual toda variación que sufra la imagen será considerada parte del *foreground*. Según la complejidad del método este patrón de fondo que el algoritmo crea tendrá en cuenta más o menos aspectos. Existen métodos simples que solo buscan las diferencias entre una imagen y la siguiente, sistemas basados en aspectos estadísticos de los píxeles o en distintas texturas o colores que presenta la imagen entre otros [16].

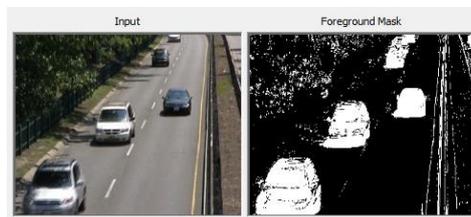


Figura 2: Ejemplo de sustracción de fondo y detección de elementos móviles

### 2.2 Estado del arte

El funcionamiento de los distintos métodos sigue el esquema presentado en la Figura 3, donde un decisor separa los píxeles de fondo de los de frente (primer plano) en función de las imágenes recibidas y del modelo de fondo que ha creado, además cada frame analizado hará que el modelo de fondo se actualice. La diferencia entre los distintos métodos reside en el decisor empleado.

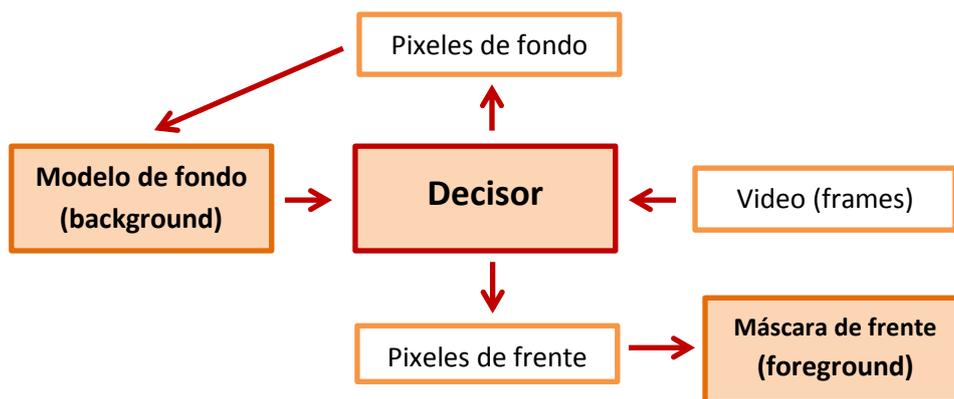


Figura 3: Esquema del funcionamiento de la sustracción de fondo

A continuación se explicarán los distintos métodos que incluye la *librería BGS*, dedicada a *Background Subtraction*, o lo que es lo mismo, a la sustracción del fondo [17, 18].

Estos métodos se pueden clasificar en diferentes categorías, formando cinco grandes grupos de modelos: básicos, estadísticos basados en uno o múltiples gaussianos, basados en características como texturas y colores, difusos y neuronales-difusos y por último, los modelos no paramétricos.

### Modelos básicos

Los modelos básicos se caracterizan por crear un patrón de fondo de manera muy básica, por lo que no suelen presentar gran exactitud. Para crear este fondo existen diversas técnicas, hay modelos que toman como fondo la primera imagen lo que puede suponer grandes problemas en caso de que haya movimiento en el primer frame, lo denominado como inicio en caliente.

Otros toman una media o mediana de  $n$  imágenes y utilizan la imagen resultante como fondo. Algo similar ocurre en el método de mediana aproximado (*Approximated Median Filter*), donde se almacena en un buffer los valores de los píxeles de un conjunto de imágenes. El fondo se define en cada píxel como la mediana de los valores almacenados en el buffer. Es un método robusto cuyos resultados se asemejan a los de los métodos complejos.

Un método algo más avanzado es el conocido como *Frame Differenc*; en este caso no se creará una imagen de fondo si no que cada frame se comparará con el frame anterior. Se producen grandes errores cuando el movimiento para bruscamente. La gran ventaja de este método es su velocidad de procesado [19, 20].

### Modelos basados en características como texturas y colores

Otro tipo de métodos que mejoran los modelos básicos se basan en un fondo que se va adaptando constantemente. Hay varios métodos que realizan esta adaptación basada en diferentes características de este, combinaciones de textura, color y formas fundamentalmente. La detección del movimiento se producirá de una manera similar a los modelos anteriores. Estos métodos son muy utilizados en escenas dónde el fondo sufre cambios.

Otro de sus puntos fuertes es la capacidad de realizar una detección precisa cuando el fondo y el objeto a detectar presentan gran similitud en cuanto a color, lo que en otros casos genera graves problemas. Su principal inconveniente radica en el tiempo necesario para analizar y comparar las texturas [21, 22].

### Modelos estadísticos

Este grupo es el más usado actualmente, realizan una estimación del fondo más probable y determinan para cada píxel de la imagen que probabilidad tiene de pertenecer al fondo. A su vez, necesitan de parámetros que permitan actualizar el modelo, para ir ajustándolo a lo largo de una secuencia.

Esto permite que se consideren las variaciones leves y las duraderas como cambios del fondo y se descarten los cambios bruscos y poco duraderos, funcionan bien en las escenas de exteriores mientras que su principal problema radica en secuencias complejas como por ejemplo aquellas con grandes cambios de luz. [23]

El más sencillo de estos modelos es el de la gaussiana simple (*Simple Gaussian*), que representa cada píxel con una distribución unimodal Gaussiana definida por su media y su desviación típica.

Dicha gaussiana define la probabilidad del valor de gris de cada píxel en cada instante de tiempo. Para determinar si un píxel pertenece al modelo de fondo se compara el valor del píxel de la imagen con el valor definido por la gaussiana, si la diferencia entre el valor del píxel y el de la media modelada para dicho píxel es inferior a  $K$  veces la desviación típica, se considerará que el píxel pertenece al modelo de fondo.

Este principio es usado en gran cantidad de algoritmos como base. Otros métodos más avanzados calculan una gaussiana para cada canal de color RGB para cada píxel. Otra evolución del método es la mezcla de gaussianas que propone modelar la intensidad de los píxeles con una mezcla de  $k$  distribuciones Gaussianas (donde  $k$  es un número pequeño, frecuentemente se utiliza de 3 a 5), esto presenta una gran ventaja en fondos donde, por el viento o condiciones similares, se producen pequeños movimientos que hacen que los píxeles correspondientes varíen entre un rango de valores.

#### Modelos difusos (*fuzzy*)

Este tipo de modelos realizan la segmentación de fondo teniendo en cuenta la similitud medida de color y textura de la imagen a analizar y el modelo de fondo utilizado.

Algo más avanzados son los modelos neuronales-difusos, donde cada píxel sigue un modelo neuronal a partir del cual se clasifica entre fondo y movimiento, para ello se utiliza una red de neuronas y un clasificador bayesiano, para modelar las probabilidades no conocidas, sabiendo o estimando las probabilidades condicionadas y las probabilidades a priori. Este modelo neuronal se mejora utilizando el modelo difuso para la etapa de aprendizaje de fondo.

#### Modelos no paramétricos

Por último, también existen otro tipo de modelos como pueden ser aquellos basados en los valores y vectores propios o a partir de histogramas e interferencias bayesianas. El modelo VuMeter se basa en una estimación discreta de la distribución de probabilidad.

La siguiente tabla recoge los distintos métodos que se analizarán de la librería y el modelo al que pertenecen, así como una referencia bibliográfica a la publicación en la que se presentaron:

**Tabla 1:** Métodos a analizar de la librería BGS [10].

Nombre en la librería	Tipo de método	Autor
StaticFrameDifferenceBGS	Básico	-
MixtureOfGaussianV1BGS	Estadístico de múltiples gaussianos	[23]
MixtureOfGaussianV2BGS	Estadístico de múltiples gaussianos	[24]
AdaptiveBackgroundLearning	Básico	-
GMG	No paramétrico	[25]
DPAdaptiveMedianBGS	Básico	[26]
DPGrimsonGMMBGS	Estadístico de múltiples gaussianos	[27]
DPZivkovicAGMMBGS	Estadístico de múltiples gaussianos	[27]
DPMeanBGS	Básico	-
DPWrenGABGS	Estadístico de un gaussiano	[28]
DPPratiMediodBGS	Básico	[29]
DPEigenbackgroundBGS	Basado en vectores y valores propios	[30]
DPTtextureBGS	Estadístico usando color y texturas	[22]
T2FGMM_UM	Difuso	[31, 32, 33]
T2FGMM_UV	Difuso	[31, 32, 33]
T2FMRF_UM	Difuso	[34]
T2FMRF_UV	Difuso	[34]
MultiLayerBGS	Múltiples características (color, textura,..)	[35]
LBSimpleGaussian	Estadístico de un gaussiano	[23]
LBFuzzyGaussian	Estadístico de múltiples gaussianos	[36]
LBMixtureOfGaussians	Estadístico de múltiples gaussianos	[37]
LBAadaptiveSOM	Neuronal / neuronal-difuso	[38]
LBFuzzyAdaptiveSOM	Neuronal / neuronal-difuso	[39]
VuMeter	No paramétrico	[40]
IndependentMultimodalBGS	Múltiples características (color, textura,..)	[41]
SJN_MultiCueBGS	Múltiples características (color, textura,..)	[42]
SuBSENSEBGS()	Múltiples características (color, textura,..)	[43]
LOBSTERBGS()	Múltiples características (color, textura,..)	[44]

### **2.3 Procedimiento de evaluación**

Para evaluar los distintos métodos se ha implementado un programa que ejecuta la librería para los frames que se seleccionen. El programa abre las imágenes de entrada y aplica el método elegido, mostrando en pantalla tanto las entradas como la máscara de salida a lo largo de los frames para que el usuario pueda observar el movimiento detectado. El programa también es capaz de medir el tiempo empleado en la ejecución de cada frame, así como la precisión de la sustracción realizada.

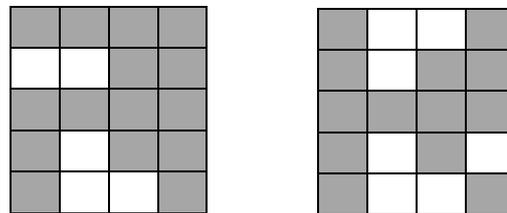
Para ejecutar dicha evaluación, se realiza una comparación con las “soluciones” (imágenes incluidas en el *data set*), de esta forma se podrá medir la precisión de los distintos métodos. Ambas imágenes son del mismo tamaño, por lo que cada pixel de una se corresponde con uno de la otra. Se comparará cada uno de los pixeles diferenciando los negros, que representan el



		<u>Valores reales</u>		Total:
		p	n	
<u>Predicción</u>	p'	Verdaderos Positivos	Falsos Positivos	P'
	n'	Falsos Negativos	Verdaderos Negativos	N'
Total:		P	N	

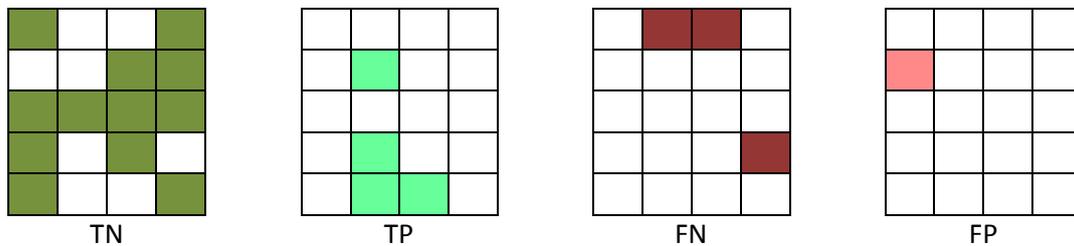
**Figura 5:** Esquema de la clasificación de resultados, dónde p y n representan los positivos y negativos reales, p', n' los detectados. P y N son los positivos y negativos totales reales y P',N' los totales predichos

Se explicará con un ejemplo: supongamos que a la izquierda se muestra la imagen obtenida mediante la librería mientras a la derecha se muestra la solución.



**Figura 6:** Ejemplo de imagen a evaluar

Como podemos ver son de tipo binario, solo distinguen dos estados (blanco y gris en este caso). Atendiendo a la clasificación expuesta anteriormente, pixel a pixel la matriz quedaría de la siguiente forma:



**Figura 7:** Ejemplo de imagen evaluada

De izquierda a la derecha se pueden apreciar: verdaderos negativos (*true negatives*, TN), verdaderos positivos (*true positives*, TP), falsos negativos (*false negatives*, FN) y falsos positivos (*false positives*, FP).

Para comparar los resultados en los distintos métodos se utilizan los siguientes parámetros de desempeño:

$$\text{Tasa de FP} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

$$\text{Tasa de FN} = \frac{\text{FN}}{\text{FN} + \text{TP}}$$

$$\text{Sensibilidad} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Especificidad} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

La tasa de FP representa los negativos que el método identifica como positivos frente al total de negativos. De manera análoga, la tasa de FN muestra los positivos en los que el método comete error frente a los positivos totales a analizar. Un bajo valor de dichas tasas implica una mayor precisión.

La sensibilidad, también es conocida como recall, muestra los positivos que el algoritmo identifica correctamente, frente a los positivos totales; de la misma forma, la especificidad muestra los negativos identificados correctamente frente al total de ellos.

También se calcularán dos valores predictivos:

La probabilidad de resultado negativo (PVN), que, de la misma forma, será la probabilidad de que obteniendo un resultado negativo en el test, el valor real sea negativo:

$$\text{PVN} = \frac{\text{TN}}{\text{TN} + \text{FN}}$$

Y la probabilidad de resultado positivo (PVP), es decir, la probabilidad de que obteniendo un resultado positivo con el método utilizado el positivo ocurra en la realidad:

$$\text{PVP} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Este valor también es conocido como precisión. Este valor, junto con el recall (o exhaustividad) son claves en la medición del rendimiento, en el mejor de los casos ambos deben rondar valores cercanos a 1.

El sistema de medición en base a estos dos parámetros es conocido como de precisión y exhaustividad y tiene un gran uso en sistemas de búsqueda y recuperación de información y reconocimiento de patrones [45].

En dicho sistema también es muy utilizado el parámetro F-measure que se define como:

$$\text{F - measure} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{(\text{precision} + \text{recall})}$$

Este valor, también conocido como medida-F, está relacionado con la precisión que tiene un test o un modelo [46]. En este caso se da la misma importancia a la variable de precisión que a la de exhaustividad, aunque podría añadirse un parámetro auxiliar, beta, que hiciera a alguno de ellos gozar de mayor importancia.

Por último se trazarán los puntos en el espacio ROC (acrónimo de *Receiver Operating Characteristic*, o Característica Operativa del Receptor), esta curva representa gráficamente la sensibilidad frente a  $(1 - \text{especificidad})$ .

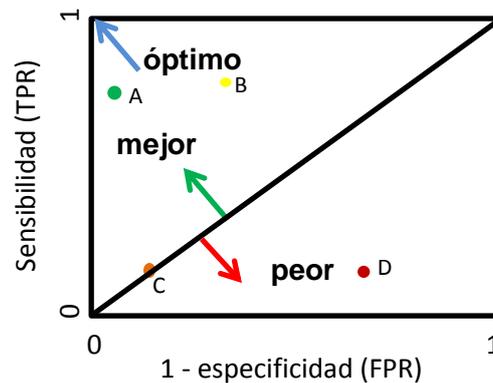


Figura 8: Ejemplos de distintos puntos en el espacio ROC.

En la figura 8 se pueden ver varios ejemplos de puntos en el espacio ROC, un punto situado en la zona superior izquierda representa un buen resultado, cuanto más abajo y más a la derecha se sitúe el resultado será peor, en este caso el mejor resultado será el representado en el punto A y el peor el D. Un punto que se encuentre por debajo de la diagonal trazada indicará un modelo muy poco preciso. La unión de los puntos obtenidos experimentalmente con el punto (0,0) y el (1,1) crea las denominadas curvas ROC, la óptima será aquella que pase por las tres esquinas ya que será más preciso aquel modelo cuyo área encerrada entre dicha curva y la diagonal trazada sea mayor.

## **2.5 Resultados experimentales**

Los resultados completos (tiempos, positivos y negativos, parámetros y representación en el espacio ROC) para cada una de las secuencias se pueden consultar en la Tabla 8.

La Tabla 2 muestra el tiempo medio (en segundos) que tarda cada método en analizar un frame así como los frames que es capaz de analizar cada segundo (promedio de las once secuencias analizadas):

**Tabla 2:** Clasificación de los métodos en función del tiempo medio de ejecución.

	Tiempo/frame (s)	Nombre del método	fps
<b>1</b>	<b>0,0181</b>	<b>StaticFrameDifferenceBGS</b>	<b>55,31</b>
<b>2</b>	0,0321	AdaptiveBackgroundLearning	<b>31,16</b>
<b>3</b>	0,0339	MixtureOfGaussianV1BGS	<b>29,51</b>
<b>4</b>	0,0401	VuMeter	<b>24,92</b>
<b>5</b>	0,0455	DPAdaptiveMedianBGS	<b>21,97</b>
<b>6</b>	0,0502	MixtureOfGaussianV2BGS	<b>19,91</b>
<b>7</b>	0,0641	GMG	<b>15,60</b>
<b>8</b>	0,0716	LBSimpleGaussian	<b>13,97</b>
<b>9</b>	0,0739	LBFuzzyGaussian	<b>13,54</b>
<b>10</b>	0,0798	DPZivkovicAGMMBGS	<b>12,53</b>
<b>11</b>	0,0868	DPWrenGABGS	<b>11,52</b>
<b>12</b>	0,0939	IndependentMultimodalBGS	<b>10,65</b>
<b>13</b>	0,1104	DPEigenbackgroundBGS	<b>9,05</b>
<b>14</b>	0,1174	DPMeanBGS	<b>8,52</b>
<b>15</b>	0,1231	LBMixtureOfGaussians	<b>8,12</b>
<b>16</b>	0,1383	SJN_MultiCueBGS	<b>7,23</b>
<b>17</b>	0,1428	T2FGMM_UM	<b>7,00</b>
<b>18</b>	0,1813	T2FGMM_UV	<b>5,52</b>
<b>19</b>	0,2062	LBAadaptiveSOM	<b>4,85</b>
<b>20</b>	0,2099	DPGrimsonGMMBGS	<b>4,77</b>
<b>21</b>	0,2204	LBFuzzyAdaptiveSOM	<b>4,54</b>
<b>22</b>	0,2711	T2FMRF_UM	<b>3,69</b>
<b>23</b>	0,3118	DPPratiMediodBGS	<b>3,21</b>
<b>24</b>	0,3173	T2FMRF_UV	<b>3,15</b>
<b>25</b>	0,3557	MultiLayerBGS	<b>2,81</b>
<b>26</b>	1,1991	LOBSTERBGS()	<b>0,83</b>
<b>27</b>	2,0313	SuBSENSEBGS()	<b>0,49</b>
<b>28</b>	2,1729	DPTtextureBGS	<b>0,46</b>

Si se tiene en cuenta que la velocidad de grabación de una cámara de videovigilancia suele estar entre 24 y 50 fps y que, incluso, existen cámaras más rápidas capaces de grabar a una velocidad bastante superior. Si se desea implementar estos sistemas en videovigilancia “en vivo”, será necesario utilizar alguno de los métodos que ocupan los primeros puestos de la tabla y/o utilizar una máquina muy potente.

A la vista de los distintos parámetros (Tabla 3 y Tabla 7) y de la representación en el espacio ROC (Tabla 8 en el Anexo 2), teniendo en cuenta tanto la velocidad como la precisión de la detección, se puede observar como los mejores resultados se dan para los siguientes métodos:

LBSimpleGaussian

LBFuzzyGaussian

LBMixtureOfGaussians

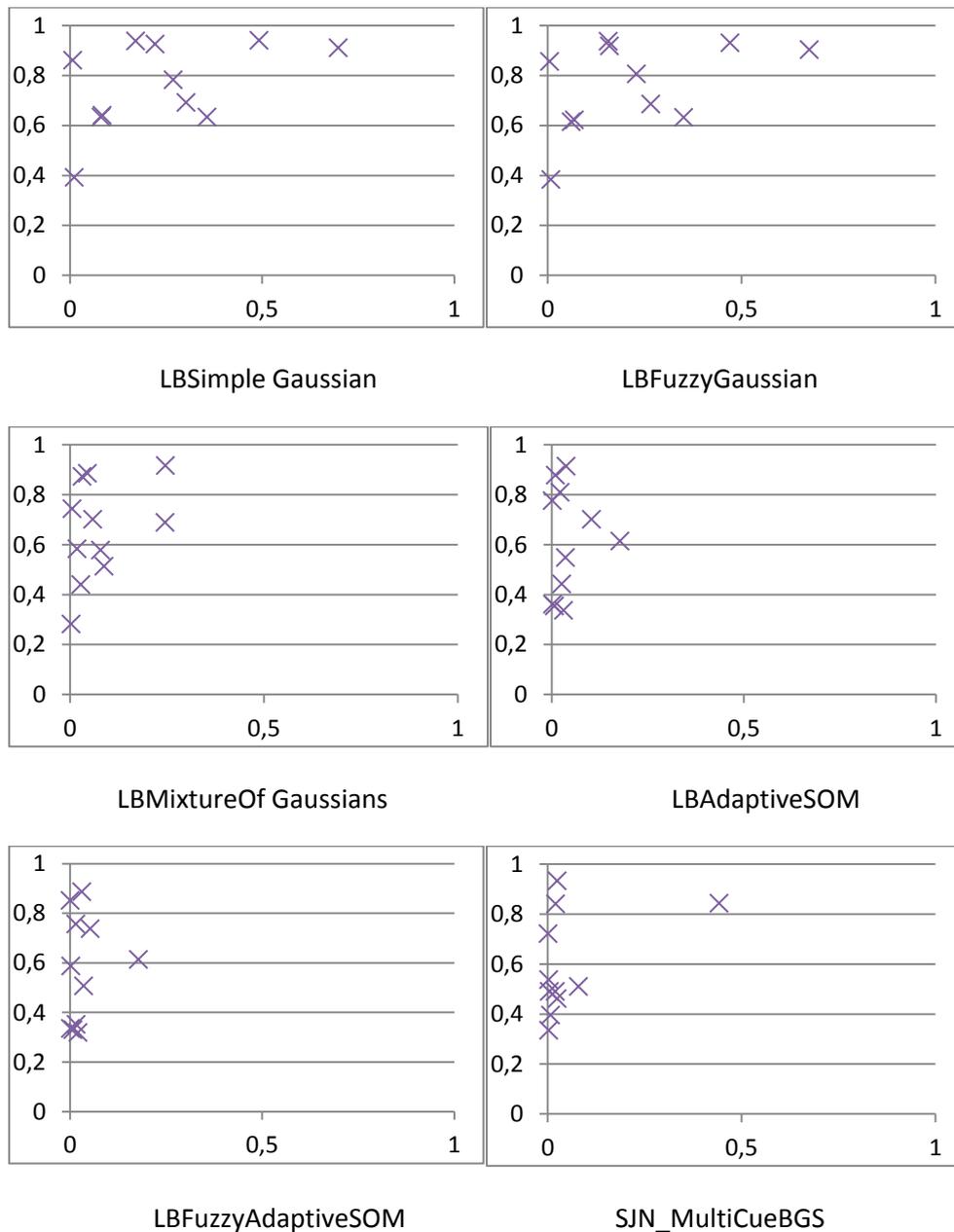
LBAadaptiveSOM

LBFuzzyAdaptiveSOM

SJN\_MultiCueBGS

Hay gran diferencia en los resultados obtenidos para estos métodos y los restantes. El parámetro F-measure, uno de los más representativos y cuyo alto valor asegura una gran eficacia, en los métodos seleccionados toma valores comprendidos entre 0.43 y 0.46 frente al resto dónde nunca supera 0.17 como se puede apreciar en la Tabla 8.

De la misma forma, se puede apreciar claramente observando los puntos en el espacio ROC, en este caso, unos valores situados en la zona superior izquierda en las gráficas aseguran buena precisión para el método representado. En la siguiente figura se puede observar los puntos en el espacio ROC de los métodos seleccionados, el resto se pueden encontrar en la Figura 22 perteneciente al Anexo 2:



**Figura 9:** Puntos en el espacio ROC para los métodos con mejores resultados, el espacio ROC representa gráficamente la sensibilidad frente a  $(1 - \text{especificidad})$ .

**Tabla 3:** Resultados de los métodos más significativos

	LBSimpleGaussian	LBFuzzyGaussian	LBmixtureOfGaussians	LBAdaptiveSOM	LBFuzzyAdaptiveSOM	SJN_MultiCueBGS
<b>Frames</b>	300	300	300	300	300	300
<b>Tiempo total</b>	<b>21,33</b>	<b>22,01</b>	<b>36,69</b>	61,45	65,67	<b>41,21</b>
<b>Tiempo medio</b>	0,07	0,07	0,12	0,21	0,22	0,14
<b>Tiempo máximo</b>	0,22	0,27	0,35	0,52	0,54	0,45
<b>Desviación media</b>	0,02	0,01	0,01	0,01	0,01	0,03
<b>Tasa de FP</b>	0,24	0,22	<b>0,08</b>	<b>0,04</b>	<b>0,03</b>	<b>0,06</b>
<b>Tasa de FN</b>	<b>0,24</b>	<b>0,25</b>	0,35	0,39	0,43	0,40
<b>Sensibilidad / Recall</b>	<b>0,76</b>	<b>0,75</b>	0,65	0,61	0,57	0,60
<b>Especificidad</b>	0,76	0,78	0,92	<b>0,96</b>	<b>0,97</b>	0,94
<b>PVP / Precision</b>	0,30	0,33	0,48	0,55	0,59	<b>0,67</b>
<b>PVN</b>	0,98	0,98	0,98	0,97	0,97	0,97
<b>F-Measure</b>	<b>0,43</b>	<b>0,46</b>	<b>0,56</b>	<b>0,58</b>	<b>0,58</b>	<b>0,63</b>

Los dos primeros modelos seleccionados, LBSimpleGaussian y LBFuzzyGaussian, tienen por ventaja el tiempo de ejecución, el modelo LBmixtureOfGaussians mejora la especificidad y con ello la F-measure pero aumenta también el tiempo de análisis.

LBAdaptiveSOM así como LBFuzzyAdaptiveSOM obtienen tasas de acierto superiores, en cambio, el tiempo que necesita para analizar las imágenes es algo excesivo para que opere en tiempo real. Y en último lugar, SJN\_MultiCueBGS, presenta la mayor precisión y mejora en tiempo a los dos anteriores, aunque no mejora en tiempo a los modelos gaussianos.

La Tabla 4 muestra los resultados promedio para los métodos analizados de cada gran grupo. La pertenencia de cada algoritmo a los distintos grupos está dispuesta en la Tabla 1.

Los modelos sencillos no alcanzan valores de precisión comparables debido a su simplicidad mientras los tiempos de ejecución son similares a los de otros modelos más elaborados. Ocurre algo similar con los modelos paramétricos, cuyos resultados no son nada precisos, así como con los difusos, que ya han quedado obsoletos siendo sustituidos por los neuronales-difusos que partiendo de esas bases fueron mejorados.

En cuanto a aquellos basados en características tales como texturas y colores tienen el gran inconveniente del tiempo, ya que su ejecución es muy lenta, estos ocurre para todos los métodos analizados menos para SJN\_MultiCueBGS, que es uno de los más rápidos y además logra una gran precisión.

**Tabla 4:** Resultados de los distintos grupos de modelos

	Básico	Estadístico de un gaussiano	Estadístico de múltiples gaussianos	Basado en vectores y valores propios	No paramétrico	Difuso	Neuronal / neuronal-difuso	Múltiples características (color, textura,..)
<b>Frames</b>	300	300	300	300	300	300	300	300
<b>Tiempo total</b>	31,28	<b>23,60</b>	28,35	32,91	<b>15,53</b>	67,99	63,56	227,57
<b>Tiempo medio</b>	0,10	0,08	0,10	0,11	0,05	0,23	0,21	0,76
<b>Tiempo máximo</b>	0,55	0,23	0,32	1,67	0,20	0,51	0,53	2,24
<b>Desviación media</b>	0,02	0,02	0,01	0,02	0,01	0,03	0,02	0,06
<b>Tasa de FP</b>	0,11	0,17	0,10	0,15	0,09	<b>0,07</b>	<b>0,04</b>	<b>0,06</b>
<b>Tasa de FN</b>	0,81	0,53	0,66	0,73	0,83	0,88	<b>0,41</b>	0,76
<b>Sensibilidad</b>	0,19	<b>0,47</b>	<b>0,34</b>	0,27	0,17	0,12	<b>0,59</b>	0,24
<b>Especificidad</b>	0,89	0,83	<b>0,90</b>	0,85	<b>0,91</b>	<b>0,93</b>	<b>0,96</b>	<b>0,94</b>
<b>PVP</b>	0,13	<b>0,22</b>	<b>0,22</b>	0,13	0,13	0,13	<b>0,57</b>	0,26
<b>PVN</b>	0,94	<b>0,96</b>	<b>0,95</b>	0,94	0,94	0,94	<b>0,97</b>	0,95
<b>F-Measure</b>	0,14	0,29	0,25	0,18	0,14	0,11	<b>0,58</b>	0,25

De los seis métodos seleccionados en la Tabla 3 como los mejores, tres de ellos pertenecen al grupo de los gaussianos (uno de ellos solo utiliza uno, mientras los otros dos utiliza múltiples), otros dos métodos pertenecen al grupo neuronal/neuronal-difuso, mientras el último de ellos pertenece al gran grupo de métodos basados en distintas características tales como texturas o colores.

En la Tabla 5 se muestra la F-measure obtenida para las distintas secuencias en función de los métodos más significativos, el promedio para los 28 métodos se puede encontrar en la Tabla 7 dentro del Anexo 2. A partir de dicha tabla se puede concluir que tipo de secuencias son las que más dificultad presentan para los distintos métodos.

El video que más problemas ha generado es aquel en el que se utiliza el zoom. Es el único video en el que la cámara no permanece fija y esto hace que los errores sean elevadísimos.

Le sigue el video de la fuente, esta secuencia muestra la fuente en primer plano, por lo que son muchos los pixeles que esta ocupa e induce error a los distintos métodos con los distintos cambios de altura del agua y similares. El error de esta secuencia afecta a todos los métodos excepto al método SJN\_MultiCueBGS, basado en texturas que funciona con una tasa de acierto aceptable. Algo similar ocurre en la secuencia que recoge un paisaje con un clima fatal (turbulencias) dónde hay gran movimiento del fondo: hojas volando, árboles que se mueven,...

Otros dos videos con resultados similares al de las turbulencias, son el nocturno, donde la iluminación cambia y el del sofá, que a diferencia de los anteriores, su dificultad radica en distintos objetos que se van depositando y permanecen un tiempo en sobre el sofá o la mesa, lo que hace los métodos consideren dichos objetos como parte del fondo.

El resto de secuencias presentan resultados similares, la secuencia de cámara térmica debe ser destacada, ya que no produce problemas en los métodos estadísticos ni neuronales pero afecta en gran medida al método de SJN\_MultiCueBGS.

La secuencia de los esquiadores sobre fondo nevado, al ser la nieve ligera, parece no dificultar la detección de movimiento para muchos métodos, es probable que el fondo nevado, homogéneo, ayude a ello. No obstante, existen métodos para los que presenta grandes dificultades como se puede apreciar en la Figura 10.

**Tabla 5:** F-measure de los mejores métodos para los distintos videos

Secuencia	F-Measure					
	LBSimpleGaussian	LBuzzyGaussian	LBmixtureOfGaussians	LBAadaptiveSOM	LBuzzyAdaptiveSOM	SJN_MultiCueBGS
PeopleInShade	0,50	0,54	<b>0,65</b>	0,56	0,53	0,53
Highway	0,459	0,535	<b>0,81</b>	<b>0,8</b>	<b>0,80</b>	<b>0,86</b>
Sofa	0,51	0,51	0,43	0,52	0,49	<b>0,53</b>
Turbulence03	0,106	0,117	0,214	0,267	0,303	<b>0,464</b>
ZoomInZoomOut	0,03	0,03	<b>0,06</b>	<b>0,07</b>	<b>0,07</b>	0,04
diningRoom	<b>0,89</b>	<b>0,89</b>	0,82	<b>0,86</b>	0,73	0,69
Traffic	0,27	0,27	0,41	0,50	<b>0,65</b>	0,44
Fountain02	0,02	0,03	0,08	0,16	0,18	<b>0,72</b>
Skating	0,48	0,50	0,74	<b>0,88</b>	<b>0,89</b>	0,81
Turnpike 0.5 fps	0,37	0,37	<b>0,59</b>	0,51	0,48	<b>0,62</b>
Street corner at night	0,26	0,28	0,35	0,35	0,35	<b>0,59</b>

## 2.6 Conclusiones

En primer lugar, debe tenerse en cuenta que el análisis ha sido realizado utilizando distintas secuencias con distintas dificultades, solo dos de ellas son tomadas con una cámara fija estándar en la calle similar a las que pueden ser utilizadas en la vida real, mientras las otras nueve han sido tomadas en situaciones de gran dificultad como mal clima, esquiadores, cámara nocturna, cámara de calor,...

La elección del método de análisis dependerá de los requisitos de aplicación. En general, la decisión presentará un compromiso entre velocidad y precisión, aunque también se deberá tener en cuenta el entorno en el que trabajará el sistema. En caso de que la precisión sea muy importante, el método que debemos aplicar será el de SJN\_MultiCueBGS (especialmente si el fondo en el que trabajará es de tipo dinámico, para aplicaciones con fondos más sencillos se podrá utilizar logrando gran precisión cualquiera de los dos métodos gaussianos mejorando la velocidad de procesado.

Por otra parte, analizando los métodos por grupos según el tipo de algoritmo que utilizan, descubrimos que los mejores métodos analizados se reparten entre los distintos grupos, por lo que, tanto los métodos basados en gaussianos como los neuronales-difusos o las nuevas técnicas basadas en distintas características pueden lograr grandes resultados.



**Figura 10:** Ejemplo de problemas generados en fondos dinámicos (para el método de VuMeter, uno de los más afectados).

La Tabla 5 indica que la precisión es mayor para los modelos neuronales y neuronales difusos, se debe tener en cuenta que en esta categoría la librería solo incluye los métodos LBAadaptiveSOM y LBFuzzyAdaptiveSOM, que, como se ha comentado anteriormente, han demostrado una gran precisión en el estudio. No obstante, estos modelos son los más novedosos y se espera, en los próximos años sean los más populares juntos a los métodos basados en características.

Por último, los gaussianos, que, actualmente, son los más usados, poseen una gran velocidad y muchos de sus métodos también gran precisión lo que los convierten en muy útiles para aplicaciones donde sea prioritaria la velocidad o donde el fondo no sea de gran complejidad.

Si analizamos los resultados según las distintas secuencias utilizadas, buscando los entornos favorables a la detección de movimiento y tratando de explicar aquellos en los que la problemática es mayor, que será donde, en trabajos futuros, se deberá mejorar. Las conclusiones se extraen de las Tabla 6 y Tabla 7 donde se presenta la F-measure, la medida más inmediata y sencilla para analizar precisión.

En dichas tablas se puede observar como aquellas secuencias con fondos dinámicos, donde hay movimiento de árboles, fuentes, hojas, viento o similares son los que presentan una gran dificultad, especialmente en los modelos gaussianos. Además, objetos que permanecen mucho tiempo en un sitio pueden generar problemas en la clasificación entre fondo o movimiento para todos los modelos. Y, por descontado, una cámara móvil, aunque solo sea por el hecho de aumentar y disminuir el zoom, supone un gran problema para todos los métodos.

## 3. Seguimiento de elementos móviles

### 3.1 Introducción. Objetivos

La segunda etapa en cualquier sistema de visión artificial es la de realizar un seguimiento del movimiento localizado.

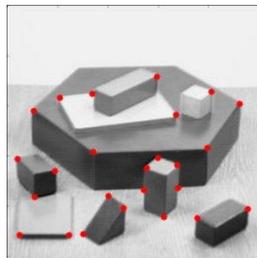
Esta etapa es imprescindible para cualquier aplicación, ya que puede permitir una identificación del movimiento detectado, identificando si se trata de un determinado objeto o de una persona, así como interpretar la acción que está realizando a partir de su trayectoria [47].

Debido a su gran utilidad existen multitud de algoritmos creados para resolver el problema, en función de las técnicas utilizadas se pueden clasificar en tres grandes grupos:

- Seguimiento de características (o de puntos): Cada elemento es representado por uno o varios puntos, se realizará un seguimiento de dichos puntos a lo largo de imágenes consecutivas. Se pueden clasificar en deterministas o estadísticos.
- Seguimiento de núcleo (kernel): En estas técnicas el objeto está representado por una región inicial a partir de la cual se realiza el cálculo del movimiento del objeto.
- Seguimiento de siluetas: Se realiza una valoración de la región del objeto de cada imagen utilizando la información que contiene en forma de densidad de aspecto o de modelos de forma.

Este trabajo se centra en el seguimiento de puntos debido a su sencillez y popularidad. Este tipo de procesos se puede dividir en dos fases:

1. Extracción de puntos característicos: Los puntos deben ser lo más robustos posible, o lo que es lo mismo, puntos que sean fácilmente detectables a lo largo de las imágenes.



**Figura 11:** Ejemplo de extracción de puntos característicos

2. Correspondencia de puntos, en inglés, *matching*: Existen dos formas de realizarlo, basándose en el flujo óptico o en la similitud de las características de un punto recogidas en su descriptor [48].

El objetivo de esta sección del trabajo será el de implementar y evaluar el funcionamiento de un sistema de seguimiento basado en el flujo óptico. Se experimentará con distintas secuencias para determinar en qué entornos es adecuado utilizar dicho método y en cuáles puede presentar dificultades.

### **3.2 Estado del arte**

En cuanto a la extracción de puntos existen diferentes algoritmos, uno de los primeros es el que Harris publicó en 1988 [49]. Basado en la detección de esquinas, su rapidez de cómputo es una de sus grandes ventajas.

Otro extractor de puntos característicos es el detector de SUSAN [50] (*Smallest Univalued Segment Assimilating Nucleus*), funciona aplicando una máscara circular sobre el píxel. Un píxel es punto característico en función del número de píxeles similares al píxel central situados dentro.

Uno de los que mayor popularidad tiene es el SIFT, en dicho algoritmo los puntos característicos son los máximos y los mínimos de las diferencias entre funciones gaussianas aplicadas a distintas escalas en la imagen suavizada [51], existe una versión más ligera conocida como SURF [52], basado en la matriz Hessiana que depende de imágenes integrales para reducir el coste computacional.

Recientemente, ha crecido el uso de los puntos ORB y BRIEF. BRIEF [53] (*Binary Robust Independent Elementary Features*) es un descriptor binario, el más sencillo de ellos, en primer lugar, ocupa los puntos de interés encontrados por un detector de los anteriores (una gran ventaja de este método es que permite combinarse con numerosos detectores). Tras ello, busca  $n$  pares de puntos en la región de interés haciendo una comparación de interés.

ORB [54] (*Oriented-FAST and Rotated-BRIEF*), como su nombre indica es una variación de BRIEF, que busca resolver el problema de la orientación. Adicionalmente, no ocupa muestras aleatorias como BRIEF, sino que trata de aprender cuales son los mejores pares para hacer la comparación.

Entre los algoritmos encargados de buscar correspondencias, los que lo hacen buscando la similitud de descriptores, cada punto característico obtenido debe tener asociado un descriptor. Este descriptor contiene información acerca de la escala y orientación. Seguidamente se calculan las distancias entre descriptores de un punto característico en la imagen fuente frente a todos los puntos característicos de la imagen destino. Aquella distancia mínima de entre todas servirá para identificar el par de puntos correspondidos entre la imagen fuente y la de destino. El proceso se repite para el resto de puntos característicos de la imagen fuente.

El otro tipo de métodos utiliza la idea del flujo óptico para realizar el *matching* de los puntos. En lugar de obtener una ecuación para cada píxel de la imagen, se obtiene una ecuación por cada punto característico de la imagen, entre estos algoritmos destacan el método de mínimos cuadrados, el de Horn y Schunk [55] y el de Lucas-Kanade [56], que en el siguiente epígrafe será explicado detenidamente, la diferencia en estos algoritmos radica en la forma en que cada uno de ellos soluciona el denominado problema de apertura que presenta el flujo óptico.

### 3.3 Algoritmo de Lucas y Kanade

El flujo óptico (*Optical Flow*) de una imagen se puede definir como el movimiento relativo de los píxeles de una imagen. En el caso ideal, en el que la iluminación se mantiene a lo largo de una secuencia, el flujo óptico se corresponde con el campo de velocidades. Suponiendo que la iluminación es constante, un punto de la imagen conserva su intensidad al moverse. En la Figura 12 se puede observar un ejemplo del cálculo de flujo óptico realizado en una imagen.

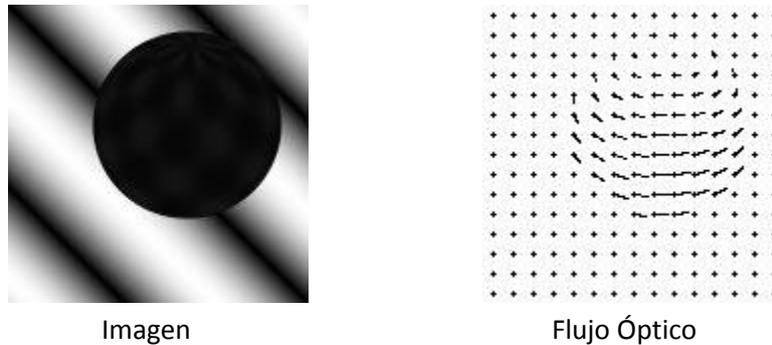


Figura 12: Ejemplo de flujo óptico para una esfera que gira en sentido horario

Desarrollando este concepto, se puede obtener la componente de la velocidad en la dirección del gradiente de intensidad, esto es, perpendicularmente a los contornos. No se podrá determinar la componente del flujo óptico en la dirección del contorno, lo que se conoce como problema de la apertura.

El problema de la apertura surge de la imposibilidad de asignar un valor unívoco al desplazamiento de un punto debido a que la región analizada es demasiado pequeña, también ocurre en la vista humana como se puede comprobar en la siguiente figura, también conocido como el ejemplo de “ojo de buey”:

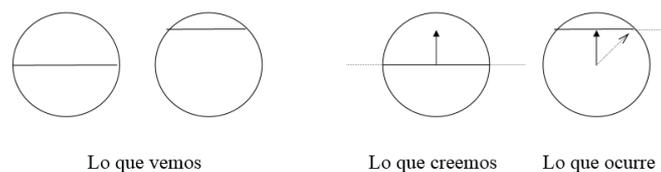
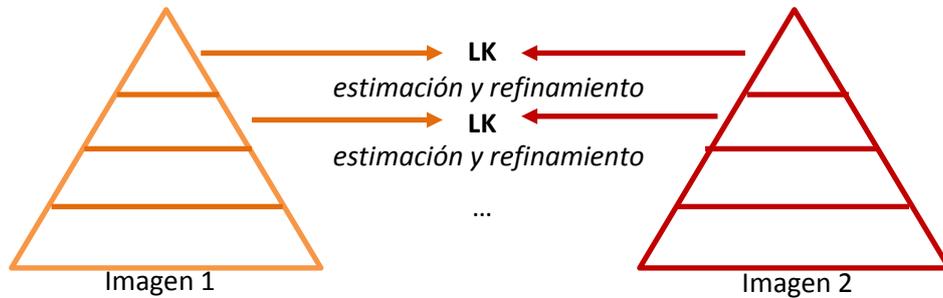


Figura 13: Ejemplo de problema de apertura, dónde a la vista únicamente del círculo, el observador cree que la línea se ha desplazado en vertical mientras en realidad esta lo ha hecho en diagonal.

El algoritmo de Lucas y Kanade [56] resuelve este problema recurriendo a la aplicación de criterios de vecindad. Se asume que el movimiento entre dos imágenes consecutivas para píxeles cercanos es pequeño, lo que permite considerar que el flujo óptico en un pequeño bloque es constante.

Se suele implantar como un método piramidal [57], así la representación de la imagen se calcula con varios niveles. La imagen resultado es una suma de los distintos niveles calculados: En el primer nivel se encuentra la imagen, en el segundo nivel se submuestra este resultado con un filtro gaussiano, y así consecutivamente. En el último nivel, con un barrido de píxeles, se calcula el flujo de movimiento, este se transporta a niveles más bajos dónde se vuelve a estimar el flujo de movimiento, se repite este procedimiento hasta el primer nivel dónde se obtiene el desplazamiento. La Figura 14 muestra un esquema del método piramidal:



**Figura 14:** Esquema del funcionamiento piramidal del método de Lucas y Kanade, donde las distintas alturas de las pirámides representan la imagen una vez aplicados los distintos filtros de distorsión.

Este método consigue hallar los más pequeños movimientos, este es también su principal problema, el cálculo de pequeñas variaciones en la imagen que puedan ser debidas a errores en la captación de las imágenes. Por otra parte, cuando el movimiento no es pequeño, cuando un punto no se mueve como sus vecinos o cuando el brillo no es constante también se inducen altos niveles de error.

### **3.4 Procedimiento de evaluación**

El algoritmo encargado de realizar el seguimiento de los objetos móviles ha de ser programado en dos etapas:

En primer lugar, mediante la función incluida en opencv *“goodFeaturesToTrack”* se detectarán los puntos significativos (esquinas) en la imagen, en este caso se tomarán 20 puntos.

Se utilizará la función *“calcOpticalFlowPyrLK”*, que incluye el algoritmo de Lucas Kanade lo que permite localizar los puntos significativos de la primera imagen en la segunda; con ello podremos calcular el desplazamiento del punto entre ambos frames. Este procedimiento se repite para todas las imágenes de la secuencia, en el siguiente esquema se puede apreciar gráficamente:

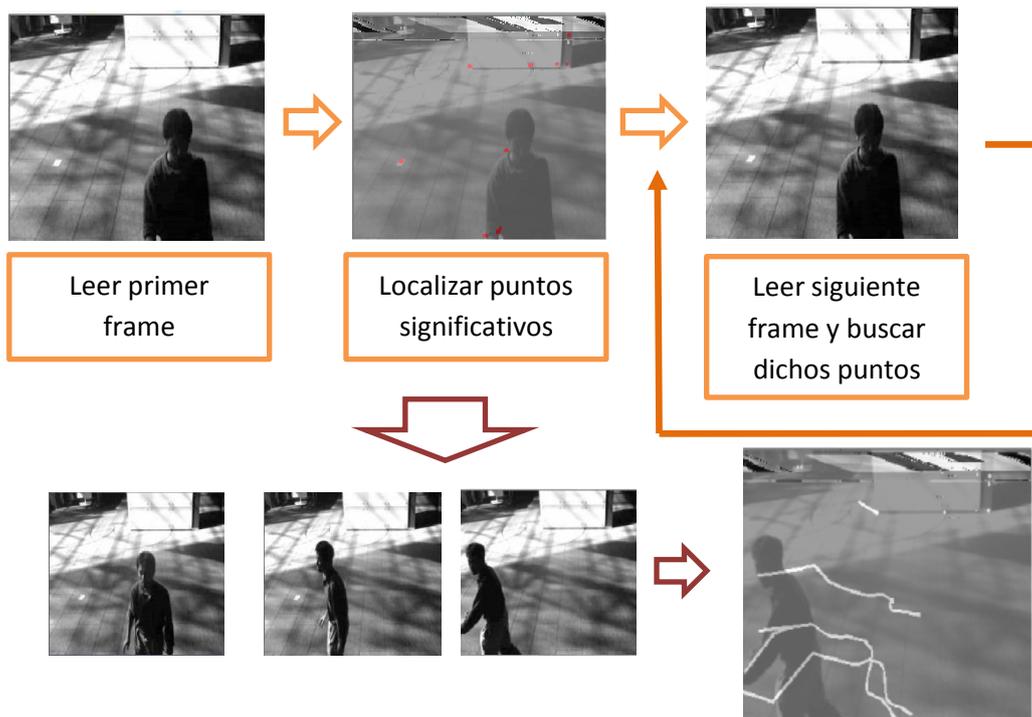


Figura 15: Esquema del procedimiento para el seguimiento.

El código implantado, en C++, se puede encontrar en el Anexo 1.

### **3.5 Método de evaluación**

En este caso, la evaluación se realizará de forma manual, se analizarán pequeñas secuencias dónde ocurra un determinado evento (persona pasando, personas que se cruzan, coche atravesando la carretera,...), se medirá el número de puntos característicos que el algoritmo sitúa en la región de movimiento al comienzo y si estos se siguen mostrando correctamente a lo largo de los frames. Se evaluará cada 10 frames y se anotará en qué circunstancias se pierden dichos frames.

### **3.6 Resultados experimentales**

En la Tabla 9 del Anexo 2 se recogen los resultados completos de los experimentos realizados, en las siguientes figuras se muestran ejemplos visuales de secuencias en las que el método falla.

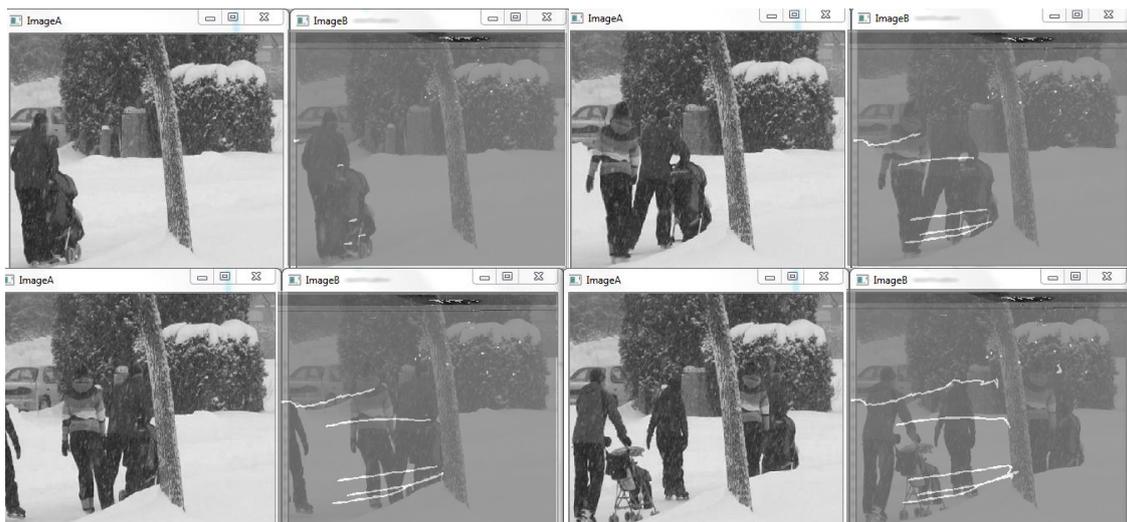
La Figura 16 muestra un ejemplo del cruce entre personas: al comienzo uno de los individuos se encuentra en el sofá sentado y el otro a su izquierda, de pie, 2 puntos a seguir se encuentran en el torso del hombre sentado (nótese que en el primer intervalo cuando se acaba de sentar le siguen en el movimiento de su torso hacia arriba), mientras 5 puntos siguen al hombre que camina.

Cuando se produce el cruce todos los puntos (los 7) siguen al hombre que camina, que es el que realiza el cruce por delante, siguiéndole correctamente hasta que abandona la escena. Otras secuencias que presentan cruces estudiadas en este trabajo también generan este problema donde los puntos del elemento que queda en la parte trasera en el cruce se pierden.



**Figura 16:** Ejemplo de error en el seguimiento al cruzarse dos personas

Las Figura 17 y 18 muestran otros errores que ocurren en el seguimiento de elementos como pueden ser la pérdida de puntos al desaparecer el movimiento de la escena un breve periodo de tiempo o la pérdida de puntos al realizar una persona un movimiento que cambie su forma como un giro.



**Figura 17:** Ejemplo de error de seguimiento al pasar por detrás de un árbol



**Figura 18:** Ejemplo de error en el seguimiento de un giro

### **3.7 Conclusiones**

En primer lugar, podemos concluir que el seguimiento se realiza mejor en coches/objetos que en personas; estas presentan mayor dificultad debido a los movimientos de las distintas partes, giros, etc. En cambio los coches se mueven de forma constante como un bloque.

Los fondos dinámicos también presentan una gran dificultad, como se puede comprobar tanto en los videos de la fuente o de la nevada donde se pierden gran cantidad de puntos en movimientos sencillos que no presentan dificultad en otros entornos.

Se han estudiado gran cantidad de secuencias dónde se producen cruces tanto entre personas como entre coches y se puede concluir que este es otro de los grandes problemas ya que en todas ellas el sujeto que pasa por detrás en el cruce se pierde su seguimiento y en alguno de los casos también el del otro, esto se podría solucionar implementando un filtro de kalman que nos permitiera predecir los puntos en los frames siguientes. Un problema similar se puede observar en el video de las turbulencias donde al pasar las personas por detrás de un gran poste muchas veces se pierde también el seguimiento.

Por último, se observa que de los 20 puntos característicos que se toman en cada imagen nunca más de 10 pertenecen a las regiones a estudiar, aquellas dónde existe movimiento.

## **4. Implementación del sistema completo**

### **4.1 Introducción. Objetivos**

Una vez analizadas ambas etapas del sistema de videovigilancia, ya se conocen los métodos más precisos y más rápidos que incluye la librería BGS. También se conocen aquellas secuencias que presentan más problemas a la hora de aplicar visión artificial, tanto en la detección de elementos móviles como en el seguimiento de estos.

Como final del trabajo se crea un algoritmo que combina los utilizados en las dos etapas anteriores que sirva como punto de partida para nuevas aplicaciones de videovigilancia en trabajos futuros. Se aplicarán los conocimientos extraídos de los anteriores apartados del trabajo.

### **4.2 Implementación**

En primer lugar se implementará el código utilizado para la detección de movimiento, se ejecutará durante 10 frames para lograr una mejor detección de elementos móviles (con más frames en varios de los métodos se puede lograr un mejor conocimiento del fondo).

Se utilizarán los siguientes métodos de sustracción de fondo: LBSimpleGaussian, LBFuzzyGaussian, LBMixtureOfGaussians, LBAadaptiveSOM, LBFuzzyAdaptiveSOM y SJN\_MultiCueBGS, especialmente el último ya que fue el que mejores resultados presentó para los experimentos realizados durante este trabajo.

Con la máscara generada en el último de los frames (el décimo), se detectarán los bordes de los distintos movimientos utilizando la función *findContours*. Esto se combinará con el código utilizado para el seguimiento de objetos, de esta forma, los puntos característicos que se utilizarán para el *tracking* no solo dependerán de la imagen sino que también se tendrá en cuenta el movimiento que está ocurriendo en la escena. Esto logrará una mayor precisión de los puntos seleccionados para el seguimiento, detectándose un mayor número de ellos en las zonas de mayor interés.

El resto del programa coincide con el utilizado en el seguimiento de objetos, estos 20 puntos de la imagen se localizarán en los siguientes frames, realizando así un seguimiento del movimiento detectado. En la siguiente figura se muestra un esquema del funcionamiento de este sistema completo:

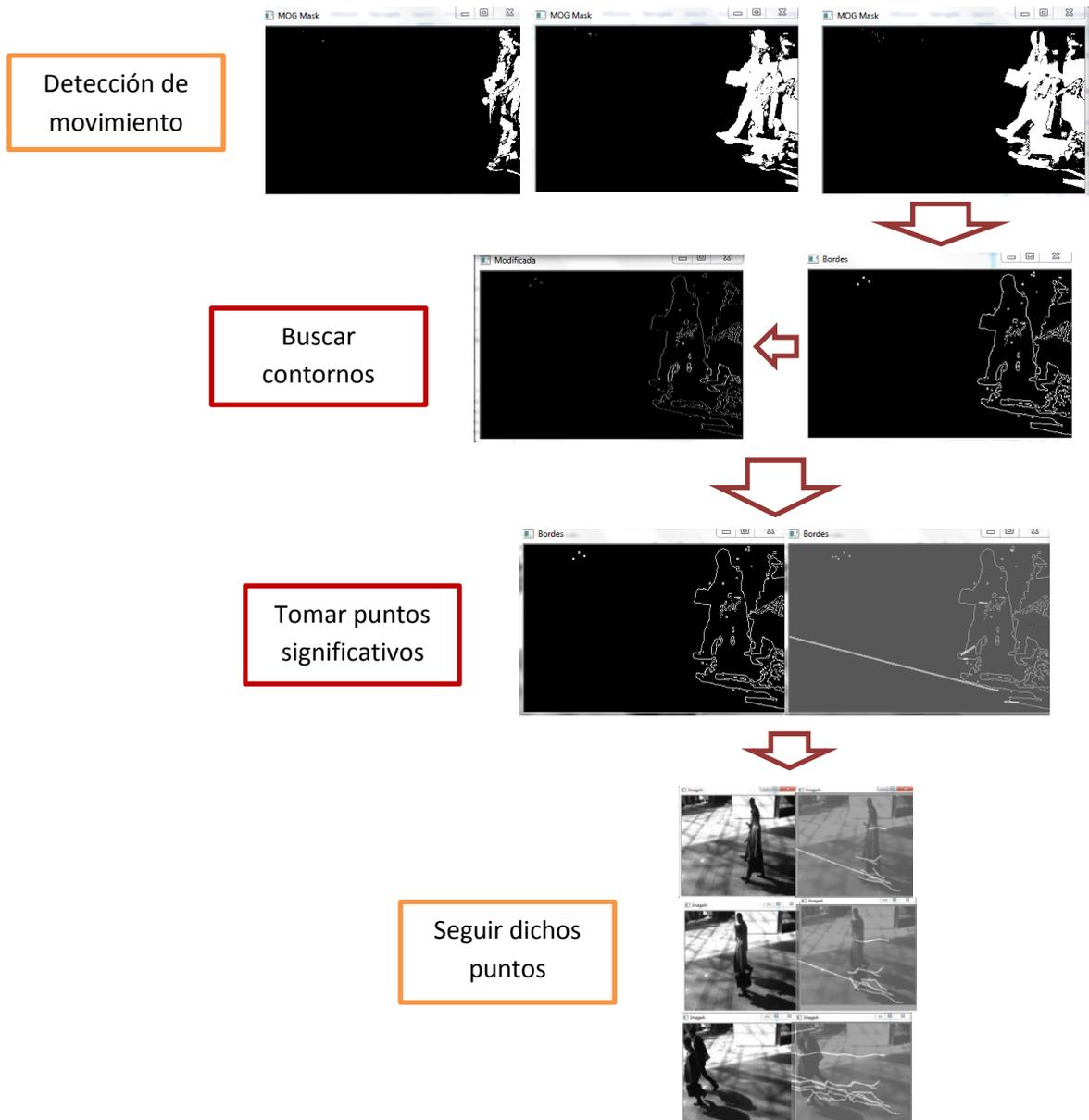


Figura 19: Esquema de la implementación de un sistema completo de vigilancia

### **4.3 Resultados y conclusiones**

En las Figura 20 y Figura 15 se puede observar el funcionamiento del sistema completo. La primera de ellas muestra una secuencia dónde un hombre entra en escena, se agacha y recoge objetos antes de salir de nuevo mientras la segunda muestra una carretera por la que van avanzando diversos coches.

En la Figura 20 se puede observar como existen problemas con el seguimiento de algunos puntos cuando el hombre se agacha para recoger los objetos como se estudió en el apartado 3, el seguimiento de los coches (Figura 15) funciona sin ningún tipo de problemas.

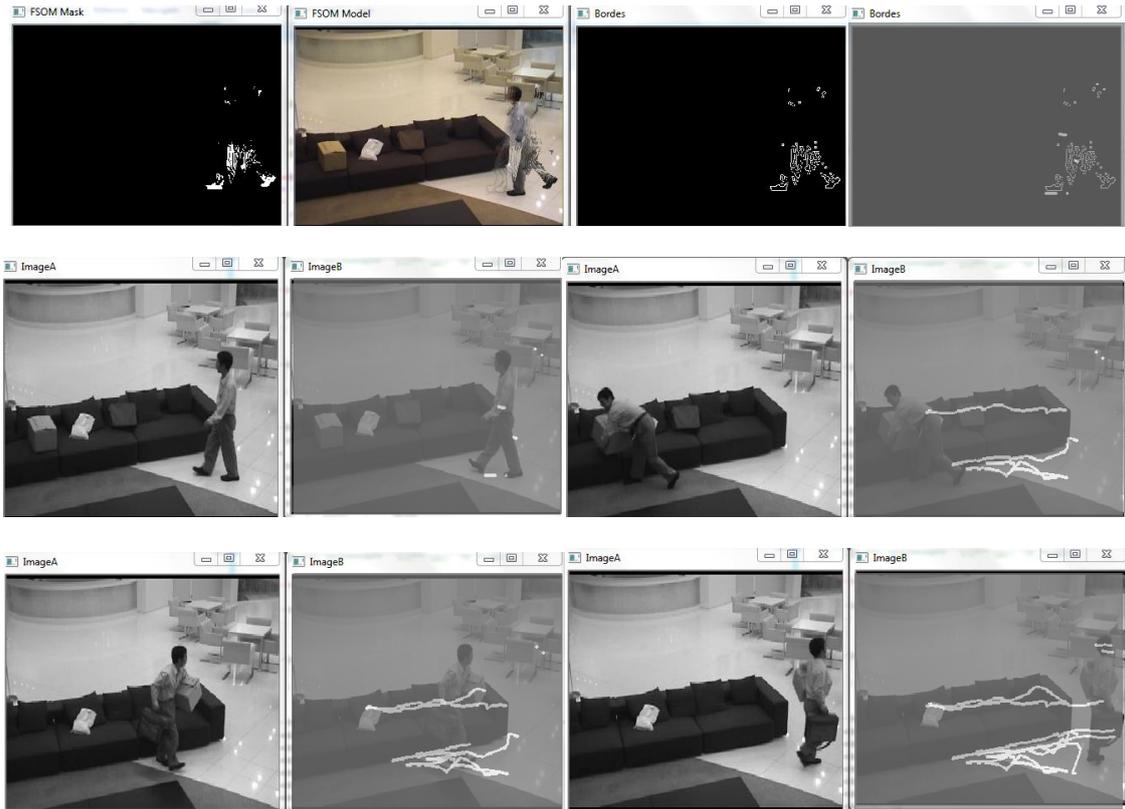


Figura 20: Ejemplo de implantación del algoritmo completo en la secuencia del sofá



Figura 21: Ejemplo de implantación del sistema completo en secuencia de carretera (highway).

La única diferencia que existe con respecto a los seguimientos estudiados en el apartado 3 es la mayor presencia de puntos en las zonas que presentan movimiento. El seguimiento sigue presentando problemas similares a los analizados con anterioridad.

## **5. Conclusiones y trabajo futuro**

En primer lugar, se ha realizado un estudio de los distintos métodos de detección de elementos móviles, se ha evaluado cada uno de ellos atendiendo tanto al tiempo de procesado como a su precisión a la hora de separar fondo de elementos que presentan movimiento.

Se han obtenido buenos resultados, especialmente en cuanto a tiempo de procesado, para algunos métodos gaussianos (LBSimpleGaussian y LBFuzzyGaussian principalmente), aunque, en cuanto a precisión se han obtenido mejores resultados para nuevos modelos tanto neuronales-difusos como basados en distintas características (color, texturas, formas,...), estos modelos generan un gran avance, especialmente para fondos dinámicos (viento, agua, nieve,...) para los que los métodos estadísticos presentan grandes dificultades, aunque el tiempo de procesado sigue siendo un problema.

El método de SJN\_MultiCueBGS, englobado dentro de los que basan la detección en distintas características presenta el mejor resultado en cuanto a precisión, en cambio, el tiempo de procesado es superior al de los métodos gaussianos, por lo que para entornos sencillos siguen siendo estos últimos los más adecuados.

En segundo lugar, se han analizado distintos modelos de seguimiento de elementos móviles, tanto en su etapa de detección de puntos como en la de *matching*. Se ha estudiado en profundidad el algoritmo de Lucas-Kanade, uno de los más utilizados basado en el flujo óptico. Se han analizado numerosas secuencias y se ha podido determinar que dicho método, al igual que los métodos de detección, presenta dificultades en entornos dinámicos. También sufre graves problemas en el seguimiento de objetos o personas que se cruzaban entre sí o que pasan por detrás de objetos fijos.

Por último, se ha implantado un sistema de videovigilancia completo, capaz de detectar movimiento con los mejores algoritmos analizados y de realizar su seguimiento con el algoritmo de Lucas-Kanade. Este sistema puede ser utilizado como base para aplicaciones más complejas como la detección de objetos abandonados o la regulación del tráfico.

También puede ser de gran utilidad, como trabajo futuro, la mejora de los algoritmos o la creación de otros nuevos para su utilización en entornos complicados, especialmente aquellos con fondos dinámicos, que, como se ha analizado, presentan dificultad para los métodos actuales.

## ANEXO 1. Código C++

En primer lugar, se muestran los códigos utilizados para el análisis de las distintas librerías de sustracción de fondo. Se ha implementado un contador que comienza a contar al inicio de cada ciclo, ejecuta el modelo a analizar y finalmente memoriza el tiempo que se ha tardado en realizar.

Se guardará tanto el tiempo total como el máximo, se descartarán las primeras iteraciones de tiempo ya que sus valores son superiores al funcionamiento normal.

```
//Definimos
double performancecounter_diff(LARGE_INTEGER *a, LARGE_INTEGER *b){
    LARGE_INTEGER freq;
    QueryPerformanceFrequency(&freq);
    return (double)(a->QuadPart - b->QuadPart) / (double)freq.QuadPart;
}

//main

while( frameNumber<frameUltimo+1)
{
    LARGE_INTEGER iniTime, endTime;
    QueryPerformanceCounter(&iniTime);

    //Programa

    QueryPerformanceCounter(&endTime);
    //Imprime tiempo del ultimo ciclo:
    printf("%.16g segundos\n", performancecounter_diff(&endTime,
&iniTime));
    //Actualiza tiempo maximo y tiempo total
    if (frameNumber>frames+1) //Eliminamos los primeros valores ya que
son los más lentos.
    {
        tiempototal= tiempototal +
            (performancecounter_diff(&endTime, &iniTime));
        if ((performancecounter_diff(&endTime, &iniTime))>tiempomax)
            {tiempomax=(performancecounter_diff(&endTime, &iniTime));
            }
    }
}
```

Una vez calculados estos tiempos, se implementará el cálculo de la desviación total como la suma de las diferencias de estas con el tiempo medio (en valor absoluto), para luego calcular la desviación media.

Para realizar el análisis cualitativo se utilizarán 4 contadores, cada uno de ellos comienza a cero, se analiza tanto el frame de la imagen como el de la solución. Al estar estas en binario tomará valor 0 cuando el pixel es negro y 1 si este es blanco. De esta forma según las condiciones ya comentadas se podrá clasificar en FP, TP, FN o TN.

Se sumarán los valores obtenidos calculándose la media para toda la secuencia.

```

//Compara pixel a pixel con la solución: (contadores a 0)
TP=0;
FP=0;
TN=0;
FP=0;

for(int row = 0; row < img_sol.rows; ++row)
{
    const int* inp = img_sol.ptr<int>(row);
    const int* inp2 = img_mask.ptr<int>(row);
    for (int col = 0; col < img_sol.cols; ++col)
    {
        //Establece valor para
        if(inp[col]==0) //El pixel es negro
        {
            pixelsol=0;
        } else //El pixel NO es negro
        {
            pixelsol=1;
        }

        if (inp2[col] == 0) //El pixel es negro
        {
            pixel=0;
        } else //El pixel NO es negro
        {
            pixel=1;
        }

        //Compara el pixel obtenido con el de la solución
        if ((pixel==0) & (pixelsol==0)) //Ambos negro, TN
        {
            TN=TN+1;
        }
        if ((pixel==0) & (pixelsol==1)) //Sale negro y en la solución lo
        hace blanco, FN
        {
            FN=FN+1;
        }
        if ((pixel==1) & (pixelsol==1)) //Ambos blancos, TP
        {
            TP=TP+1;
        }
        if ((pixel==1) & (pixelsol==0)) //Detecta movimiento pero no lo
        hay, FP
        {
            FP=FP+1;
        }
    }
}

```

En cuanto al código utilizado en el *tracking*:

En una primera parte se leen dos imágenes, se buscan los puntos significativos de esta (20) con la función *goodFeaturesToTrack*, dichos puntos se buscarán en la segunda imagen utilizando el algoritmo de lucas kanade, para eso se utilizará el comando *calcOpticalFlowPyrLK*.

De esta forma los dos primeros frames de la secuencia ya pueden ser analizados.

```
Mat img_1 = imread( img1, CV_LOAD_IMAGE_GRAYSCALE );
Mat img_2 = imread( img2, CV_LOAD_IMAGE_GRAYSCALE );

Size img_sz = img_1.size();
Mat img_3(img_sz,1);
int win_size = 15;
int maxCorners = 20;
int esquinas = 20;
double qualityLevel = 0.05;
double minDistance = 5.0;

std::vector<cv::Point2f> corners1;
corners1.reserve(maxCorners);

std::vector<cv::Point2f> corners2;
corners2.reserve(maxCorners);

goodFeaturesToTrack(img_1,corners1,maxCorners,qualityLevel,minDistance,cv::
Mat());
cornerSubPix( img_1, corners1, Size( win_size, win_size ), Size( -1, -1 ),
TermCriteria( CV_TERMCRIT_ITER | CV_TERMCRIT_EPS, 20, 0.03 ) );

// Call Lucas Kanade algorithm
CvSize pyr_sz = Size( img_sz.width+8, img_sz.height/3 );
std::vector<uchar> features_found;
features_found.reserve(maxCorners);

std::vector<float> feature_errors;
feature_errors.reserve(maxCorners);

calcOpticalFlowPyrLK( img_1, img_2, corners1, corners2, features_found,
feature_errors ,Size( win_size, win_size ), 5,cvTermCriteria( CV_TERMCRIT_ITER |
CV_TERMCRIT_EPS, 20, 0.3 ), 0 );

// Make an image of the results

for( int i=0; i < features_found.size(); i++ ){
    cout<<"Distancia en el punto: "<< i << " : " <<
feature_errors[i]<<endl;

    Point p0( ceil( corners1[i].x ), ceil( corners1[i].y ) );
    Point p1( ceil( corners2[i].x ), ceil( corners2[i].y ) );
    line( img_3, p0, p1, CV_RGB(255,255,255), 2 );
}
}
```

La segunda parte del código incluye un bucle que se repetirá para el resto de frames. El funcionamiento es idéntico a la primera parte, pero en este caso no será necesario buscar puntos, simplemente la segunda imagen se copia en el lugar de la primera y se buscan en ella los puntos significativos (los mismos que ya se leyeron en el primer frame).

Este procedimiento se repetirá sucesivamente hasta que decidamos interrumpir el seguimiento de los objetos del principio.

```

while( frameNumber<frameUltimo+1) //Para el resto lo mismo pero 2 pasa a ser 1.
{
    Mat img_2 = imread( img2, CV_LOAD_IMAGE_GRAYSCALE );
    // Call Lucas Kanade algorithm
    CvSize pyr sz = Size( img_sz.width+8, img_sz.height/3 );
    std::vector<uchar> features_found;
    features_found.reserve(maxCorners);

    std::vector<float> feature_errors;
    feature_errors.reserve(maxCorners);

    calcOpticalFlowPyrLK( img_1, img_2, corners1, corners2,
features_found, feature_errors ,Size( win_size, win_size ), 5,cvTermCriteria(
CV_TERMCRIT_ITER | CV_TERMCRIT_EPS, 20, 0.3 ), 0 );
    for( int i=0; i < features_found.size(); i++ )
    {
        Point p0( ceil( corners1[i].x ), ceil( corners1[i].y ) );
        Point p1( ceil( corners2[i].x ), ceil( corners2[i].y ) );

        line( img_3, p0, p1, CV_RGB(255,255,255), 2 );

        dx=(corners1[i].x-corners2[i].x);
        dy=(corners1[i].y-corners2[i].y);
        d=sqrt(dx*dx+dy*dy);
        cout<<"Distancia en el punto: "<< i << " :" <<d<<endl;
    }
    frameNumber++;
    waitKey(0);
    img_2.copyTo(img_1);
}
}

```

Por último, en la implementación de un sistema completo de videovigilancia, que detecta movimiento y realiza su seguimiento, el código implementado es la suma de los códigos anteriores. Para combinarlos, se hace que los puntos elegidos para realizar el seguimiento sean elegidos a partir de la máscara creada por el contorno del movimiento detectado con la librería bgs. El código utilizado para dicho algoritmo es el siguiente:

```
//ANALIZA MOVIMIENTO ccon bgs, analizamos 10 frames:

while (frameNumber<(frames+10)){
    std::stringstream ss;
    ss << frameNumber;
        //Leyendo frames de entrada
    std::string fileName = "in"
        "00" + ss.str() + ".jpg";
    std::cout << "leyendo " << fileName << std::endl;
    Mat img_2bgs = imread( fileName, CV_LOAD_IMAGE_COLOR );

    cv::Mat img_bkgmodel;
    bgs->process(img_2bgs, img_mask, img_bkgmodel); // by default, it shows
automatically the foreground mask image
    frameNumber++;
    cvWaitKey(0);
}

std::cout << "---Análisis bgs realizado---"<< std::endl;
// Buscamos contornos
//Una vez utilizada la librería BGS buscamos contornos:
Mat canny_output;
vector<vector<Point> > contours;
vector<Vec4i> hierarchy;
int thresh = 100;
// Detectar los bordes con un umbral min = 100 y max = 200
Canny(img_mask, canny_output, thresh, thresh * 2);
// Mostrar los bordes detectados con Canny
imshow("Bordes", canny_output);
// Buscar los contornos de la imagen, se almacenan en contours
findContours(canny_output, contours, hierarchy, CV_RETR_TREE,
CV_CHAIN_APPROX_SIMPLE, Point(0, 0));
// Mostrar la Imagen modificada por la funcion findContours
imshow("Modificada", canny_output);
// Dibujar los contornos encontrados
Mat drawing = Mat::zeros(canny_output.size(), CV_8UC3);
for (size_t i = 0; i< contours.size(); i++)
{
    Scalar color = CV_RGB(0, 255, 0);
    drawContours(drawing, contours, (int)i, color, 2, 8, hierarchy, 0,
Point());
}

Mat img_1 = imread( fileName, CV_LOAD_IMAGE_GRAYSCALE );
Mat img_2 = imread( img2, CV_LOAD_IMAGE_GRAYSCALE);
std::vector<cv::Point2f> corners1;
corners1.reserve(maxCorners);
std::vector<cv::Point2f> corners2;
corners2.reserve(maxCorners);

goodFeaturesToTrack(img_1,corners1,maxCorners,qualityLevel,minDistance,canny_outpu
t);
```

## ANEXO 2. Resultados completos

**Tabla 6:** Clasificación en función del tiempo medio de ejecución para los videos de 76800 pixeles.

	Tiempo/frame (s)	Método	fps
1	0,0120	StaticFrameDifferenceBGS	83,39
2	0,0212	AdaptiveBackgroundLearning	47,14
3	0,0260	MixtureOfGaussianV1BGS	38,47
4	0,0262	VuMeter	38,11
5	0,0288	DPAadaptiveMedianBGS	34,69
6	0,0380	MixtureOfGaussianV2BGS	26,33
7	0,0444	LBSimpleGaussian	22,54
8	0,0459	GMG	21,77
9	0,0485	DPWrenGABGS	20,62
10	0,0495	DPZivkovicAGMMBGS	20,18
11	0,0507	LBFuzzyGaussian	19,73
12	0,0635	IndependentMultimodalBGS	15,75
13	0,0659	DPMeanBGS	15,18
14	0,0698	DPEigenbackgroundBGS	14,33
15	0,0886	LBMixtureOfGaussians	11,29
16	0,1073	T2FGMM_UM	9,32
17	0,1261	DPGrimsonGMMBGS	7,93
18	0,1311	LBAadaptiveSOM	7,63
19	0,1347	T2FGMM_UV	7,43
20	0,1387	LBFuzzyAdaptiveSOM	7,21
21	0,1511	SJN_MultiCueBGS	6,62
22	0,1805	DPPratiMediodBGS	5,54
23	0,2191	T2FMRF_UM	4,56
24	0,2653	MultiLayerBGS	3,77
25	0,2806	T2FMRF_UV	3,56
26	0,8583	LOBSTERBGS()	1,17
27	1,2827	SuBSENSEBGS()	0,78
28	1,3538	DPTtextureBGS	0,74

**Tabla 7:** F-Measure en los distintos videos (promedio de todos los métodos)

Secuencia	F-Measure
PeopleInShade	0,23
Highway	0,257
Sofa	0,14
Turbulence03	0,1302
ZoomInZoomOut	0,02
diningRoom	0,292
Traffic	0,25
Fountain02	0,0986
Skating	0,34
Turnpike 0.5 fps	0,2485
Street corner at night	0,13

La Tabla 6 recoge el tiempo de procesado para aquellas secuencias analizadas filmadas en un tamaño de 76800 pixeles, en la columna de la derecha muestra a cuantos frames por segundo (fps) equivale dicho tiempo. Para que un sistema de videovigilancia pueda funcionar “en vivo” será necesario que la velocidad de procesado sea superior a 25 fps, ya que esta es la velocidad mínima a la que graban las cámaras habitualmente.

Debe tenerse siempre en cuenta que la velocidad también depende de la máquina que ha de procesar las imágenes por lo que se deberá probar con el sistema que se encargará de hacer funcionar la aplicación.

La Tabla 7 muestra la F-measure media de los 28 métodos analizados para cada una de las secuencias, se observa como los peores resultados ocurren para fondos dinámicos ya que estos presentan grandes dificultades en la sustracción para casi todos los métodos excepto aquellos basados en texturas.

La siguiente tabla recoge los resultados completos para cada una de las secuencias analizadas: Los positivos y negativos (tanto verdaderos como falsos) promedio en la secuencia así como los distintos parámetros a los que se alude a lo largo del trabajo:

**Tabla 8:** Resultados completos para cada una de las secuencias.

PeopleInSha de (650 a 949)	StaticFrameDifferenceBGS	MixtureOfGaussianV1BGS;	MixtureOfGaussianV2BGS	AdaptiveBackgroundLearning	GMG	DPAdaptiveMedianBGS	DPGrimsonGMMBGS	DPZivkovicAGMMBGS	DPMeanBGS	DPWrenGABGS	DPPratiMediodBGS	DPEigenbackgroundBGS	DPTextureBGS	TZFGMM_UM	TZFGMM_UV	TZFMRF_UM	TZFMRF_UV	MultiLayerBGS	LBSimpleGaussian	LBfuzzyGaussian	LBmixtureOfGaussians	LBAdaptiveSOM	LBfuzzyAdaptiveSOM	VuMeter	IndependentMultimodalBGS	SJN_MultiCueBGS	SUBSENSEBGS()	LOBSTERBGS()	
Frames	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300
Tiempo total	2,64	5,93	6,898	8,625	21,72	16,9	55,33	15,26	22,98	18,82	68,35	23,02	478,7	47,02	57,01	70,91	73,04	94,48	15,49	19,93	29,57	40,17	50	10,61	25,22	37,75	425,4	238,9	
Tiempo medio	0,009	0,02	0,023	0,029	0,073	0,057	0,186	0,051	0,077	0,063	0,229	0,077	1,606	0,158	0,191	0,238	0,245	0,317	0,052	0,067	0,099	0,135	0,168	0,036	0,085	0,127	1,427	0,802	
Tiempo máximo	0,038	0,145	0,074	0,015	0,196	0,2	0,888	0,112	0,218	0,156	1,527	1,134	2,687	0,401	0,698	0,435	0,603	0,591	0,147	0,334	0,251	0,251	0,321	0,084	0,383	0,323	2,691	1,339	
TP	1593	421,1	535,7	1549	1167	773,5	1109	1310	291,6	1127	1076	1559	1168	485,4	1261	390,8	872,9	1243	4834	4696	4454	4190	3869	606,2	762,7	3519	1267	1599	
FP	10305	2176	2875	8503	6545	4622	5424	7981	1361	6779	5173	9194	6270	2292	6556	1830	4200	5851	6886	5201	1556	3088	3014	3287	4124	2061	6437	8595	
TN	74772	82901	82202	76574	78532	80455	79653	77096	83716	78298	79904	75883	78807	82785	78521	83247	80877	79226	78191	79876	83521	81989	82063	81790	80953	83016	78640	76482	
FN	6050	7222	7107	6093	6476	6869	6534	6332	7351	6516	6567	6084	6475	7157	6382	7252	6770	6399	2809	2947	3189	3453	3774	7037	6880	4124	6376	6044	
TP+FN	7643	7643	7643	7643	7643	7643	7643	7643	7643	7643	7643	7643	7643	7643	7643	7643	7643	7643	7643	7643	7643	7643	7643	7643	7643	7643	7643	7643	
TN+FP	85077	85077	85077	85077	85077	85077	85077	85077	85077	85077	85077	85077	85077	85077	85077	85077	85077	85077	85077	85077	85077	85077	85077	85077	85077	85077	85077	85077	
Total frames	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	9272 0	
Tasa de FP	0,121	0,026	0,034	0,1	0,077	0,054	0,064	0,094	0,016	0,08	0,061	0,108	0,074	0,027	0,077	0,022	0,049	0,069	0,081	0,061	0,018	0,036	0,035	0,039	0,048	0,024	0,076	0,101	
Tasa de FN	0,792	0,945	0,93	0,797	0,847	0,899	0,855	0,829	0,962	0,853	0,859	0,796	0,847	0,936	0,835	0,949	0,886	0,837	0,367	0,386	0,417	0,452	0,494	0,921	0,9	0,54	0,834	0,791	
Sensibilidad	0,208	0,055	0,07	0,203	0,153	0,101	0,145	0,171	0,038	0,147	0,141	0,204	0,153	0,064	0,165	0,051	0,114	0,163	0,633	0,614	0,583	0,548	0,506	0,079	0,1	0,46	0,166	0,209	
Especificidad	0,879	0,974	0,966	0,9	0,923	0,946	0,936	0,906	0,984	0,92	0,939	0,892	0,926	0,973	0,923	0,978	0,951	0,931	0,919	0,939	0,982	0,964	0,965	0,961	0,952	0,976	0,924	0,899	
PVP	0,134	0,162	0,157	0,154	0,151	0,143	0,17	0,141	0,176	0,142	0,172	0,145	0,157	0,175	0,161	0,176	0,172	0,175	0,412	0,474	0,741	0,576	0,562	0,156	0,156	0,631	0,164	0,157	
PVN	0,925	0,92	0,92	0,926	0,924	0,921	0,924	0,924	0,919	0,923	0,924	0,926	0,924	0,92	0,925	0,92	0,923	0,925	0,965	0,964	0,963	0,96	0,956	0,921	0,922	0,953	0,925	0,927	
F-Measure	0,163	0,082	0,097	0,175	0,152	0,119	0,156	0,155	0,063	0,145	0,155	0,169	0,155	0,093	0,163	0,079	0,137	0,169	0,499	0,535	0,652	0,562	0,533	0,105	0,122	0,532	0,165	0,179	

**Highway  
(600 a 899)**

	StaticFrameDifferenceBGS	MixtureOfGaussianV1BGS;	MixtureOfGaussianV2BGS	AdaptiveBackgroundLearning	GMG	DPAdaptiveMedianBGS	DPGrimsonGMMBGS	DPZivkovicAGMMBGS	DPMeanBGS	DPWrenGABGS	DPPratiMediodBGS	DPEigenbackgroundBGS	DPTextureBGS	TZFGMM_UM	TZFGMM_UV	TZFMRF_UM	TZFMRF_UV	MultilayerBGS	LBSimpleGaussian	LBfuzzyGaussian	LBmixtureOfGaussians	LBAdaptiveSOM	LBfuzzyAdaptiveSOM	VuMeter	IndependentMultimodalBGS	SJN_MultiCueBGS	SubSENSEBGS(!)	LOBSTERBGS(!)
Frames	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300
Tiempo total	4,304	5,177	8,839	5,464	13,65	8,743	34,51	13,62	19,96	17,57	53,38	24,19	375,6	23,95	36,31	54,78	70,2	76,35	9,062	12,33	17,75	30,24	39,75	6,418	19,68	38,51	364,7	248,5
Tiempo medio	0,014	0,017	0,03	0,018	0,046	0,029	0,116	0,046	0,067	0,059	0,179	0,081	1,26	0,08	0,122	0,184	0,236	0,256	0,03	0,041	0,06	0,101	0,133	0,022	0,066	0,129	1,224	0,834
Tiempo máximo	0,336	0,062	0,169	0,139	0,123	0,189	0,309	0,091	0,159	0,279	1,255	1,363	2,63	0,138	0,453	0,349	0,567	0,692	0,051	0,088	0,14	0,209	0,318	0,044	0,271	0,269	3,406	1,394
TP	2301	427,8	923,7	1824	1438	710,4	1229	1171	525,1	1017	1047	1618	1027	193,1	1473	126,9	897,9	1167	6760	6707	6381	5917	5529	781,8	1085	6821	1011	993,9
FP	15704	2275	5408	12639	9374	4226	7555	7345	2566	6239	6171	10324	5617	1247	9733	909,1	5454	6056	15395	11073	2150	1565	1015	4511	5996	1728	5032	5673
TN	53785	67214	64081	56850	60115	65263	61934	62144	66924	63250	63318	59166	63872	68242	59756	68580	64035	63433	54094	58416	67339	67924	68474	64978	63493	67761	64457	63816
FN	5010	6883	6387	5487	5873	6601	6082	6140	6786	6294	6264	5693	6284	7118	5838	7184	6413	6145	551	603,5	930,4	1394	1782	6529	6226	489,7	6300	6317
TP+FN	7311	7311	7311	7311	7311	7311	7311	7311	7311	7311	7311	7311	7311	7311	7311	7311	7311	7311	7311	7311	7311	7311	7311	7311	7311	7311	7311	7311
TN+FP	69489	69489	69489	69489	69489	69489	69489	69489	69489	69489	69489	69489	69489	69489	69489	69489	69489	69489	69489	69489	69489	69489	69489	69489	69489	69489	69489	69489
	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800
Tasa de FP	0,226	0,033	0,078	0,182	0,135	0,061	0,109	0,106	0,037	0,09	0,089	0,149	0,081	0,018	0,14	0,013	0,078	0,087	0,222	0,159	0,031	0,023	0,015	0,065	0,086	0,025	0,072	0,082
Tasa de FN	0,685	0,941	0,874	0,751	0,803	0,903	0,832	0,84	0,928	0,861	0,857	0,779	0,86	0,974	0,799	0,983	0,877	0,84	0,075	0,083	0,127	0,191	0,244	0,893	0,852	0,067	0,862	0,864
Sensibilidad	0,315	0,059	0,126	0,249	0,197	0,097	0,168	0,16	0,072	0,139	0,143	0,221	0,14	0,026	0,201	0,017	0,123	0,16	0,925	0,917	0,873	0,809	0,756	0,107	0,148	0,933	0,138	0,136
Especificidad	0,774	0,967	0,922	0,818	0,865	0,939	0,891	0,894	0,963	0,91	0,911	0,851	0,919	0,982	0,86	0,987	0,922	0,913	0,778	0,841	0,969	0,977	0,985	0,935	0,914	0,975	0,928	0,918
PVP	0,128	0,158	0,146	0,126	0,133	0,144	0,14	0,138	0,17	0,14	0,145	0,135	0,155	0,134	0,131	0,122	0,141	0,162	0,305	0,377	0,748	0,791	0,845	0,148	0,153	0,798	0,167	0,149
PVN	0,915	0,907	0,909	0,912	0,911	0,908	0,911	0,91	0,908	0,909	0,91	0,912	0,91	0,906	0,911	0,905	0,909	0,912	0,99	0,99	0,986	0,98	0,975	0,909	0,911	0,993	0,911	0,91
F-Measure	0,182	0,085	0,135	0,168	0,159	0,116	0,153	0,148	0,101	0,14	0,144	0,168	0,147	0,044	0,159	0,03	0,131	0,161	0,459	0,535	0,806	0,8	0,798	0,124	0,151	0,86	0,151	0,142

**Sofa (550 a 849)**

	StaticFrameDifferenceBGS	MixtureOfGaussianV1BGS;	MixtureOfGaussianV2BGS	AdaptiveBackgroundLearning	GMG	DPAadaptiveMedianBGS	DPRimsonGMMBGS	DPZivkovicAGMMBGS	DPMeanBGS	DPWrenGABGS	DPPratiMediodBGS	DPEigenbackgroundBGS	DPTextureBGS	T2FGMM_UM	T2FGMM_UV	T2FMRF_UM	T2FMRF_UV	MultiLayerBGS	LBSimpleGaussian	LBFuzzyGaussian	LBMixtureOfGaussians	LBAadaptiveSOM	LBFuzzyAdaptiveSOM	VuMeter	IndependentMultimodalBGS	SIN_MultiCueBGS	SUBSENSEBGS()	LOBSTERBGS()	
Frames	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300
Tiempo total	2,847	8,261	10,09	7,954	17,36	15,28	48,76	19,56	32,23	24,81	86,6	34,43	581,8	61,4	66,08	91,09	154,8	112,4	18,25	26,14	38,41	63,73	63,63	11,07	20,43	44,4	472,3	335,3	
Tiempo medio	0,01	0,028	0,034	0,027	0,058	0,051	0,164	0,066	0,108	0,083	0,291	0,116	1,952	0,206	0,222	0,306	0,519	0,377	0,061	0,088	0,129	0,214	0,214	0,037	0,069	0,149	1,585	1,125	
Tiempo máximo	0,041	0,171	0,137	0,198	0,298	0,179	0,474	0,188	0,27	0,253	3,728	1,875	3,737	0,424	0,548	0,866	1,447	0,803	0,199	0,766	0,41	0,214	0,584	0,129	0,394	0,436	2,638	3,27	
TP	278,2	70,81	106,9	271,3	309,6	73,49	276,4	176,2	34,2	141,1	109,4	216	208,4	93,39	318,2	80,42	167,2	171,2	2128	2082	1534	1966	1818	169,1	59,25	2148	182,9	174,6	
FP	2877	579	1088	3129	2661	927,6	2126	2082	353,3	1502	1053	2237	2505	928,9	2728	816,9	1787	1374	803,6	601,6	172,2	194,1	107,7	1733	2041	546,3	2234	2083	
TN	68489	70787	70278	68237	68705	70438	69240	69284	71013	69864	70313	69129	68861	70437	68638	70549	69579	69992	70562	70764	71194	71172	71258	69633	69325	70820	69132	69283	
FN	5156	5363	5327	5163	5124	5361	5158	5258	5400	5293	5325	5218	5226	5341	5116	5354	5267	5263	3306	3352	3900	3468	3616	5265	5375	3286	5251	5259	
TP+FN	5434	5434	5434	5434	5434	5434	5434	5434	5434	5434	5434	5434	5434	5434	5434	5434	5434	5434	5434	5434	5434	5434	5434	5434	5434	5434	5434	5434	
TN+FP	71366	71366	71366	71366	71366	71366	71366	71366	71366	71366	71366	71366	71366	71366	71366	71366	71366	71366	71366	71366	71366	71366	71366	71366	71366	71366	71366	71366	
	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	
Tasa de FP	0,04	0,008	0,015	0,044	0,037	0,013	0,03	0,029	0,005	0,021	0,015	0,031	0,035	0,013	0,038	0,011	0,025	0,019	0,011	0,008	0,002	0,003	0,002	0,024	0,029	0,008	0,031	0,029	
Tasa de FN	0,949	0,987	0,98	0,95	0,943	0,986	0,949	0,968	0,994	0,974	0,98	0,96	0,962	0,983	0,941	0,985	0,969	0,969	0,608	0,617	0,718	0,638	0,665	0,969	0,989	0,605	0,966	0,968	
Sensibilidad	0,051	0,013	0,02	0,05	0,057	0,014	0,051	0,032	0,006	0,026	0,02	0,04	0,038	0,017	0,059	0,015	0,031	0,031	0,392	0,383	0,282	0,362	0,335	0,031	0,011	0,395	0,034	0,032	
Especificidad	0,96	0,992	0,985	0,956	0,963	0,987	0,97	0,971	0,995	0,979	0,985	0,969	0,965	0,987	0,962	0,989	0,975	0,981	0,989	0,992	0,998	0,997	0,998	0,976	0,971	0,992	0,969	0,971	
PVP	0,088	0,109	0,089	0,08	0,104	0,073	0,115	0,078	0,088	0,086	0,094	0,088	0,077	0,091	0,104	0,09	0,086	0,111	0,726	0,776	0,899	0,91	0,944	0,089	0,028	0,797	0,076	0,077	
PVN	0,93	0,93	0,93	0,93	0,931	0,929	0,931	0,929	0,929	0,93	0,93	0,93	0,929	0,93	0,931	0,929	0,93	0,93	0,955	0,955	0,948	0,954	0,952	0,93	0,928	0,956	0,929	0,929	
F-Measure	0,065	0,023	0,032	0,061	0,074	0,023	0,071	0,046	0,012	0,04	0,033	0,055	0,051	0,029	0,075	0,025	0,045	0,049	0,509	0,513	0,43	0,518	0,494	0,046	0,016	0,529	0,047	0,045	

Turbulence03 (1000 a 1299)	StaticFrameDifferenceBGS	MixtureOfGaussianV1BGS;	MixtureOfGaussianV2BGS	AdaptiveBackgroundLearning	GMG	DPAdaptiveMedianBGS	DPGrimsonGMMBGS	DPZivkovicAGMMBGS	DPMeanBGS	DPWrenGABGS	DPPratiMediodBGS	DPEigenbackgroundBGS	DPTextureBGS	T2FGMM_UM	T2FGMM_UV	T2FMRF_UM	T2FMRF_UV	MultilayerBGS	LBSimpleGaussian	LBFuzzyGaussian	LBmixtureOfGaussians	LBAaptiveSOM	LBFuzzyAdaptiveSOM	VuiMeter	IndependentMultimodalBGS	SJN_MultiCueBGS	SubSENSEBGS(!)	LOBSTERBGS()	
Frames	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	
Tiempo total	13,79	33,03	52,13	35,02	49,36	36,05	236,7	101,4	162,6	114,2	446,9	157,5	2742	129,9	147	249,6	270,3	338,5	102,1	81,07	138,5	269,9	289,8	51,39	101,5	68,04	2486	1257	
Tiempo medio	0,046	0,111	0,175	0,118	0,166	0,121	0,794	0,34	0,546	0,383	1,5	0,529	9,201	0,436	0,493	0,838	0,907	1,136	0,343	0,272	0,465	0,906	0,973	0,172	0,341	0,228	8,344	4,219	
Tiempo máximo	0,306	0,572	0,474	0,639	0,363	0,66	1,498	0,725	1,183	0,701	4,365	6,397	15,85	0,84	0,708	1,401	1,513	2,58	0,724	0,562	0,719	1,865	2,104	0,54	1,11	0,571	17,34	27,71	
TP	4036	604,9	1042	2809	793,4	870	1768	2384	624,6	1495	620,3	2094	1659	297,1	2465	254,2	1117	1037	6279	6223	4671	3055	2895	1016	766,9	3032	604,6	1292	
FP	76455	3021	12086	33878	11265	7950	21725	27873	5254	14923	3139	27560	16844	2301	36559	1906	11859	5077	1E+05	90793	29810	10738	7116	10822	7755	945,7	2499	10279	
TN	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	2E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05
FN	5054	8485	8048	6281	8297	8220	7322	6706	8465	7595	8470	6996	7431	8793	6625	8836	7973	8053	2811	2867	4419	6035	6195	8074	8323	6058	8485	7798	
TP+FN	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090	9090
TN+FP	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05
	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05	3E+05
Tasa de FP	0,224	0,009	0,035	0,099	0,033	0,023	0,064	0,082	0,015	0,044	0,009	0,081	0,049	0,007	0,107	0,006	0,035	0,015	0,302	0,266	0,087	0,032	0,021	0,032	0,023	0,003	0,007	0,03	
Tasa de FN	0,556	0,933	0,885	0,691	0,913	0,904	0,805	0,738	0,931	0,836	0,932	0,77	0,817	0,967	0,729	0,972	0,877	0,886	0,309	0,315	0,486	0,664	0,682	0,888	0,916	0,666	0,933	0,858	
Sensibilidad	0,444	0,067	0,115	0,309	0,087	0,096	0,195	0,262	0,069	0,164	0,068	0,23	0,183	0,033	0,271	0,028	0,123	0,114	0,691	0,685	0,514	0,336	0,318	0,112	0,084	0,334	0,067	0,142	
Especificidad	0,776	0,991	0,965	0,901	0,967	0,977	0,936	0,918	0,985	0,956	0,991	0,919	0,951	0,993	0,893	0,994	0,965	0,985	0,698	0,734	0,913	0,968	0,979	0,968	0,977	0,997	0,993	0,97	
PVP	0,05	0,167	0,079	0,077	0,066	0,099	0,075	0,079	0,106	0,091	0,165	0,071	0,09	0,114	0,063	0,118	0,086	0,17	0,058	0,064	0,135	0,221	0,289	0,086	0,09	0,762	0,195	0,112	
PVN	0,981	0,975	0,976	0,98	0,975	0,976	0,978	0,979	0,975	0,977	0,976	0,978	0,978	0,975	0,979	0,975	0,976	0,977	0,988	0,989	0,986	0,982	0,982	0,976	0,976	0,982	0,976	0,977	
F-Measure	0,09	0,095	0,094	0,123	0,075	0,097	0,109	0,121	0,083	0,117	0,097	0,108	0,12	0,051	0,102	0,045	0,101	0,136	0,106	0,117	0,214	0,267	0,303	0,097	0,087	0,464	0,099	0,125	

ZoomInZoomOut (500 a 799)	StaticFrameDifferenceBGS	MixtureOfGaussianV1BGS;	MixtureOfGaussianV2BGS	AdaptiveBackgroundLearning	GMG	DPA daptiveMedianBGS	DPGrimsonGMMBGS	DPZivkovicAGMMBGS	DPMeanBGS	DPWrenGABGS	DPPratilMediodBGS	DPEigenbackgroundBGS	DPTextureBGS	T2FGMM_UM	T2FGMM_LUV	T2FMRF_UM	T2FMRF_UV	MultiLayerBGS	LB SimpleGaussian	LB FuzzyGaussian	LB MixtureOfGaussians	LB AdaptiveSOM	LB FuzzyAdaptiveSOM	VuMeter	IndependentMultimodalBGS	SJN_MultiCueBGS	SuBSENSEBGS()	LOBSTERBGS()	
Frames	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300
Tiempo total	3,431	8,956	12,08	5,675	17,84	7,938	45	15,46	15,78	9,207	52,58	17,97	464	30,21	33,56	80,48	91,69	81,44	20,27	15,55	30,68	36,96	47,73	9,675	17,86	78,96	435,6	292,7	
Tiempo medio	0,012	0,03	0,041	0,019	0,06	0,027	0,151	0,052	0,053	0,031	0,176	0,06	1,557	0,101	0,113	0,27	0,308	0,273	0,068	0,052	0,103	0,124	0,16	0,032	0,06	0,265	1,462	0,982	
Tiempo máximo	0,192	0,128	0,143	0,135	0,414	0,185	1,76	0,295	0,225	0,093	2,659	1,041	2,74	0,174	0,339	0,831	1,096	0,882	0,302	0,244	0,551	0,683	0,354	0,11	0,379	1,002	2,803	1,69	
TP	653,5	23,58	64,53	252,2	189,8	78,88	181,8	360,8	18,15	185,3	83,34	498,2	75,31	11,91	286,7	9	84,34	68,76	824,8	818,2	623,4	555,8	555,8	49,21	109,6	763,4	41,43	46,04	
FP	51371	3052	6443	18375	17393	16719	15939	34550	1059	24167	10952	42008	16648	1702	24675	1678	11436	6964	53009	51234	18581	13505	13505	3567	4917	33558	4047	18822	
TN	24523	72842	69451	57519	58501	59175	59955	41344	74835	51727	64942	33886	59246	74192	51220	74216	64458	68930	22885	24660	57313	62390	62390	72327	70977	42337	71848	57072	
FN	252,5	882,4	841,5	653,8	716,2	827,1	724,2	545,2	887,9	720,7	822,7	407,8	830,7	894,1	619,4	897	821,7	837,2	81,2	87,8	282,6	350,2	350,2	856,8	796,4	142,6	864,6	860	
TP+FN	906	906	906	906	906	906	906	906	906	906	906	906	906	906	906	906	906	906	906	906	906	906	906	906	906	906	906	906	906
TN+FP	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894	75894
	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800
Tasa de FP	0,677	0,04	0,085	0,242	0,229	0,22	0,21	0,455	0,014	0,318	0,144	0,554	0,219	0,022	0,325	0,022	0,151	0,092	0,698	0,675	0,245	0,178	0,178	0,047	0,065	0,442	0,053	0,248	
Tasa de FN	0,279	0,974	0,929	0,722	0,791	0,913	0,799	0,602	0,98	0,795	0,908	0,45	0,917	0,987	0,684	0,99	0,907	0,924	0,09	0,097	0,312	0,387	0,387	0,946	0,879	0,157	0,954	0,949	
Sensibilidad	0,721	0,026	0,071	0,278	0,209	0,087	0,201	0,398	0,02	0,205	0,092	0,55	0,083	0,013	0,316	0,01	0,093	0,076	0,91	0,903	0,688	0,613	0,613	0,054	0,121	0,843	0,046	0,051	
Especificidad	0,323	0,96	0,915	0,758	0,771	0,78	0,79	0,545	0,986	0,682	0,856	0,446	0,781	0,978	0,675	0,978	0,849	0,908	0,302	0,325	0,755	0,822	0,822	0,953	0,935	0,558	0,947	0,752	
PVP	0,013	0,008	0,01	0,014	0,011	0,005	0,011	0,01	0,017	0,008	0,008	0,012	0,005	0,007	0,011	0,005	0,007	0,01	0,015	0,016	0,032	0,04	0,04	0,014	0,022	0,022	0,01	0,002	
PVN	0,99	0,988	0,988	0,989	0,988	0,986	0,988	0,987	0,988	0,986	0,987	0,988	0,986	0,988	0,988	0,988	0,987	0,988	0,996	0,996	0,995	0,994	0,994	0,988	0,989	0,997	0,988	0,985	
F-Measure	0,025	0,012	0,017	0,026	0,021	0,009	0,021	0,02	0,018	0,015	0,014	0,023	0,009	0,009	0,022	0,007	0,014	0,017	0,03	0,031	0,062	0,074	0,074	0,022	0,037	0,043	0,017	0,005	

diningRoom (700 a 999)	StaticFrameDifferenceBGS	MixtureOfGaussianV1BGS;	MixtureOfGaussianV2BGS	AdaptiveBackgroundLearning	GMG	DPAdaptiveMedianBGS	DPGrinsonGMMBGS	DPZivkovicAGMMBGS	DPMeanBGS	DPWrenGABGS	DPPratiMediodBGS	DPEigenbackgroundBGS	DPTtextureBGS	T2FGMM_UM	T2FGMM_UV	T2FMRF_UM	T2FMRF_UV	MultiLayerBGS	LBSimpleGaussian	LBfuzzyGaussian	LBmixtureOfGaussians	LBAaptiveSOM	LBfuzzyAdaptiveSOM	VuMeter	IndependentMultimodalBGS	SJN_MultiCueBGS	SubSENSEBGS()	LOBSTERBGS()
Frames	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300
Tiempo total	2,576	6,145	7,11	4,863	10,56	7,612	33,61	15,82	19,78	14,13	55,19	20,08	406,7	32,13	37,87	75,21	79,27	77,22	16,36	13,51	25,19	41,18	40,24	8,113	16,78	34,58	409,1	258,4
Tiempo medio	0,009	0,021	0,024	0,016	0,035	0,026	0,113	0,053	0,066	0,047	0,185	0,067	1,365	0,108	0,127	0,252	0,266	0,259	0,055	0,045	0,085	0,138	0,135	0,027	0,056	0,116	1,373	0,867
Tiempo máximo	0,047	0,076	0,128	0,034	0,069	0,109	0,27	0,243	0,157	0,125	1,531	1,123	2,168	0,303	0,255	0,529	0,463	0,579	0,349	0,147	0,255	0,466	0,351	0,123	0,278	0,253	2,524	2,615
TP	1591	211,9	364,7	1421	662,6	231,2	1156	1356	22,51	980,8	630,7	1413	1372	196,1	1569	103,3	1081	683,4	5181	5152	4482	4675	3534	540,3	728,3	3237	1314	1361
FP	6149	890,9	1872	5588	2998	860,4	4863	5300	113,7	3752	2377	5346	5349	998,9	6934	695,8	4381	2716	504,8	391,8	387,1	161	115,3	2921	2824	192,4	4932	5023
TN	64625	69883	68902	65186	67776	69914	65911	65474	70660	67022	68397	65428	65425	69775	63840	70078	66393	68058	70269	70382	70387	70613	70659	67853	67950	70582	65842	65751
FN	4435	5814	5661	4605	5363	5795	4871	4670	6003	5045	5395	4613	4654	5830	4457	5923	4945	5343	844,6	874	1544	1351	2492	5486	5298	2789	4712	4665
TP+FN	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026	6026
TN+FP	70774	70774	70774	70774	70774	70774	70774	70774	70774	70774	70774	70774	70774	70774	70774	70774	70774	70774	70774	70774	70774	70774	70774	70774	70774	70774	70774	70774
	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800
Tasa de FP	0,087	0,013	0,026	0,079	0,042	0,012	0,069	0,075	0,002	0,053	0,034	0,076	0,076	0,014	0,098	0,01	0,062	0,038	0,007	0,006	0,005	0,002	0,002	0,041	0,04	0,003	0,07	0,071
Tasa de FN	0,736	0,965	0,939	0,764	0,89	0,962	0,808	0,775	0,996	0,837	0,895	0,766	0,772	0,967	0,74	0,983	0,821	0,887	0,14	0,145	0,256	0,224	0,413	0,91	0,879	0,463	0,782	0,774
Sensibilidad	0,264	0,035	0,061	0,236	0,11	0,038	0,192	0,225	0,004	0,163	0,105	0,234	0,228	0,033	0,26	0,017	0,179	0,113	0,86	0,855	0,744	0,776	0,587	0,09	0,121	0,537	0,218	0,226
Especificidad	0,913	0,987	0,974	0,921	0,958	0,988	0,931	0,925	0,998	0,947	0,966	0,924	0,924	0,986	0,902	0,99	0,938	0,962	0,993	0,994	0,995	0,998	0,998	0,959	0,96	0,997	0,93	0,929
PVP	0,206	0,192	0,163	0,203	0,181	0,212	0,192	0,204	0,165	0,207	0,21	0,209	0,204	0,164	0,185	0,129	0,198	0,201	0,911	0,929	0,92	0,967	0,968	0,156	0,205	0,944	0,21	0,213
PVN	0,936	0,923	0,924	0,934	0,927	0,923	0,931	0,933	0,922	0,93	0,927	0,934	0,934	0,923	0,935	0,922	0,931	0,927	0,988	0,988	0,979	0,981	0,966	0,925	0,928	0,962	0,933	0,934
F-Measure	0,231	0,059	0,088	0,218	0,137	0,065	0,192	0,214	0,007	0,182	0,14	0,221	0,215	0,054	0,216	0,03	0,188	0,145	0,885	0,891	0,823	0,861	0,731	0,114	0,152	0,685	0,214	0,219

Traffic ( 1250 a 1550)	StaticFrameDifferenceBGS	MixtureOfGaussianV1BGS;	MixtureOfGaussianV2BGS	AdaptiveBackgroundLearning	GMG	DPA daptiveMedianBGS	DPGrimsonGMMBGS	DPZivkovicAGMMBGS	DPMeanBGS	DPWrenGABGS	DPPratilMediodBGS	DPEigenbackgroun dBGS	DPTextureBGS	TZFGMM_UM	TZFGMM_LUV	TZFMRF_UM	TZFMRF_UV	MultiLayerBGS	LB SimpleGaussian	LB FuzzyGaussian	LB MixtureOfGaussians	LB AdaptiveSOM	LB FuzzyAdaptiveSOM	VuMeter	IndependentMultimodalBGS	SJN_MultiCueBGS	SuBSENSEBGS()	LOBSTERBGS()	
Frames	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300
Tiempo total	5,496	12,03	20,78	9,42	10,57	3,474	21,08	7,393	10,52	8,148	27,08	10,23	189,2	10,01	15,44	34,85	38,36	65,92	7,663	11,78	27,44	32,94	32,6	7,164	26,09	48,11	368,6	219,2	
Tiempo medio	0,018	0,04	0,07	0,032	0,035	0,012	0,071	0,025	0,035	0,027	0,091	0,034	0,635	0,034	0,052	0,117	0,129	0,221	0,026	0,04	0,092	0,111	0,109	0,024	0,088	0,161	1,237	0,736	
Tiempo máximo	0,234	0,346	0,345	0,114	0,704	0,031	0,214	0,091	0,094	0,099	0,64	0,465	1,09	0,111	0,23	0,28	0,271	0,537	0,134	0,17	0,353	0,412	0,271	0,13	0,279	0,912	2,405	1,481	
TP	4348	1128	1980	3818	3765	1820	2743	3307	1465	2799	2206	3416	2986	482,3	3270	347,3	1916	2344	6347	6287	6189	4734	4977	2187	2217	3445	1860	2510	
FP	34619	3518	7949	29939	26289	9950	16304	22809	8763	16586	8274	21472	21138	1278	24011	1250	11308	8129	34476	32988	17203	7289	3652	7379	6861	5584	4878	11765	
TN	35425	66526	62095	40105	43755	60094	53741	47235	61281	53458	61770	48572	48906	68766	46033	68794	58736	61915	35569	37056	52841	62755	66392	62665	63183	64460	65166	58279	
FN	2408	5628	4776	2938	2991	4936	4013	3449	5291	3957	4550	3340	3770	6274	3486	6409	4840	4412	409,1	468,7	567,1	2022	1779	4569	4539	3311	4896	4246	
TP+FN	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756	6756
TN+FP	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044	70044
	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800
Tasa de FP	0,494	0,05	0,113	0,427	0,375	0,142	0,233	0,326	0,125	0,237	0,118	0,307	0,302	0,018	0,343	0,018	0,161	0,116	0,492	0,471	0,246	0,104	0,052	0,105	0,098	0,08	0,07	0,168	
Tasa de FN	0,356	0,833	0,707	0,435	0,443	0,731	0,594	0,51	0,783	0,586	0,674	0,494	0,558	0,929	0,516	0,949	0,716	0,653	0,061	0,069	0,084	0,299	0,263	0,676	0,672	0,49	0,725	0,629	
Sensibilidad	0,644	0,167	0,293	0,565	0,557	0,269	0,406	0,49	0,217	0,414	0,326	0,506	0,442	0,071	0,484	0,051	0,284	0,347	0,939	0,931	0,916	0,701	0,737	0,324	0,328	0,51	0,275	0,371	
Especificidad	0,506	0,95	0,887	0,573	0,625	0,858	0,767	0,674	0,875	0,763	0,882	0,693	0,698	0,982	0,657	0,982	0,839	0,884	0,508	0,529	0,754	0,896	0,948	0,895	0,902	0,92	0,93	0,832	
PVP	0,112	0,243	0,199	0,113	0,125	0,155	0,144	0,127	0,143	0,144	0,21	0,137	0,124	0,274	0,12	0,217	0,145	0,224	0,155	0,16	0,265	0,394	0,577	0,229	0,244	0,382	0,276	0,176	
PVN	0,936	0,922	0,929	0,932	0,936	0,924	0,931	0,932	0,921	0,931	0,931	0,936	0,928	0,916	0,93	0,915	0,924	0,933	0,989	0,988	0,989	0,969	0,974	0,932	0,933	0,951	0,93	0,932	
F-Measure	0,19	0,198	0,237	0,188	0,205	0,196	0,213	0,201	0,172	0,214	0,256	0,216	0,193	0,113	0,192	0,083	0,192	0,272	0,267	0,273	0,411	0,504	0,647	0,268	0,28	0,436	0,276	0,239	

Fountain02 (1050 a 1349)	StaticFrameDifferenceBGS	MixtureOfGaussianV1BGS;	MixtureOfGaussianV2BGS	AdaptiveBackgroundLearning	GMG	DPAdaptiveMedianBGS	DPGrimsonGMMBGS	DPZivkovicAGMMBGS	DPMeanBGS	DPWrenGABGS	DPPratiMediodBGS	DPEigenbackgroundBGS	DPTextureBGS	T2FGMM_UM	T2FGMM_UV	T2FMRF_UM	T2FMRF_UV	MultilayerBGS	LBSimpleGaussian	LBfuzzyGaussian	LBmixtureOfGaussians	LBAaptiveSOM	LBfuzzyAdaptiveSOM	VuMeter	IndependentMultimodalBGS	SJN_MultiCueBGS	SubSENSEBGS()	LOBSTERBGS()	
Frames	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300
Tiempo total	8,769	7,551	8,628	6,243	14,01	9,549	36,53	10,15	18,22	16,21	57,03	16,7	565	11,95	15,65	0	0	68,21	7,657	12,07	16,61	25,98	26,73	5,321	22,6	26,08	460,4	333,7	
Tiempo medio	0,029	0,025	0,029	0,021	0,047	0,032	0,123	0,034	0,061	0,054	0,191	0,056	1,896	0,04	0,053	0	0	0,229	0,026	0,04	0,056	0,087	0,09	0,018	0,076	0,088	1,545	1,12	
Tiempo máximo	0,118	0,107	0,095	0,054	0,103	0,073	0,198	0,135	0,191	0,139	1,119	1,184	3,877	0,113	0,045	0	0	0,295	0,046	0,095	0,105	0,22	0,16	0,032	0,28	0,198	2,626	1,95	
TP	188,5	36,34	47,69	144,9	142,9	27,48	74,1	72,64	31,3	55,37	51,7	139	25,09	28,51	120,3	15,68	86,08	139,9	364,6	375,4	326,8	426,1	413	140,1	140,6	335,8	139,7	64,21	
FP	28167	474,8	3189	8832	9021	216,3	5312	5786	669,7	2202	708,5	1658	955,5	587,9	13106	1059	1753	942,5	33263	28426	7220	4700	3767	2147	5722	136,8	293,7	341	
TN	95783	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	90687	95524	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	
FN	277,5	429,7	418,3	321,1	323,1	438,5	391,9	393,4	434,7	410,6	414,3	327	440,9	437,5	345,8	450,3	379,9	326,1	101,4	90,58	139,2	39,95	53,05	325,9	325,4	130,2	326,3	401,8	
TP+FN	466	466	466	466	466	466	466	466	466	466	466	466	466	466	466	466	466	466	466	466	466	466	466	466	466	466	466	466	466
TN+FP	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05
	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05
Tasa de FP	0,227	0,004	0,026	0,071	0,073	0,002	0,043	0,047	0,005	0,018	0,006	0,013	0,008	0,005	0,106	0,009	0,014	0,008	0,268	0,229	0,058	0,038	0,03	0,017	0,046	0,001	0,002	0,003	
Tasa de FN	0,596	0,922	0,898	0,689	0,693	0,941	0,841	0,844	0,933	0,881	0,889	0,702	0,946	0,939	0,742	0,966	0,815	0,7	0,218	0,194	0,299	0,086	0,114	0,699	0,698	0,279	0,7	0,862	
Sensibilidad	0,404	0,078	0,102	0,311	0,307	0,059	0,159	0,156	0,067	0,119	0,111	0,298	0,054	0,061	0,258	0,034	0,185	0,3	0,782	0,806	0,701	0,914	0,886	0,301	0,302	0,721	0,3	0,138	
Especificidad	0,773	0,996	0,974	0,929	0,927	0,998	0,957	0,953	0,995	0,982	0,994	0,987	0,992	0,995	0,894	0,991	0,986	0,992	0,732	0,771	0,942	0,962	0,97	0,983	0,954	0,999	0,998	0,997	
PVP	0,007	0,071	0,015	0,016	0,016	0,113	0,014	0,012	0,045	0,025	0,068	0,077	0,026	0,046	0,009	0,015	0,047	0,129	0,011	0,013	0,043	0,083	0,099	0,061	0,024	0,71	0,322	0,158	
PVN	0,997	0,997	0,997	0,997	0,997	0,996	0,997	0,997	0,996	0,997	0,997	0,997	0,996	0,996	0,997	0,996	0,997	0,997	0,999	0,999	0,999	1	1	0,997	0,997	0,999	0,997	0,997	
F-Measure	0,013	0,074	0,026	0,031	0,03	0,077	0,025	0,023	0,054	0,041	0,084	0,123	0,035	0,053	0,018	0,02	0,075	0,181	0,021	0,026	0,082	0,152	0,178	0,102	0,044	0,716	0,311	0,147	

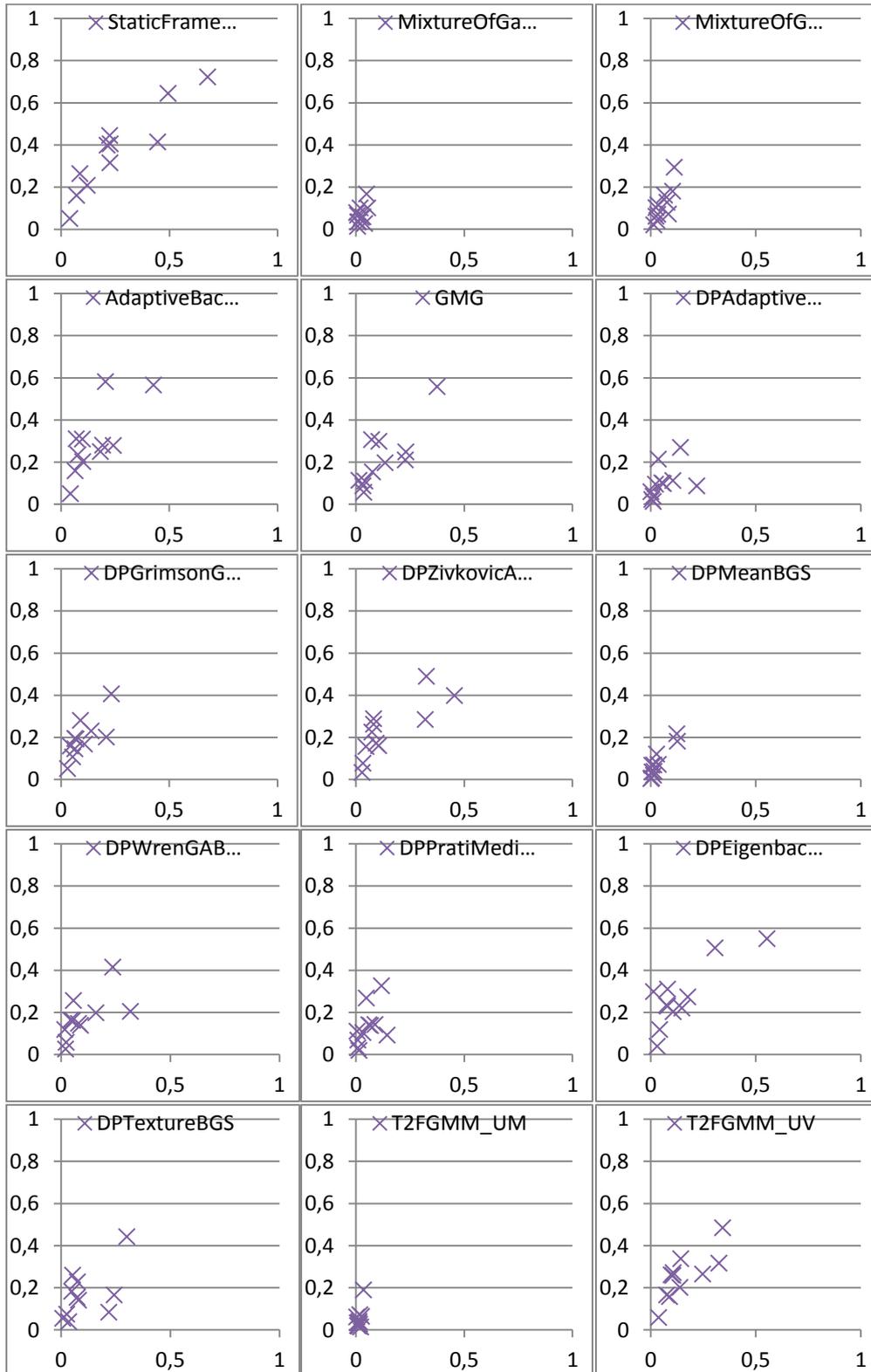
Skating (1750 a 2049)	StaticFrameDifferenceBGS	MixtureOfGaussianV1BGS;	MixtureOfGaussianV2BGS	AdaptiveBackgroundLearning	GMG	DPAdaptiveMedianBGS	DPGrimsonGMMBGS	DPZivkovicAGMMBGS	DPMeanBGS	DPWrenGABGS	DPPratiMediodBGS	DPEigenbackgroundBGS	DPTextureBGS	T2FGMM_UM	T2FGMM_UV	T2FMRF_UM	T2FMRF_UV	MultilayerBGS	LBSimpleGaussian	LBfuzzyGaussian	LBmixtureOfGaussians	LBAdaptiveSOM	LBfuzzyAdaptiveSOM	VuMeter	IndependentMultimodalBGS	SJN_MultiCueBGS	SuBSENSEBGS()	LOBSTERBGS()
Frames	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300
Tiempo total	8,826	9,841	13,83	9,788	21,59	13,87	63,29	21,7	24,73	20,22	64,65	18,44	459,5	42,98	66,53	70,35	80,77	90,4	12,79	17,92	30,18	50,36	55,36	8,772	29,64	30,37	612,4	337,5
Tiempo medio	0,03	0,033	0,046	0,033	0,072	0,047	0,212	0,073	0,083	0,068	0,217	0,062	1,542	0,144	0,223	0,236	0,271	0,303	0,043	0,06	0,101	0,169	0,186	0,029	0,099	0,102	2,055	1,132
Tiempo máximo	0,125	0,137	0,135	0,118	0,238	0,183	0,466	0,251	0,12	0,19	1,332	1,015	2,19	0,344	0,532	0,429	0,671	0,753	0,122	0,13	0,244	0,388	0,555	0,103	0,396	0,358	10,5	7,468
TP	6272	1589	2544	9151	4709	3374	4418	4543	1889	4033	4220	4891	4076	2974	5303	2786	4438	3621	14768	14736	13943	13809	13400	2279	2700	13228	3964	3804
FP	38174	3556	11690	36796	19068	6462	16052	14661	5137	10127	8603	14342	9404	6435	25747	5999	14007	9230	30451	27915	7938	1847	1024	7747	6975	3658	8387	7282
TN	1E+05	2E+05	2E+05	1E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	1E+05	2E+05	2E+05	2E+05	2E+06	2E+05	2E+05	2E+05	2E+05	2E+05
FN	9470	14154	13199	6591	11034	12369	11325	11200	13854	11710	11523	10852	11667	12769	10440	12956	11305	12122	974,8	1006	1799	1934	2342	13464	13043	2514	11779	11939
TP+FN	15743	15743	15743	15743	15743	15743	15743	15743	15743	15743	15743	15743	15743	15743	15743	15743	15743	15743	15743	15743	15743	15743	15743	15743	15743	15743	15743	15743
TN+FP	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+06	2E+05	2E+05	2E+05	2E+05	2E+05
	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+05	2E+06	2E+05	2E+05	2E+05	2E+05	2E+05
Tasa de FP	0,214	0,02	0,065	0,206	0,107	0,036	0,09	0,082	0,029	0,057	0,048	0,08	0,053	0,036	0,144	0,034	0,078	0,052	0,17	0,156	0,044	0,01	6E-04	0,043	0,039	0,02	0,047	0,041
Tasa de FN	0,602	0,899	0,838	0,419	0,701	0,786	0,719	0,711	0,88	0,744	0,732	0,689	0,741	0,811	0,663	0,823	0,718	0,77	0,062	0,064	0,114	0,123	0,149	0,855	0,828	0,16	0,748	0,758
Sensibilidad	0,398	0,101	0,162	0,581	0,299	0,214	0,281	0,289	0,12	0,256	0,268	0,311	0,259	0,189	0,337	0,177	0,282	0,23	0,938	0,936	0,886	0,877	0,851	0,145	0,172	0,84	0,252	0,242
Especificidad	0,786	0,98	0,935	0,794	0,893	0,964	0,91	0,918	0,971	0,943	0,952	0,92	0,947	0,964	0,856	0,966	0,922	0,948	0,83	0,844	0,956	0,99	0,999	0,957	0,961	0,98	0,953	0,959
PVP	0,141	0,309	0,179	0,199	0,198	0,343	0,216	0,237	0,269	0,285	0,329	0,254	0,302	0,316	0,171	0,317	0,241	0,282	0,327	0,346	0,637	0,882	0,929	0,227	0,279	0,783	0,321	0,343
PVN	0,937	0,925	0,927	0,956	0,935	0,933	0,935	0,936	0,926	0,935	0,937	0,938	0,936	0,931	0,936	0,93	0,936	0,933	0,993	0,993	0,99	0,989	0,999	0,927	0,929	0,986	0,935	0,935
F-Measure	0,208	0,152	0,17	0,297	0,238	0,264	0,244	0,26	0,166	0,27	0,295	0,28	0,279	0,236	0,227	0,227	0,26	0,253	0,484	0,505	0,741	0,88	0,888	0,177	0,212	0,811	0,282	0,284

Turnpike 0.5 fps (800 a 1099)	StaticFrameDifferenceBGS	MixtureOfGaussianV1BGS;	MixtureOfGaussianV2BGS	AdaptiveBackgroundLearning	GMG	DPA daptiveMedianBGS	DPGrimsonGMMBGS	DPZivkovicAGMMBGS	DPMeanBGS	DPWrenGABGS	DPPratilMediodBGS	DPEigenbackgroudBGS	DPTextureBGS	T2FGMM_UM	T2FGMM_LUV	T2FMRF_UM	T2FMRF_LUV	MultiLayerBGS	LB SimpleGaussian	LB FuzzyGaussian	LB MixtureOfGaussians	LB AdaptiveSOM	LB FuzzyAdaptiveSOM	VuMeter	Independent MultimodalBGS	SJN_MultiCueBGS	SuBSENSEBGS()	LOBSTERBGS()	
Frames	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300
Tiempo total	2,786	5,904	9,002	4,556	12,17	8,497	42,59	16,73	19,5	12,84	47,98	17,88	403,2	34,14	51,5	55,41	67,42	61	7,742	11,29	18,86	29,39	24,05	4,467	12,71	25,63	243,2	180,5	
Tiempo medio	0,009	0,02	0,03	0,015	0,041	0,029	0,143	0,056	0,065	0,043	0,161	0,06	1,353	0,115	0,173	0,186	0,226	0,205	0,026	0,038	0,063	0,099	0,081	0,015	0,043	0,086	0,816	0,606	
Tiempo máximo	0,026	0,057	0,065	0,071	0,099	0,125	0,206	0,134	0,139	0,176	1,295	0,582	1,343	0,147	0,217	0,356	0,334	0,34	0,079	0,136	0,113	0,257	0,124	0,021	0,106	0,167	1,132	1,067	
TP	5322	1293	2314	3602	3196	1446	2954	3661	2326	2557	1771	3518	2122	505,5	3407	430	2109	2459	8134	8116	7435	4557	4244	1762	1690	6304	1185	1776	
FP	28556	3512	6724	12326	14821	6749	8931	20517	8151	10270	4338	11355	15604	933,4	15853	892,1	6515	5996	22787	22443	5019	501,6	505	5193	3882	1250	3425	8154	
TN	35384	60428	57216	51614	49119	57191	55009	43423	55789	53671	59602	52585	48336	63007	48087	63048	57425	57944	41153	41497	58921	63438	63435	58747	60058	62690	60515	55786	
FN	7538	11567	10546	9258	9664	11414	9906	9199	10534	10303	11089	9342	10738	12354	9453	12430	10751	10401	4726	4744	5425	8303	8616	11098	11170	6556	11675	11084	
TP+FN	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860	12860
TN+FP	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940	63940
	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800	76800
Tasa de FP	0,447	0,055	0,105	0,193	0,232	0,106	0,14	0,321	0,127	0,161	0,068	0,178	0,244	0,015	0,248	0,014	0,102	0,094	0,356	0,351	0,078	0,008	0,008	0,081	0,061	0,02	0,054	0,128	
Tasa de FN	0,586	0,899	0,82	0,72	0,751	0,888	0,77	0,715	0,819	0,801	0,862	0,726	0,835	0,961	0,735	0,967	0,836	0,809	0,367	0,369	0,422	0,646	0,67	0,863	0,869	0,51	0,908	0,862	
Sensibilidad	0,414	0,101	0,18	0,28	0,249	0,112	0,23	0,285	0,181	0,199	0,138	0,274	0,165	0,039	0,265	0,033	0,164	0,191	0,633	0,631	0,578	0,354	0,33	0,137	0,131	0,49	0,092	0,138	
Especificidad	0,553	0,945	0,895	0,807	0,768	0,894	0,86	0,679	0,873	0,839	0,932	0,822	0,756	0,985	0,752	0,986	0,898	0,906	0,644	0,649	0,922	0,992	0,992	0,919	0,939	0,98	0,946	0,872	
PVP	0,157	0,269	0,256	0,226	0,177	0,176	0,249	0,151	0,222	0,199	0,29	0,237	0,12	0,351	0,177	0,325	0,245	0,291	0,263	0,266	0,597	0,901	0,894	0,253	0,303	0,835	0,257	0,179	
PVN	0,824	0,839	0,844	0,848	0,836	0,834	0,847	0,825	0,841	0,839	0,843	0,849	0,818	0,836	0,836	0,835	0,842	0,848	0,897	0,897	0,916	0,884	0,88	0,841	0,843	0,905	0,838	0,834	
F-Measure	0,228	0,146	0,211	0,25	0,207	0,137	0,239	0,198	0,199	0,199	0,187	0,254	0,139	0,071	0,212	0,061	0,196	0,231	0,372	0,374	0,587	0,509	0,482	0,178	0,183	0,618	0,136	0,156	

Street corner  
at night  
(2700 a 2999)

	StaticFrameDifferenceBGS	MixtureOfGaussianV1BGS;	MixtureOfGaussianV2BGS	AdaptiveBackgroundLearning	GMG	DPAdaptiveMedianBGS	DPGrimsonGMIMBGS	DPZivkovicAGMMBGS	DPMeanBGS	DPWrenGABGS	DPPratiMediodBGS	DPEigenbackgroundBGS	DPTtextureBGS	T2FGMM_UM	T2FGMM_UV	T2FMRF_UM	T2FMRF_UV	MultiLayerBGS	LBSimpleGaussian	LBfuzzyGaussian	LBmixtureOfGaussians	LBAdaptiveSOM	LBfuzzyAdaptiveSOM	VuMeter	IndependentMultimodalBGS	SJN_MultiCueBGS	SUBSENSEBGS()	LOBSTERBGS()	
Frames	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	300	
Tiempo total	3,809	8,261	15,24	7,577	21,36	21,29	70,5	24,46	38,43	28,37	62,27	21,53	457,2	44,54	67,44	106	114,4	100,3	17,24	20,52	30,37	55,1	52,51	8,524	15,25	20,83	380,3	228,7	
Tiempo medio	0,013	0,028	0,051	0,025	0,072	0,071	0,237	0,082	0,129	0,095	0,209	0,072	1,534	0,149	0,226	0,356	0,384	0,336	0,058	0,069	0,102	0,185	0,176	0,029	0,051	0,07	1,276	0,767	
Tiempo máximo	0,025	0,242	0,33	0,198	0,248	0,384	1,356	0,166	0,423	0,359	1,839	2,208	24,5	0,403	0,686	0,85	0,976	1,72	0,253	0,317	0,663	0,787	0,77	0,134	0,545	0,454	4,074	2,166	
TP	567,2	118,3	143,1	564,1	402,4	78,85	381	270,4	124,2	206,2	111,8	422,6	264,4	68,26	555,3	49,65	286,9	315	2271	2204	1560	1570	1242	319,5	243,6	1737	181,5	161,9	
FP	10283	1360	4267	9333	2060	765,2	7581	4734	1046	3287	1396	6132	3711	1244	12840	873,9	6111	3793	11900	9834	3930	3791	2252	7274	3459	620,6	2458	1903	
TN	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05
FN	2978	3427	3402	2981	3143	3466	3164	3275	3421	3339	3433	3122	3281	3477	2990	3495	3258	3230	1274	1341	1985	1975	2303	3226	3301	1808	3364	3383	
TP+FN	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545	3545
TN+FP	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05
	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05	1E+05
Tasa de FP	0,072	0,01	0,03	0,066	0,014	0,005	0,053	0,033	0,007	0,023	0,01	0,043	0,026	0,009	0,09	0,006	0,043	0,027	0,084	0,069	0,028	0,027	0,016	0,051	0,024	0,004	0,017	0,013	
Tasa de FN	0,84	0,967	0,96	0,841	0,887	0,978	0,893	0,924	0,965	0,942	0,968	0,881	0,925	0,981	0,843	0,986	0,919	0,911	0,359	0,378	0,56	0,557	0,65	0,91	0,931	0,51	0,949	0,954	
Sensibilidad	0,16	0,033	0,04	0,159	0,113	0,022	0,107	0,076	0,035	0,058	0,032	0,119	0,075	0,019	0,157	0,014	0,081	0,089	0,641	0,622	0,44	0,443	0,35	0,09	0,069	0,49	0,051	0,046	
Especificidad	0,928	0,99	0,97	0,934	0,986	0,995	0,947	0,967	0,993	0,977	0,99	0,957	0,974	0,991	0,91	0,994	0,957	0,973	0,916	0,931	0,972	0,973	0,984	0,949	0,976	0,996	0,983	0,987	
PVP	0,052	0,08	0,032	0,057	0,163	0,093	0,048	0,054	0,106	0,059	0,074	0,064	0,066	0,052	0,041	0,054	0,045	0,077	0,16	0,183	0,284	0,293	0,356	0,042	0,066	0,737	0,069	0,078	
PVN	0,978	0,976	0,976	0,978	0,978	0,976	0,977	0,977	0,976	0,977	0,976	0,978	0,977	0,976	0,977	0,976	0,977	0,977	0,99	0,99	0,986	0,986	0,984	0,977	0,977	0,987	0,977	0,976	
F-Measure	0,079	0,047	0,036	0,084	0,134	0,036	0,066	0,063	0,053	0,059	0,044	0,084	0,07	0,028	0,066	0,022	0,058	0,082	0,256	0,283	0,345	0,353	0,353	0,057	0,067	0,589	0,059	0,058	

A partir de los resultados anteriores, el trazado de los puntos obtenidos en el espacio ROC para cada uno de los métodos seleccionados queda de la siguiente forma:



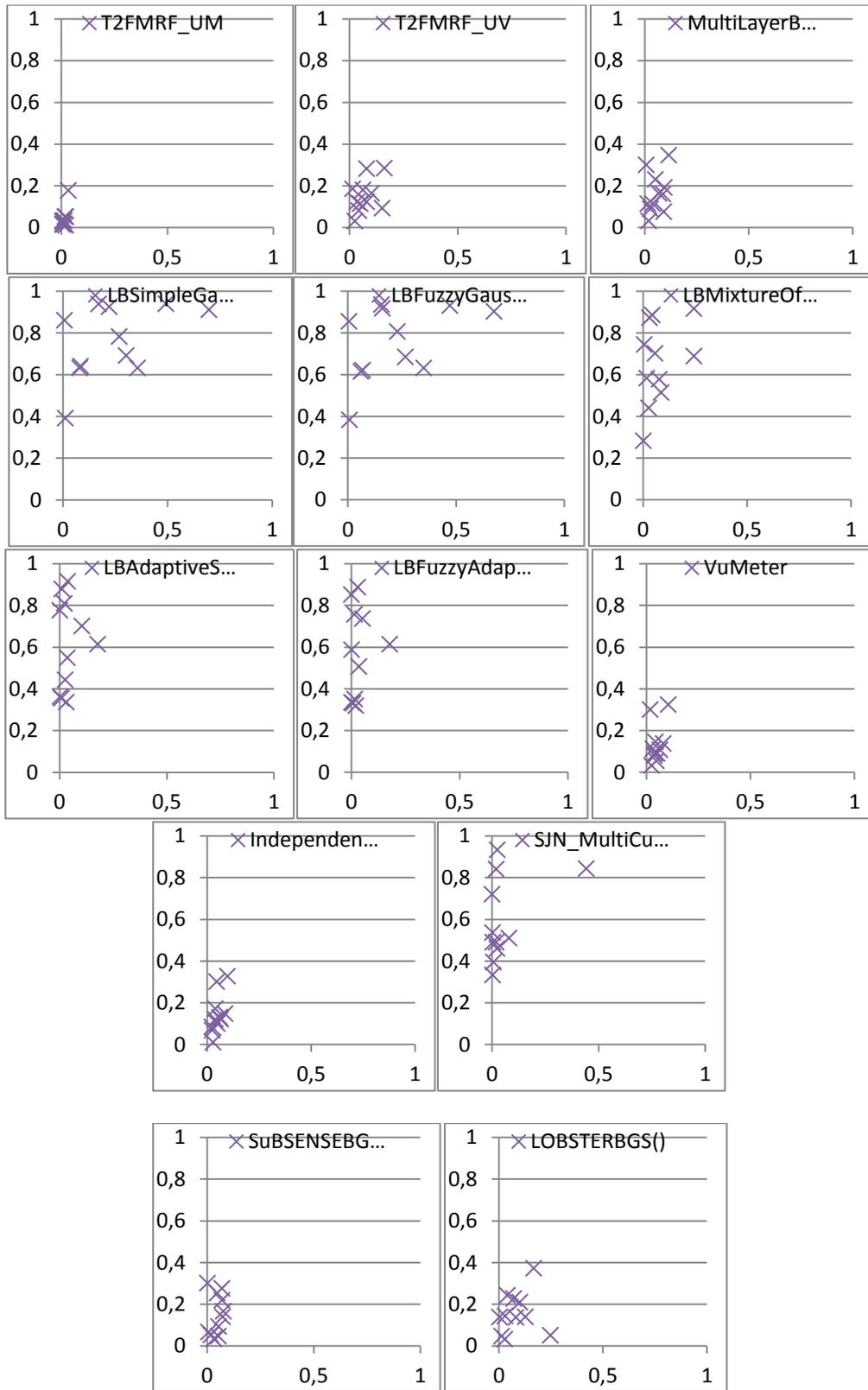


Figura 22: Puntos en el espacio ROC para todos los métodos analizados

En la tabla 9 se muestra el análisis realizado para el método de seguimiento de Lucas y Kanade, se indica el video al que pertenece la secuencia, así como una breve descripción de la misma; en caso de que a lo largo de los frames se pierdan puntos injustificadamente (cuándo la persona/coche permanece en escena) se muestra coloreado de rojo. En la columna de observaciones se explica en qué momento se pierden y a que es debido:

**Tabla 9:** Análisis del método de Lucas y Kanade

Video	Secuencia (frame de inicio)	Puntos														Observaciones	
		Ini	10	20	30	40	50	60	70	80	90	100	110	120	130		140
Highway	5 coches en la carretera, trayectoria recta (1000)	10	10	10	8	8	8	8	6	6	6	6	6				Quando se pierde el pierde el primer coche se van dos puntos y con el segundo otros dos.
Highway	Gran densidad de tráfico, hay adelantamientos (1620)	9	9	9	9	5	4	4	4	4							Se pierden cuando salen los coches de escena. Ningún problema
Traffic (jitter)	Coche pasando (1058)	3	2	2													Uno de los puntos se pierde en los primeros movimientos. El resto se mantienen aunque al ser la camara jitter la trayectoria es en zig-zag (el coche va recto)
People in shade	Hombre pasa por delante, parandose (700)	3	3	3	3	3	3	3	3	2							Quando el hombre camina de perfil un punto cambia de lugar
People in shade	Hombre pasa por delante, parandose (850)	5	5	5	5	5	5	5	5	4	4	4					Al realizar el giro (de frente a perfil) un punto cambia de lugar)
People in shade	Pareja pasa por delante (1070)	9	8	7	7	7											Un punto con el paso del timpo va pasa de una de las personas a la otra y otro cambia del maletín a la pierna.
Fountain02	Pasa un coche (1225)	1	0	0	0												Quando pasa el coche por delante de alguno de los puntos pertenecientes a la fuente estos comienzan a seguirlo
Sofa	Hombre entra en escena y se sienta en el sofá (1140)	7	5	4	4	4	3	3	3	3	3						Los puntos se pierden al girar el individuo y agacharse
Sofa	Hombre entra, se agacha y cambia una caja de lugar (1540)	7	7	6	6	6	6	6	6	6	6	6	6	6	6	6	Se pierde un punto al entrar en escena y realizar el giro (de espaldas a de costado). Cuando el individuo se agacha y levanta el seguimiento es correcto.
Sofa	Hombre entra, recoge una caja y una bolsa y se marcha (1900)	5	5	5	5	5	5	2	2	2	2						Se pierden tres puntos cuando el hombre se agacha a recoger la caja (coincide con que gira el cuerpo)
Sofa	Un hombre recoge cosas del sofa mientras otro se sienta en el y permanece sentado(se cruzan) (2450)	Sentado	2	2	0	0	0	0	0								Quando el otro hombre pasa por delante del que permanece sentado los puntos de este le siguen también a el
		De pie	4	4	4	4	4	4	4								
Dinning Room	Mujer gira 180º y deja un objeto cerca de la camara (1000)	4	3	3	3	3	3	3	3	3							Al comienzo del giro un punto perteneciente a la mano cambia de lugar
Dinning Room	Mujer va de espaldas a por un objeto, lo recoge, gira 180º y lo deposita cerca de la camara (2360)	4	4	4	4	4	4	4	4	4	4	4					En esta ocasión no hay problema al realizar el giro. Funcionamiento correcto.
Skaitting	Pasa un grupo de personas esquiando (1350)	5	5	5	5	5	5	0	0	0	0						Se pierde el seguimiento al pasar las personas por detrás de un árbol

Skaitting	Pasa otro grupo de personas (en este caso no pasan por detrás del árbol) (2400)	6 6 6 6 6 4 4 4		Funciona correctamente. Cuando las primeras personas desaparecen de la imagen lo hacen sus puntos de seguimiento.
Skaitting	Pasa una persona cerca de la camara con capucha (delante del árbol) (2660)	2 2 2 2 2 2 2		Funciona correctamente.
Skaitting	Pasa un grupo de personas lejanas y una persona cercana (se cruzan) (3135)	Lejos	5 3 3 3	Cuando se produce el cruce gran cantidad de puntos se muestran incorrectamente, unicamente conservandose 2 del grupo grande de personas.
		Cerca	2 0 0 0	
Skaitting	Pasa un esquiador lejano (por detrás del árbol) (3835)	4 3 3 3 0 0 0 0		Un punto al comienzo de movimiento baja del torso a las piernas y al pasar por detrás del árbol el resto también se pierden.
Street corner at night	Un coche atraviesa la carretera de arriba a abajo (1145)	1 1 1		Funcionamiento correcto.
Street corner at night	Un coche atraviesa el cruce entrando por la izq y saliendo por arriba (1235)	1 1 1 0 0		Cuando el coche gira se pierde el punto.
Street corner at night	Un coche atraviesa de cruce de arriba abajo y otro en sentido opuesto (2775)	7 7 7 5 5		Funcionamiento correcto. Los dos puntos del coche que primero sale de imagen desaparecen con el.
Turbulence03	Pasa un coche (1140)	4 3 1 1		Se van perdiendo puntos.
Turbulence03	Se cruzan dos coches (uno en cada sentido) (1165)	Izq->Der	1 1 1	Al realizarse el cruce el coche que pasa por detrás pierde su punto.
		Der->Izq	1 0 0	
Turbulence03	Se cruzan dos coches (uno en cada sentido) (1270)	Izq->Der	3 2 2	En el cruce uno de los puntos del que pasa por detrás cambia de sentido siguiendo al otro coche. Uno de los del coche delantero también se pierde.
		Der->Izq	1 0	
Turbulence03	Pasa un coche (1460)	1 1 0 0		Se pierde el punto al pasar por un poste (el punto sigue el ruido del movimiento de dicho poste)
Turbulence03	Se cruzan dos coches (uno en cada sentido) (1550)	Izq->Der	1 1 0 0	Los tres puntos del coche que pasa por detrás se pierden, más tarde, aquel que seguía al delantero cambia de sentido siguiendo al otro coche.
		Der->Izq	3 0 0 0	
ZoomInZoomOut	Persona andando mientras se aleja el zoom (300)	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		Funcionamiento correcto del punto posicionado en la persona, aunque se registran movimientos de puntos del edificio (se van alejando de la camara)
ZoomInZoomOut	Persona andando mientras se acerca el zoom (740)	1 1 1 0 0 0		Se pierde el punto perteneciente a la persona con el acercamiento del zoom.

## Referencias

---

- [1] K. Toyama, J. Krumm, B. Brumitt, B. Meyers, Wallflower: principles and practice of background maintenance, in: International Conference on Computer Vision (ICCV), 1999, pp. 255–261.
- [2] M. Ota. Obstacle Detection System with Stereo Cameras for Level Cross-ings. Railway Technology Avalanche , 5:32. (2004)
- [3] Control inteligente en las grandes ciudades, 2010. Revista DGT, Núm. 201 (<http://www.dgt.es/revista/archivo/pdf/num201-2010-vigilancia.pdf>)
- [4] S.A. Velastin, B.A. Boghossian, B.P.L. Lo, J. Sun, y M.A.Vicencio-Silva. Prismatic: Toward ambient intelligence in public transport environments. IEEE Transactions on Systems, Man and Cybernetics, Part A, 35(1):164–182 , 2005.
- [5] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, C. Rosenberger, Review and evaluation of commonly-implemented background subtraction algorithms. In: IEEE International Conference on Pattern Recognition (ICPR), 2008, pp. 1–4.
- [6] Y. Dhome, N. Tronson, A. Vacavant, T. Chateau, C. Gabard, Y. Goyat, D. Gruyer, A benchmark for background subtraction algorithms in monocular vision: a comparative study, in: IEEE International Conference on Image Processing Theory, Tools and Applications (IPTA), 2010, pp. 66–71.
- [7] A. Prati, I. Mikic, M. Trivedi, R. Cucchiara, Detecting moving shadows: algorithms and evaluation, IEEE Trans. Pattern Anal. Mach. Intell. 25 (7) (2003) 918–923.
- [8] A. Bruhn, J. Weickert, C. Schnörr. "Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Methods", International Journal of Computer Vision, 61, 3, 211-321, 2005.
- [9] S. Baker, I. Matthews, "Lucas-Kanade 20 Years On: A Unifying Framework", International Journal of Computer Vision, 56, 3, 221-255, 2004.
- [10] A. Sobral, An OpenCV C++ Background Subtraction Library in IX Workshop de Visão Computacional (WVC'2013) Rio de Janeiro, Brazil, 2013.
- [11] A. Sobral, A. Vacavant. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. Computer Vision and Image Understanding (CVIU), 2014.
- [12] A. Sobral, L. Oliveira, L. Schnitman, F. Souza, (Best Paper Award) Highway Traffic Congestion Classification Using Holistic Properties. In International Conference on Signal Processing, Pattern Recognition and Applications (SPPRA'2013), Innsbruck, Austria, Feb 2013
- [13] Open CV (Open Source Computer Vision). Disponible en: [www.opencv.org](http://www.opencv.org)
- [14] Eclipse (C/C++ Development Tooling). Disponible en: [www.eclipse.org](http://www.eclipse.org)
- [15] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, P. Ishwar, Changedetection.net: a new change detection benchmark dataset, in: Workshop on Change Detection (CDW) at the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 1–8.
- [16] A. Sobral, A. Vacavant: Computer Vision and Image Understanding 122 (2014) 4–21
- [17] T. Bouwmans, Recent advanced statistical background modeling for foreground detection: A systematic survey, Recent Patents Comput. Sci. 4 (3) (2011) 147–176.
- [18] T. Bouwmans, F. El Baf, B. Vachon, Statistical background modeling for foreground detection: a survey, Handbook of Pattern Recognition and Computer Vision, vol. 4, World Scientific Publishing, 2010, pp. 181–199. Ch. 3.
- [19] A. de la Escalera Hueso. Visión por Computador, Fundamentos y Métodos. Prentice Hall, 2001.

- 
- [20] P. KaewTraKulPong y R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *VideoBased Surveillance Systems*, pages 135{144. Springer, 2002.
- [21] M. Heikkila y M. Pietikainen. A texture-based method for modeling the background and detecting moving objects. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):657{662, 2006
- [22] M. Heikkila, M. Pietikainen, y J. Heikkila. A texture-based method for detecting moving objects. In *BMVC*, pages 1{10, 2004.
- [23] P. Kaewtrakulpong, R. Bowden, An improved adaptive background mixture model for realtime tracking with shadow detection, *European Workshop on Advanced Video Based Surveillance Systems (AVSS)*, 2001
- [24] Z. Zivkovic, F.V. D. Heijden, Efficient adaptive density estimation per image pixel for the task of background subtraction, *Pattern Recog. Lett.* 27 (7) (2006) 773–780.
- [25] A.B. Godbehere, A. Matsukawa, K. Goldberg, Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation, in: *American Control Conference (ACC)*, 2012.
- [26] N. J. B. McFarlane, C. P. Schofield, Segmentation and tracking of piglets in images, *Mach. Vis. Appl.* 8 (3) (1995) 187–193.
- [27] C. Stauffer, W.E.L. Grimson, Adaptive background mixture models for a realtime tracking, in: *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999.
- [28] C. Wren, A. Azarbayejani, T. Darrell, A. Pentland, Pfinder: real-time tracking of the human body, *IEEE Trans. Pattern Anal. Mach. Intell.* 19 (7) (1997) 780–785.
- [29] S. Calderara, R. Melli, A. Prati, R. Cucchiara, Reliable back ground suppression for complex scenes, in: *ACM International Workshop on Video Surveillance and Sensor Networks*, 2006.
- [30] A. Prati, I. Mikic, M. Trivedi, R. Cucchiara, Detecting moving shadows: algorithms and evaluation, *IEEE Trans. Pattern Anal. Mach. Intell.* 25 (7) (2003) 918–923.
- [31] F. El Baf, T. Bouwmans, B. Vachon, Type-2 fuzzy mixture of Gaussians model: Application to background modeling, in: *International Symposium on Advances in Visual Computing (ISVC)*, 2008.
- [32] F. El Baf, T. Bouwmans, B. Vachon, Fuzzy statistical modeling of dynamic backgrounds for moving object detection in infrared videos, in: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2009, pp. 60–65.
- [33] T. Bouwmans, F. El Baf, Modeling of dynamic backgrounds by type-2 fuzzy Gaussian mixture models, *MASAUM J. Basic Appl. Sci.* (2009).
- [34] Z. Zhao, T. Bouwmans, X. Zhang, Y. Fang, A fuzzy background modeling approach for motion detection in dynamic backgrounds, in: *International*.
- [35] J. Yao, J. marc Odobez, Multi-layer background subtraction based on color and texture, in: *IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, 2007.
- [36] L. Maddalena, A. Petrosino, A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection, *Neural Comput. Appl.* 19 (2) (2010) 179–186.
- [37] T. Bouwmans, F. El Baf, B. Vachon, Background modeling using mixture of Gaussians for foreground detection - a survey, *Recent Patents Comput. Sci.* 1 (3) (2008) 219–237.
- [38] L. Maddalena, A. Petrosino, A self-organizing approach to background subtraction for visual surveillance applications, *IEEE Trans. Image Process.* 17 (7) (2008) 1168–1177.
- [39] L. Maddalena, A. Petrosino, A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection, *Neural Comput. Appl.* 19 (2) (2010) 179–186.

- 
- [40] Y. Goyat, T. Chateau, L. Malaterre, L. Trassoudaine, Vehicle trajectories evaluation by static video sensors, in: IEEE International Conference on Intelligent Transportation Systems, 2006.
- [41] B. Lorenço, T. Henrique, R. Goularte, Video Scene Detection by Multimodal Bag of Features Journal of Information and Data Management, Vol. 5, No. 2, June 2014.
- [42] S. Noh and M. Jeon, "A new framework for background subtraction using multiple cues," in Computer Vision ACCV 2012, ser. Lecture Notes in Computer Science, K. Lee, Y. Matsushita, J. Rehg, and Z. Hu, Eds. Springer Berlin Heidelberg, 2013, vol. 7726, pp. 493–506.
- [43] P.L.St-Charles, G.A. Bilodeau, "Flexible Background Subtraction With Self-Balanced Local Sensitivity" LITIV lab., Dept. of Computer & Software Eng. École Polytechnique de Montreal, Montreal, QC, Canada (2014).
- [44] P.L.St-Charles, G.A. Bilodeau, "Improving Background Subtraction using Local Binary Similarity Patterns" LITIV lab., Dept. of Computer & Software Eng. École Polytechnique de Montreal, Montreal, QC, Canada (2014).
- [45] Salton, G. y M. J. McGill., Introduction to Modern Information Retrieval. New York: McGraw Hill (1983)
- [46] M. Beitzel. On Understanding and Classifying Web Queries (Ph.D. thesis). IIT. CiteSeerX: 10.1.1.127.634. (2006).
- [47] T.Ellis B.James y P.Rosin. A novel method for video tracking performance evaluation. Pattern Recognition, pages 125–132, 2003
- [48] I.Saleemi S.Ali S.Bhattacharya, H. Idrees y M.Shah. Moving object detection and tracking in forward looking infra-red aerial imagery. 2010.
- [49] C. Harris y M. Stephens). «A combined corner and edge detector». Proceedings of the 4th Alvey Vision Conference. pp. 147–151. 1988
- [50] S. M. Smith y J. M. Brady. Susan - a new approach to low level image processing. International Journal of Computer Vision, 23(1):45–78, 1997.
- [51] D. G. Lowe. Object recognition from local scale-invariant features. Proceedings of the International Conference on Computer Vision., 2:1150–1157, 1999.
- [52] B. Herbert, T. Tuytelaars y L. Van Gool (2006). SURF: Speeded up robust features. In: European Conference on Computer Vision
- [53] M. Calonder, V. Lepetit, C. Strecha y P. Fua, "BRIEF: Binary Robust Independent Elementary Features", Proceedings of the European Conference on Computer Vision, 2010
- [54] E. Rublee, V. Rabaud, K. Konolige y G. Bradski, "ORB: an efficient alternative to SIFT and SURF", IEEE International Conference on Computer Vision, 2011
- [55] Horn, B.K.P. y Schunk, B.G. 'Determining optical flow', Artificial Intelligence, vol. 17, 185-203. (1981)
- [56] B. D. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision. 1981.
- [57] J. Y. Bouguet, . Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. Intel Corporation, 5. 2001.