Junio de 2009

Análisis y detección de objetos de primer plano en secuencias de video

Proyecto Final de Carrera

Oscar Mateu Garcia
Tutora: Montse Pardàs
Tutora: Glòria Haro
Junio de 2009

Agradecimientos

A Montse Pardàs y a Glòria Haro, quienes han dirigido el proyecto y siempre me han ayudado con gran profesionalidad y paciencia.

A mis compañeros de laboratorio, especialmente a Albert, Marcel y Jaime, quienes siempre han estado disponibles para echarme una mano cuando lo he necesitado.

Y especialmente, ya no sólo en el desarrollo de proyecto, sino también durante toda la carrera:

A mis amigos, especialmente Gisela, David, Cali y Javi

A mi familia, en especial a mi madre y a David

Por haber aguantado mis cambios de humor, por haber estado siempre ahí y por haber aguantado mis conversaciones monotemáticas.

Muchas gracias a todos.

Índice

I Introducción 1
II Detección de objetos de primer plano 3
1 Sistemas de detección de primer plano clásicos....5
1.1 Running Gaussian average 6
1.2 Múltiples Gaussianas (Stauffer & Grimson) 8
1.3 Histograma 12
1.4 Kernel Density Estimation (KDE)....13
1.5 Comparativa 21
2 Sistema post-procesado: Corrección de sombras 23
2.1 Método luminancia normalizada....24
2.2 Método híbrido 28
2.3 Método reflectancia 32
2.4 Comparativa 44
3 Sistemas de detección de primer plano con regularidad espacial....45
3.1 Inclusión de información espacial en el modelo: KDE 5D 46
3.2 Estimación Bayesiana mediante dos modelos 52
3.3 Estimación Bayesiana mediante varios modelos (Seguimiento) 54
III Conclusiones....63
IV Futuras líneas de trabajo 65
V Referencias....67

"La imaginación es más importante que el conocimiento."

Albert Einstein (1879-1955)

I Introducción

En la actualidad, la seguridad se está convirtiendo en un servicio de primera necesidad, tanto en espacios públicos como privados. Para garantizarla, se utiliza una gran cantidad de desarrollos tecnológicos como los sistemas de videovigilancia, los cuales tienen por objetivo facilitar el trabajo a los entes de seguridad mediante el análisis de las secuencias de video.

Además de éstas, también existen otras aplicaciones donde es necesaria la interacción entre máquinas y hombre mediante secuencias de video, como pueden ser las *smart-rooms* o habitaciones inteligentes, siendo necesario analizar el contenido de las secuencias de video para que sistemas autónomos puedan tomar decisiones de forma automática.

Teniendo en cuenta el marco de aplicaciones en el que nos encontramos, vamos a trabajar con sistemas capaces de detectar objetos en movimiento e incluso seguirlos para poder ser utilizados en protocolos de seguridad. Por tanto, gracias a algoritmos que procesan las secuencias de video captadas por las cámaras fijas de videovigilancia instaladas en el perímetro de seguridad, vamos a trabajar con sistemas que son capaces de denunciar comportamientos extraños a las competencias pertinentes, aumentando así la eficiencia de los sistemas de seguridad.

Dicho esto, este proyecto tiene por objetivo presentar algunas de las técnicas que se utilizan para detectar objetos de primer plano en una secuencia de video, siendo utilizadas en algunos de los escenarios descritos anteriormente. Para ello, se realizará un análisis comparativo entre distintas técnicas utilizadas, estudiando las ventajas e inconvenientes de cada una de ellas.

Además, se prestará especial atención a los algoritmos basados en un modelo de fondo a nivel de imagen, con el fin de abrir la puerta a futuras líneas de investigación que puedan aportar mejores soluciones a la sociedad.

II Detección de objetos de primer plano

En procesado de imagen se entiende por detección de primer plano o de foreground al conjunto de técnicas que tienen por objetivo detectar objetos en movimiento que aparecen en la secuencia de video sobre la que se trabaja.

Para ello, teniendo presente que las cámaras utilizadas en las aplicaciones prácticas mencionadas son fijas y las restricciones impuestas por la necesidad de trabajar en tiempo real, la mayoría de estos algoritmos se basan en las distribuciones estadísticas de color estimadas para cada píxel, siendo conocidas también como *modelado* del píxel, técnicas que serán agrupadas y explicadas en el apartado *Sistemas de detección de primer plano clásicos*.

Aún así, otras de las técnicas desarrolladas en este proyecto incorporan también información espacial del píxel con el fin de desarrollar un algoritmo más robusto. Éstas serán explicadas en el apartado *Sistemas de detección de primer plano con regularidad espacial*, mostrando algunas de las mejoras aportadas con respecto a los sistemas clásicos.

En cualquier caso, todas ellas obtienen un modelo estadístico del fondo, ya sea considerando cada píxel o toda la imagen completa como una variable aleatoria independiente. Además, se ha intentado cumplir con las siguientes premisas de diseño con el fin de conseguir aumentar la robustez del sistema de detección de objetos de primer plano:

- **Modelo de fondo actualizable:** el modelo de fondo utilizado para detectar los objetos debe poder evolucionar junto con la secuencia, con el fin de adaptarse a los cambios observados en ella, como pueden ser los cambios de iluminación o la detención de objetos de primer plano en el fondo, situación en que el objeto de primer plano pasaría a ser un objeto inmóvil de fondo.

- **Reducir falsas detecciones de primer plano / maximizar las detecciones correctas:** debido a que la estimación de la distribución estadística del fondo tiene como objetivo calcular la probabilidad de que una nueva muestra pertenezca a este modelo, es muy importante elegir correctamente un umbral de decisión que permita discernir entre primer plano y fondo. Pero también hay otras técnicas que consiguen el mismo objetivo, como son incluir información espacial en los modelos de fondo.

- **Eliminar sombras o brillos de las detecciones de primer plano:** independientemente del método de detección de primer plano, debido a que los modelos están basados en el color, es fácil detectar erróneamente sombras de objetos o brillos, pudiéndose aplicar otros algoritmos para disminuir la aparición de estas falsas detecciones.

Para realizar el modelado del fondo en el conjunto de sistemas de detección clásicos, se consideran los distintos píxeles como variables aleatorias independientes. Es decir, dada una secuencia de video cuyo fotograma tiene un tamaño de MxP píxeles, se considera que la secuencia está formada por MxP variables aleatorias independientes. En cambio, en el caso de los sistemas de detección con regularidad espacial se considera cada imagen como una única variable aleatoria, modelando el fondo a partir de las N imágenes del periodo de aprendizaje.

En cualquier caso, cada una de las técnicas permitirá obtener una función de densidad de probabilidad de fondo estimada a partir de las muestras conocidas durante el periodo de aprendizaje, lo que nos permitirá detectar los objetos de primer plano.

Explicadas las premisas de diseño, todo sistema de detección de objetos de primer plano se basa en:

1. Crear el modelo de fondo: la parte inicial de la secuencia, generalmente ausente de objetos a detectar, se utiliza con el fin de poder estimar correctamente la distribución estadística del fondo a partir de los valores observados. Por tanto, a lo largo de este periodo no será posible detectar objetos puesto que el único objetivo es construir el modelo que posteriormente será usado para realizar dicha acción, por lo que todo valor será considerado siempre fondo. Este periodo es conocido como *periodo de aprendizaje o entrenamiento.*

2. A partir del modelo generado, se procederá a detectar objetos de primer plano en la secuencia de trabajo según la metodología propia del sistema utilizado.

3. Finalmente, todos los píxeles que hayan sido asignados al fondo se incluirán en el correspondiente modelo de fondo (se actualizará con el nuevo valor).

En el siguiente esquema se puede observar gráficamente cada una de las fases en las que se basa todo sistema de detección de primer plano:

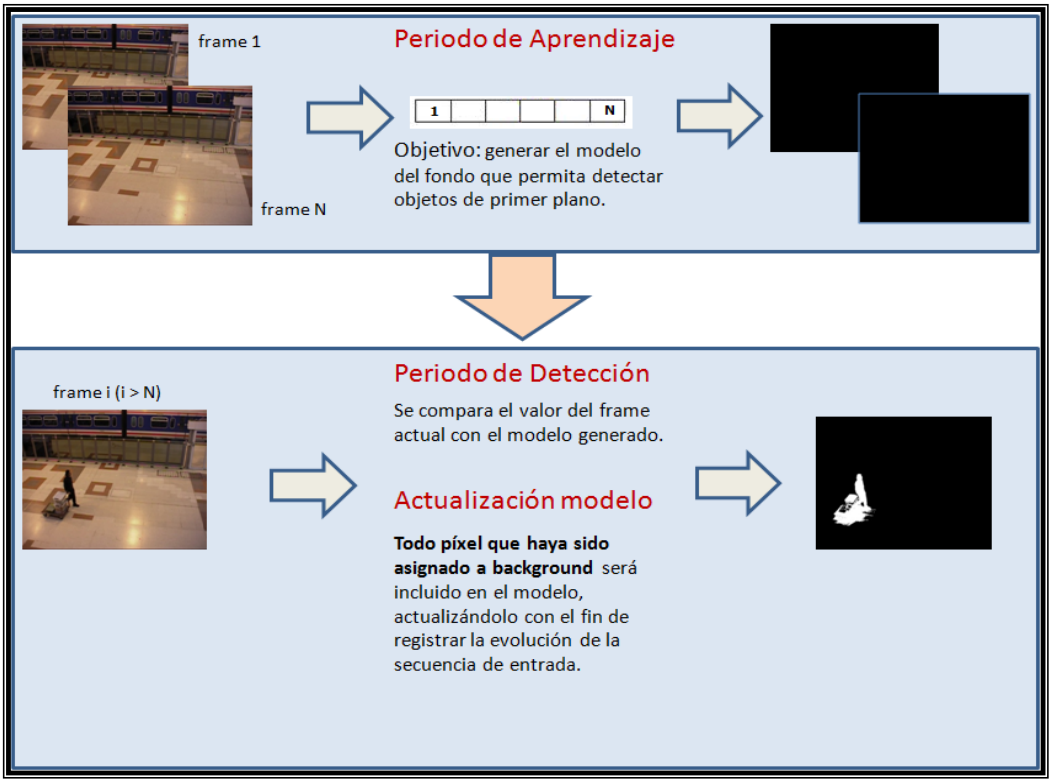


Figura 1 – Diagrama de funcionamiento de un sistema detector de primer plano genérico

1 Sistemas de detección de primer plano clásicos

Como ya se ha comentado, en este capítulo se explicarán distintos sistemas de detección de primer plano cuyos modelos acumulados del fondo están basados únicamente en el color de cada píxel.

Formalmente, existen dos familias de estimadores: los estimadores paramétricos y los no paramétricos. Los primeros sólo pueden usarse en el caso de que la función de densidad de probabilidad a estimar se asemeje con alguna de las distribuciones conocidas (distribución Gaussiana, de Poisson,...), es decir, todo estimador paramétrico sólo es válido para una cierta distribución estadística. En cambio, los segundos estimadores son válidos independientemente de la forma que tome la función de densidad de probabilidad (también conocida como *pdf* del inglés *probability density function*).

Hecha esta breve introducción, en este capítulo se procederá a explicar cada uno de los métodos usados para detectar objetos de primer plano, clasificándose en:

- Estimadores paramétricos
  - Running Gaussiana Average
  - Múltiples Gaussianas (S & G – Stauffer & Grimson)
- Estimadores no paramétricos
  - Histograma
  - Kernel Densisty Estimation (KDE)

Finalmente, el capítulo lo cerrará una comparativa entre las distintas técnicas a modo de conclusión.

1.1 Running Gaussian average

Ésta es una de las primeras técnicas surgidas con el fin de detectar objetos de primer plano en una secuencia de vídeo. Wren et al en [1] proponen modelar el fondo de cada píxel mediante una función de densidad de probabilidad conocida, la función Gaussiana, la cual queda determinada mediante sus dos parámetros: media μ y varianza σ^2. La forma de esta función matemática puede observarse en la siguiente figura.

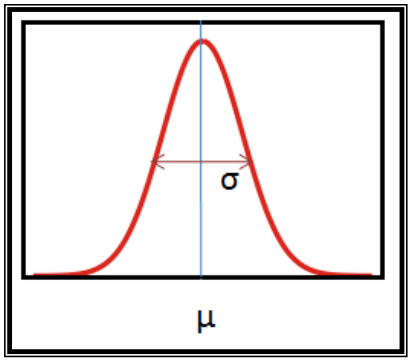


Figura 2 – Función Gaussiana con parámetros estadísticos

De este modo, para estimar la función de densidad de probabilidad del modelo a partir de las muestras obtenidas durante el periodo de entrenamiento, basta con calcular los parámetros característicos de la curva a partir de éstas.

Para ello, se trabaja con cada uno de los canales de color (R,G,B) que se utilizan para obtener los más de 16 millones de colores que se es capaz de representar en un píxel (los cuales surgen de todas las combinaciones que existen con los 256 posibles valores que puede tomar cada uno de los canales R,G,B).

En general, podremos considerar que los canales de color son estadísticamente independientes. Dicha independencia se traduce en que la pdf del píxel en cuestión se puede obtener como el producto de las pdf's marginales que se pueden estimar a partir de la estimación de cada uno de los mencionados canales por separado, es decir, $f(z)=f_R(R)\cdot f_G(G)\cdot f_B(B)$, donde z=(R,G,B).

De este modo, a partir de los N valores observados en cada uno de los tres canales de color durante el periodo de entrenamiento, se puede calcular una media μ y una varianza σ^2 que sea capaz de representarlos a todos ellos.

Una vez encontrada cada una de las Gaussianas marginales, se obtendrá la pdf del píxel que modele el fondo observado como el producto de éstas. De este modo, habiendo estimado la pdf del fondo de cada uno de los píxeles, se utilizará para decidir si un píxel de valor *x* se corresponde con el modelo obtenido mediante el siguiente criterio:

$$|x-\mu| > k\sigma \Rightarrow \text{foreground}$$
$$\text{Resto} \Rightarrow \text{background}$$

Expresión 1.1

donde k es una constante que puede valer entre 3 y 5.

Finalmente, sólo queda aclarar el método usado para realizar la actualización de los parámetros de la Gaussiana una vez obtenido el modelo de fondo. Esta actualización se realiza a lo largo de toda la secuencia con el fin de adaptar la estimación a la propia evolución de la secuencia, mediante las siguientes ecuaciones:

$$\mu_t = \rho x_t + (1-\rho)\mu_{t-1}$$

Expresión 1.2

$$\sigma_t^2 = \rho(x_t - \mu_t)^2 + (1-\rho)\sigma_{t-1}^2$$

Expresión 1.3

donde x_t es el valor del píxel en la imagen actual, μ_t y σ_t son la media y la desviación estándar que caracterizan la función de densidad de probabilidad del píxel en ese mismo instante, y ρ es el parámetro de absorción, el cual determina la velocidad con la que se actualiza el modelo en cada imagen (valor usual de ρ=0.01). En este sentido, resaltar la importancia de actualizar el modelo sólo en el caso de que se haya decidido que el píxel pertenece al fondo, la cual fue apuntada por Koller et al. en [2].

Sobre este sistema, cabe destacar la elevada velocidad de procesado y los bajos requerimientos de memoria y coste computacional asociados debido a su simplicidad. Aún así, en este proyecto no se ha realizado un estudio exhaustivo del mismo, sino que se ha trabajado con la evolución natural del mismo (el cual se explicará en el siguiente punto) debido a que el objetivo de este proyecto es obtener una buena detección de primer plano independientemente de los recursos usados. Así que la única finalidad de este apartado es introducir y facilitar la comprensión del método de detección de primer plano de múltiples Gaussianas.

1.2 Múltiples Gaussianas (Stauffer & Grimson)

Este método es un estimador paramétrico que proviene de la evolución del método anterior (running gaussian average), puesto que utiliza hasta K Gaussianas por canal de color para estimar la pdf del modelo de cada píxel. Surge con la finalidad de mejorar la detección de objetos en secuencias en las que aparecen variaciones periódicas en el fondo (movimiento de objetos del fondo con el viento como árboles, banderas,…).

Stauffer and Grimson [3] proponen este método para detectar objetos de primer plano, mediante la suma ponderada de funciones Gaussianas por canal y píxel:

$$\hat{f}(z) = \sum_{i=1}^{k} \omega_i \cdot G(z, \mu_i, \Sigma_i)$$

Expresión 1.4

donde K es el número máximo de Gaussianas utilizadas para realizar la estimación del modelo (normalmente se suele optar por usar entre tres y cinco funciones distintas), G(·) representa cada una de las Gaussianas utilizadas, ω_i son las ponderaciones que tienen cada una de las curvas, μ_i y Σ_i son los parámetros estadísticos de cada una estas Gaussianas y z es el valor del píxel en cuestión (magnitud vectorial perteneciente a R^3).

Dicha estimación se utiliza para, al igual que con el resto de sistemas de detección de primer plano, obtener un modelo de fondo y decidir si el píxel en cuestión se corresponde con el modelo o con un objeto de primer plano.

En este caso, las Gaussianas representan tanto el fondo como el primer plano y se aplicará un criterio para decidir de manera dinámica cuáles modelan el fondo y cuáles el primer plano. De esta forma será posible introducir los objetos de primer plano en el modelo de fondo cuando permanezcan estáticos en la escena, asignando la Gaussiana correspondiente a fondo.

Para ello, cada píxel (i,j) de la imagen (donde i y j son los índices que representan la posición de fila y columna, respectivamente, que ocupa el píxel en cuestión dentro de la imagen) se modela como una combinación de distribuciones Gaussianas (el número típico suele variar entre 3 y 5), combinándose entre sí a partir de los factores de ponderación $\omega i,t$, que se modifican iterativamente en función de las veces que se da el valor que recogen, formando así el modelo probabilístico del píxel.

Los factores de ponderación se normalizan según:

$$\omega_{i,t} = \frac{\omega_{i,t}}{\sum_{j=1}^{k} \omega_{j,t}}$$

Expresión 1.5

A fin de conseguir:

$$\sum_{i=1}^{K} \omega_{i,t} = 1$$

Expresión 1.6

Para la clasificación de un píxel como primer plano o fondo, se considera que el fondo queda modelado por las B distribuciones Gaussianas de mayor peso y menor varianza según la resolución de la siguiente inecuación:

$$B = \arg\min_b \left(\sum_{i=1}^{b} \omega_{i,t} > T \right)$$ Expresión 1.7

donde *T* es el umbral de decisión asignado (usualmente 0.6), y *B* es el número mínimo de distribuciones a incluir en el sumatorio (ordenadas según ω/σ), para que se cumpla la inecuación.

De este modo, las B primeras distribuciones Gaussianas serán las que modelarán la función de densidad de probabilidad del fondo, puesto que normalmente es más estático y aparece con más frecuencia, correspondiéndose con aquellas Gaussianas que han sido utilizadas más veces y que son más compactas.

Para comprobar si el valor del píxel de entrada encaja en alguna distribución Gaussiana del modelo probabilístico del píxel, se evalúa la siguiente inecuación:

$$\left| x_t - \mu_{i,t} \right| > k\sigma_{i,t}$$ Expresión 1.8

Donde x_t es el valor de un canal de color del píxel estudiado en el fotograma t, $\mu_{i,t}$ es la media de la Gaussiana i-ésima y $\sigma_{i,t}$ la desviación estándar de la Gaussiana i-ésima.

- Si la inecuación se cumple para todas las Gaussianas, se decide primer plano, puesto que se debe a que no encaja con el modelo probabilístico del píxel aprendido hasta el momento. En este caso, se generará una nueva Gaussiana con el fin de permitir adaptar el modelo al fondo, eliminando la distribución con menor peso.

- Si la inecuación no se cumple para una Gaussiana, se supone que el valor encaja con el modelo probabilístico modelado mediante la misma (si la inecuación no se cumple para más de una Gaussiana, se supondrá que la función que mejor representa al píxel es la de menor varianza). En este caso, si el valor del píxel encaja con una de las B primeras distribuciones Gaussianas se asignará a fondo, debido a que la frecuencia con la que ha aparecido es lo suficientemente elevada como para que sea considerado fondo.

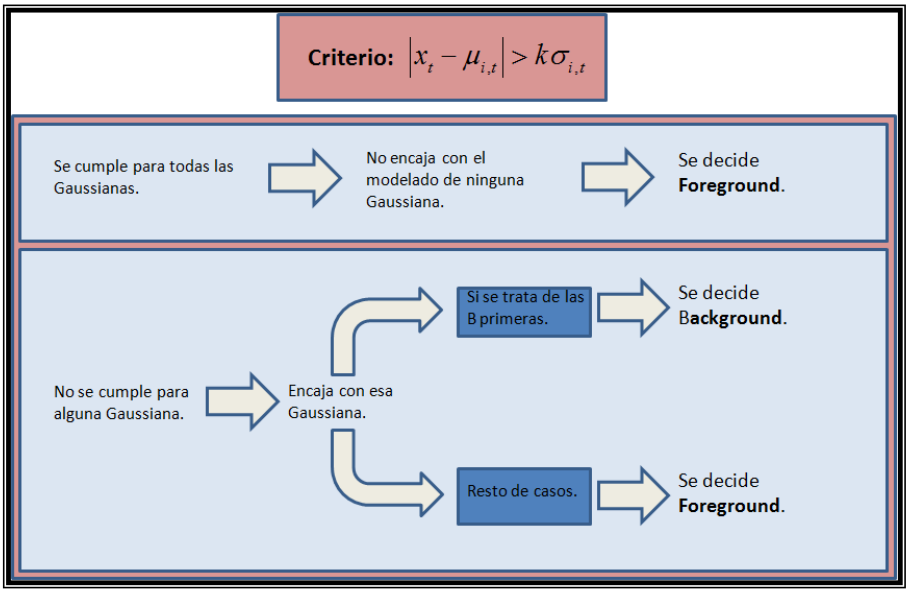


Figura 3 – Esquema de decisión del detector Stauffer and Grimson

En cuanto a la actualización del modelo, decir que se realizará mediante las siguientes expresiones:

Actualización de la media y la varianza (afecta sólo a la distribución Gaussiana que ha encajado):

$$\mu_t = \rho x_t + (1-\rho)\mu_{t-1}$$

$$\sigma_t^2 = \rho(x_t - \mu_t)^2 + (1-\rho)\sigma_{t-1}^2 \qquad \text{Expresión 1.9}$$

Actualización de los pesos (afecta a todas las distribuciones que conforman el modelo):

$$\omega_{k,t} = (1-\alpha)\omega_{k,t-1} + \alpha(M_{k,t}) \qquad \text{Expresión 1.10}$$

donde α es un coeficiente de absorción (distinto a ρ, normalmente se suele usar un valor fijo de 0,005) y $M_{k,t}$ es 1 para la función Gaussiana que ha encajado y 0 para las que no.

Resultados:

Éste es un sistema de detección de primer plano muy estudiado y utilizado. Por este motivo se ha decido realizar una serie de pruebas de detección con este método, para poder ser comparado a posteriori con los otros sistemas desarrollados, utilizándose como referencia para resaltar las mejoras y deficiencias aportados por cada uno de ellos con respecto a éste. A continuación, se mostrarán algunos de los resultados obtenidos con distintas secuencias:

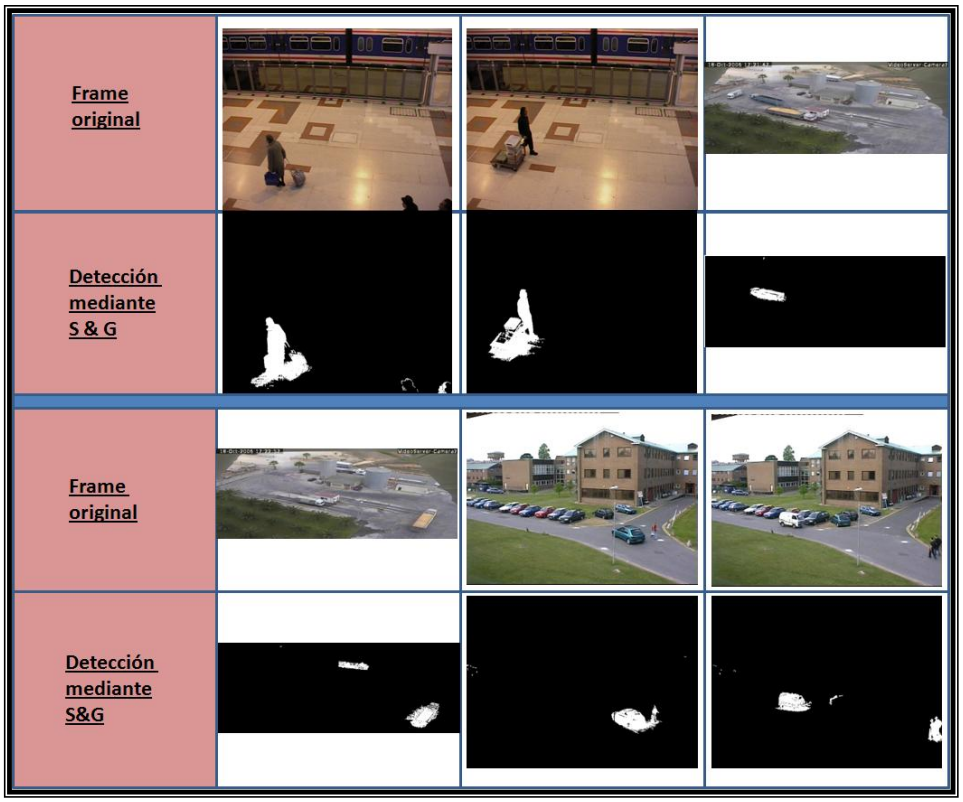


Figura 4 – Resultados Stauffer and Grimson

Además de la tradición histórica, destacar también que se ha usado como referencia debido a la calidad de los resultados (obsérvese la poca cantidad de falsas detecciones) y debido a la estabilidad de la implementación desarrollada en la librería en la que se ha trabajado.

A medida que se necesite comparar estos resultados con algún otro método, se rescatarán con el fin de poder comparar ambas detecciones.

1.3 Histograma

El ejemplo más sencillo de estimador no paramétrico es el histograma. Partiendo del supuesto de que debemos estimar la función de densidad de probabilidad de una variable aleatoria cuyo espacio muestral es discreto (un número finito de posibles valores), los histogramas se basan en contar el número de veces o la frecuencia con la que aparece cada uno de los valores de dicho espacio muestral en las muestras que se tienen en cuenta para realizar la estimación.

Por tanto, cada vez que se requiera actualizar el histograma se incrementa en 1 el valor que se corresponda con la muestra utilizada, como si de una concatenación de N contadores independientes se tratara (uno por cada posible valor).

Por otro lado, en el supuesto caso de que fuera necesario utilizar un histograma para realizar la estimación de una variable aleatoria continua (el espacio muestral está formado por un conjunto infinito de valores contenidos dentro de un intervalo), simplemente sería necesario decidir el número de intervalos en los que dividir el espacio muestral y la longitud de los mismos, cuantificando los infinitos valores en L niveles. En este sentido, normalmente también en el caso de una variable aleatoria discreta es necesario cuantificar los valores, ya que no se dispone de suficientes datos de entrenamiento para estimar correctamente la probabilidad de cada uno de los valores.

En cualquier caso, el resultado es un gráfico escalonado con distintas alturas para cada posible valor o rango de valores, dando una idea de la probabilidad que tendrá asociada cada uno: los valores que más se hayan repetido entre las muestras tendrán asociada una probabilidad mayor que aquellos que casi no hayan aparecido.

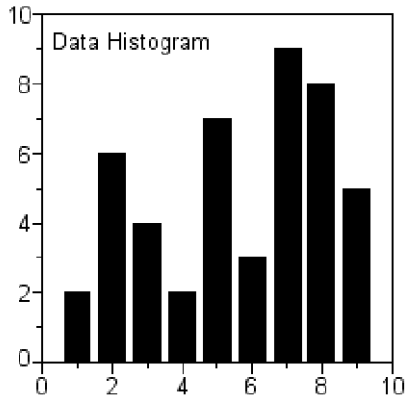


Figura 5 – Ejemplo histograma

En la figura adjunta hay un ejemplo de histograma. Como se puede ver, existen tres problemas asociados a este estimador:

a) Es necesario decidir el número de niveles a representar.

b) El resultado es una estimación discretizada o escalonada de la función de densidad de probabilidad, cosa que no tiene sentido en una pdf continua.

c) A priori no cumplen una de las condiciones de las funciones de densidad de probabilidad (Área = 1), aunque existe la posibilidad de normalizarlo con ese fin.

También existe solución a los dos primeros problemas, tomar un número elevado de niveles L con el fin de obtener una pdf lo suficientemente continua y con un error de cuantificación mínimo. Por este motivo el histograma es un estimador no paramétrico también válido para pdf's continuas. Sin embargo, se requieren muchas muestras para estimar correctamente la pdf.

1.4 Kernel Density Estimation (KDE)

El Kernel Density Estimation es un método no paramétrico que permite estimar la función de densidad de probabilidad de una variable aleatoria a partir de algunas de sus muestras.

Su principio de funcionamiento está basado en el uso de funciones Kernel (de ahí proviene su nombre), cuya denominación se refiere a toda función integrable real definida no-negativa que cumple:

1.- Área unitaria: $\int_{-\infty}^{\infty} K(u)du = 1$ **Expresión 1.11**

2.- Simétrica: $K(-u) = K(u) \forall u$ **Expresión 1.12**

Sea x(0),…,x(N-1) el conjunto de muestras conocidas de nuestra variable aleatoria X, y sea K(x) la función kernel utilizada, obtenemos la estimación de la pdf de nuestra variable como la suma promediada de todas las funciones kernel centradas, cada una de ellas, en cada una de las muestras conocidas, es decir:

$$\hat{f}(x) = \frac{1}{N}\sum_{i=0}^{N-1}\left(\frac{K(x-x_i)}{h}\right) \quad \textbf{Expresión 1.13}$$

donde h es el ancho de la función kernel utilizada.

Normalmente, las funciones kernel utilizadas son las Gaussianas por su simplicidad y suavidad, aunque, en este sentido, cabe destacar que la calidad de la estimación no reside principalmente en la función kernel utilizada, sino en el ancho seleccionado. A continuación se podrá observar un ejemplo que ilustra la importancia de la correcta selección del ancho de la función kernel.

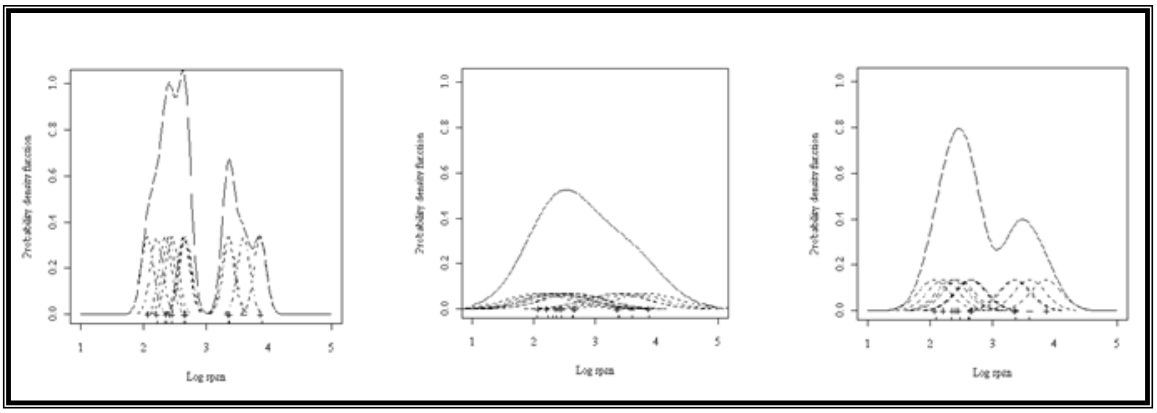


Figura 6 – Elección de ancho de banda para un estimador KDE.

Como puede observarse en este ejemplo, para un mismo número de Gaussianas (y por tanto muestras), la elección del ancho de la función utilizada para realizar la estimación puede variar de forma notable el resultado.

En la primera figura un ancho demasiado pequeño permite obtener una estimación demasiado variable, obteniendo picos mayores que en la verdadera pdf. En cambio, en la segunda, un ancho demasiado grande enmascara la verdadera forma de la pdf. Por último, a la derecha se puede observar el resultado de la elección óptima.

Aplicación de KDE para modelar el fondo

A. Elgammal, R. Duraiswami [4] proponen usar este estimador no paramétrico para implementar un sistema de detección de primer plano. El motivo por el que justifican su uso es que es adaptable a cualquier fondo (debido a que es un estimador no paramétrico) y, por tanto, adecuado para cualquier aplicación de visión artificial (no habrá limitaciones de Gaussianas para modelar el fondo).

Asumiéndose la independencia entre los canales de color, se puede resumir el funcionamiento del sistema en:

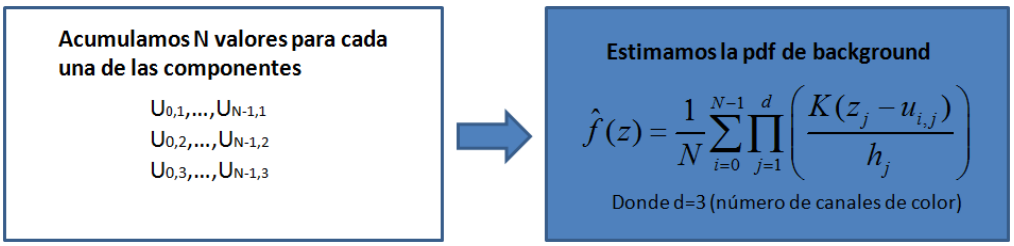


Figura 7 – Estimación de la función de densidad de probabilidad del fondo según KDE

Respecto a la expresión utilizada para realizar la estimación, es destacable:

- El uso del producto de cada una de las componentes (R,G,B). Puesto que son variables aleatorias independientes, se tiene que $f(z)=f(z_1,z_2,z_3) = f_R(z_1)\cdot f_G(z_2)\cdot f_B(z_3)$
- El uso de 1/N. Tiene por objetivo normalizar la pdf resultante, es decir, puesto que se usan N funciones kernel para estimar la pdf del fondo y que cada una tiene área 1, necesitamos ponderar cada una de ellas por 1/N con el fin de asegurar que el área de la pdf estimada resultante esté normalizada a 1.

Finalmente, cabe mencionar que la implementación realizada en este proyecto, como en la mayoría de aplicaciones prácticas, emplea Gaussianas como funciones Kernel, por lo que la expresión anterior queda traducida a:

$$\hat{f}(z)=\frac{1}{N}\sum_{i=0}^{N-1}\prod_{j=1}^{d}\frac{1}{\sqrt{2\pi\sigma_j^2}}e^{-\frac{1}{2}\frac{\left(z_j-u_{ij}\right)^2}{\sigma_j^2}} \quad \text{Expresión 1.14}$$

Decisión entre fondo y primer plano

En este apartado se va a detallar cómo usar el sistema para realizar la detección de primer plano, que es el objetivo de este capítulo.

Puesto que para cada uno de los píxeles tenemos una estimación de la pdf del modelo de fondo, es posible calcular la probabilidad de que un nuevo valor pertenezca a este modelo evaluando dicha estimación en el nuevo valor z, es decir

$$prob(z\in background): p(z)=\frac{1}{N}\sum_{i=0}^{N-1}\prod_{j=1}^{d}\frac{1}{\sqrt{2\pi\sigma_j^2}}e^{-\frac{1}{2}\frac{\left(z_j-u_{ij}\right)^2}{\sigma_j^2}} \quad \text{Expresión 1.15}$$

Hay que remarcar que el resultado de esta expresión es un número (una probabilidad) y no una expresión que modela una pdf (pese a la semejanza con la expresión anterior). Dicho esto, para una cierta imagen de entrada se utiliza esta probabilidad para determinar si cada uno de los píxeles corresponden o no al modelo acumulado (se corresponde con fondo/primer plano), mediante el siguiente criterio

$$\text{si } p(z) < th \text{ (umbral)} \Rightarrow \text{se decide primer plano}$$
$$\text{si } p(z) > th \text{ (umbral)} \Rightarrow \text{se decide fondo}$$

donde th es un umbral, un parámetro de entrada del sistema (el mismo para todos los píxeles) que habrá sido ajustado previamente mediante testeo.

Parámetros del sistema

De igual forma que el umbral de decisión, el algoritmo implementado en este proyecto tiene una serie de parámetros de entrada que permiten adaptarlo a la secuencia sobre la que se quiere trabajar. A continuación se detallarán algunas de las funcionalidades implementadas.

Actualización del modelo de fondo durante la detección

La implementación utilizada permite decidir si actualizar o no el modelo del fondo. Como en el resto de sistemas, la importancia de trabajar con algoritmos que son capaces de actualizar el modelo reside en que tienen la virtud de adaptarse a ciertos cambios del fondo, como puede ser la absorción de objetos de primer plano por el fondo por su inmovilidad, los cambios suaves de luminancia... El sistema actualizará el modelo sólo en el caso de haber decidido fondo.

Corrección de sombras

En ocasiones, la sombra de un objeto pude ser considerada como un objeto propiamente, debido a su no correspondencia con el modelo acumulado del fondo. En ciertas aplicaciones, como en las de tracking[1], estas detecciones pueden ser perjudiciales, debido a que pueden distorsionar las características del propio objeto a seguir. Por eso, se ha incorporado la posibilidad de comprobar si la detección se corresponde con objeto o con sombra, suprimiéndose en ese último caso.

En este apartado no se explicarán los métodos utilizados para suprimir la detección de sombras, puesto que el capítulo 2 está centrado íntegramente en ello.

Umbral de decisión entre fondo y primer plano (Threshold o Th)

Este parámetro permite al sistema trabajar con un umbral de decisión fondo/primer plano más alto o bajo, en función de la secuencia con la que se trabaje.

Un umbral suficientemente pequeño hace que no sólo los objetos sean detectados correctamente, sino que también aparezcan una gran cantidad de falsas detecciones (algunos píxeles aislados que son asignados a primer plano indebidamente).

[1] Se entiende por tracking al conjunto de técnicas que tienen por objetivo realizar un seguimiento de todos los objetos en movimiento que aparecen en la secuencia de video sobre la que se trabaja.

Por el contrario, un umbral suficientemente elevado impide la aparición de dichas falsas detecciones, aunque también provoca una distorsión en la detección del primer plano puesto que no asigna algunos píxeles que realmente deberían estar asignados a foreground, obteniendo como resultado figuras "partidas", agujereadas o poco claras. De ahí la importancia de encontrar un compromiso entre ambos extremos (ajuste del umbral), cosa que se consigue mediante el testeo.

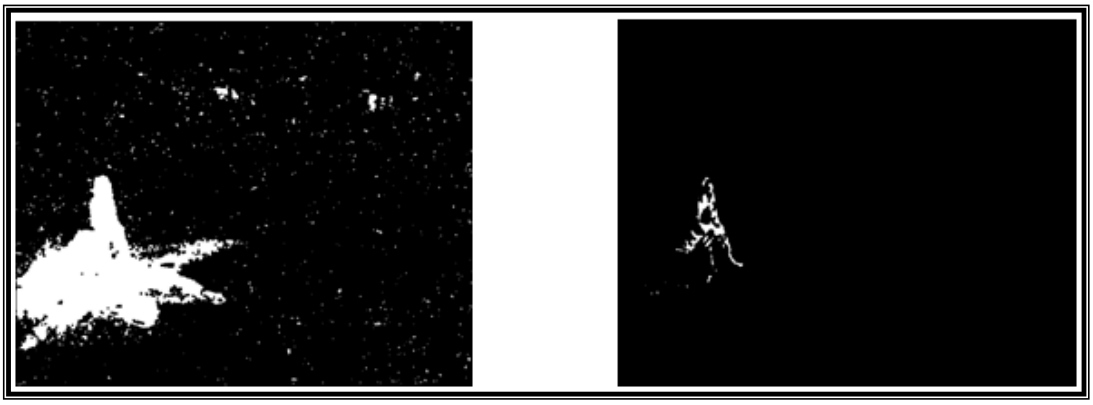


Figura 8 – Influencia del umbral de decisión en la máscara obtenida

En esta figura se puede observar la importancia de elegir un correcto umbral de decisión. A la izquierda, un umbral excesivamente alto permite la asignación de muchos píxeles a primer plano indebidamente (falsas detecciones), mientras que a la derecha se puede observar el resultado de elegir un umbral excesivamente bajo (hay muchos píxeles de foreground que tienen asociada una probabilidad menor al umbral, debido a que éste es muy bajo).

Estimación de la desviación estándar de la función kernel

Como ya se ha mencionado en el inicio de este estimador, la importancia de elegir un buen ancho de la función kernel (en nuestro caso, una buena desviación estándar de la función Gaussiana) es fundamental para una correcta estimación de la pdf. Por este motivo, se ha implementado una funcionalidad que permite optimizar dicha desviación estándar con el fin de ajustar al máximo la estimación de la pdf del modelo de fondo a la realidad. A continuación vamos a explicar cómo optimizamos la desviación estándar de cada Gaussiana.

Teniendo en cuenta que un mismo píxel puede representar distintos objetos en distintos instantes de tiempo (cielo, hojas y ramas de un árbol en distintos instantes), la estimación está basada en un histograma que trabaja con la distancia entre valores consecutivos $|x(i)-x(i+1)|$, con el fin de poder evitar la aparición de saltos de luminancia (relacionados con la representación de distintos objetos). De este modo, si se asume que la luminancia de cada canal sigue una distribución localmente en el tiempo normal $N(\mu,\sigma^2)$, es fácil demostrar que la diferencia entre muestras consecutivas $(x(i)-x(i+1))$ sigue una distribución normal $N(0,2\sigma^2)$.

Así que existe una función probabilística conocida, la Gaussiana, que es capaz de modelar el comportamiento de las diferencias entre luminancias de imágenes consecutivas. Por otro lado, para aumentar la robustez frente a estos cambios bruscos, se calcula la mediana del histograma de cada canal de color para cada píxel y se utiliza para estimar la desviación estándar óptima que mejor pueda representar a las muestras acumuladas.

Es decir, asumiendo que la luminancia diferencia $(x(i)-x(i+1))$ sigue la distribución $N(0,2\sigma^2)$, y teniendo en cuenta que lo que en realidad necesitamos es la distancia entre valores consecutivos (sólo tenemos en cuenta los valores positivos, o lo que es lo mismo, el 50% de los posibles valores), la mediana m de esta

variable es equivalente al 50% de las muestras acumuladas, o sea, el 25% de la distribución de la desviación estándar $N(0,2\sigma^2)$, por lo que:

$$(\Pr\left(N\left(0,2\sigma^2\right)>m\right)=0.25 \qquad \text{Expresión 1.16}$$

De este modo, el estimador de la desviación estándar utilizado, basado en la mediana calculada gracias al histograma acumulado de la distancia entre muestras consecutivas, es:

$$\hat{\sigma}=\frac{m}{0.68\sqrt{2}} \qquad \text{Expresión 1.17}$$

Mediante este estimador, el sistema es capaz de conseguir adaptar la desviación típica de las funciones Gaussianas de cada uno de los canales de los píxeles y así estimar de forma óptima la pdf del fondo para cada uno de los píxeles que conforman la imagen de entrada.

Implementación del algoritmo

En la inmensa mayoría de los diseños realizados en ingeniería se debe encontrar un compromiso entre coste y calidad. Y el sistema desarrollado en este proyecto no es una excepción.

Un ejemplo es el número de muestras utilizadas para realizar la estimación de la pdf, puesto que un número elevado puede hacer que la estimación sea muy buena, aunque el coste computacional sea muy alto, y viceversa. Por eso es necesario encontrar un compromiso que permita detectar correctamente las formas de los objetos evitando velocidades de procesado muy bajas. Estos compromisos dependen de la secuencia con la que se trabaje y puede oscilar entre 10 y 150 muestras, dependiendo de la calidad que se quiera obtener en la detección y la secuencia con que trabajar.

Aún así, dejando de lado los compromisos, hay implementadas técnicas con el fin de optimizar el código y así aumentar la eficiencia del algoritmo, pudiendo obtener una mayor calidad con la misma velocidad de procesado. A continuación se detallarán algunos ejemplos.

Actualización del modelo

Debido a que este sistema depende del número de muestras con las que se desea modelar la pdf de fondo, el algoritmo implementado en este proyecto utiliza un buffer cíclico para acumular dichas muestras, por lo que para actualizar el modelo sólo hay que sustituir la muestra más antigua por la más nueva.

Con el fin de hacerlo de una forma rápida, utiliza un puntero que se incrementa módulo N (donde N es la longitud del mismo, o lo que es lo mismo, el número de muestras con las que se trabaja) para indicar cuál es la muestra más vieja, y sustituyendo dicha muestra por la más nueva. De esta forma se evita desplazar las N-1 muestras restantes 1 posición a la izquierda para posteriormente guardar en la última posición la muestra de entrada.

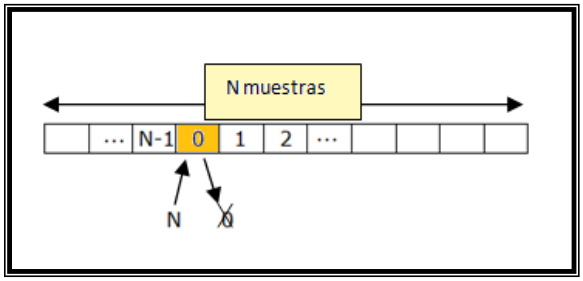


Figura 9 – Representación del bucle cíclico

A priori, la velocidad de la actualización del modelo varía en función del número de muestras, puesto que habría que realizar N-1 desplazamientos en memoria para guardar la última muestra en la última posición. De este modo conseguimos independizar la velocidad de actualización del modelo del número de muestras utilizadas.

Uso de Lookup Tables

Otro de los cuellos de botella del algoritmo relacionados con el número de muestras utilizadas es el cálculo de la probabilidad de pertenecer a fondo. La expresión utilizada en este cálculo es la correspondiente a la expresión 1.15, por lo que se puede deducir que será más lento cuantas más muestras sean utilizadas.

Pese a que esto es inevitable, se ha optado por usar lookup tables para aumentar la eficiencia en el cálculo de estas probabilidades con el fin de que, al tratarse de la parte fundamental del algoritmo, puedan usarse el número de muestras que se consideren necesarias.

Lookup tables no es más que una colección de tablas donde hay una serie de cálculos costosos (en nuestro caso, probabilidades) ordenados de manera que sea fácil de indexar. En nuestra implementación, es una matriz de 80 filas x 511 columnas donde hay acumuladas 80 Gaussianas (ordenadas de menor a mayor desviación), cada una de ellas con 511 muestras.

Tanto el número de Gaussianas a muestrear (80 filas), como la desviación estándar mínima y máxima pueden ser cambiadas puesto que son parámetros de la implementación. En cambio, el número de columnas no, puesto que, como consecuencia de trabajar con valores de color de 7 bits (256 posibles valores), 511 son todos los posibles valores de restar dos luminancias que pueden ser cualquier valor contenido en [0,255], es decir, [-255,255], o lo que es lo mismo, 511 valores.

Por tanto, para calcular $e^{\frac{x-x_i}{\sigma_j^2}}$ sólo nos hace falta acceder a la fila donde se encuentra muestreada la Gaussiana cuya desviación estándar es σj , y a la columna que resulta de la operación (x - x(i) + 255) para obtener el valor sin tener que calcularlo. Cabe destacar la importancia del 255, el cual nos sitúa en el centro de la Gaussiana muestreada para que los posibles valores [-255,255] sean utilizados como posiciones de un vector, es decir, [0,510].

Mediante este método se consigue aumentar la eficiencia del cálculo de estas probabilidades, disminuyendo la importancia de elegir un número elevado de muestras.

Resultados

Como ya se ha comentado, hay 2 parámetros de entrada al sistema que determinan la calidad de la detección realizada por el algoritmo: el umbral de decisión entre fondo/primer plano y el número de muestras utilizadas para hacer la estimación. A continuación se realizará un análisis de ambos parámetros para ver la influencia de ambos en la detección.

Umbral de decisión (threshold)

Se ha realizado una serie de pruebas de testeo para ajustar correctamente el umbral de decisión. En la siguiente figura se puede comprobar algunas de las pruebas realizadas y cuál es el correcto orden de magnitud del mismo, junto con los efectos explicados anteriormente de elegir uno excesivamente superior o inferior al mismo.

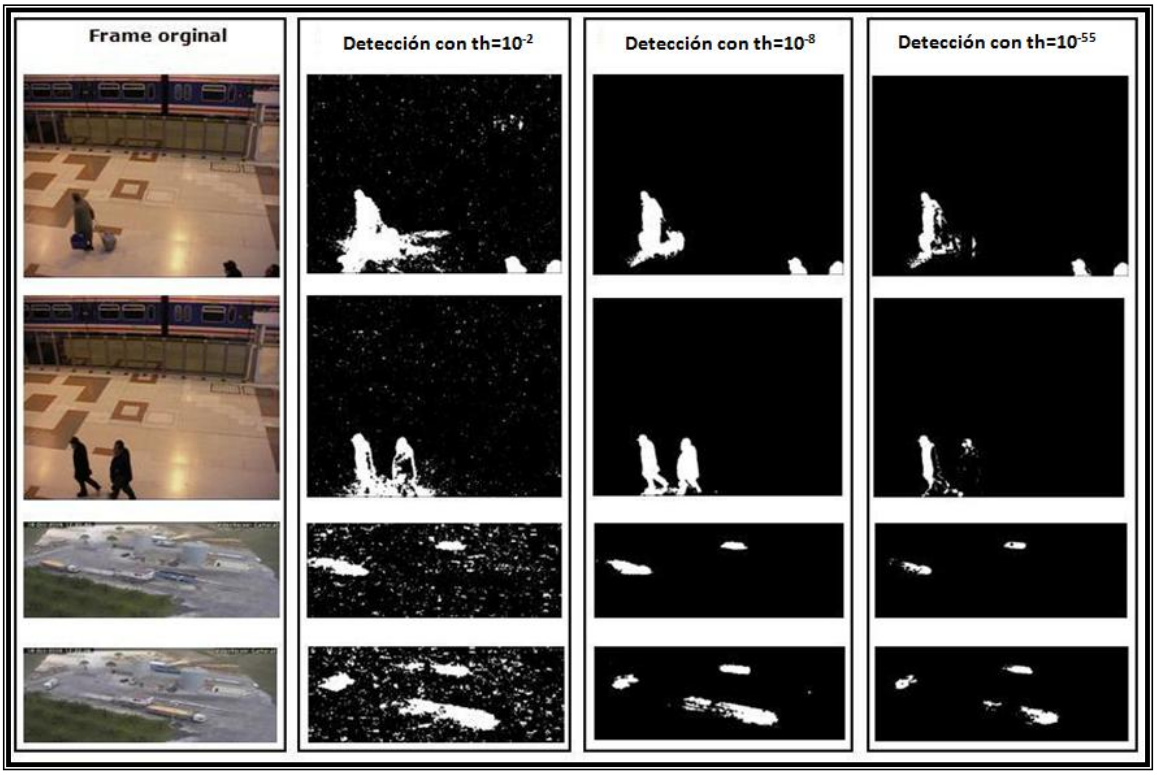


Figura 10 – Resultados obtenidos mediante KDE

Puede sorprender el uso de umbrales tan bajos, pero teniendo en cuenta la expresión utilizada para calcular la probabilidad, puede resultar comprensible puesto que es el sumatorio de N factores, donde cada uno de ellos es el producto de 3 Gaussianas, cada una de ellas de 511 posibles valores y con área normalizada. Por tanto, en el momento en que una de las componentes diste un valor superior a 6, el producto tiende rápidamente a 0.

Viendo los resultados mostrados en la figura 10 se puede deducir que el orden de magnitud del umbral de decisión se encuentra entorno a 10^{-8}, aunque puede variar un poco en función de la secuencia con la que se trabaje y de si se quiere dar prioridad a la no aparición de falsas detecciones o a la correcta detección de todo el objeto, independientemente de las falsas detecciones. Esta decisión se tomará en función de la aplicación que se quiera implementar o los filtros o post-procesados que se dispongan para realizar correcciones.

En este sentido, cabe comentar que las imágenes mostradas en la figura anterior son el resultado de un post-procesado teniendo en cuenta los píxeles vecinos, donde se eliminan falsas detecciones aisladas y se rellenan pequeños agujeros en las detecciones.

Número de muestras

Por otro lado, el número de muestras utilizadas para estimar la pdf de fondo puede variar en ciertas condiciones la calidad de la detección, puesto que cuantas más muestras se utilicen más memoria tiene el sistema.

Esta memoria puede hacer que objetos de fondo no estáticos (como por ejemplo un árbol oscilante bajo la influencia del viento), sean siempre asignados al modelo de fondo, mientras que en otro tipo de algoritmos provoquen la aparición de elevadas falsas detecciones.

Debido a que se utilizan un conjunto de Gaussianas para modelar cada una de las representaciones del píxel y no una única forma para todo el píxel se soluciona el problema anterior. Por tanto, desde ese punto de vista, cuantas más muestras sean utilizadas para realizar la estimación mejor, puesto que se podrán tener en cuenta más objetos no estáticos como fondo y eliminar así falsas detecciones.

Aún así, el uso de un número elevado de muestras tiene efectos contraproducentes: la velocidad de procesado disminuye (ya comentado anteriormente) y la absorción de objetos en fondo es más lenta. En función de la aplicación (si aparecen objetos en el fondo no estáticos o si se puede dar el caso de que objetos sean absorbidos por el fondo) será más conveniente un caso u otro.

1.5 Comparativa

En este apartado se va a comparar dos métodos detectores de primer plano, el método de S&G y el KDE, comentándose algunos resultados obtenidos y destacándose los puntos fuertes y débiles de cada uno de ellos. Para ello, obsérvese la siguiente colección de imágenes:

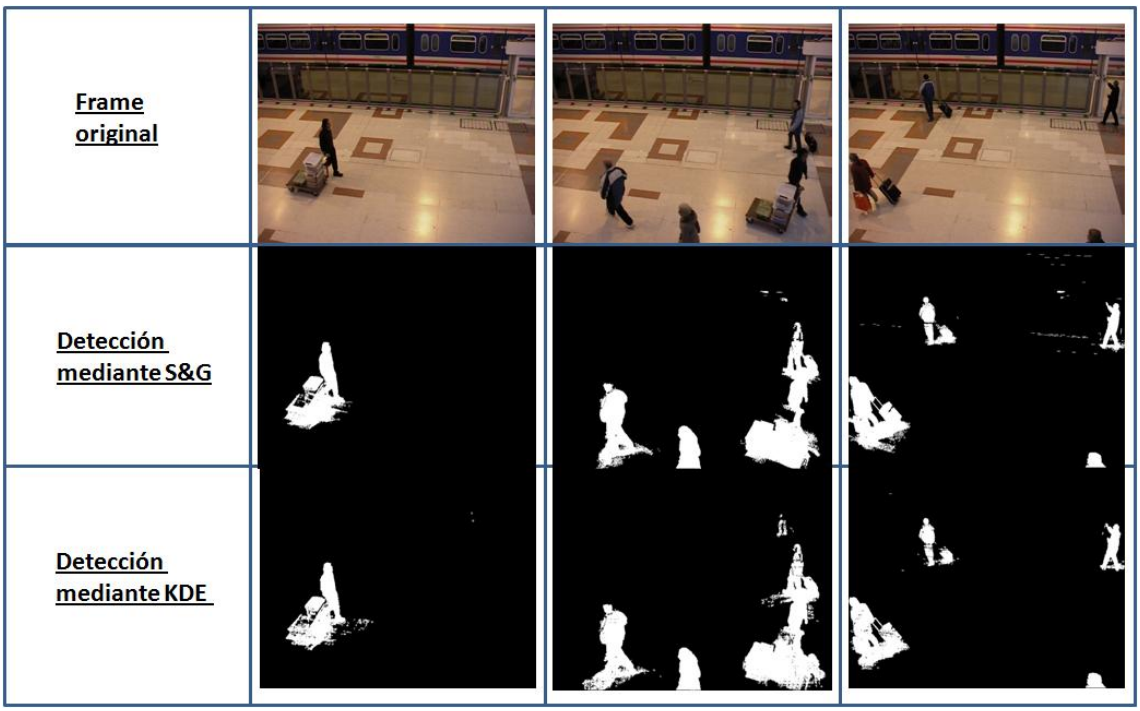


Figura 11 – Resultados comparativos Stauffer and Grimson y KDE

A primera vista, se puede notar que los resultados obtenidos mediante ambos métodos son parecidos, puesto que no se puede apreciar notables diferencias entre las máscaras obtenidas. Esto se traduce en que el sistema KDE también se puede considerar un buen detector de primer plano.

Aún así, entrando en un poco más de detalle, se pueden observar algunas diferencias. Por un lado, la aparición de falsas detecciones asociadas a las sombras de los objetos varía en ambos métodos. Pero este hecho puede solventarse utilizando un sistema corrector de sombras (los cuales serán explicados posteriormente).

Por otro lado, si se observa la última imagen mostrada en la anterior figura, se puede comprobar que el método KDE es menos susceptible a la aparición de pequeñas falsas detecciones aisladas con respecto a S&G, debido a que la memoria del sistema es mayor, por lo que pequeños cambios de luminancia pueden no ser detectados. Aún así, esta diferencia, también puede solucionarse mediante el uso de filtrados posteriores que permitan eliminar aquellas detecciones suficientemente pequeñas.

Por último, en la siguiente figura se pretende mostrar algunos resultados más, habiendo de prestar especial atención a la primera y última imagen.

Figura 12 – Resultados comparativos Stauffer and Grimson y KDE

En estas imágenes se puede observar cómo el método KDE puede ser más lento que S&G en la absorción de objetos de primer plano que de repente pasan a ser estáticos por el modelo de fondo. El motivo es que en KDE no se trabaja con un parámetro de absorción, sino que se trabaja con N Gaussianas, cada una de ellas centrada en cada una de las N muestras anteriores del píxel. De este modo, se puede considerar que un objeto ha sido absorbido por el fondo en el momento en que hay un número elevado de Gaussianas que lo modelan como parte del fondo, por lo que debe haber estado detenido durante un número de imágenes lo suficientemente elevado.

Teniendo en cuenta que el tiempo de absorción, por parte del modelo de fondo, de objetos de primer plano que pasan a ser inmóviles se rigen por distintos parámetros en ambos métodos, se puede considerar que los resultados obtenidos por ambos métodos son semejantes, con unas pequeñas diferencias que dependen de la secuencia con la que se trabaje.

2 Sistema post-procesado: Corrección de sombras

Como ya se ha comentado anteriormente, los sistemas basados en substracción de fondo (es decir, los sistemas que asignan a primer plano todos aquellos píxeles que no se corresponden con el fondo aprendido) tienen el riesgo de detectar las sombras como parte del objeto, debido a la no correspondencia con el modelo aprendido hasta el momento. Esta asignación incorrecta suele ser problemática en la mayoría de las aplicaciones prácticas en las que se utiliza este tipo de algoritmos, puesto que distorsionan las características propias de los objetos.

Debido a que las sombras tienen las características del modelo de fondo aunque con ciertos tonos más oscuros, hay distintas técnicas que utilizan esta propiedad con el fin de eliminar parcial o totalmente las falsas detecciones producidas por este fenómeno.

Mientras que algunos de estos métodos son algoritmos aplicados a la salida del sistema detector para corregir la máscara obtenida, otros se basan en realizar una transformación a la entrada del mismo detector para reducir la aparición de sombras en la detección.

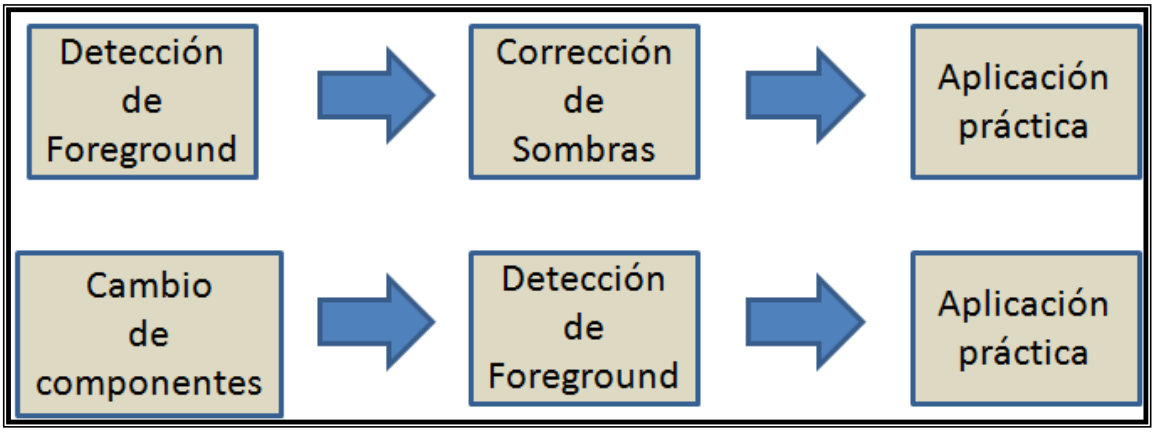


Figura 13 – Diagrama de bloques de los dos posibles correctores de sombras

En este capítulo se centrará la atención en este tipo de algoritmos, concretamente en tres de ellos, explicándose cada uno de ellos y realizando un análisis comparativo:

- Método de luminancia normalizada
- Método híbrido
- Método reflectancia

2.1 Método luminancia normalizada

Partiendo de la base de que, en procesado de imagen, todo color se descompone en 3 componentes de un espacio vectorial (normalmente R,G,B), y que cada uno de ellos puede tener 256 valores, es posible representar 256^3 colores, o lo que es lo mismo, unos 16 millones de colores.

Pero (R,G,B) no es la única manera de representar estos 16 millones de colores: cada uno de dichos colores se puede descomponer en crominancia y luminancia, entendiéndose por crominancia la componente que aporta toda la información de color, es decir, lo que permite al receptor discernir entre un color amarillo y uno rojo, motivo por el que se expresa mediante un vector de 3 componentes, dando una idea de la proporción entre las distintas componentes del espacio RGB.

En cambio, la luminancia es la parte que aporta toda la información de luminosidad, lo que se suele denominar brillo, cuya representación se realiza mediante un escalar y con la finalidad de diferenciar entre un amarillo claro y uno oscuro.

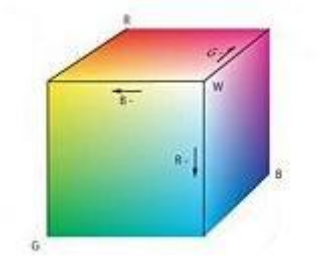


Figura 14 – Descomposición del color en las componentes R,G,B

A partir de estas premisas, A. Elgammal y R. Duraiswami proponen en [4] descomponer cada uno de los colores con los que se trabaja en luminancia y crominancia y utilizarlas como entrada al sistema con el fin de identificar la sombra del objeto como todos aquellos píxeles que tienen la misma crominancia y distinta luminancia que el fondo.

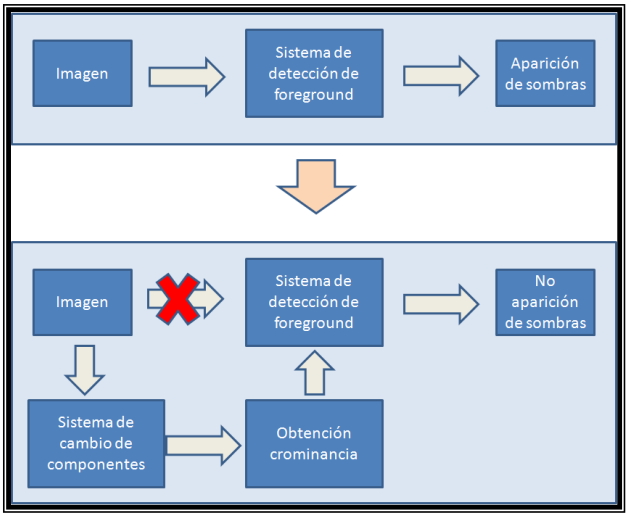


Figura 15 – Diagrama de bloques del corrector de sombras mediante luminancia normalizada

Hay distintos métodos que realizan la descomposición de un color en crominancia y luminancia, pero todas tienen en común que intentan calcular la luminancia como la cantidad de blanco contenido en el color, mientras que intentan eliminar *la componente de blanco* de cada una de las componentes de color en el caso de la crominancia.

Dadas las 3 variables de color (R,G,B), en la implementación realizada hemos definido las respectivas componentes de crominancia (r,g,b) como:

$$\left.\begin{aligned} r &= \frac{R}{R+G+B} \\ g &= \frac{G}{R+G+B} \\ b &= \frac{B}{R+G+B} \end{aligned}\right\}$$ Por lo que se puede observar que r+g+b=1

Expresión 2.1

Trabajando con estas coordenadas, el sistema es capaz de detectar objetos sin sombras debido a que es insensible a los cambios de brillo de la secuencia de entrada (no habrá diferencia entre fondo y sombras), puesto que no se tiene en cuenta la luminancia de los colores.

Pero no tener en cuenta la luminancia implica perder información, por lo que puede significar confundir dos colores distintos como el mismo (debido a que tengan la misma crominancia pero distinta luminancia). Es decir, eliminar la información de la secuencia que hace que se detecten las sombras puede empeorar la detección de los objetos en sí.

Un ejemplo del problema mencionado puede ser intentar detectar una persona que viste una camiseta blanca en un fondo gris. El resultado será la no detección de la misma debido a que los dos colores, blanco y gris, comparten la misma crominancia. Por eso, debe ser incorporada también la información de luminancia, para no confundir colores distintos como el mismo una vez hecha la conversión de componentes.

Para ello, considerando la luminancia como s=(R+G+B)/3, es posible representar cualquier color en el espacio crominancia conociendo la luminancia y cualesquiera 2 componentes de (r,g,b), puesto que la tercera estará unívocamente representada mediante la expresión r+g+b=1. Por lo que un color puede ser unívocamente identificado como, por ejemplo, (s,r,g).

Además, cabe destacar que trabajando con (r,g,b) no es posible volver al espacio (R,G,B) una vez realizada la conversión (otro hecho que justifica la no identificación unívoca entre ambos espacios), mientras que sí que lo es en el caso de trabajar con (s,r,g).

Dicho esto, siendo x_i (con i=0,…,N-1, cada una representada mediante (s_i,r_i,g_i)) el conjunto *A* de las N muestras acumuladas para modelar el fondo, y siendo x_t el valor actual del mismo píxel, representado mediante (s_t,r_t,g_t), el algoritmo consiste en:

1.- considerar **sólo** un subconjunto de muestras $B \subseteq A$, formado por todos los valores de A que cumplen que

$\alpha \leq \frac{s_t}{s_i} \leq \beta$, donde α y β son dos umbrales idénticos para todos los píxeles (en la implementación realizada, después de haber realizado un testeo, se ha empleado α=0,9 y β=1,1), y despreciando el resto de muestras, es decir

$$B = \left\{ x_j \middle| x_j \in A \wedge \alpha \leq \frac{s_t}{s_i} \leq \beta \right\}$$ Expresión 2.2

2.- de entre todas las muestras de *B* acumuladas, calcular la probabilidad de pertenecer a fondo como:

$$prob(x_t \in background) : p(x_t) = \frac{1}{N} \sum_{m=0}^{L-1} \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{1}{2}\frac{(x_{tj}-x_{mj})^2}{\sigma_j^2}}$$ Expresión 2.3

siendo *L* el número de Gaussianas empleadas (las que están centradas en las muestras contenidas en el conjunto *B*) y *d* las dos componentes de crominancia utilizadas. En este sentido, cuando B sea cero, se asignará la probabilidad cero (se asignará a primer plano).

Para entender este cálculo, hay que recordar que la luminancia es la componente de color que provoca la detección de las sombras, motivo por el que no es incorporada en el cálculo de la probabilidad (simplemente es el producto de las dos Gaussianas de la crominancia). En cuanto al punto 1, decir que es el criterio que se ha optado para evitar utilizar crominancias pertenecientes a muestras de color distinto a x_t, para evitar no detectar objetos de la misma crominancia que el fondo de la secuencia (notar la explicación realizada en el ejemplo anterior donde se ha justificado que no se detectaría un objeto blanco en un fondo gris).

Resultados

Se ha realizado una serie de pruebas con distintas secuencias con el fin de comprobar la mejora aportada por el sistema. A continuación se mostrarán y se comentarán algunos ejemplos ilustrativos.

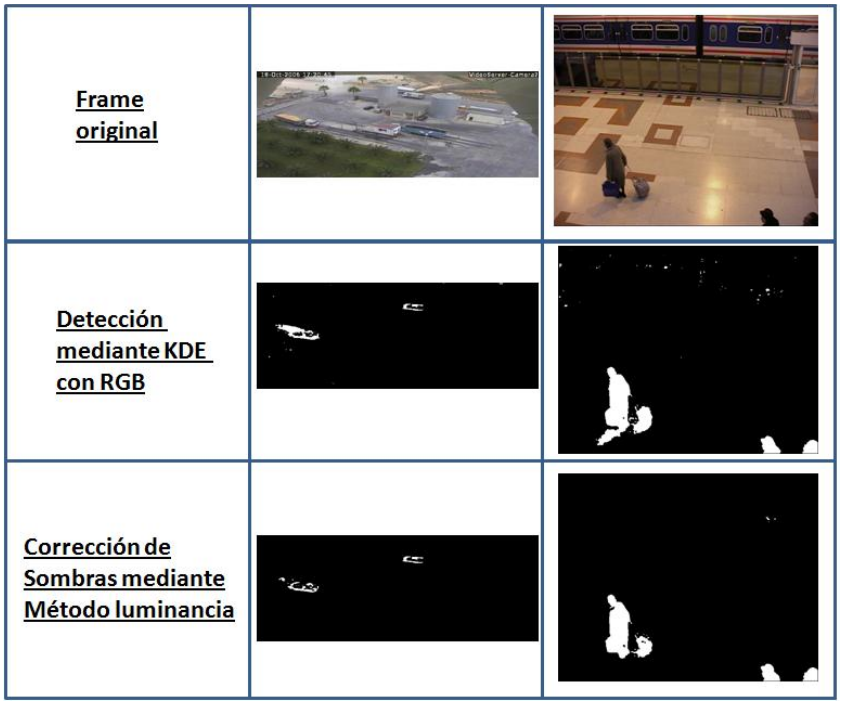


Figura 16 – Resultados método luminancia normalizada

En estos dos ejemplos se puede observar un inconveniente del método corrector. Por un lado, si nos fijamos en el segundo fotograma, se puede observar que la corrección de sombras aporta una considerable mejora a la detección. Además de eliminar parte de falsas detecciones aisladas (poco importantes, puesto que mediante un filtro de apertura aplicado a la máscara también pueden ser eliminadas), también elimina gran parte de las sombras aparecidas. En este sentido, hay que destacar que ya se ha elegido una imagen donde se pudiera notar la mejora, pero no siempre la mejora es tan notable.

En cambio, si nos fijamos en la primera imagen se puede observar como el trabajar con estas componentes puede implicar no detectar parte del objeto. Éste es un riesgo que se corre cuando el método corrector de sombras consiste en cambiar las componentes de color con las que trabajar, que el para corregir la aparición de sombras en la detección suprimamos la detección del objeto en sí. Por tanto, este no es un buen método corrector de sombras, puesto que puede significar perder parte de la información del objeto a detectar.

2.2 Método híbrido

Este método tiene la particularidad de que corrige la detección de primer plano obtenida mediante cualquier algoritmo. Es decir, el método propuesto por M.Pardàs y J.L. Landabaso en [5] no hace ninguna transformación de componentes para ser menos sensible a la aparición de sombras y luego utilizarlas como entrada al sistema, sino que valida si la detección realizada con el anterior método es correcta o susceptible de corrección.

Por tanto, el bloque corrector de sombras se sitúa a la salida de cualquier bloque detector de objetos de primer plano, donde se determina si la detección obtenida en el primer bloque corresponde efectivamente con el objeto o con su sombra. De este modo, el diagrama de bloques es:

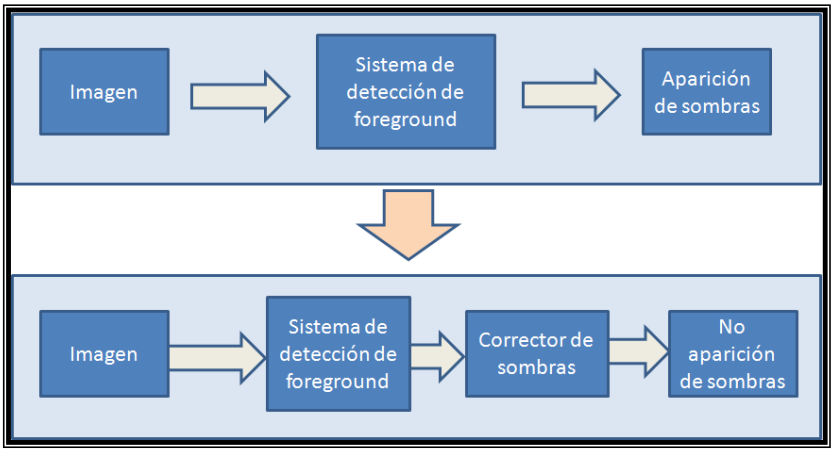
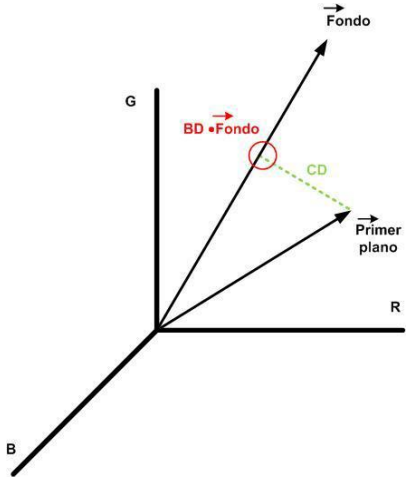


Figura 17 – Diagrama de bloques del método híbrido

Internamente, este bloque corrector consta de dos partes: una primera de comprobación y una segunda de corrección.

Por un lado, la comprobación que se realiza consiste en calcular la distorsión de color y brillo de cada píxel detectado como primer plano y determinar si se encuentra dentro de un cierto margen para determinar si es sombra u objeto. En [6] Horpraset, Harwood y Davis definen la Distorsión de Color (CD) como la distancia ortogonal entre el color esperado del fondo y la línea de cromaticidad observada, mientras que se entiende por Distorsión de Brillo (BD) el escalar que acerca el valor observado a la línea de cromaticidad esperada del fondo. Ambas medidas son mostradas en la siguiente figura, siendo calculadas mediante las expresiones:



$$BD = \min_{\alpha} \left\| \vec{p}rimer\_plano - \alpha \cdot \vec{f}ondo \right\|^2$$

$$CD = \left\| \vec{p}rimer\_plano - BD \cdot \vec{F}ondo \right\|$$

Expresión 2.4

Los valores de distorsión de brillo (BD) por encima de 1 corresponden a un fondo iluminado, mientras que el primer plano es más oscuro cuando BD está por debajo de 1.

De este modo, se define un conjunto de umbrales para validar la clasificación de píxel de primer plano, y obtener primer plano, brillo o sombra. Los criterios de decisión utilizados son:

De entre los píxeles asignados a primer plano, si CD < 10 entonces:

- Si 0,5 < BD < 1 $\Longrightarrow$ **SOMBRA**
- Si 1 < BD < 1,25 $\Longrightarrow$ **BRILLO**
- En otro caso $\Longrightarrow$ **PRIMER PLANO**

Pero trabajar con este método implica generar anulaciones en la detección de primer plano en situaciones en las que los objetos a detectar tengan colores similares a las regiones identificadas como fondo sombreado. Es aquí donde interviene la segunda parte, la de corrección.

Denominaremos *máscara original* la que se obtiene a partir del sistema detector primer plano sin realizar ninguna corrección de sombras, y *máscara corregida* a la que se obtiene a la salida del anterior bloque corrector de sombras. Teniendo presente estos términos, esta parte consiste en aplicar una dilatación a la máscara corregida y aplicar la operación intersección con la máscara original, con el fin de mantener la detección del objeto de la máscara original y corregir una parte de las sombras que aparecen corregidas en la máscara dilatada. Por tanto, el diagrama de bloques del sistema será:

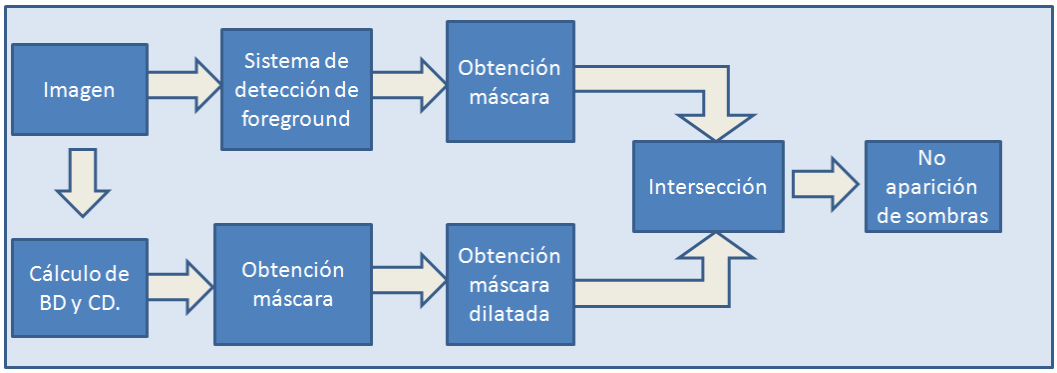


Figura 18 – Funcionamiento método híbrido

En la siguiente figura se van a mostrar algunos de los resultados obtenidos mediante este método de corrección de sombras. Como se puede observar en los tres ejemplos, en todos los casos se produce una pequeña corrección de las falsas detecciones aparecidas como consecuencia de las sombras.

El ejemplo más claro lo podemos encontrar en la imagen del medio, donde la persona ubicada en el marco inferior izquierdo aparece nítidamente detectada mediante el método híbrido, mientras que en el método KDE aparece una gran mancha en el suelo correspondiente a la sombra de la persona. Aún así, en el resto de imágenes sucede el mismo fenómeno.

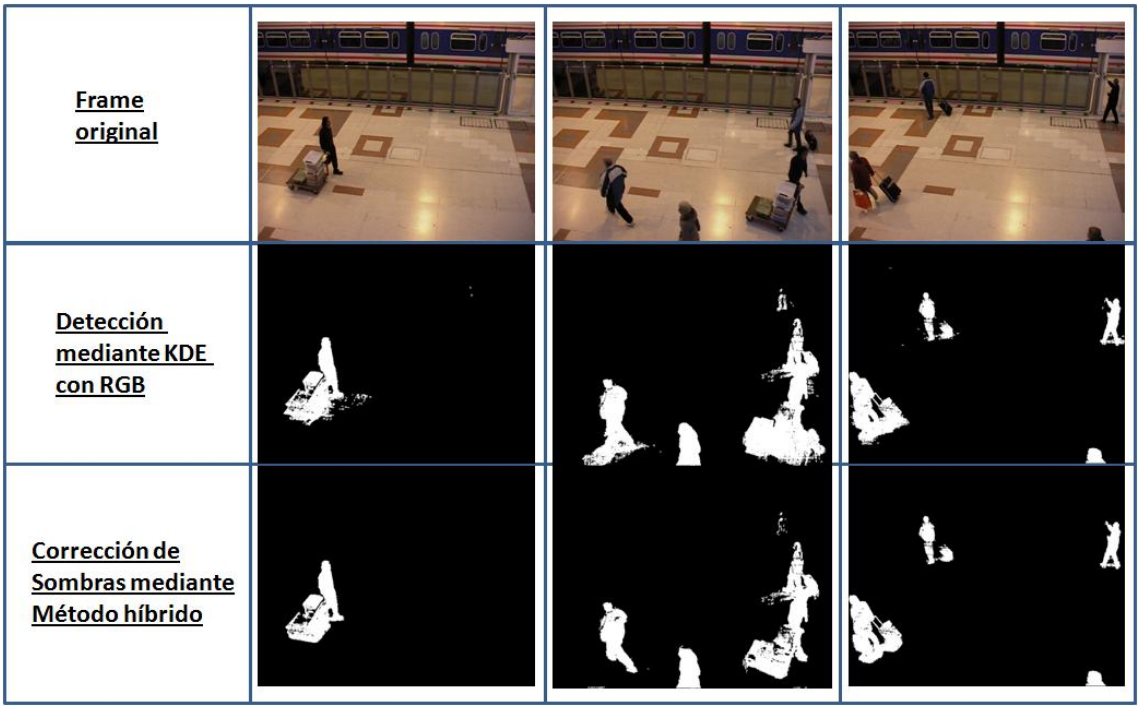


Figura 19 – Resultados comparativos KDE y KDE + método híbrido

Hay que hacer hincapié otra vez en que constantemente se utiliza esta secuencia para realizar los test de corrección de sombras puesto que es una secuencia donde aparecen una gran cantidad de sombras, por lo que es una secuencia muy ilustrativa. Aún así, no sólo ha sido utilizada en ésta, prueba de ello puede observarse también la siguiente figura.

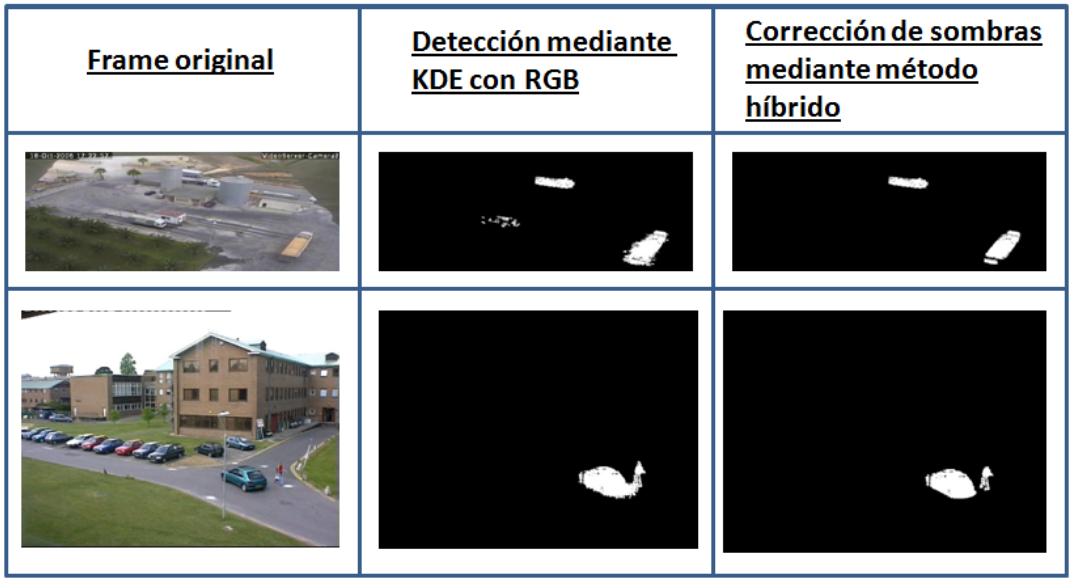


Figura 19 – Resultados comparativos KDE y KDE + método híbrido

El segundo ejemplo ilustra perfectamente la importancia de corregir correctamente las falsas detecciones de las sombras, puesto que los dos objetos detectados, persona y coche, aparecen conexos

por la sombra del segundo, pudiéndose considerar como un único objeto. En cambio, al corregir las sombras aparecidas, los objetos aparecen disjuntos, por lo que las características de ambos objetos no serán mezclados en uno sólo, y se podrá realizar el seguimiento de ambos correctamente por separado.

Así que se puede concluir que este método es muy correcto para realizar las correcciones de sombras que aparecen en la mayoría de métodos de detección de primer plano, además de la gran velocidad de procesado que tiene con respecto a otros métodos.

2.3 Método reflectancia

Teniendo presente que la imagen de un objeto resulta de proyectar luz sobre él, existe otra descomposición de una imagen basada en esta idea. Se define luminancia como el flujo de luz que incide en la superficie del objeto, y reflectancia como la proporción de flujo de luz reflejada. Por tanto, la luminancia dependerá únicamente de la luz ambiente, mientras que la reflectancia dependerá además de las características del objeto (color, textura,…).

En la práctica, dada una imagen, basta con aplicar un filtrado paso-bajo para encontrar la luminancia, mientras que no es posible obtener la reflectancia mediante ningún filtrado puesto que contiene altas y bajas componentes frecuenciales, tal y como proponen Lou, Yang, Hu y Tan en [7]. Esta consideración proviene de la idea de que, en la mayoría de aplicaciones, se observa que la luminancia tiene un suave degradado a lo largo del espacio por la propia naturaleza de la luz, mientras que la luz reflejada en los objetos no. Para entenderlo mejor, obsérvese el siguiente ejemplo ilustrativo.

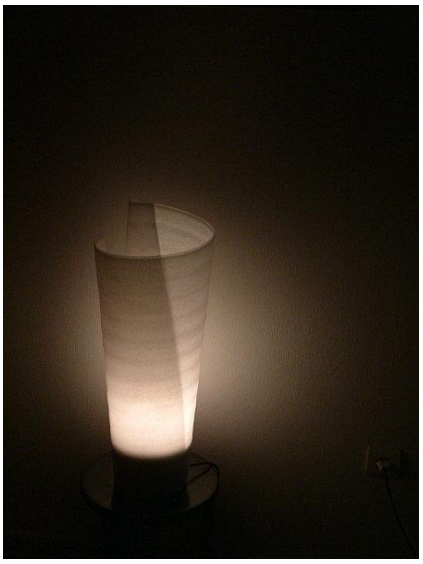


Figura 20 – Ejemplo luminancia

En la foto, se puede observar alrededor de la lámpara el suave degradado natural de la luz emitida, la cual corresponde a la luminancia puesto que depende única y exclusivamente del foco, y constituye pues la luz incidente o ambiente de la escena.

En cambio, la luz reflejada en un objeto no tiene por qué ser suave, puesto que no depende únicamente de la luz incidente sino también de las características geométricas y del material del objeto. Un ejemplo es que todo objeto está delimitado por un contorno, característica que permite diferenciarlo del fondo u otros objetos y que, en general, tiene componentes de alta frecuencia espacial en alguna de sus componentes (R,G,B) debido al cambio brusco de color entre el objeto y el fondo.

Corrector de sombras

Bajo las premisas anteriores, y considerando la iluminación incidente blanca, Zun, Feng y Tan proponen en [8] modelar la intensidad de luz captada por la cámara como

$$I(j) = e(j)\left[m_b(\vec{n},\vec{s})\cdot k_c(j) + m_s(\vec{n},\vec{s},\vec{v})\cdot f(j) \right] \qquad \text{Expresión 2.5}$$

donde *e* representa la iluminación ambiente de la secuencia, $m_s(\vec{n},\vec{s},\vec{v})$ y $m_b(\vec{n},\vec{s})$ denotan las características geométricas del cuerpo y de la superficie de reflexión del objeto, *kc* es una expresión que depende del sensor, *f* denota el parámetro de reflexión especular y *j* es el índice del píxel.

Observando la expresión anterior, es fácil ver que se puede simplificar como el producto de dos factores que representan la luz incidente y reflejada respectivamente, es decir, en luminancia y reflectancia:

$$I(j)=l(j)\cdot r(j) \qquad \text{Expresión 2.6}$$

De esta forma, el objetivo del sistema es separar la reflectancia del resto de la imagen y trabajar sólo con ella, puesto que es la única componente que nos aporta información de los objetos. De esta manera, el diagrama de bloques sería el siguiente:

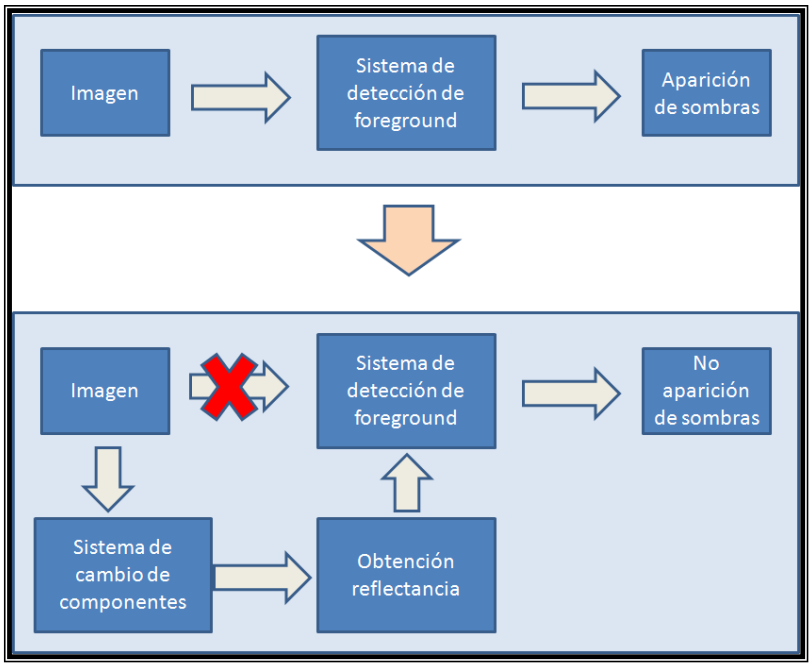


Figura 21 – Diagrama de bloques método reflectancia

La importancia de trabajar en estas componentes reside en que una sombra no es más que el mismo fondo de la secuencia pero con una luz incidente de menor intensidad, o lo que es lo mismo, menor luminancia. Por eso, el objetivo de este sistema es trabajar con la reflectancia, para poder obtener así una secuencia de entrada libre de sombras (sin degradado de luz incidente).

Para realizar dicha separación, se recurre a la definición de luminancia y reflectancia, aplicando un filtro paso-bajo y paso-alto respectivamente, siguiendo el siguiente esquema:

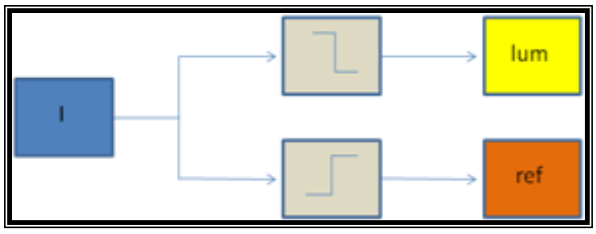


Figura 22 – Descomposición frecuencial luminancia/reflectancia

Pero el filtrado no puede aplicarse directamente sobre *I(j)=l(j)·r(j)*, puesto que la transformada de Fourier de un producto no es el producto de las transformadas sino la convolución:

$$TF\{l \cdot r\} = TF\{l\} * TF\{r\}$$

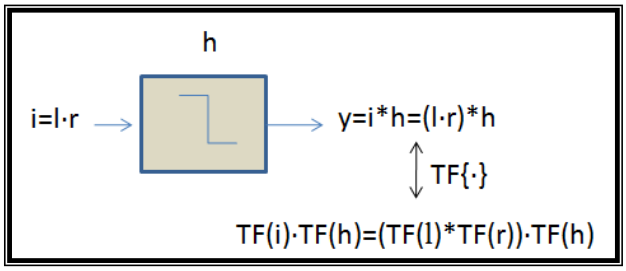


Figura 23 – Salida del filtro paso bajo con entrada lineal

Para solucionarlo, se trabaja con los logaritmos de las expresiones anteriores, con el fin de convertir los productos en sumas y, por tanto, convertir en lineal la expresión de entrada al sistema. De este modo, la salida del sistema sí que tiene relación con la luminancia[2], puesto que se trata del logaritmo neperiano de la misma, es decir:

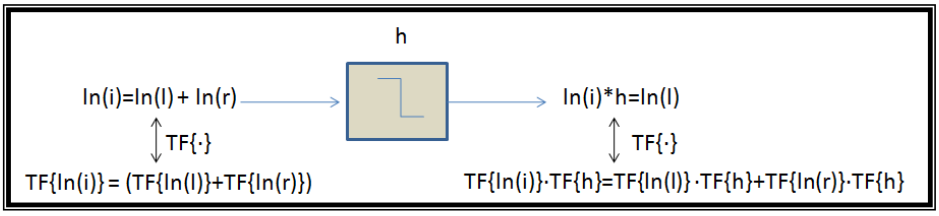


Figura 24 – Salida del filtro paso bajo con entrada logarítmica

Cabe destacar que a la salida de este filtro paso-bajo no tenemos la reflectancia sino la luminancia, por lo que la obtendremos como la diferencia entre la imagen original y la luminancia resultante. Éste es el sistema conocido como filtro homomórfico, y el esquema completo es el siguiente:

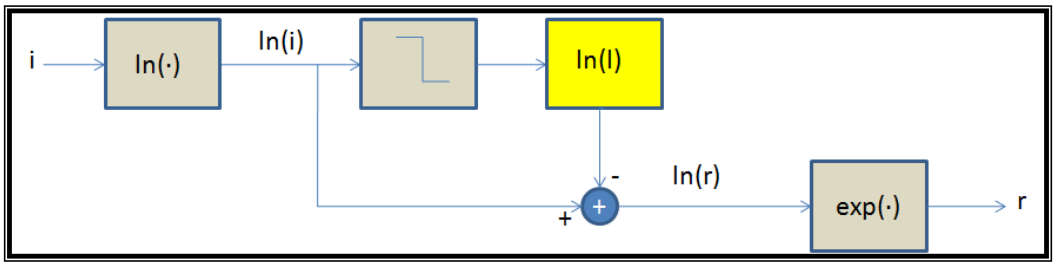


Figura 25 – Esquema filtro homomórfico

En la siguiente figura, se pueden observar dos ejemplos de imágenes originales y las reflectancias obtenidas. En ambos casos, podemos comprobar que eliminar la componente de baja frecuencia

[2] Esta explicación puede aplicarse también a la reflectancia

espacial de la secuencia supone eliminar los cambios de luminancia, como pueden ser las sombras o los aumentos de brillo.

Prueba de que se ha realizado un filtrado paso-alto es que ha disminuido considerablemente el rango dinámico de la imagen (no es que se represente la imagen en blanco y negro, sino que los valores de todas las componentes son similares) y que, por otro lado, ha aumentado el contraste en el contorno de los objetos, debido a la alta componente frecuencial que contienen.

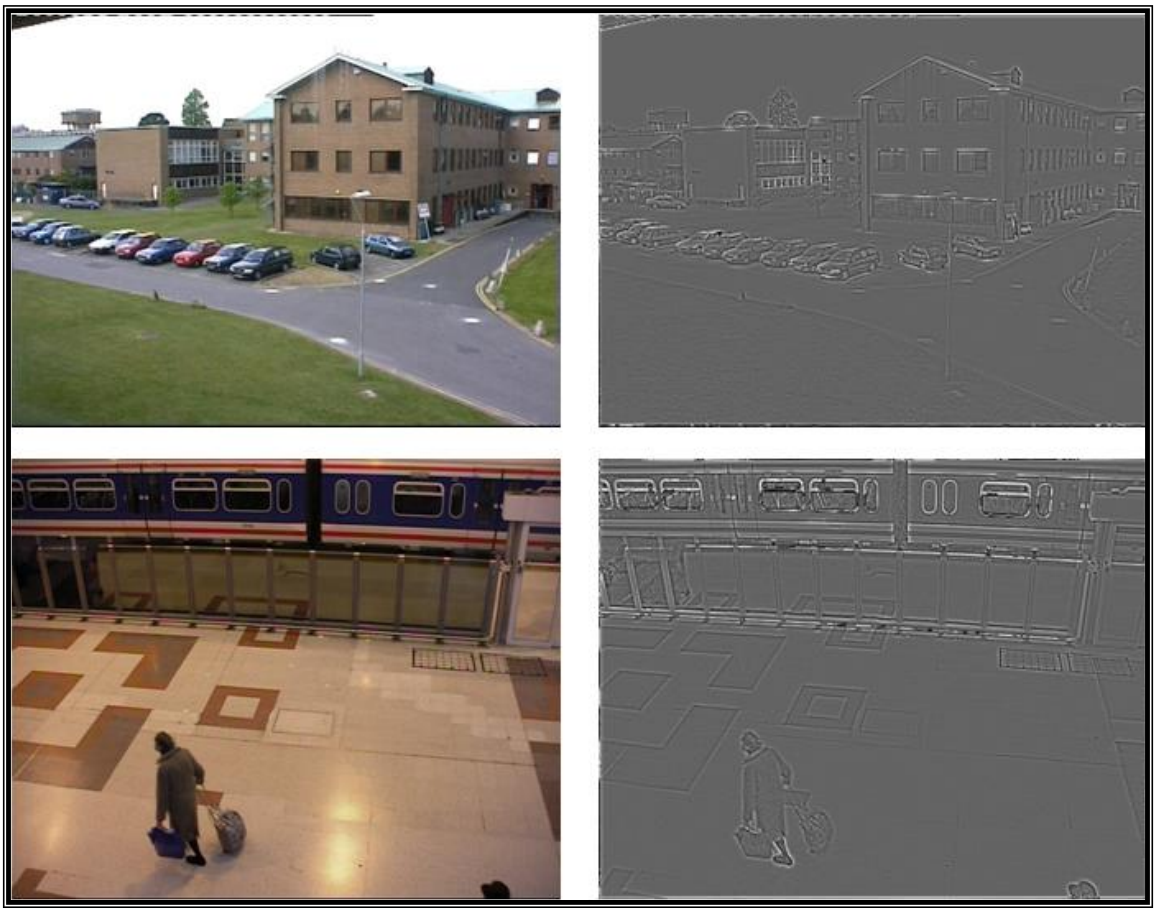


Figura 26 – Representación de reflectancias de imágenes

Implementación

Es importante mencionar algunos detalles de la implementación desarrollada en el proyecto, puesto que son básicos para entender la estructura de pruebas que se ha realizado y los resultados obtenidos.

Por un lado, se ha optado por utilizar un filtro homomórfico con una Gaussiana 2D de hasta r vecinos de influencia como filtro paso-bajo, donde r es un parámetro de entrada al sistema, y se corresponde con la semi-longitud positiva no nula de una de las dimensiones de la ventana, es decir, *el radio de influencia de la misma*.

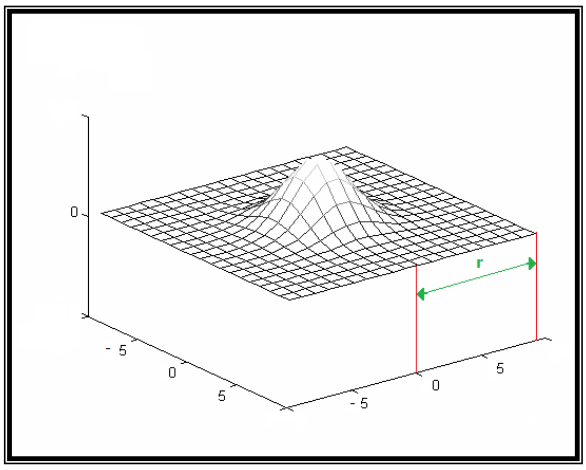


Figura 27 – Gaussiana 2D con parámetro r

De este modo, es posible testear el método con distintos tamaños de ventana y asegurar así la optimización del resultado para cada secuencia de trabajo, descartando la posibilidad de que un mal filtrado afecte negativamente a la posterior detección de objetos. En este sentido, hay que remarcar la importancia de tener valores no nulos en toda la ventana de filtrado sin llegar a distorsionar la Gaussiana en ningún caso.

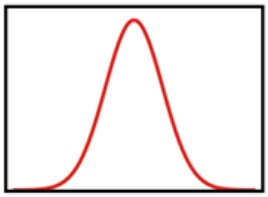
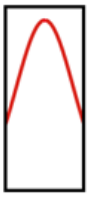
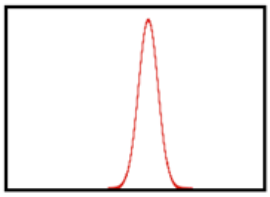
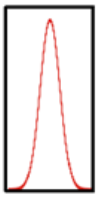
	r=7	r=3
σ grande	 filtro óptimo	 r=3: filtro distorsionado, Gaussiana incompleta
σ pequeña	 r=7: filtro con mismo efecto que r=3(resto de valores nulos)	 filtro óptimo

Figura 28 – Gaussiana enventanada en función del parámetro r

Este hecho implica que la implementación de la Gaussiana esté condicionada al tamaño de la ventana. Para hacerlo, se ha optado por asegurar que el filtro contenga siempre, independientemente de r, el 99% de la función de densidad, es decir, se ha impuesto el criterio $r=3\sigma$, criterio matemático que asegura que entre $\mu-3\sigma$ y $\mu+3\sigma$ se encuentran el 99% de las muestras.

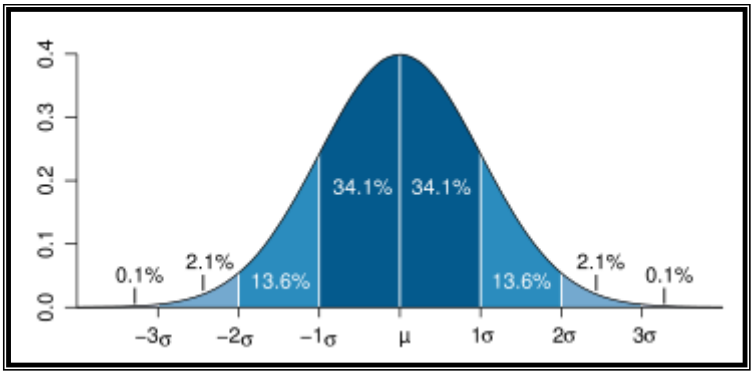


Figura 29 – Representación de probabilidades en función de σ

Por otro lado, dada una imagen de NxM píxeles, el primer píxel de la imagen en el que se puede centrar la Gaussiana para calcular la convolución es en el píxel (r,r), y el último el (N-r, M-r), debido a la no causalidad del filtro Gaussiano. Por tanto, la reflectancia no puede ser calculada en todos los píxeles con componentes (i,j), con i,j = 0,..,r ó j=M-r+1,…, M-1 ó i=N-r+1,…,N-1 (el marco exterior de la imagen de tamaño r píxeles).

Para solucionarlo, se ha optado por ubicar la imagen dentro un marco negro de r píxeles. De esta forma, se podrá calcular la reflectancia de todos los píxeles de la imagen y, además, sin distorsión, puesto que al ser el marco negro, estos valores no serán utilizados en la convolución. Para entender mejor esta idea, obsérvese el siguiente esquema:

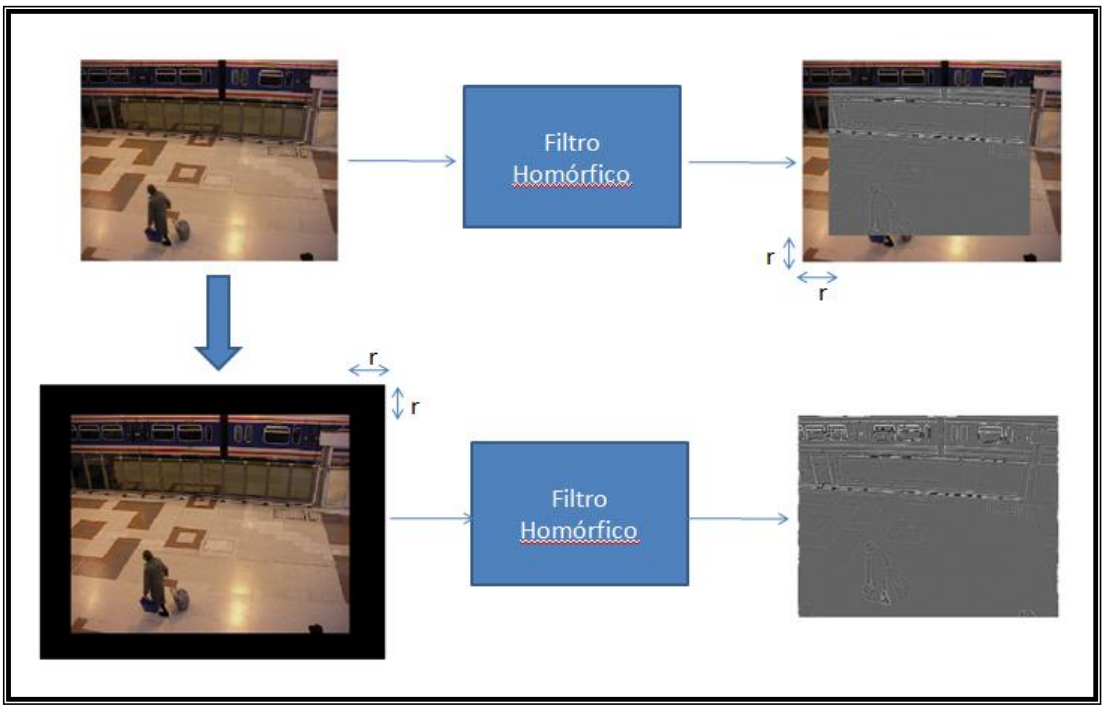


Figura 30 – Adaptación de la imagen para la correcta obtención de la reflectancia

Resultados en función de r

Como ya se ha mencionado anteriormente, se ha introducido la posibilidad de controlar r mediante un parámetro con el fin de asegurar la optimización del filtro paso-alto para cada secuencia. Esto nos ha

permitido obtener distintas reflectancias de una misma secuencia, pudiendo analizar la calidad de la detección de foreground obtenida en cada caso. En la siguiente figura puede observarse un caso particular del tipo de test realizado.

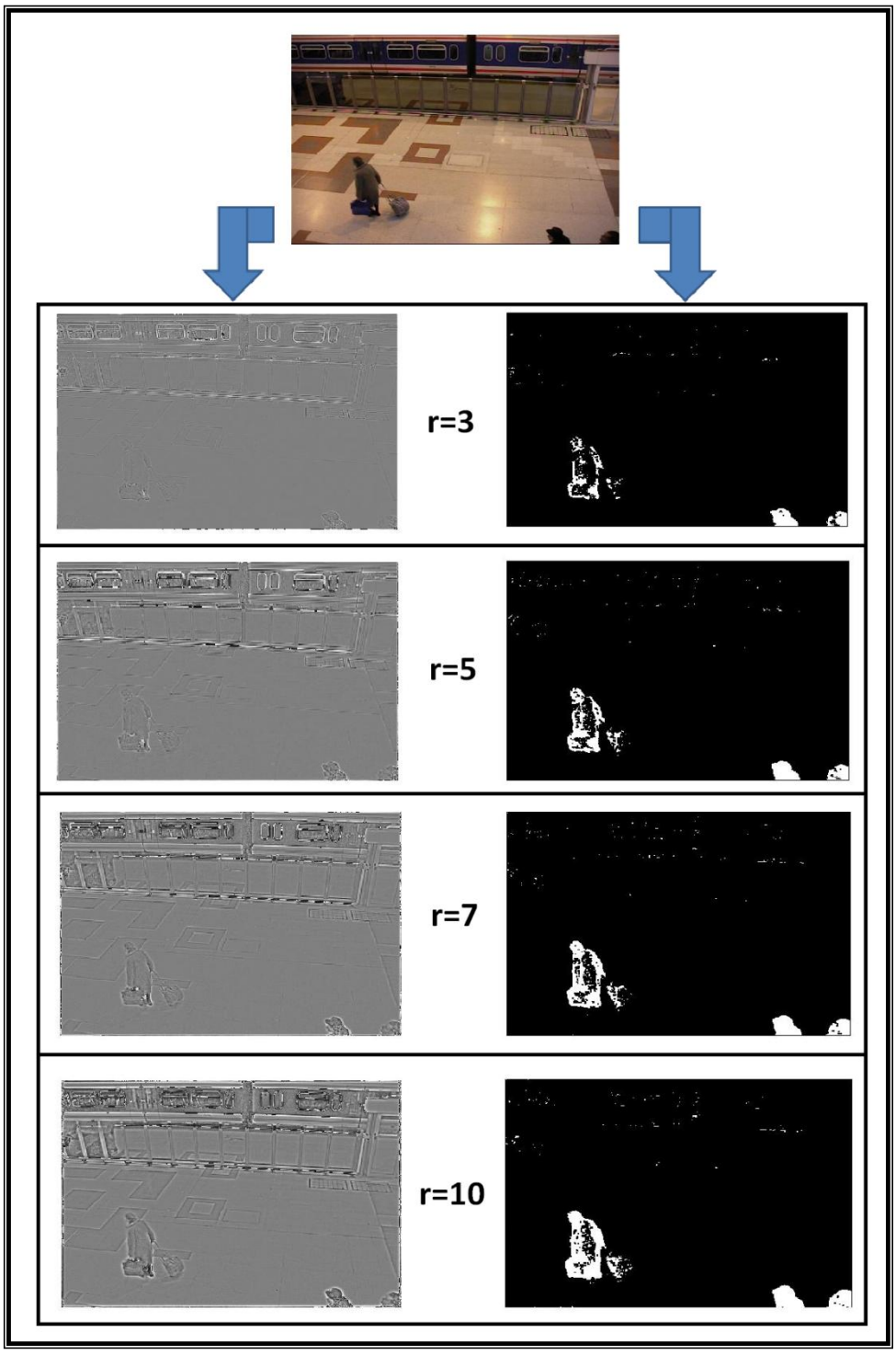


Figura 31 – Influencia en la detección del parámetro r

Cabe destacar que, a pesar de que la figura representa el caso particular de una secuencia, el método se comporta de forma similar con el resto de secuencias con las que ha sido probado, así que se analizará este caso particular pudiéndose aplicar a cualquier otra secuencia, debido a que es una secuencia donde

aparecen una gran cantidad de falsas detecciones asociadas a las sombras y se puede apreciar fácilmente la corrección realizada.

Dicho esto, para empezar se puede comprobar que, efectivamente, a medida que aumenta r también lo hace la resolución del filtro paso-alto, puesto que se remarcan más los contornos de los objetos. En cuanto a la detección, existe una relación directa entre el valor usado para el filtro paso-alto y la calidad de la detección (mejora con el aumento de r), aunque la diferencia entre la detección usando una r=7 y otra de r=10 no es muy elevada. En cuanto a las falsas detecciones obtenidas, comentar que no son objeto de preocupación debido al pequeño tamaño que tienen con respecto a los objetos útiles a detectar, por lo que mediante un filtro de apertura pueden ser eliminados cómodamente sin deterioros en los objetos.

En este sentido, es posible que la detección obtenida con una r mayor fuera ligeramente mejor, pero, debido a que la luminancia debe tener unas variaciones mínimas, no tiene mucho sentido el hecho de filtrar con una r mayor que 10. Dicho de otra forma, hacer que r tienda al tamaño de la imagen implica que la reflectancia tienda a la imagen original, por lo que volverían a aparecer las sombras.

Esto último, junto con el hecho de que a medida que aumentamos el número de vecinos utilizados para calcular la reflectancia aumenta considerablemente el coste computacional (disminuye la velocidad de procesado), hace que nos decantemos por trabajar con una reflectancia de 7 vecinos, puesto que se puede considerar el punto óptimo entre buena detección y coste computacional. Por tanto, a partir de ahora, las pruebas que se vayan a presentar habrán sido obtenidas con r=7.

Aún así, en ningún caso se obtiene una detección correcta, puesto que el objeto a detectar aparece con no detecciones en su interior, produciendo un efecto de objeto agujereado. Esto se debe a que el filtro homomórfico, como ya se ha comentado, elimina las componentes espaciales de baja frecuencia de la imagen, por lo que elimina la luminancia y una parte de la reflectancia, información característica necesaria para detectar correctamente el objeto.

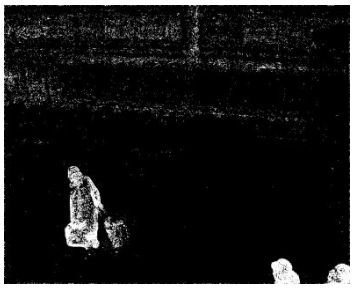


Figura 32 – Detección mediante reflectancia

Para solucionarlo, se optó por introducir en el cálculo de la probabilidad de primer plano información de color, con el fin de no confundir distintos colores una vez realizado el filtrado. Para ello, fue necesario realizar una serie de pruebas que consistieron en detectar primer plano con cada una de las componentes por separado (cada una de las componentes de color y luminancia), y decidir así qué tipo de información incluir. Los resultados obtenidos fueron los siguientes.

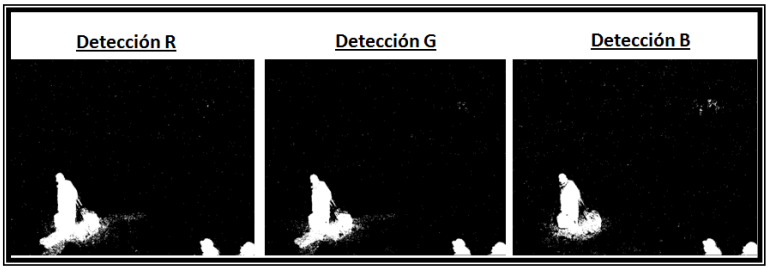


Figura 33 – Detección trabajando con una única componente de color en la entrada

En la figura se puede observar que la detección de primer plano obtenida depende ligeramente de la componente que se elije para hacer el cálculo de probabilidades (la detección con B es ligeramente mejor a las otras dos). Pero esta dependencia en la calidad no es exclusiva de la componente usada, sino que es función de la secuencia, es decir, a pesar de que en esta secuencia el resultado mejora si se trabaja con la componente B con respecto a si se hace con R o con G, puede ser que en otra secuencia sea mejor trabajar con R. No parece, pues, una buena elección incorporar una única componente (R,G,B) en el cálculo, puesto que la elección será buena o mala en función de la secuencia.

La elección fue incorporar la luminancia, entendiéndose por luminancia como la cantidad de blanco que contiene cada píxel, calculada como s=(R+G+B)/3 tal y como se explicó en el método de crominancia:

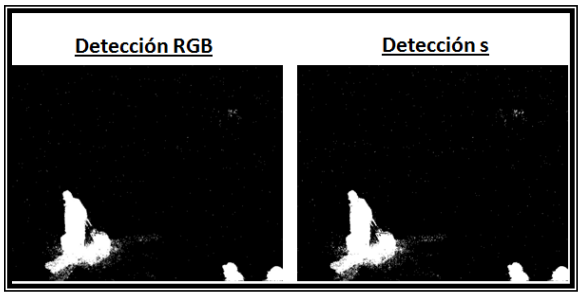


Figura 34 – Comparación de la detección usando luminancia (s) y usando las componentes R,G,B

Se obtiene la misma detección de primer plano modelando el fondo a nivel de píxel con (R,G,B) que trabajando con la luminancia del mismo píxel, es decir *s*, puesto que ésta recoge información de todas las componentes por igual. De esta forma, modelando cada píxel mediante la luminancia y la reflectancia de la misma, el sistema es capaz de eliminar las no detecciones del interior de los objetos, a pesar de que también aumentan las falsas detecciones y que no se corrige tanto la aparición de sombras.

Por tanto, la mejora aportada en la detección de primer plano al modelar el píxel mediante las tres componentes de reflectancia y la de luminancia se puede observar en la siguiente figura:

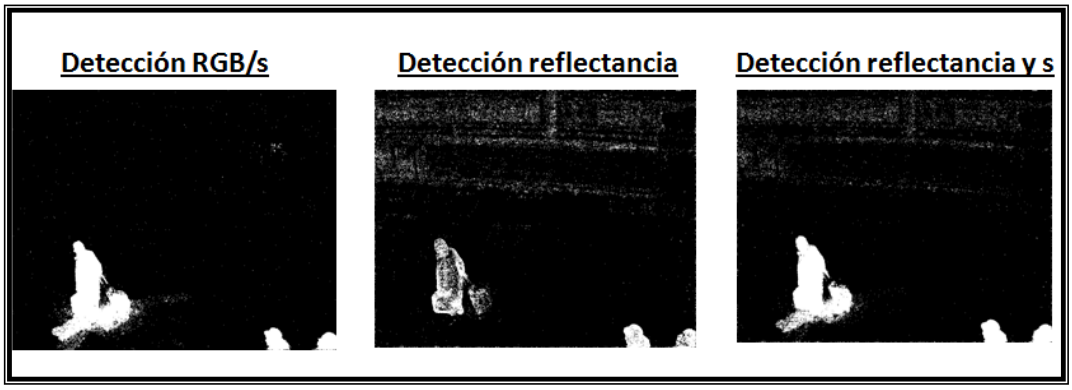


Figura 35 – Comparativa entre la detección original, la detección con la reflectancia y la combinación de ambas

Como puede observarse, trabajar con la combinanción entre reflectancia y luminancia implica eliminar las no detecciones de los objetos que se producen al trabajar sólo con la reflectancia, a pesar de que también implica la aparición otra vez de sombras.

Si se comparan la primera y la última imagen, se puede ver la mejora aportada por el método de corrección de sombras. Por tanto, se puede concluir que el método es capaz de eliminar una pequeña parte de las sombras que aparecen, aunque tiene como contrapartida que provoca la aparición de gran cantidad de falsas detecciones que, debido al pequeño tamaño de las mismas, pueden ser eliminadas mediante el uso de un filtro de apertura. Por tanto, la mejora aportada por el método no es demasiado buena.

Aún así, se quiso probar una última modificación del sistema con el fin de intentar mejorar el método de corrección de sombras. Si a la máscara de primer plano obtenida al trabajar con la reflectancia se le aplica un filtro de apertura (para eliminar las falsas detecciones de pequeño tamaño que han aparecido) y, posteriormente, se aplica una dilatación con el fin de *rellenar los agujeros* de los objetos detectados, es posible combinar esta máscara con la detección original mediante la intersección, obteniendo así el objeto completo con una pequeña parte de sombra (de forma análoga a como se explicó en el método híbrido).

Por tanto, esta modificación consiste en trabajar paralelamente con la reflectancia y las componentes de color, utilizándose la reflectancia como un método puramente corrector a la salida del detector (R,G,B), es decir, el diagrama de bloques pasaría a ser:

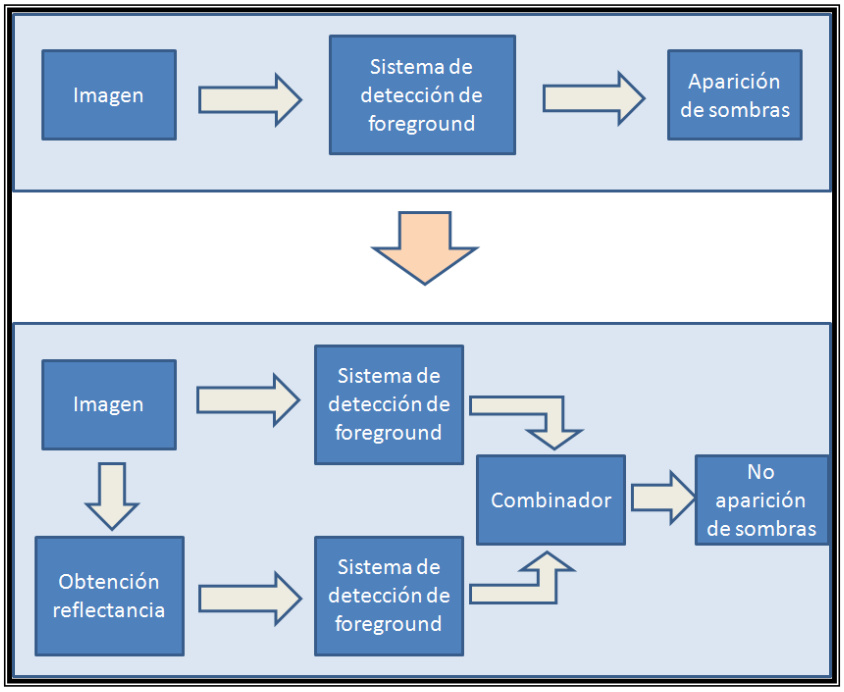


Figura 36 - Diagrama de bloques combinador detector de primer plano según R,G,B y reflectancia

Para empezar, este método tiene el inconveniente de que la velocidad de procesado es menor que la del algoritmo usado como detector de primer plano (sin corrector de sombras), puesto que el sistema tiene que realizar la detección de primer plano no sólo en (R,G,B), sino también en reflectancias. Aún así, no se ha tenido en cuenta este inconveniente y se ha analizado el resultado con el fin de comprobar si aporta alguna mejora.

En la siguiente figura puede observarse la evolución que ha sufrido el método de corrección de sombras mediante el uso de la reflectancia, por lo que al final se puede observar los resultados obtenidos mediante esta última metdología:

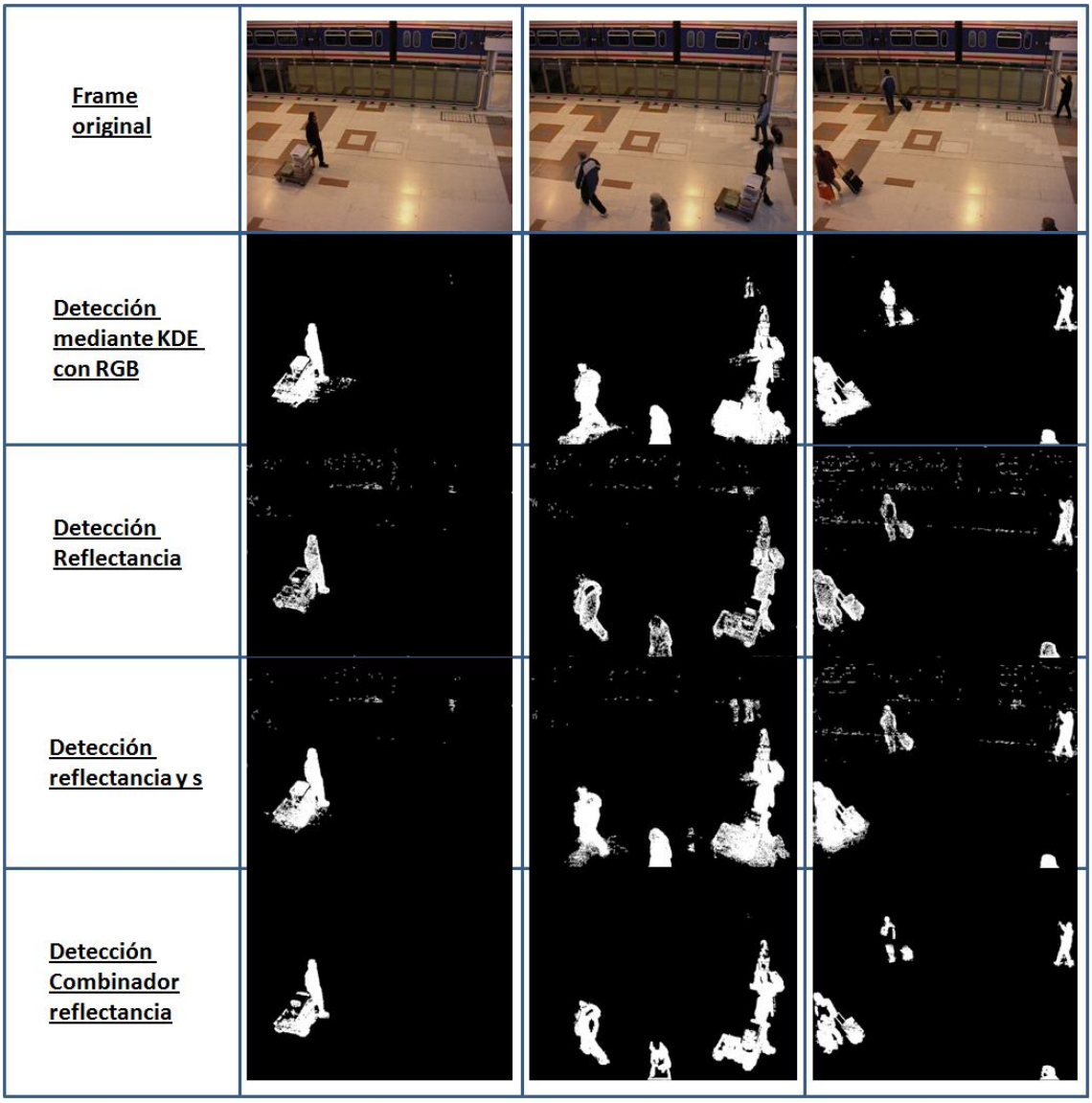


Figura 37 – Resultados comparativos entre KDE, uso de reflectancias y sus combinaciones

Como se puede observar en la figura, trabajar con esta última metodología implica corregir gran parte de las falsas detecciones asociadas a las sombras, a pesar de que tiene como contrapartida la aparición de no detecciones en el interior de los objetos. Para independizar el método corrector de sombras del sistema detector de objetos, se probó también esta corrección mediante el detector S&G, apareciendo también en este caso el mismo fenómeno (no detecciones en el interior de los objetos detectados).

De este modo, al obtener los mismos resultados mediante S&G, podemos confirmar que dicho método corrector de sombras elimina parte del interior del objeto para eliminar las sombras detectadas.

Por tanto, de entre los métodos de corrección de sombras basados en la reflectancia, el que más se acerca al objetivo de este capítulo es el último, el que combina la detección usando la reflectancia con la detección de la propia imagen, puesto que a pesar de que también elimina detecciones propias del objeto, es capaz de corregir una gran cantidad de sombras y casi no aparecen falsas detecciones.

2.4 Comparativa

Para hacer el análisis comparativo más visual, se presenta el siguiente esquema que intenta recoger las ventajas e inconvenientes de cada método de corrección de sombras:

	Ventajas	Inconvenientes
Método de Luminancia normalizada	- Muy rápido - Sin post-procesado	-Basado en cambio de componentes a entrada. - Dependencia de secuencia de entrada (no detección objetos).
Método Híbrido	-No elevado coste computacional. - Respeta detecciones originales. - Independientemente del método, se basa en un post-procesado (sin cambio de componentes).	- No eliminación por completo de las sombras (sobretodo las cercanas a los objetos).
Método de Reflectancia	- Eliminación por completo de las sombras	-Elevado coste computacional. - No respeta detecciones originales (elimina objetos). - Además del post-procesado, es necesario hacer un cambio de componentes y una doble detección.

Figura 38 – Ventajas e inconvenientes de los métodos correctores de sombras

Por tanto, teniendo presente el cuadro anterior, si se observan los distintos resultados obtenidos con los distintos métodos y los comentarios expuestos en dichos apartados, es fácil concluir que el método óptimo como corrector de sombras es el método híbrido, puesto que:

- Es un método que corrige sombras, no hace un cambio de componentes para ser menos sensible a la aparición de sombras. Esto lo hace menos sensible al cambio de secuencia de trabajo.
- La velocidad de procesado es relativamente alta (aunque no tanto como el primer método).
- Siempre respeta la detección de los objetos realizada por el sistema detector de primer plano, es decir, el hecho de corregir sombras no implica que se anule parte del objeto detectado originalmente.
- Es válido para cualquier método detector de objetos de primer plano.

3 Sistemas de detección de primer plano con regularidad espacial

Como ya se ha comentado, todos los *sistemas de detección de primer plano clásicos* tienen en común que están basados únicamente en la información de un píxel. Es decir, en las implementaciones explicadas, se trata cada uno de los píxeles de la imagen como una variable aleatoria independiente, modelando para cada uno de ellos su propio fondo en función de los valores acumulados. De modo que, una vez obtenido el modelo, se procede a decidir si el respectivo valor del píxel en la nueva imagen corresponde al fondo o a un objeto de primer plano.

Esta metodología surge como consecuencia de optimizar la velocidad de procesado en aplicaciones que tienen como premisa fundamental la ejecución en tiempo real. Aún así, esta técnica es poco intuitiva (tiene poca relación con la realidad), debido a que la percepción humana está basada en identificar los objetos no por el color de cada uno de los píxeles que lo representan sino por el color del objeto en global, la posición que ocupa, el tamaño,…

Con el fin de aumentar la eficiencia en la detección de objetos, existen técnicas que intentan aprovechar la información contenida en el entorno de cada píxel, obteniendo un modelo más complejo. Para ello, pretenden aprovechar la persistencia temporal de los objetos reales en la secuencia, es decir, que *los objetos a detectar tienden a permanecer en la misma vecindad espacial y a conservar el color coherente*.

Para hacerlo, en [9] y [10], se propone utilizar un único modelo de fondo en vez de LxM modelos independientes (correspondientes a los píxeles). Para ello, se toma un vector de características de 5 dimensiones, acumulando información de color y de posición de cada uno de los píxeles (r,g,b,x,y).

Teniendo claro que no existe ninguna relación entre la posición *x* e *y* que ocupa un píxel en la secuencia con los canales de color observados, se podrá asumir también en este caso la independencia de cada una de las componentes usadas. Por tanto, la función de densidad de probabilidad del fondo que se pretende estimar será el resultado de multiplicar cada una de sus marginales $f_{R,G,B,X,Y}$ (r,g,b,x,y)= $f_R(r)\cdot f_G(g)\cdot f_B(b)\cdot f_X(x)\cdot f_Y(y)$, pudiendo estimar cada una de ellas a partir de las muestras acumuladas. Finalmente, se clasificará cada uno de los píxeles teniendo en cuenta la pdf de fondo obtenida.

A continuación se explicarán algunos métodos de esta índole y se compararán con los métodos explicados anteriormente, con el fin de poder notar las mejoras aportadas. Concretamente, se analizará dos métodos, los cuales han sido implementados en este proyecto:

- Inclusión de información espacial en el modelo: KDE 5D
- Estimación Bayesiana mediante varios modelos (Seguimiento de objetos)

3.1 Inclusión de información espacial en el modelo: KDE 5D

A raíz de la necesidad de implementar una técnica de detección de primer plano que hiciera una clasificación en función del entorno del píxel, surgió la idea de adaptar el algoritmo KDE explicado en el apartado 1.3 (de ahora en adelante, KDE 3D) para que también incluyera información espacial, pudiendo estimar así una función de densidad de probabilidad del fondo de toda la imagen. Para ello, tal y como se explica en [9] y [10], basta con trabajar con muestras pertenecientes a R^5 de la forma z = (r,g,b,x,y).

Haremos una pequeña diferenciación entre las componentes de color y las de posición, puesto que será necesario diferenciar ambos elementos en posteriores explicaciones debido a su significado. Así que, de ahora en adelante, las componentes (R,G,B) serán denominadas componentes de *color* o *rango*, mientras que (x,y) serán denominadas componentes de *espacio* o *dominio*.

A pesar de la diferencia conceptual entre ambos elementos (componentes de color y de dominio), se trabajará con ambos elementos de la misma forma, es decir, se tratarán las componentes de dominio de la misma forma que las de color, dando una idea de la importancia o influencia que deben tener los vecinos en el píxel a clasificar (dando mayor importancia a los píxeles más cercanos). De este modo, se obtendrá un único modelo que será capaz de estimar la pdf del fondo de toda la imagen, puesto que se tendrá en cuenta la posición mediante las componentes *x* e *y*.

Funcionamiento

El funcionamiento del sistema KDE 5D, como en todo sistema de detección de primer plano, está basado en tres fases: aprendizaje, decisión y actualización.

Durante el periodo de aprendizaje, es posible obtener un único modelo del fondo de la secuencia a partir de la acumulación de N imágenes. Mediante esta acumulación se consigue ahorrar espacio en memoria, puesto que el valor de las componentes x e y quedará implícito según la posición donde se guarden los valores de color.

Una vez acumulados los N fotogramas, se procederá a obtener el modelo de fondo mediante el estimador KDE, tratándose cada una de las componentes de forma independiente a través de la expresión:

$$\hat{f}(z) = \frac{1}{N}\sum_{i=0}^{P-1}\prod_{j=1}^{d}\left(\frac{K\left(z_j - u_{i,j}\right)}{h_j}\right) \qquad \text{Expresión 3.1}$$

donde P representa el número de muestras con las que se modela el fondo (las cuales pertenecen a R^5, por lo que d=5) y h_j el ancho de banda de la función kernel.

Aún así, como ya se ha comentado, se diferenciarán las componentes de color de las de dominio, puesto que dependiendo de la componente los límites del respectivo sumando cambiarán.

Teniendo en cuenta que la secuencia de video con la que se trabaje no tiene por qué tener el mismo número de filas, de columnas que de imágenes de aprendizaje, el número de funciones kernel a utilizar para estimar la pdf del fondo puede variar en función de la componente. Por eso, la implementación realizada del algoritmo contempla una ligera transformación en la expresión anterior, que se puede traducir en:

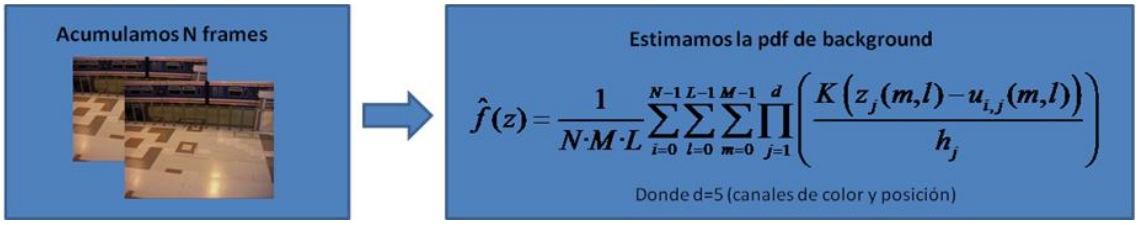


Figura 39 – Funcionamiento de KDE 5D

Donde $z_j(m,l)$ queda definida por el canal de color j-ésimo, la fila m-ésima y la columna l-ésima. En este sentido, decir que ésta será la notación usada de ahora en adelante (seguiremos diferenciando las componentes de color de las de posición).

De la misma forma a cómo sucedía en KDE 3D, la función kernel utilizada es la Gaussiana, por lo que la expresión anterior queda traducida a:

$$\hat{f}(z)=\frac{1}{N\cdot M\cdot L}\sum_{i=0}^{N-1}\sum_{l=0}^{L-1}\sum_{m=0}^{M-1}\prod_{j=1}^{d}\frac{1}{\sqrt{2\pi\sigma_j^2}}e^{-\frac{1}{2}\frac{\left(z_j(m,l)-u_{i,j}(m,l)\right)^2}{\sigma_j^2}} \qquad \text{Expresión 3.2}$$

Decisión entre fondo y primer plano

Una vez modelado el fondo de la secuencia, se procederá a la detección de objetos de primer plano. Para ello, análogamente a su antecesor, es necesario calcular la probabilidad que el píxel pertenezca a fondo teniéndose en cuenta también los valores anteriores de todo el entorno, por lo que:

$$p(z\in bg)=\frac{1}{N\cdot M\cdot L}\sum_{m=0}^{M-1}\sum_{l=0}^{L-1}\sum_{i=0}^{N-1}\prod_{j=1}^{d}\frac{1}{\sqrt{2\pi\sigma_j^2}}e^{-\frac{1}{2}\frac{\left(z_j(m,l)-u_{i,j}(m,l)\right)^2}{\sigma_j^2}} \qquad \text{Expresión 3.3}$$

donde L y M son el número de filas y columnas del fotograma respectivamente, utilizándose un umbral de decisión como criterio clasificador entre fondo y primer plano (el mismo para todos los píxeles),

si p(z) < th (umbral) $\Longrightarrow$ se decide primer plano
si p(z) > th (umbral) $\Longrightarrow$ se decide fondo

Expresión 3.4

Interpretación del sistema

Dejando de lado la definición estrictamente matemática de KDE 5D (como ya se ha explicado en 1.3, KDE es un estimador no paramétrico, en este caso de muestras pertenecientes a R^5), este sistema se puede interpretar como un *promediador de valores vecinos*.

Analíticamente, lo podemos comprobar mediante la expresión 3.3. Como consecuencia de que las componentes de dominio son independientes de la imagen seleccionada entre las N acumuladas, en el sumando con índice *i* se puede sacar factor común el producto de las Gaussianas asociadas a las componentes de dominio. Vamos a verlo: separando las componentes de dominio de las de color se tiene que:

$$p(z \in bg) = \frac{1}{N \cdot M \cdot L} \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} \sum_{i=0}^{N-1} \prod_{j=1}^{3} \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{1}{2}\frac{\left(z_j(m,l)-u_{i,j}(m,l)\right)^2}{\sigma_j^2}} \cdot \frac{e^{-\frac{1}{2}\frac{\left(z_x(m,l)-u_{i,x}(m,l)\right)^2}{\sigma_x^2}}}{\sqrt{2\pi\sigma_x^2}} \cdot \frac{e^{-\frac{1}{2}\frac{\left(z_y(m,l)-u_{i,y}(m,l)\right)^2}{\sigma_y^2}}}{\sqrt{2\pi\sigma_y^2}}$$

Expresión 3.5

Puesto que para cada píxel los valores ui,x(m,l) y ui,y(m,l) corresponden, respectivamente, a m y l (la posición que ocupa el píxel en la imagen), cada una de las gaussianas de dominio se puede sustituir por las expresiones ωx(m,l) y ωy(m,l), puesto que no dependen de la variable temporal i.

Por tanto,

$$p(z \in bg) = \frac{1}{N \cdot M \cdot L} \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} \sum_{i=0}^{N-1} \prod_{j=1}^{3} \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{1}{2}\frac{\left(z_j(m,l)-u_{i,j}(m,l)\right)^2}{\sigma_j^2}} \cdot \omega_x(m,l) \cdot \omega_y(m,l)$$

Expresión 3.6

por lo que

$$p(z \in bg) = \frac{1}{M} \cdot \frac{1}{L} \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} \omega_x(m,l) \cdot \omega_y(m,l) \cdot \frac{1}{N} \sum_{i=0}^{N-1} \prod_{j=1}^{3} \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{1}{2}\frac{\left(z_j(m,l)-u_{i,j}(m,l)\right)^2}{\sigma_j^2}}$$

Expresión 3.7

Por este motivo, podemos ver el cálculo como un promediado ponderado de la probabilidad de que distintos píxeles pertenezcan a fondo, donde cada una de ellas se calcula como en KDE 3D (para todos los píxeles), y donde las Gaussianas de dominio hacen el papel de factores de ponderación. De este modo, es posible dar un mayor peso a la probabilidad asociada al propio píxel que al resto, disminuyendo el peso proporcionalmente a la distancia que los separa (debido a que se rigen por el producto de Gaussianas de dominio, es decir, por la función $e^{-dis\tan cia^2}$).

Por otro lado, si intentamos razonar el resultado obtenido es lógico pensar que, a pesar de que se tenga en cuenta el entorno del píxel, el peso asociado al mismo sea mayor que el de cualquier otro píxel, teniendo a su vez mayor importancia los píxeles más cercanos que los lejanos.

Por tanto, se tienen en cuenta un número de vecinos del entorno para clasificar cada píxel. De esta forma, si mediante KDE 3D un píxel es asignado a fondo indebidamente y el resto de los vecinos han sido asignados a primer plano correctamente, este sistema es capaz de corregir la no detección del píxel en cuestión, sin necesidad de aplicar ninguna técnica de post-procesado, principal objetivo de este tipo de algoritmos.

Por analogía, se puede afirmar también que en el caso de que con KDE 3D un píxel fuera clasificado indebidamente como primer plano (falsa detección) y la mayoría de sus vecinos fueran asignados correctamente a fondo, el sistema sería capaz de corregirla a partir de la correcta clasificación de los vecinos.

La aparición de gran cantidad de pequeñas falsas detecciones puede deberse a la acción de algún agente externo sobre el fondo de la secuencia, como puede ser el viento sobre árboles, banderas, corriente de agua o incluso sobre la propia cámara (provocando el mismo efecto). Por tanto, un modelado del fondo

genérico (no centrado en el píxel) puede aportar gran cantidad de mejoras en secuencias donde el fondo es dinámico.

Finalmente, teniendo presente que cada una de las Gaussianas son función kernel (con área unitaria), el uso de estas componentes de dominio no modifica el área de la pdf del modelo. Esto se debe a que la suma de todos los valores del kernel resulta uno, por lo que el resultado total será una probabilidad del orden de magnitud de las de KDE 3D.

Por tanto, se suma de forma ponderada las distintas probabilidades obtenidas mediante KDE 3D con el fin de asegurar una cierta regularidad espacial en el entorno del píxel. Como consecuencia, el umbral de decisión en este caso coincide con el utilizado en KDE 3D.

Implementación del algoritmo

Según las expresiones mostradas anteriormente, sería necesario comparar la muestra actual (R,G,B) del píxel a clasificar, ubicado en un píxel genérico (i,j), con el resto de píxeles de la imagen, a pesar de que muchos de ellos tuvieran una aportación nula. Es decir, hay que notar que se modela el fondo como el producto de las cinco funciones marginales estimadas a partir de las muestras, por lo que basta con que una componente esté a una distancia lo suficientemente mayor con respecto a las muestras acumuladas para que la aportación de ese píxel sea cero (puesto que una de ellas lo es).

Por este motivo, decidimos que no era necesario hacer la comparación con todos los píxeles de la imagen, sino que bastaba con hacerlo en una ventana de P píxeles centrada en el píxel a clasificar, puesto que fuera de ésta la aportación sería nula.

Como consecuencia del uso del parámetro P, fue necesario también normalizar las funciones Gaussianas que se utilizaban para el cálculo de probabilidad de x e y, para que todos los valores no nulos estuvieran contenidos en esa ventana, puesto que en caso contario dejaría de trabajarse con funciones Gaussianas y se haría con otra función, la cual correspondería a una Gaussiana distorsionada.

El criterio que se ha utilizado en esta normalización es el mismo que se utilizó en el método corrector de sombras mediante reflectancias explicado en el apartado 2.3. Consiste en asumir que en las (2P+1) muestras se encuentre el 99% de las muestras no nulas, por lo que P=3σ.

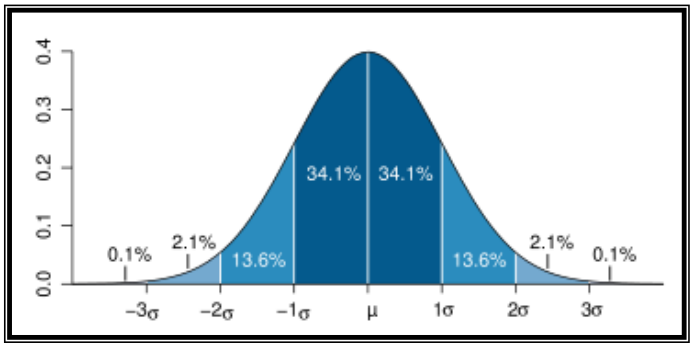


Figura 40 - Representación de probabilidades en función de σ

Por tanto, en función del parámetro P que se quiere utilizar, se recalcularán las Gaussianas con el fin de contener todos los valores no nulos en (2P+1) muestras y asegurar también que son de área unitaria.

Pero a esta ventana de influencia P también se le puede dar una interpretación y es que, por ejemplo, no tiene sentido utilizar valores de píxeles que representan una persona para decidir si un píxel que representa un coche en el otro extremo de la imagen pertenece o no al modelo de fondo. Teniendo en cuenta la explicación introductoria de estas técnicas, hay que ver el objeto como el conjunto de píxeles que lo forman, siendo necesario reducir al máximo la influencia de píxeles que no tengan nada que ver, ya que podrían interferir de forma negativa en la decisión.

Así que, con el fin de poder hacer una serie de pruebas para estudiar la influencia en los resultados de esta ventana y la correcta elección de dicho valor, decidimos implementar el algoritmo de manera que se pudiera variar la medida de la ventana a utilizar. En el siguiente apartado se podrán observar un ejemplo que muestra el impacto del uso de distintas desviaciones espaciales en la detección obtenida.

Figura 41 – Detección de KDE 5D en función de la desviación estándar espacial

Llegamos a la conclusión de que, como consecuencia de su funcionamiento como promediador, con una desviación estándar espacial pequeña conseguimos nuestro propósito sin eliminar detecciones correctas, mientras que el uso de una desviación grande hace que eliminemos en mayor grado las falsas detecciones, pero con el inconveniente de que también elimina algunas detecciones correctas (tal y como puede observarse en el ejemplo de la figura).

Por último, destacar también que, para comprobar la correcta implementación del algoritmo, probamos el sistema con una ventana de tamaño cero, concluyendo que el resultado coincidía con el obtenido con KDE 3D. Esto se correspondía con lo esperado, puesto que tomar una ventana nula se puede modelar como unas Gaussianas de dominio tan estrechas que se pueden asumir como deltas digitales, por lo que la expresión anterior sólo tendría sumandos no nulos en los valores donde estén centradas las deltas. Es decir, esto reduciría el sistema al KDE 3D visto en el apartado 1.3 (no se tendría en cuenta ningún vecino).

Resultados

En este apartado se mostrarán algunos resultados con el fin de destacar la mejora aportada con respecto a los métodos explicados en el apartado 1 (métodos clásicos). Cabe destacar que en ninguno de los casos que se van a observar se ha aplicado ningún algoritmo de post-procesado cuyo fin sea eliminar las pequeñas falsas detecciones, a pesar de la considerable disminución de falsas detecciones en KDE 5D con respecto a KDE 3D. En esta figura se recogen algunas limitaciones de la detección mediante KDE (aparición de una gran cantidad de falsas detecciones por distintos motivos).

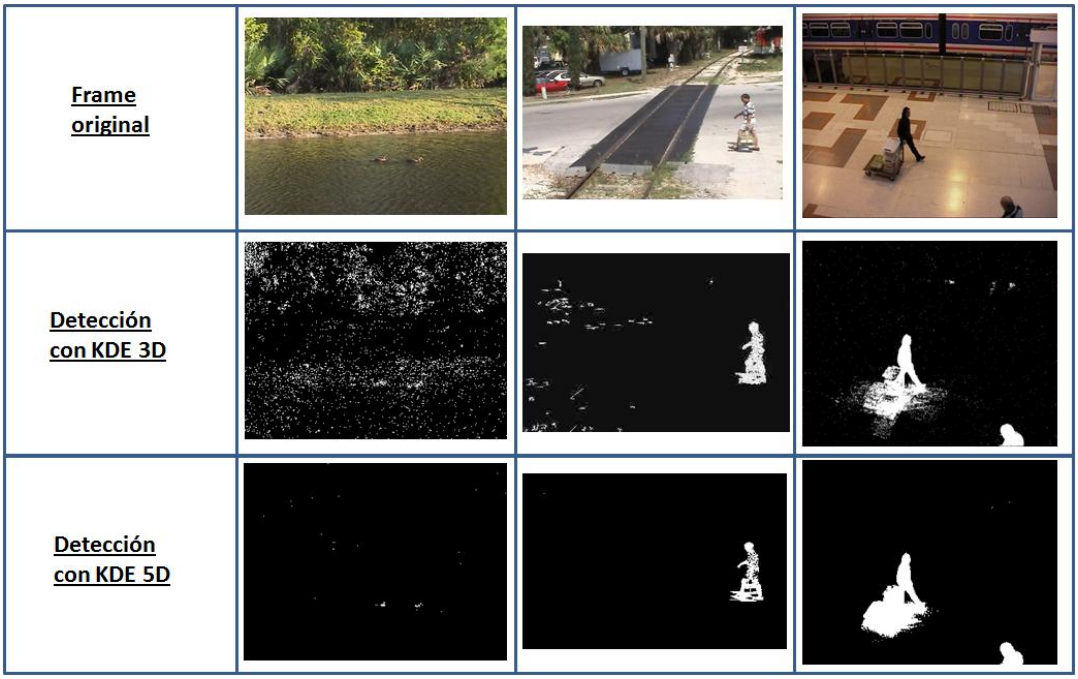


Figura 42 – Resultados comparativos KDE 3D / KDE 5D

En la primera imagen aparecen una gran cantidad de falsas detecciones debido al movimiento de los elementos que conforman el fondo por la acción del viento (los arbustos y el agua del río), mientras que en la segunda situación el movimiento de la cámara hace que aparezcan falsas detecciones debido a que el modelo de fondo está basado en el píxel. Finalmente, la detección obtenida en la tercera secuencia no sólo funciona correctamente sino que además es capaz de obtener una máscara más realista puesto que ha eliminado parte de las sombras del objeto (a pesar de que el único fin de esta imagen era comprobar también el correcto funcionamiento de KDE 5D en una secuencia donde el resultado ya era bueno con KDE 3D o S&G).

Observando los resultados de la figura anterior, se puede comprobar cómo el uso de un único modelo de fondo permite una considerable disminución de falsas detecciones, debido a la regularidad espacial introducida. Además, fijándonos en el resultado obtenido en la última imagen, el sistema KDE 5D no disminuye la calidad de la máscara obtenida mediante KDE 3D (incluso la llega a mejorar un poco).

Por tanto, trabajar con este detector de primer plano hace posible una óptima detección en situaciones donde el sistema KDE 3D o Stauffer and Grimson detectarían una gran cantidad de falsas detecciones, no empeorando, en ningún caso, la detección original.

Aún así, a pesar de que no esté contemplado entre los objetivos de este proyecto, la implementación realizada tiene el inconveniente de que tiene un elevado coste computacional con respecto a KDE 3D o Stauffer and Grimson, puesto que realiza una gran cantidad de operaciones para decidir sobre un píxel.

3.2 Estimación Bayesiana mediante dos modelos

Otra metodología distinta es la que proponen Sheikh y Shah en [10], la cual consiste en generar dos modelos para toda la imagen, uno para modelar el fondo y otro para el primer plano, y clasificar a partir de éstos cada uno de los píxeles al modelo adecuado.

Recordando que en este algoritmo las muestras son de la forma z = (r,g,b,x,y), en dicha implementación se modela el fondo de la secuencia mediante una variante KDE con información espacial, a partir de las N muestras acumuladas durante el aprendizaje

$$\hat{f}(z \mid \psi_b) = \frac{1}{N}\sum_{i=1}^{N} K_b(z - m_i)$$

Expresión 3.8 – Modelo de estimación del fondo

En cambio, se usa una variante para el modelo de los objetos de primer plano, puesto que consiste en la suma de dos distribuciones estadísticas.

Por un lado, una distribución uniforme representada mediante $\gamma = \frac{1}{R \cdot G \cdot B \cdot L \cdot M}$, donde L es el número de columnas del fotograma, M el número de filas y R,G,B representan los posibles valores de cada uno de los canales de color. Por otro lado, tenemos la distribución proveniente de la estimación KDE con información espacial (análoga a KDE 5D), ponderadas ambas por α y (1-α), respectivamente, es decir:

$$\hat{f}(z \mid \psi_f) = \alpha\gamma + \frac{1-\alpha}{N}\sum_{i=1}^{N} K_f(z - m_i)$$

Expresión 3.9 – Modelo de estimación de primer plano

Esta combinación tiene por objetivo detectar y aprender de los objetos de primer plano. Mientras la distribución uniforme pretende tener en cuenta todo valor de (R,G,B) para todo píxel, con el fin de poder representar a todo objeto nuevo (detectar objetos nuevos), la distribución KDE tiene como objetivo aprender de él y adaptar la pdf estimada para poder ser detectado más fácilmente en posteriores imágenes.

Es decir, suponiendo que en un instante t aparece un objeto en la secuencia, éste será asignado al modelo de primer plano, debido a la no coincidencia con el fondo observado y a que el modelo de primer plano tiene en cuenta todos los posibles valores de color por píxel. De este modo, una vez asignado al modelo, éste se actualizará acumulando el valor observado en el píxel (i,j) de dicho modelo, por lo que la probabilidad del mismo para una similar distribución de color en sus proximidades aumentará (debido al aprendizaje aportado por KDE).

Para entenderse mejor, obsérvese la siguiente figura, la cual muestra ambas distribuciones para una de las pdf's marginales.

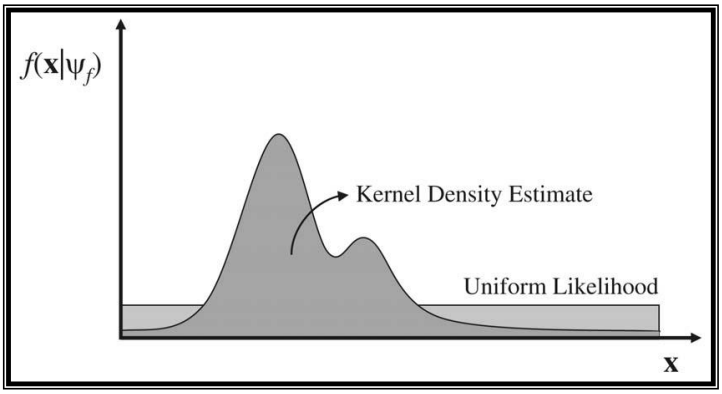


Figura 43 – Representación estimador de los objetos de primer plano

Una vez explicados los modelos, el criterio de decisión entre ambos consistirá en comprobar la siguiente relación:

$$\tau = -\ln \frac{\hat{f}(z|\psi_b)}{\hat{f}(z|\psi_f)} = -\ln \frac{\frac{1}{N}\sum_{i=1}^{N} K_b(z-m_i)}{\alpha\gamma + \frac{1-\alpha}{N}\sum_{i=1}^{N} K_f(z-m_i)}$$ **Expresión 3.10**

Asignándose a la etiqueta l = 0 si τ > k y l = 1 en el resto, donde k es un umbral.

En la práctica, no se decide directamente entre primer plano/fondo ya que, tal y como se ha razonado anteriormente, no hay total independencia entre píxeles vecinos. Para incluir esta dependencia, se maximiza la probabilidad a posteriori asumiendo un modelo Ising como probabilidad a priori de las etiquetas:

$$\sum_{i=1}^{P} \ln\left(\frac{\hat{f}(z|\psi_b)}{\hat{f}(z|\psi_f)}\right) l_i + \sum_{i=1}^{P}\sum_{j=1}^{P} \lambda\left(l_i l_j + (1-l_i)(1-l_j)\right)$$ **Expresión 3.11**

donde li corresponden a las etiquetas, P es el número de píxeles y λ es una constante positiva.

De este modo, considerando una región de la imagen, el sistema es capaz de detectar objetos de primer plano mediante la optimización de una función que depende de más de un píxel, rompiéndose con la idea de que los píxeles son independientes. Esta optimización se hace mediante la técnica de graph-cuts.

En el siguiente apartado proponemos una variante de este algoritmo que utiliza varios modelos para los objetos de primer plano, uno para cada uno de los objetos de primer plano y otro uniforme que permita detectar la aparición de objetos nuevos en la secuencia.

3.3 Estimación Bayesiana mediante varios modelos (Seguimiento)

Basándonos en el método *Estimación Bayesiana mediante dos modelos* explicado en el apartado anterior, surgió la idea de hacer una implementación que, además de mantener una cierta regularidad espacial entorno a cada píxel, aportara una importante funcionalidad con respecto al resto. Se trata de hacer detección de objetos y seguimiento de los mismos a la vez, cosa que puede ser interesante de explorar en futuros proyectos o líneas de investigación.

Su funcionamiento es parecido al anterior, puesto que se basa en la combinación de una distribución uniforme para detectar nuevos objetos, y en KDE para aprender de ellos y realizar el seguimiento. Para ello, hay que aumentar el número de modelos utilizados al número de objetos a seguir, modificando así el sistema antecesor.

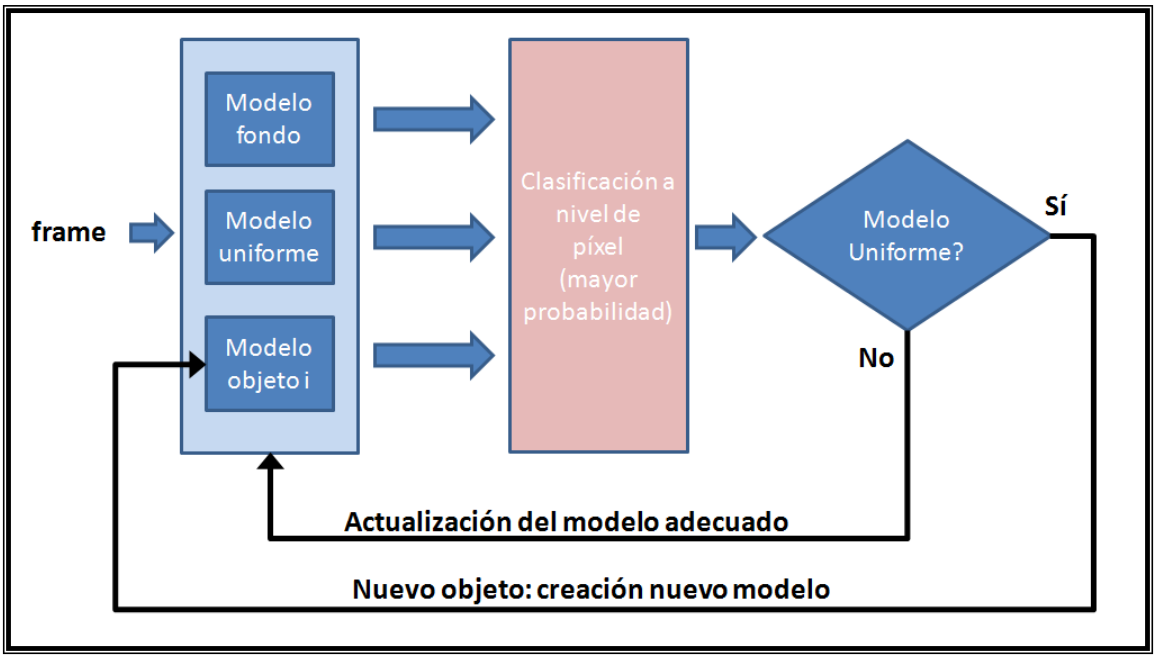


Figura 44 – Diagrama de bloques del estimador bayesiano mediante varios modelos

Sistemas de seguimiento de objetos

Para entender un poco la mejora aportada con respecto a los sistemas de seguimiento de objetos, vamos a explicar las características más importantes de éstos. Situados a la salida de los sistemas de detección de primer plano, se basan en el reconocimiento de los objetos a posteriori, es decir, a partir de la máscara obtenida por el sistema detector, intentan relacionar objetos detectados en imágenes anteriores con los que han sido detectados en el actual a partir de un conjunto de características conocidas como pueden ser el color y tamaño del objeto, la posición de su centroide dentro del fotograma (para determinar su ubicación),…

Una vez encontrada esta relación, se le asigna una etiqueta, la cual no es más que un identificador que tiene como objetivo agrupar los píxeles que pertenecen a un mismo objeto, pudiéndose realizar un seguimiento del objeto imagen a imagen.

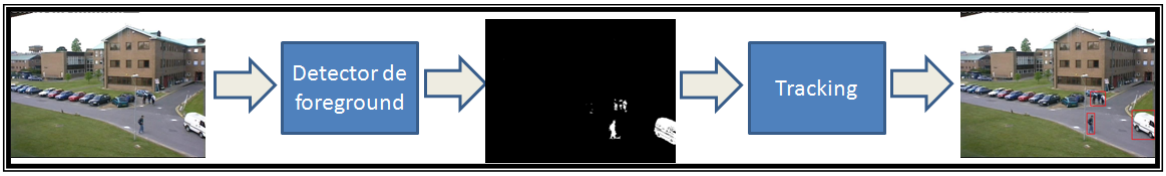


Figura 45 – Diagrama de bloques y resultados tradicionales

A pesar de que no vamos a entrar en más detalle, es importante destacar que uno de los puntos críticos de cualquier método de seguimiento es la colisión de objetos, entendiéndose como tal a la acción de que dos objetos distintos coincidan en una misma ubicación de la escena, puesto que una vez se separen pueden detectarse problemas fácilmente como pueden ser perder a uno de los objetos, cruzar etiquetas,…

La principal diferencia de este método con respecto a los métodos genéricos de seguimiento es que no es necesario conocer las características de más alto nivel para realizar el seguimiento de objetos, puesto que está basado en un análisis estadístico. Esto puede propiciar su uso como sistema de seguimiento o como condicionador, es decir, no utilizarlo como un método de seguimiento tal y como lo conocemos, sino dar una información a priori al sistema de seguimiento para poder afinar mejor el resultado obtenido, utilizándose únicamente como detector de primer plano con cierta información adicional.

De este modo, los métodos de seguimiento utilizarían la máscara que se obtiene a la salida de este detector con la ventaja de tener una información a priori de la relación entre objetos que ayude a realizar el seguimiento con mayor exactitud o de una forma más rápida.

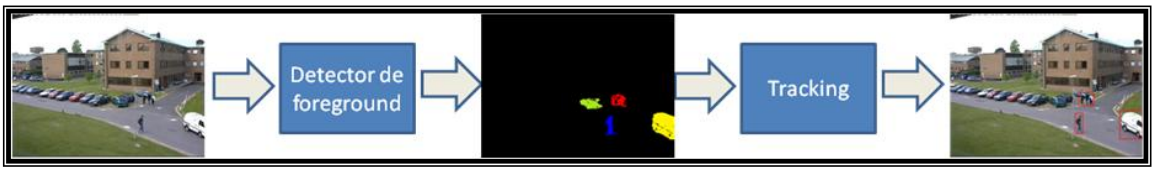


Figura 46 – Diagrama de bloques y resultados mediante estimador multi-modelo

Funcionamiento

Pese a tener otras funcionalidades, no hay que perder de vista que éste es un sistema detector de objetos de primer plano, por lo que consta de una fase de aprendizaje, otra de clasificación y una última de actualización.

El aprendizaje, produciéndose al inicio de la secuencia, tiene como objetivo crear un modelo del fondo, que consiste en acumular las primeras N imágenes tal y como sucedía con KDE 5D. En este caso, para poder evaluar la mejora aportada por el sistema, el modelo de fondo se obtiene a partir del estimador KDE 3D (visto en el apartado 1.3), aunque podría modificarse para utilizarse el modelo 5D.

Una vez transcurrido el periodo de aprendizaje, empezará la clasificación utilizando dos modelos: el modelo de fondo (modelado con KDE 3D) y el de detección de objetos, el cual está modelado mediante una pdf uniforme de la forma:

$$f_{RGBXY}(r,g,b,x,y) = \frac{1}{R \cdot G \cdot B \cdot L \cdot M}$$

Expresión 3.12

donde R·G·B corresponden a los posibles valores de cada una de las componentes de color, y L y M corresponden al número de filas y columnas de la imagen, respectivamente.

El objetivo de este modelo no es otro que detectar objetos nuevos y crear un modelo nuevo que sea capaz de aprender mediante KDE, realizando así su seguimiento a partir de entonces. Es decir, dado un instante t en el que irrumpe un objeto en la escena, el modelo uniforme es capaz de encajar mejor dichas muestras en él (asignándose a él) y crear un nuevo modelo a partir de las mismas con el fin de realizar el seguimiento del objeto. Por tanto, a partir de ese instante, cada píxel debería de clasificarse entre cada uno de los tres modelos obtenidos (los dos iniciales y el del objeto detectado).

De este modo, para cualquier instante genérico, dados N+2 modelos estadísticos (N correspondientes a los objetos, uno al fondo y otro al modelo uniforme), la metodología que se usa para clasificar un píxel entre los distintos modelos consiste en calcular la probabilidad de pertenecer a cada uno de ellos y asignarlo a aquél que tenga asociada una mayor probabilidad.

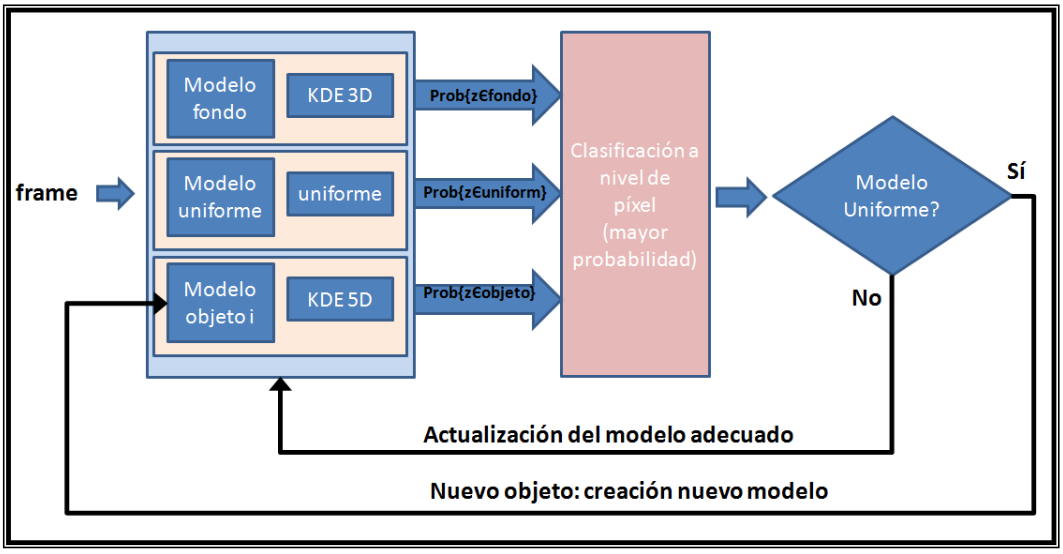


Figura 47 – Diagrama de bloques detallado del sistema implementado

Finalmente, se procedería a actualizar cada uno de los modelos agregando las muestras entre las acumuladas, con el fin de tenerse en cuenta en posteriores imágenes. En este sentido, es importante destacar que para facilitar la implementación se decidió, como versión inicial, almacenar sólo las últimas muestras asignadas al modelo durante la imagen en curso (eliminando las muestras acumuladas durante imágenes anteriores), es decir, el sistema no tiene memoria. Teniendo en cuenta esta limitación, es importante destacar el correcto funcionamiento del sistema. Aún así, la aportación de memoria podría mejorar notablemente los resultados, por lo que podría desarrollarse en futuras líneas de trabajo para comprobar su correcto funcionamiento y cuantificar las mejoras aportadas.

Implementación del algoritmo

A pesar de que, teóricamente, el sistema debe comprobar para cada píxel si pertenece o no a cada modelo posible calculando la probabilidad asociada a éste, sólo se calcula la probabilidad de que un píxel pertenezca a un cierto modelo en caso de estar en la cercanía de la zona de la imagen donde se encuentra ubicado el objeto, con el fin de aumentar la eficiencia del método. En caso contrario, dicha probabilidad será cero, debido al uso de las componentes de dominio x e y.

El razonamiento de esta medida es que los objetos reales tienden a permanecer en la misma región localmente en el tiempo, por lo que no tiene sentido calcular la probabilidad de que un píxel pertenezca a un objeto que se encuentra lejos del mismo, puesto que es muy improbable que haya avanzado tanta distancia en una única imagen.

Para ilustrarlo, obsérvese la siguiente figura, donde es fácil comprobar que no tiene sentido calcular la probabilidad de que un píxel situado, por ejemplo, cerca del objeto verde pertenezca al objeto amarillo en la siguiente imagen, puesto que significaría que habría avanzado a una velocidad irreal para las aplicaciones en las que nos orientamos.

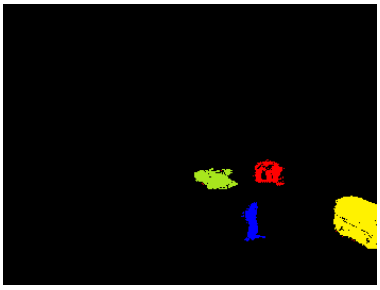


Figura 48 – Ejemplo del resultado obtenido

Una vez se ha realizado la clasificación de todos los píxeles de la imagen, se procede a hacer dos correcciones con el fin de evitar trabajar con falsas detecciones aisladas. Por un lado, a la máscara obtenida, la cual está formada por un número de etiquetas (una para cada modelo), se le puede aplicar un filtro de componentes conexas con el fin de eliminar pequeñas falsas detecciones aparecidas en el contorno de los objetos y de forma aislada (en cualquier zona).

Por otro lado, una vez aplicado el filtrado de componentes conexas, se comprueba el número de muestras que contiene cada uno de los modelos, siendo eliminados también en el caso de que sea inferior a un cierto número que dependerá del tamaño de los objetos que se quieran seguir en ella. Esta comprobación permite eliminar los objetos que han desaparecido de la secuencia, liberando los recursos no necesarios.

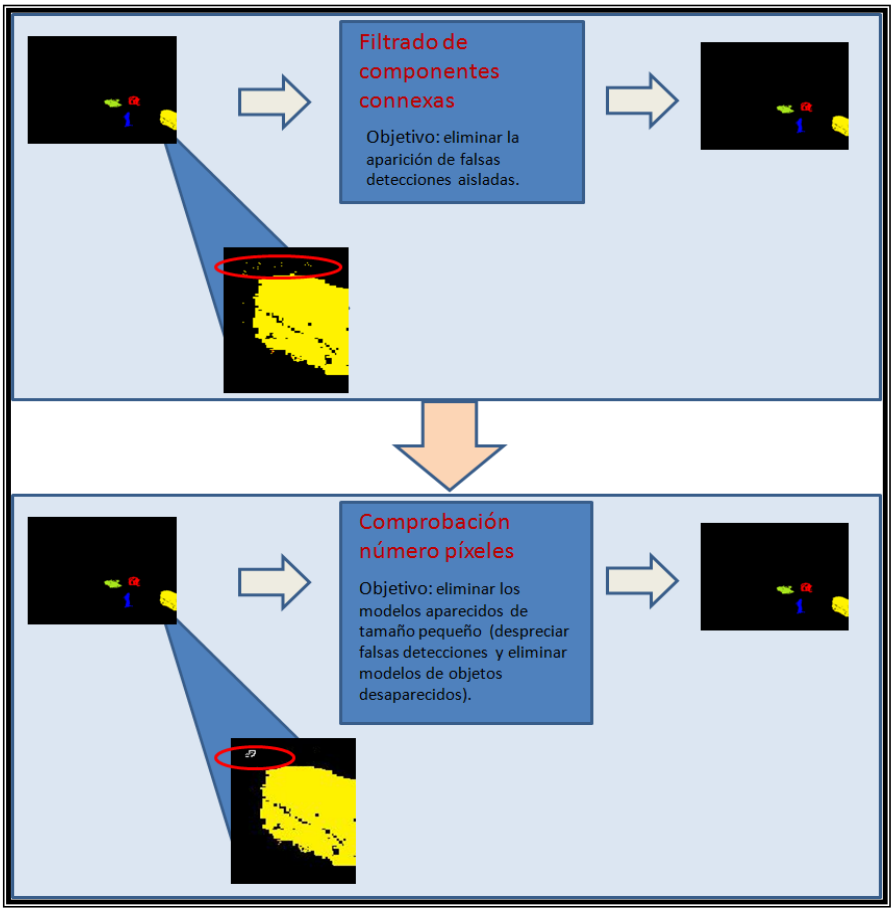


Figura 49 – Correcciones realizadas a la salida del sistema

Finalmente, como ya se ha comentado, se procedería a actualizar cada uno de los modelos con las muestras adecuadas, cosa que no se hace para simplificar la algorítmica.

Resultados

En este apartado, se analizarán los resultados obtenidos mediante esta técnica. Para entenderlos, cabe mencionar el criterio utilizado para representar un objeto. Se ha utilizado un color distinto para cada objeto, utilizándose siete colores de forma cíclica, es decir, se ha representado cada una de las etiquetas con un color distinto.

Dicho esto, obsérvese los siguientes ejemplos que intentan mostrar las ventajas del algoritmo:

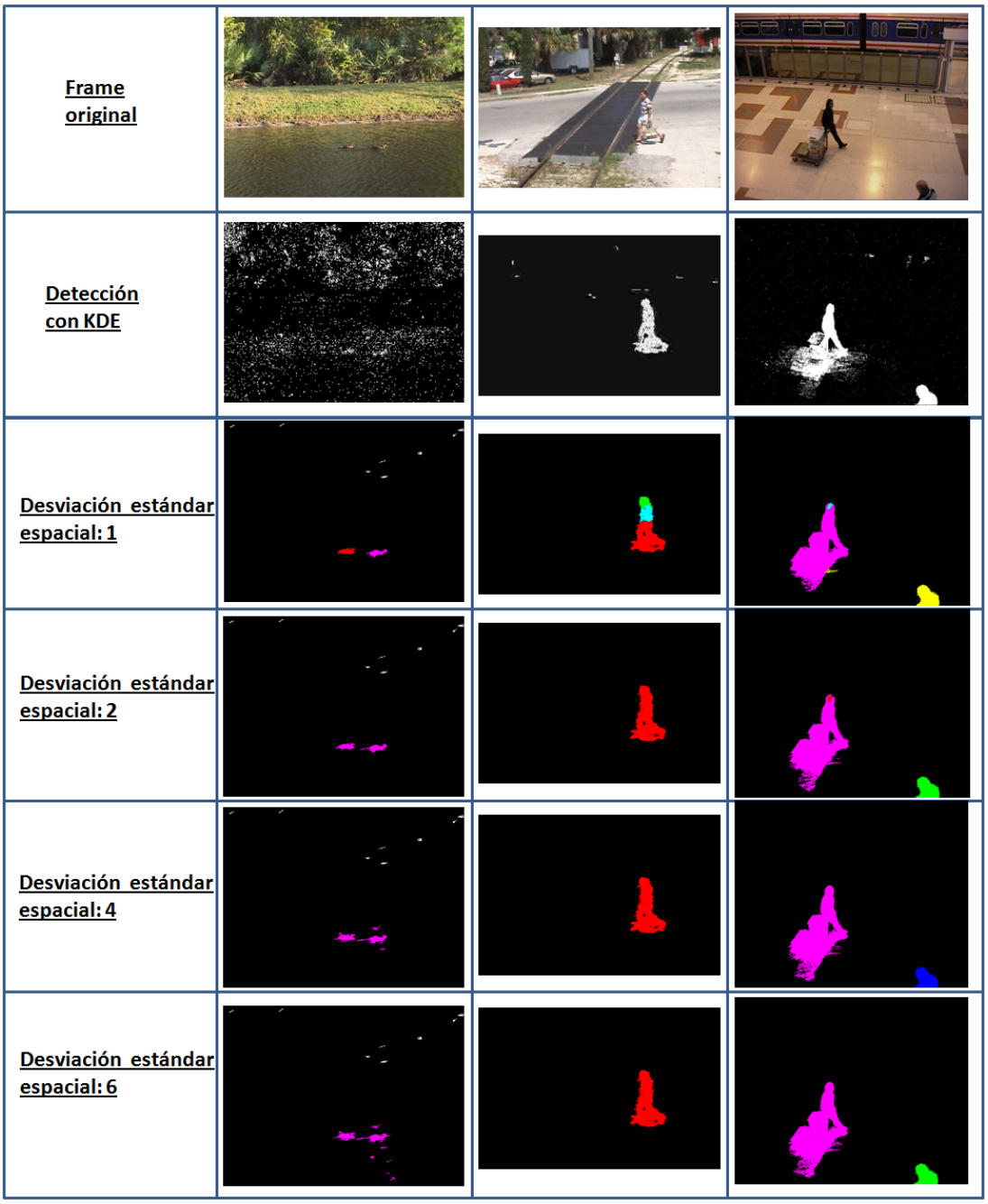


Figura 50 – Resultados obtenidos en función de la desviación estándar espacial y comparativa con KDE 3D

Estos resultados intentan mostrar la mejora aportada en secuencias donde se manifiestan las limitaciones de los detectores de objetos de primer plano que modelan el fondo a nivel de píxel (es decir, estimadores como KDE o Stauffer and Grimson vistos en apartados anteriores).

Si nos fijamos en los resultados mostrados por KDE, por ejemplo, en la primera secuencia, se puede observar la aparición de una gran cantidad de falsas detecciones producidas por el fondo no estático (el fondo está formado por árboles y arbustos que se mueven por la acción del viento). Por otro lado, en la segunda secuencia, se observa otra de las problemáticas de este tipo de algoritmos, la de no

detecciones de algunas zonas en objetos de primer plano, producidas, en este caso, por el movimiento de la cámara (produciéndose variaciones en el modelo estimado del fondo, puesto que se basa en el píxel). Finalmente, en la última secuencia se muestran los resultados donde los sistemas basados en el píxel funcionan correctamente, con el fin de evaluar si trabajar con componentes espaciales empeora en algún caso los resultados iniciales.

Contextualizadas las secuencias utilizadas, se va a analizar los resultados obtenidos como detector de primer plano. Si se considera todo lo que no es fondo como objetos de primer plano, se puede considerar el sistema implementado como un algoritmo de detección de primer plano que obtiene un único modelo de fondo para toda la imagen y otro modelo para los objetos de primer plano. Es decir, se trataría de una implementación parecida al planteado en la Estimación Bayesiana mediante dos modelos, con algunas pequeñas en la construcción de ambos modelos.

En este caso, el sistema aporta robustez en la detección de objetos de primer plano. Prueba de ello, es la drástica disminución de falsas detecciones y de no detecciones en los objetos con respecto a KDE 3D, tal y como teóricamente se había planteado inicialmente. Esto se debe al uso de las componentes de dominio, las cuales aportan una regularidad espacial al modelo de fondo estimado.

Profundizando un poco más en el análisis de los resultados, comentar que la variación de la desviación estándar utilizada para las componentes de dominio influye en la aparición de falsas detecciones, a mayor desviación estándar mayor cantidad de falsas detecciones. Es decir, en este sentido, es mejor utilizar una desviación estándar mínima para las componentes de dominio.

Por tanto, esta implementación demuestra que utilizar un detector de objetos de primer plano basado en dos únicos modelos (uno para fondo y otro para primer plano) puede optimizar la detección. Prueba de ello es que, a pesar de no tener memoria de los objetos de primer plano, éstos son detectados correctamente.

Pero como ya se ha comentado en la parte introductoria, este sistema no sólo detecta objetos de primer plano, sino que también realiza el seguimiento de los mismos. En este sentido, decir que el sistema realiza el seguimiento de objetos de forma correcta, puesto que a cada uno de ellos le asigna una etiqueta distinta, aún sin tener memoria (es decir, basándose en un modelo del objeto obtenido a partir de la imagen anterior), por lo que podría mejorarse aún más incorporando dicha característica.

Además, es destacable mencionar la relación que hay entre la desviación estándar de las componentes de dominio y el número de etiquetas observadas en los resultados. A mayor desviación estándar en las componentes de dominio, menor número de etiquetas aparecidas en la secuencia, por lo que nos interesa trabajar con la mayor desviación estándar para las componentes de dominio.

Esto es debido a que a mayor desviación estándar de las componentes espaciales, mayor influencia tienen los valores observados en píxeles vecinos, por lo que es más difícil sobresegmentar un objeto. La consecuencia de este último punto puede observarse en la penúltima secuencia de la figura 50, donde la persona ubicada en el centro de la imagen ha sido sobresegmentada en los casos de una desviación estándar pequeña, y segmentada de una forma correcta en caso contrario. Por tanto, hay que encontrar un equilibrio entre el uso de varianzas pequeñas (puesto que provoca la sobresegmentación de objetos) y el uso de varianzas grandes (puesto que facilita la aparición de falsas detecciones).

Finalmente, obsérvese los siguientes resultados, los cuales intentan remarcar la principal problemática de este algoritmo.

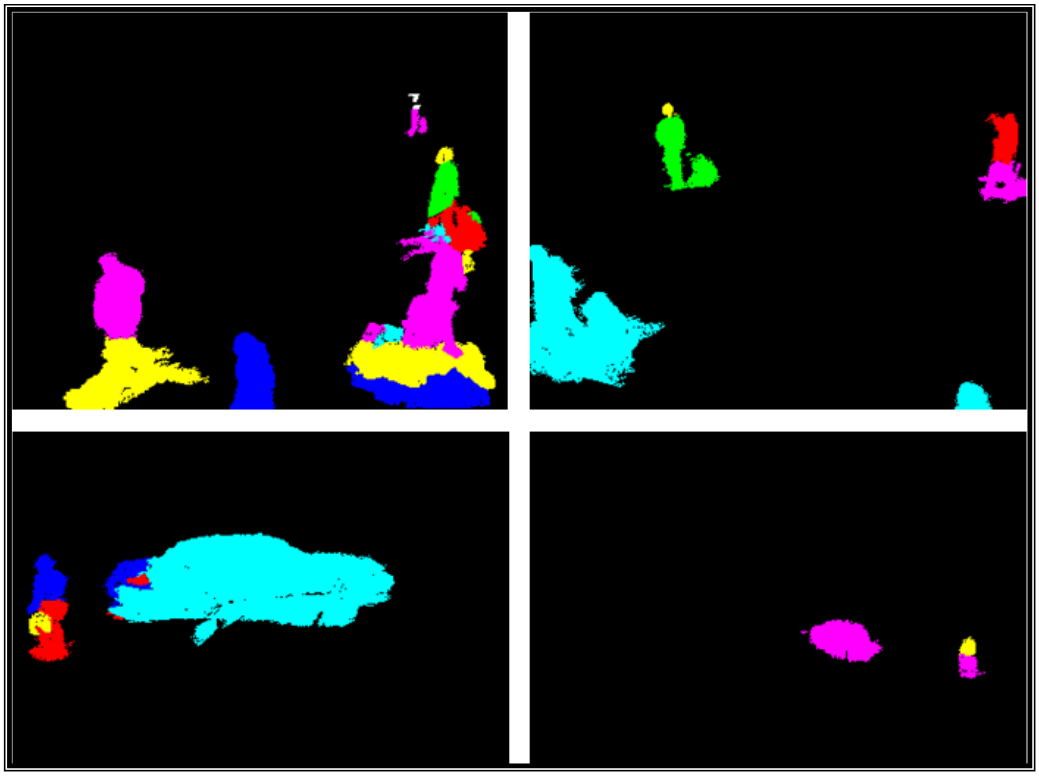


Figura 51 – Ejemplos del mal comportamiento frente a colisiones

Tal y como puede observarse, este sistema no incorpora ningún mecanismo de detección de colisión de objetos por lo que, teniendo en cuenta que el sistema no tiene memoria, los resultados en caso de colisión no son óptimos (de la misma forma a como sucede en muchos algoritmos de seguimiento). De ahí la representación de objetos mediante distintas etiquetas y viceversa (distintos objetos con la misma etiqueta), como consecuencia de las correspondientes colisiones (y de la sobresegmentción debido al uso de una desviación estándar espacial pequeña).

Por tanto, este sistema es capaz de detectar objetos de primer plano aportando una cierta regularidad espacial al modelo, y constituye una primera implementación de un sistema capaz de seguir los objetos de primer plano detectados.

Debido a que este sistema tiene algunas limitaciones, como pueden ser la implementación del algoritmo sin memoria o el no uso de técnicas de resolución de problemas para sistemas de seguimiento, hace que este sistema simplemente abra la puerta a futuras líneas de investigación en cuanto al desarrollo de un método detector y seguimiento de objetos de primer plano, pero no puede aportar ningún resultado concluyente.

III Conclusiones

En este proyecto se han presentado algunas de las técnicas aplicadas para realizar la detección de objetos de primer plano en secuencias de video, agrupadas según:

- Métodos de detección de primer plano clásicos, basados en muestras pertenecientes a R^3 (modelado del fondo a nivel de píxel)
- Métodos de detección de primer plano con regularidad espacial, basados en muestras pertenecientes a R^5 (modelado global del fondo)

Por un lado, dentro de los métodos clásicos, se ha comprobado que el uso del estimador de primer plano basado en KDE permite obtener unos resultados parecidos a los que se obtienen con Stauffer and Grimson, por lo que se puede admitir que es un buen detector de objetos de primer plano.

Por otro lado, dentro de los métodos con regularidad espacial, se ha implementado dos detectores de primer plano basados en KDE que permiten corregir algunas falsas detecciones que se obtienen cuando se trabaja con KDE o Stauffer and Grimson, como pueden ser el movimiento periódico de objetos del fondo por la acción de algún agente (árboles, agua,…).

En este sentido, se ha realizado una comparativa de métodos de ambas clases, llegando a la conclusión de que la aportación de información espacial al modelado de fondo aporta robustez a la detección. Prueba de ello es que para detectar mediante métodos clásicos objetos con la misma nitidez que con los métodos con regularidad espacial, es necesario el uso de sistemas de post procesado que permitan eliminar la gran cantidad de falsas detecciones aparecidas. Además, si se añade esa información a un modelo de primer plano, es posible aumentar considerablemente la relación obtenida entre detecciones correctas y falsas detecciones, pudiéndose llegar a realizar el seguimiento de los objetos de una forma correcta en futuras implementaciones.

Finalmente, se ha presentado también la posibilidad de utilizar un método de corrección de sombras a la salida o en paralelo al método de detección de primer plano con el fin de disminuir la aparición de falsas detecciones asociadas a éstas. En este sentido, se ha valorado que el método más eficiente es el método híbrido, el cual es capaz de corregir una gran cantidad de falsas detecciones asociadas a las sombras (no totalmente) sin eliminar parte de la correcta detección obtenida mediante el detector.

IV Futuras líneas de trabajo

Como ya se ha comentado, la principal línea de trabajo que se puede desarrollar a partir de los resultados obtenidos en este proyecto es mejorar el sistema de detección de primer plano basado en más de un modelo. A continuación se detallan algunos puntos a tener en cuenta en futuros desarrollos:

- **Implementación del algoritmo como detector de objetos de primer plano tradicional:** Para ello, basta con trabajar únicamente con dos modelos, uno para el fondo y otro para objetos de primer plano. De este modo, se conseguiría obtener los mismos resultados con una única máscara de objetos de primer plano, aumentando la eficiencia puesto que sólo habría que calcular dos probabilidades. En este caso, hay que tener en cuenta que los resultados deben mejorar puesto que el modelo de primer plano tendrá en cuenta las muestras obtenidas en las últimas imágenes en las que se haya detectado objeto de primer plano, y no sólo las muestras de la imagen anterior.

- **Incorporación de actualización del modelo de primer plano:** La idea de este algoritmo es aprender del objeto detectado para mejorar su detección en posteriores imágenes y realizar su seguimiento. Es conveniente realizar las modificaciones necesarias para poder actualizar cada uno de los modelos implicados, y así seguir la línea del algoritmo teórico planteado. Para ello, será necesario revisar toda la implementación desarrollada, puesto que hay que definir las muestras *x* e *y* de los objetos detectados a partir de su centroide, con el fin de incluir la información espacial de forma relativa a éste (y no absoluta respecto al origen de la imagen) y así poder actualizar el modelo sin problema a medida que avanza el objeto a lo largo de la secuencia.

V Referencias

[1] C.Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland, "Pfinder: real-time tracking of the human body" *IEEE Trans. On Pattern Analisis and Machine Intell*., vol. 19, nº 7, pp 780-785, 1997

[2] D. Koller, J. weber, T. Huang, J. Malik, G. Ogasakawa, B. Rao, and S. Russell, "Towards Robust Automatic Traffic Scene Analisys in Real-time", Proc. ICPR'94, pp. 126-131, Nov. 1994

[3] C.Stauffer and W.E.L. Grimson, "Adaptive fondo mixture models for real-time tracking", *Proc. IEEE CVPR 1999*, pp. 246-252, June 1999

[4] A. Elgammal, R. Duraiswami, D. Harwood and L. S. Davis, "Background and Foreground Modelling Using Nonparametric Kernel Density Estimation for Visual Surveillance", vol. 90, nº 7 pp 1151-1163, 2002

[5] M.Pardàs, J.L. Landabaso, L. Xu "Shadow Removal with Blob-Based Morphological reconstruction for error correction", *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05).* Volume: 2, On page(s): 729- 732.

[6] T. Horpraset, D. Harwood and L. Davis, "A statistical approach for real-time robust background subtraction and shadow detection", ICCV'99 FRAME-RATE Workshop

[7] J. Lou, H. Yang, W. Hu and T. Tan, "An Illuminantion Invariant Change Detection Algorithm", ACCV2002: The 5th Asian Conference on Computer Vision, 23-25 January 2002.

[8] H. Z. Sun, T. Feng and T. N. Tan, "Robust extraction of moving objectrs from video sequences", Proc. Of the Fourth Asian Conference on Computer Vision, vol nº 2, Jan 2000, pp. 961-963.

[9] A. Mittal and N. Paragios, "Motion-Based Background Subtraction using Adaptative Kernel Density Estimation", Real-Time Vision Modeling Siemens Corporate Research Princeton, NJ 08540.

[10] Y. Sheikh and M. Shah, "Bayesian Modeling of Dynamic Scenes for Object Detection", IEECS Log Number TPAMI – 0375 – 0704.

[11] R. Pless, J. Larsson, S. Siebers and B. Westover, "Evaluation of Local Models of Dynamic Background", Departament of Computer Science and Engineering Washinghton University (St. Louis).

[12] M. Piccardi, "Background subtraction techniques: a review" , In Proc. of IEEE SMC 2004 International Conference on Systems, Man and Cybernetics, volume 4, pages 3099–3104, The Hague, The Netherlands, Oct 2004.

[13] J. Gallego, "Detección y Seguimiento de Objetos de Primer Plano", PFC ETSETB Noviembre 2007.