UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

Generación de fondo de escena en secuencias de vídeo-seguridad

-PROYECTO FIN DE CARRERA-

Alberto Muñoz García
Septiembre 2012

Generación de fondo de escena en secuencias de vídeo-seguridad

Autor: Alberto Muñoz García
Tutor: Juan Carlos San Miguel Avedillo
Ponente: José María Martínez Sánchez
email: {Alberto.muoz@estudiante.uam.es,Juancarlos.Sanmiguel@uam.es,
JoseM.Martinez@uam.es}

Vídeo Processing and Understanding Lab
Departamento de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Septiembre 2012

Trabajo parcialmente financiado por el Ministerio de Economía y Competitividad bajo el proyecto TEC2011-25995 (EventVideo).

Resumen

El principal objetivo de este proyecto fin de carrera es el diseño e implementación de un algoritmo de inicialización de fondo en secuencias de vídeo-seguridad capturadas con cámara fija que permita mejorar los distintos algoritmos existentes en el estado del arte.

Proponemos una técnica de regiones para una inicialización robusta que explota la consistencia espacial y temporal en un fondo de escena estático. Primero la secuencia es dividida en regiones o "bloques" que son agrupados temporalmente para reducir el número de candidatos a fondo de escena. Posteriormente se buscan similitudes entre los candidatos que son contiguos espacialmente utilizando información de color y bordes.

Nuestro algoritmo es capaz de soportar vídeos muy poblados de frente permitiendo más de un cincuenta por ciento de objetos de primer plano en la secuencia de entrenamiento sin utilizar umbrales. Además resuelve el problema del primer plano estático y del camuflaje.

Palabras clave

Vídeo análisis, vídeo-seguridad, fondo de escena, primer plano, bloque de análisis, agrupamiento temporal, continuidad espacial.

Abstract

The main objective of this master thesis is the implementation of a new background initialization approach for video surveillance using static cameras, which improves the related state-of-the-art without using thresholds. It obtains an image that represents the background by using a training sequence.

We propose a region-based approach for robust background initialization that exploits both spatial and temporal consistency of the static background. First, the sequence is subdivided in regions that are clustered along the time-line in order to reduce the number of background candidates. Then, the background is generated by growing the current background with the best spatial continuation.

Our algorithm supports heavily cluttered video sequences allowing more than 50 per cent of foreground in the training video sequence and the effect of stationary objects that do not move for a long time (even being more visible than the background).

Keywords

Vídeo analysis, video-surveillance, scene background, scene foreground, block-based analysis, cluster temporal, spatial continuity.

Agradecimientos

Quisiera comenzar dando las gracias a mi ponente, José María Martínez, por haberme dado la oportunidad de colaborar con el grupo VPU durante estos últimos 2 años y poder haber llevado a cabo este proyecto.

Quiero agradecer de manera especial, a mi tutor del proyecto Juan Carlos San Miguel por su ayuda a la hora de realizar el PFC, por los consejos recibidos y por el apoyo recibido durante todo este tiempo.

En cuanto al resto de miembros del grupo, les quiero dar las gracias por ayudarme cuando lo he necesitado y por hacer que los días fueran más entretenidos.

Me gustaría agradecer al profesorado que trabaja en la Escuela Politécnica Superior el trato que ofrecen al alumnado. Nos hacen pasar buenos y malos momentos, pero ante todo, nos ayudan a madurar y crecer como personas.

Por supuesto no puedo olvidarme de todos esos amig@s que han hecho que los años aquí hayan sido geniales. Me llevo muy buenos recuerdos de todos tanto dentro como fuera de la universidad que jamás olvidaré. Me gustaría acordarme aquí de Alfonso Colmenarejo y Laura Valenzuela sin los cuales nada habría sido igual, gracias por todos estos años, por ser mis segundos padres, por todos los buenos momentos y por mi futuro ahijado, de Sarita por estar siempre a mi lado cuando más lo he necesitado a pesar de que quedar contigo sea lo más difícil del mundo, de Bader por las mil horas que me ha aguantado en los laboratorios y por esos fines de semana donde no has conseguido echarme de tu casa o del pizza. Gracias Chino por esos momentos de sufrimiento dentro del campo de fútbol y de risas fuera de él, a Almu por los buenos momentos que hemos pasado juntos (y los no tan buenos), por aguantar mi carácter y sobre todo por esas llamadas tan necesarias cuando estaba en Inglaterra, a Alex por amenizar cada verano con esas famosas bbq , a Luis por esos chistes tan malos y esa manera de meter la pata cada vez que habla, a Pau por aguantarnos a todos, a Puertas por hacer que sea obligado ir a EU ;), a Sas por esos ratitos de tontería que siempre son necesarios, , al “tato” por llevarme por el buen camino y enseñarme que salir a correr revitaliza cuerpo y mente (aunque tomar un caña también) y por su puesto a Juan que ha permitido que esta panda de sinvergüenzas se juntara viernes tras viernes para discutir de toros, cerdos.... y en general a todos y cada uno de los compañeros con los que he compartido algo más que la uni. Gracias a mis amig@s de

toda la vida, es especial al “cuco” y “maris” por estar a mi lado, por apoyarme, por hacerme reír en los momentos difíciles y al “enano” y “la churri” que han aparecido en el momento justo, veremos como se da el viajecito...

Quiero agradecer de manera especial el apoyo que he recibido durante toda la carrera de Alicia Beisner y toda su familia ya que aunque las cosas no siempre salen como se esperan siempre me he sentido muy arropado y muy feliz durante todos estos años y me han ayudado cuando más lo he necesitado.

Por último y más importante agradezco a toda mi familia el apoyo recibido durante la carrera y durante toda la vida ya que gracias a ellos he podido llegar hasta aquí. En especial a mis padres por la educación recibida, por apoyarme sobre todo en los primeros años o este último cuando las cosas no salían tan bien y por haberme dado la oportunidad de estudiar sin tener otras preocupaciones. Por su puesto a mi hermana y mi cuñado por toda su preocupación, por estar ahí cuando más lo necesitaba, por darme la oportunidad de pasar tres meses geniales en Inglaterra y por hacerme tío.

Alberto Muñoz García
Septiembre 2012

Índice general

Índice de figuras

Índice de cuadros

Glosario

BLOB *Binary Large OBject*

HMM *Hidden Markov Model:* A type of statistical model.

RGB *Red Green Blue Color Model:* Modelo aditivo de color donde el rojo, verde y azul son mezclados para reproducir otros colores.

PCA *Principal component Analysis:* Análisis de componentes principales, reduce un set de datos.

FG *Foreground:* Frente de escena.

BG *Background:* Fondo de escena.

GT *Ground truth:* En el caso de inicialización de fondo es el fondo real.

DCT *Discrete Cosine Transform*

HOG *Histogram of Oriented Gradients*

TP *True Positives.* Pixeles de fondo clasificados como fondo.

FP *False Positives.* Pixeles de fondo clasificados como frente.

TN *True Negatives.* Pixeles de frente clasificados como frente.

FN *False Negatives.* Pixeles de frente clasificados como fondo.

NE *Number of Error pixels*

AE *Average gray-level Error*

NC *Number of Clustered error pixel*

Capítulo 1

Introducción

1.1. Motivación

Actualmente, el uso de sistemas de análisis de secuencias de vídeo está cada vez más presente en una gran variedad de aplicaciones relacionadas con (entre otros) vídeo-vigilancia [1], indexación de contenidos [2], cine y TV [3], compresión de vídeo [4] e interacción persona-ordenador [5]. Debido a la creciente cantidad de información visual generada por las cámaras y sensores de estas aplicaciones, sea hace necesario desarrollar herramientas de análisis automático que operen en tiempo real en ciertos dominios de aplicación (por ejemplo, vídeo-vigilancia), permitiendo extraer las regiones de interés de la secuencia de vídeo analizada.

En este contexto, el primer problema es la localización espacial de la(s) región(es) donde sucede algo relevante en cada imagen (*frame*) de la secuencia de vídeo [6]. Esta operación se conoce como segmentación de objetos y tiene como objetivo discriminar las regiones u objetos del primer plano de una imagen (frente de escena o *foreground*), del resto de los objetos no relevantes (fondo de escena o *background*). En el caso de una escena grabada por una cámara fija, que es el caso analizado por este proyecto, las técnicas de segmentación más utilizadas están basadas en la aproximación *Background Subtraction [7]* que propone el modelado matemático del fondo de escena y su posterior sustracción de las imágenes de la secuencia de vídeo para identificar las regiones de primer plano.

Debido a la complejidad y heterogeneidad de la información visual, la segmentación automática de objetos es una de las tareas más complicadas dentro del procesado de vídeo cuya precisión afecta el resultado del sistema que la utiliza (pues muchas etapas internas dependen de ella). En concreto, un algoritmo de detección de objetos de primer plano debe lograr de manera eficiente, y preferentemente en tiempo real, solventar los siguientes problemas:

- *Obtener y actualizar el fondo de la escena.* El modelo de fondo se suele obtener en una fase de inicialización (por ejemplo, un número inicial de imágenes de la secuencia de vídeo), la

cual a menudo es compleja principalmente debido a que, en esa fase, existe gran cantidad de objetos en movimiento (por ejemplo, personas) que sólo permiten visualizar parcialmente el fondo [8]. Por otro lado, es necesario detectar los cambios susceptibles que pueden producirse en el fondo de la escena a lo largo del tiempo, ya que, según la complejidad del escenario, dicho fondo podría verse alterado por multitud de objetos [9] tales como objetos abandonados [10] y coches estacionados en un aparcamiento [6].

- *Sombras y reflejos presentes en la escena.* La interacción entre las fuentes luminosas y los diferentes objetos presentes en la escena pueden producir efectos (sombras y reflejos) que suelen ser categorizados como objetos del primer plano [11]. Las sombras que acompañan a los objetos en movimiento no son parte del primer plano pero tampoco forman parte del fondo de escena: deben ser eliminadas. Sin embargo, las sombras inherentes al fondo de escena (de sus objetos) deben considerarse en su modelo mediante mecanismos que permitan diferenciarlas tales como el análisis de bloques [12].

- *Fondos multimodales.* Son aquellos que podemos encontrar en secuencias con objetos cuyo movimiento es lento y/o periódico (movimiento de las hojas de los árboles, movimiento ondulatorio del agua,...) y que, desde una perspectiva semántica, habitualmente se considera que pertenecen al fondo de la escena [13].

- *Camuflaje.* Este efecto aparece cuando los objetos del primer plano poseen el mismo color y textura que el fondo de escena; por este motivo, los objetos en primer plano se confunden o camuflan con el fondo de escena [9].

- *Ruido introducido en la secuencia de imágenes.* El ruido que proviene de la captación de las imágenes por las cámaras de vídeo [14] puede producir errores en la segmentación de objetos de vídeo siendo necesaria su consideración en el modelo del fondo de escena [15].

- *Cambios de iluminación de la escena a analizar (rápidos/lentos).* Son variaciones de iluminación que se pueden producir a lo largo del tiempo (escenas en exteriores), o por la aparición/desaparición de las fuentes de iluminación (luces apagadas/encendidas en interiores). Estos cambios pueden ser malinterpretados como objetos en movimiento o hacer que el fondo de escena quede obsoleto si éste no se actualiza correctamente [16].

- *Ajuste de parámetros de funcionamiento.* Una de las tareas más importantes de un sistema automático de análisis es el ajuste de los parámetros. Identificar los parámetros más significativos (aquellos que alteren en mayor medida los resultados del sistema) es una tarea compleja, siendo deseable tener un número reducido de ellos [17], que requiere un estudio detallado de las características de los datos a analizar (secuencias de vídeo).

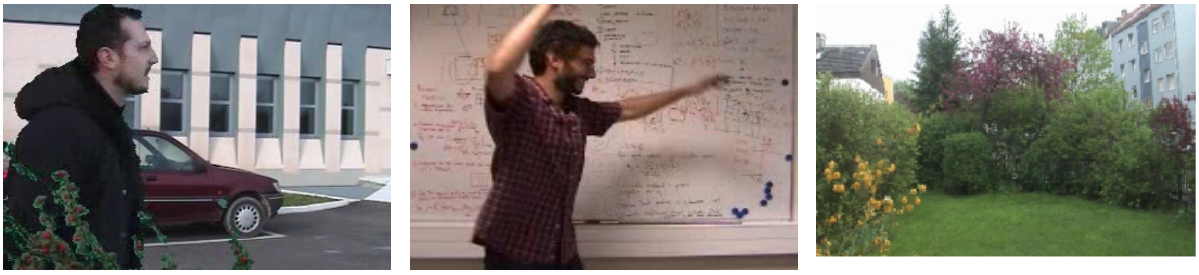


Figure 1.1: Ejemplos de problemas de segmentación de objetos que afectan a la extracción del fondo de escena (de izquierda a derecha): primer plano, sombras, fondo multimodal.

La figura 1.1 presenta un ejemplo de los problemas anteriormente mencionados que afectan a la obtención del fondo de escena. La primera imagen muestra como es posible que en la secuencia de entrada existan objetos de primer plano que reduzcan considerablemente la visibilidad fondo de escena. La segunda imagen identifica el problema de las sombras de objetos en movimiento (las cuales se suelen confundir con el fondo de escena). La última imagen muestra un fondo multimodal donde los píxeles de las ramas en movimiento pueden tomar más de un valor a lo largo del tiempo (considerándose siempre como fondo de escena). La figura 1.2 muestra problemas adicionales en la fase de inicialización para obtener el fondo de escena. En la primera fila podemos observar como, suponiendo un fondo de escena (imagen izquierda), un objeto de primer plano puede confundirse debido a que presenta un color similar (imagen derecha). En la segunda fila se puede observar el ruido introducido por la cámara que se podría confundir con objetos de primer plano (imagen derecha) sobre una imagen capturada sin ruido (imagen izquierda). Por último en la tercera fila se observa un cambio de iluminación en la escena que cambia completamente la apariencia del fondo.

A fin de solucionar estos inconvenientes y extraer las regiones de interés con máxima fiabilidad, comúnmente se definen cuatro etapas [18]: pre-procesado (tareas de procesamiento de imágenes), modelado de fondo (inicialización y mantenimiento), detección de frente (identificar objetos de primer plano) y validación de datos o pos-procesado. Aunque las etapas de modelado de fondo y detección de frente suelen mezclarse y confundirse, lo cierto es que se trata de etapas diferentes pues el modelado de fondo comprende la elaboración y actualización de un fondo de escena mientras que la detección de frente compara cada imagen del vídeo con el modelo de fondo para determinar el primer plano.

El modelado de fondo debe considerar varios aspectos tales como la representación (modelo matemático), inicialización (obtención del modelo) y actualización (adaptación) del fondo de escena. Entre ellos, la inicializacion es de especial interés pues en segmentación de objetos, dependen de ella tanto la actualización del modelo de fondo como las etapas posteriores detección de frente y post-procesado. Adicionalmente, si consideramos que la segmentación de objetos es frecuentemente utilizada por etapas de análisis de alto nivel (seguimiento, detección de activi-

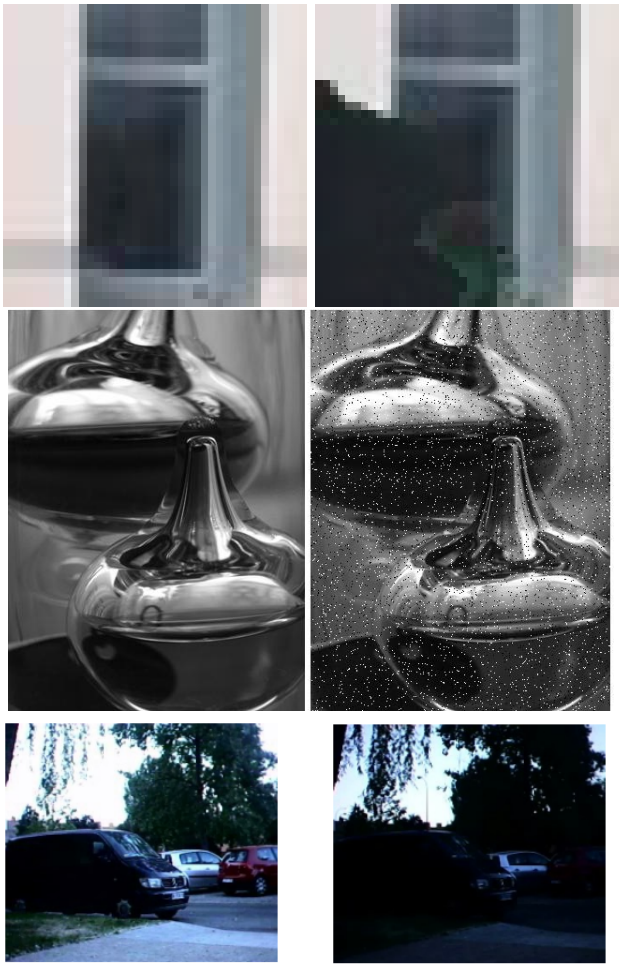


Figure 1.2: Ejemplos de problemas de segmentación de objetos que afectan a la extracción del fondo de escena (de arriba a abajo): camuflaje, ruido, cambios de iluminación.

dades, conteo de personas, etc.), la inicialización de fondo se presenta como una etapa crítica de sistemas basados en segmentación de objetos.

En general, para escenarios simples, las aproximaciones basadas en técnicas clásicas de modelado de fondo funcionan correctamente [13, 16, 7, 15], pero suelen fallar cuando nos encontramos en entornos con alta densidad de objetos en movimiento (como lugares públicos) u otros problemas anteriormente mencionados que afectan a la segmentación de objetos. La figura 1.3 muestra dos problemas típicos para extraer el fondo de escena y por consiguiente, segmentar los objetos. La primera fila representa un ejemplo de camuflaje y alta cantidad de objetos sobre un fondo de escena relativamente sencillo. La segunda fila se corresponde con el problema *sleeping person* en el cual el fondo de la escena permanece menos tiempo estático que el primer plano (la persona de la izquierda).

Figure 1.3: Ejemplos de modelado de fondo y extracción de objetos de primer plano. Fondo de escena, imagen seleccionada y frente extraído se corresponden con, respectivamente, la primera, segunda y tercera columnas.

1.2. Objetivos

El principal objetivo del trabajo presentado en este proyecto es el desarrollo de un nuevo algoritmo de inicialización de fondo en secuencias de vídeo capturadas con cámara estática. En este proyecto se hará un análisis exhaustivo de las técnicas existentes y se compararán con el algoritmo desarrollado. Para lograr los retos que se plantean en este proyecto, las tareas que se llevarán a cabo son las siguientes:

- Estudio del estado del arte actual: análisis de los algoritmos de modelado de fondo atendiendo a sus características.
- Selección e implementación de los algoritmos más relevantes de inicialización de fondo: una vez concluida la fase de recopilación de información y estudio, se implementarán y evaluarán las técnicas más relevantes.
- Desarrollo e implementación de un nuevo algoritmo: se desarrollará un nuevo algoritmo tras los resultados de la fase anterior, que consistirá en tres etapas.

    1. Etapa de estudio del tamaño óptimo de las regiones de análisis.
    2. Etapa de agrupamiento temporal.
    3. Etapa de continuidad espacial.

- Análisis de resultados y conclusiones: se extraerán una serie de conclusiones de nuestro algoritmo en función de la influencia de los diversos factores de las secuencias en los resultados y se ofrecerá una comparativa con los algoritmos existentes en el estado del arte.

1.3. Organización de la memoria

La memoria consta de los siguientes capítulos

- Capítulo 1. Este capítulo contiene la introducción, objetivos y motivación del proyecto.
- Capítulo 2. Este capítulo contiene una visión general donde la literatura existente se relaciona con el trabajo presentado en este documento.
- Capítulo 3. Este capítulo contiene una introducción sobre el algoritmo realizado.
- Capítulo 4. Este capítulo describe la segunda etapa del algoritmo realizado: agrupamiento temporal.
- Capítulo 5. Este capítulo describe la tercera etapa del algoritmo realizado: continuidad espacial.
- Capítulo 6. Este capítulo presenta los resultados.
- Capítulo 7. Este último capítulo proporciona las conclusiones y trabajo futuro.

Capítulo 2

Estado del arte

2.1. Introducción

Este capítulo describe conceptos básicos relacionados con la inicialización de fondo de escena en secuencias de vídeo, así como una revisión de las técnicas más representativas. Primero se introducirá la etapa de modelado de fondo haciendo especial atención a la parte de inicialización (sección 2.2). Posteriormente se analizará la literatura existente en el estado del arte para llevar a cabo la inicialización del fondo de escena. Se describen las diferentes características que podemos encontrar en los distintos modelos de fondo (sección 2.3) para posteriormente describir los modelos más representativos existentes en el estado del arte (sección 2.4). Por último se estudiarán los diferentes métodos de evaluación existentes (sección 2.5) así como los *datasets* disponibles (sección 2.6).

2.2. Modelado e inicialización de fondo

En segmentación de objetos utilizando cámara fija (el caso analizado por este proyecto), las técnicas más utilizadas están basadas en la aproximación *Background Subtraction [7]* cuyas etapas más comunes de análisis son las siguientes [18] (figura 2.1):

1. Pre-procesado: realiza simples tareas de procesamiento de imágenes que modifican las imágenes de entrada para mejorar su procesado por las etapas posteriores.

2. Modelado de fondo: comprende tanto la inicialización del fondo como su mantenimiento a lo largo de toda la secuencia de vídeo.

3. Detección de frente: realiza la sustracción de frente que permite obtener los objetos de interés en movimiento en la secuencia de vídeo.

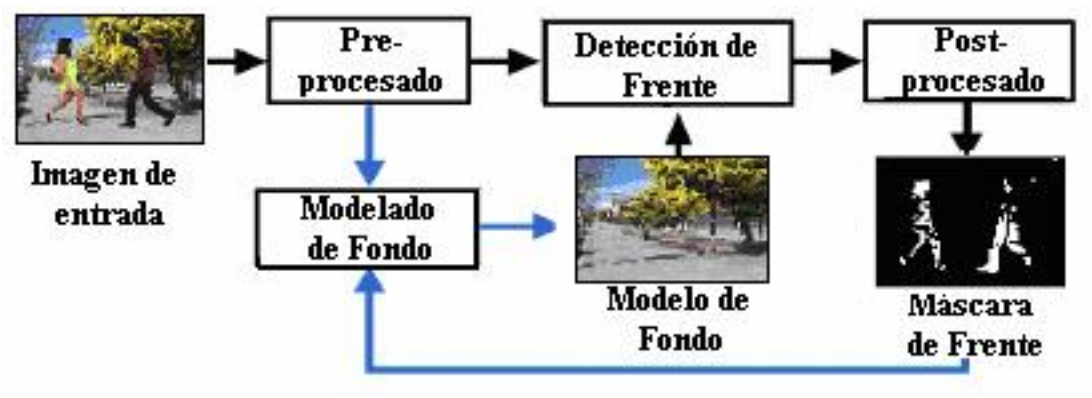


Figure 2.1: Diagrama de bloques de un algoritmo de segmentación de objetos [7].

4. Validación de datos o pos-procesado: diversas operaciones para mejorar los resultados obtenidos de las etapas anteriores.

El modelado de fondo es la etapa fundamental en los algoritmos de segmentación de objetos de primer plano. Con el modelo de fondo se compara cada imagen de un vídeo de entrada a fin de segmentar los objetos relevantes (primer plano) de la secuencia de vídeo. A su vez, el modelado de fondo comprende tres aspectos [15] [19]: representación, adaptación e inicialización.

La representación describe el tipo de modelo matemático utilizado para representar el fondo en cada instante de tiempo [15]. El modelo más simple es aquel cuyos valores son conocidos y no varían a lo largo de la secuencia (salas croma [20]). Esta suposición se suele realizar en entornos muy controlados (por ejemplo, interiores con una imagen de fondo conocida y estática y sin variaciones de iluminación) y requiere únicamente modelar factores externos como el ruido de la cámara. Este tipo de fondos se conoce como fondos unimodales. Los métodos de modelado de fondo unimodal, asumen que la distribución de cada uno de los píxeles de fondo tiene un único pico o modo (valor) y por tanto, el fondo de escena de la secuencia analizada es esencialmente estático [21]. En cambio, en entornos menos controlados, los píxeles del fondo pueden cambiar su valor debido a cambios de iluminación (como sucede, por ejemplo, en escenas capturadas al aire libre y a distintas horas del día) o debido al movimiento de los objetos que pertenecen al fondo (como ocurre en escenas con árboles agitándose u olas de mar). Este último tipo de fondos se conoce con el nombre de multimodales y existen diversas aproximaciones para su modelado utilizando mezclas finitas de Gaussianas (MOG) [21], modelos ocultos de Markov (HMM) [22] ó mediante agrupaciones (clusters) [8] que están ordenados de acuerdo a la probabilidad de que se modele el fondo.

El modelo de adaptación consiste en ir actualizando el fondo según cambios en la escena. Los métodos existentes para la adaptación de fondo pueden ser clasificados como predictivos o no predictivos [23]. Los primeros modelan la escena como una serie de tiempos basados en

	Características de los modelos
Temporal	Estabilidad temporal/Continuidad temporal.
Espacial	Continuidad espacial.
Otras	Uso de información de primer plano/Gris-Color/píxel-región Soporta primer plano estático/Output/Visibilidad fondo/Movimiento

Table 2.1: Características de los modelos de inicialización de fondo de escena

observaciones anteriores mientras que los segundos no utilizan el orden de las observaciones y construyen una representación probabilística de la observación en un píxel particular. Tradicionalmente se considera la velocidad a la que suceden los cambios en la escena (mediante un parámetro denominado factor de adaptación) para realizar esta adaptación. Además, el fondo se debe adaptar tanto a las posibles variaciones de iluminación (por ejemplo, si la secuencia es capturada a distintas horas del día o existen cambios en la fuente que ilumina la escena) como a los objetos que se depositen o desaparezcan del fondo (por ejemplo debido al robo o abandono de objetos [10]) ya que todas estas modificaciones en el fondo pueden convertirlo en otro completamente diferente al inicial.

Actualmente se ha trabajado mucho sobre adaptación y representación del modelo de fondo [24] [25] [26] [27] [28] [29] pero la inicialización ha recibido poca atención debido a su complejidad en condiciones reales. Este aspecto consiste en obtener un primer fondo de la escena que se actualiza posteriormente. La mayoría de los modelos de fondo asumen una breve secuencia en donde no hay objetos de primer plano (secuencia de entrenamiento). En cambio, los mecanismos de inicialización clásicos [15] reducen drásticamente su efectividad si se ejecutan sobre secuencias que contienen objetos en movimiento u otros problemas que afectan a la segmentación de objetos, es decir, donde no se puede obtener una secuencia de entrenamiento ideal. Este error acarrea multitud de problemas, puesto que la imagen de fondo que modelaría la secuencia tomaría objetos no deseables como parte del fondo impidiendo, por tanto, la detección correcta de posteriores objetos en movimiento.

2.3. Características de los modelos de inicialización de fondo

Entre las distintas propuestas de inicialización existentes en el estado del arte podemos encontrar diferentes características que ayudan a clasificar e identificar sus ventajas y desventajas. La mayor parte de las aproximaciones utilizan la información temporal de la secuencia de inicialización y/o la continuidad espacial entre las regiones del fondo de escena para inicializar el modelo de fondo. La tabla 2.1 presenta las distintas características que se pueden encontrar en los distintos modelos de inicialización de fondo.

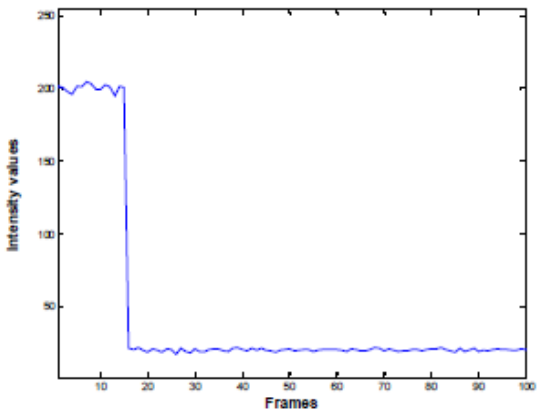


Figure 2.2: Ejemplos valores de píxel durante una secuencia de entrenamiento [30]

2.3.1. Temporales

La información temporal de la secuencia de entrenamiento se analiza desde dos puntos de vista: estabilidad y continuidad.

La estabilidad temporal analiza la variación de intensidad de un píxel/región en la secuencia de vídeo. Este criterio establece la elección de un píxel o región como fondo de escena o como candidato a fondo de escena si la variación de las intensidades de esos píxeles/regiones entre imágenes no supera un umbral sean éstas consecutivas o no consecutivas [30]. Frecuentemente, se el uso aislado de este criterio impone que el fondo de escena comprenda el intervalo más estable de la secuencia de entrenamiento (en ocasiones hasta el 50 % del tiempo total). Así pues, en aquellos píxeles/regiones donde aparezca un objeto de primer plano estático más tiempo que el verdadero fondo de escena las aproximaciones basadas en estabilidad temporal serán incapaces de reconstruir el fondo de escena correctamente. Por ejemplo, Hou et al [31] buscan los intervalos estables, entendiendo por estables aquellos en los que la intensidad del píxel no supera un determinado valor medio más un umbral,y hace una media de ellos. En la figura 2.2 se muestra la intensidad de un píxel a lo largo de una secuencia de 100 imágenes donde se busca identificar los valores de intensidades estables para construir el fondo. Se observan dos intervalos diferenciados donde cada uno puede dar lugar a un candidato a fondo de escena (o seleccionar el segundo valor por ser estable durante más tiempo).

Continuidad temporal implica en algoritmos que buscan estabilidad temporal, si dicha estabilidad es encontrada en imágenes consecutivas. Es el caso de los que por ejemplo utilizan ventanas deslizantes para obtener candidatos temporales a la imagen de fondo [30]. Existen aproximaciones donde por el contrario no necesariamente se busca la estabilidad en imágenes consecutivas como en [32], donde el candidato temporal puede formarse con una imagen del principio y otra del final. En la figura 2.3 se observa como los valores de los píxeles no son esta-

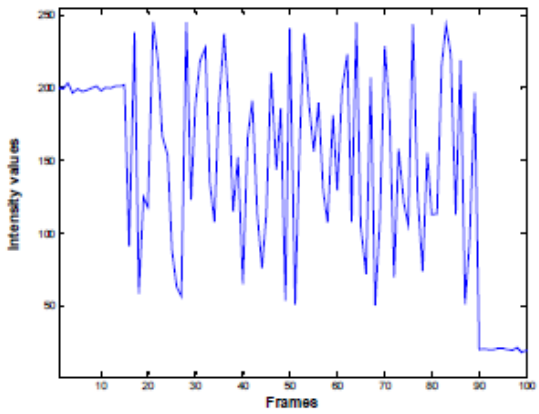


Figure 2.3: Ejemplo valores de píxel donde no existe continuidad temporal

bles en imágenes consecutivas, sin embargo se pueden encontrar valores de intensidades estables para imágenes no consecutivas en el tiempo.

2.3.2. Espacial

La información espacial para la inicialización de fondo implica utilización de información de los píxeles/regiones adyacentes que han sido previamente clasificados como fondo de escena para decidir si un píxel/región es o no fondo de escena [12] [8] [33] [17] [34] [35][36]. Normalmente existe una etapa temporal previa que aporta la primera región de fondo de escena llamada semilla. Añadir la información espacial sirve para construir el fondo alrededor de dicha región atendiendo a la continuidad espacial que debe tener el fondo de escena. Solventa el problema de los objetos de primer plano que permanecen estáticos en el fondo de escena. En [12] construyen el fondo a partir de la semilla atendiendo a la continuidad espacial medida con una DCT. En [37] usan una SVM (Support Vector Machine) basada en flujo óptico y frame difference para obtener el fondo de escena. En [33] para medir la continuidad espacial usan la transformada de Hadmard. En la figura 2.4 observamos en la primera columna un fondo de escena con una región sin reconstruir, en la segunda columna se muestran los candidatos a ocupar dicha región, mientras que en la tercera y cuarta columnas aparecen el resultado de medir la DCT para la imagen inicial con el candidato uno y dos respectivamente ocupando la región de análisis. Por tanto se mide la continuidad con el fondo de escena ya existente.

2.3.3. Otras

Adicionalmente, existen otras características

- Uso de información de primer plano. Si para la inicialización del fondo es necesaria la segmentación de primer plano. La ventaja de extraer el primer plano es que hace el algo-

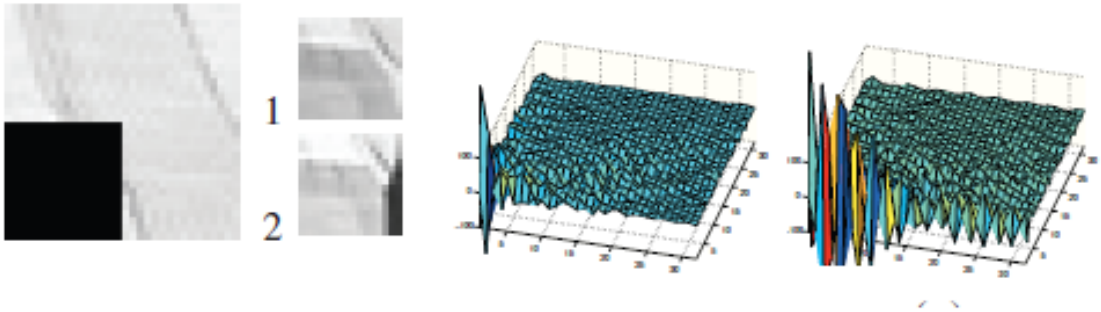


Figure 2.4: Ejemplo de continuidad espacial [12]

ritmo mucho más robusto pues permite gran cantidad de movimiento al tener identificado el frente. Sin embargo el coste computacional es mayor así como la complejidad. No soluciona el problema de los fantasmas o *sleeping person* (objetos estáticos que pasado cierto tiempo inician el movimiento) ni el fondo multimodal (olas, árboles,...). En [38] [39] aplican una segmentación a las imágenes de entrada para obtener el primer plano y fondo de escena y desestimar el primer plano en el análisis del fondo de escena.

- Gris/Color. Define la cantidad de información visual almacenada para cada píxel en un sólo canal a nivel de grises o típicamente en tres canales RGB (color). Por ejemplo la aproximación realizada en [30] se podría realizar tanto en escala de gris como en color. La ventaja de utilizar color es que se utiliza mayor cantidad de información para realizar el análisis, por ejemplo en el caso de de buscar píxeles estables ya que hay más información en tres canales que en uno. Sin embargo al hacerlo en escala de grises se reduce el coste computacional y la memoria necesaria pues sólo debe almacenar un canal. Reddy et al [12] proponen un algoritmo con imágenes de entrada a nivel de gris, ello beneficia el coste de una primera etapa donde primeramente estudia la continuidad/estabilidad temporal y posteriormente utiliza la DCT para estudiar continuidad espacial.

- Píxel/Región. Si el algoritmo analiza de manera independiente cada píxel [30] [40], región [12] o una combinación de ambos [19]. El uso de regiones implica mejor detección de movimiento en el fondo y permite incrementar la robustez de la etapa de continuidad espacial. Sin embargo el nivel de región requiere decidir su tamaño y un mayor coste computacional pues se tienen en cuenta varios píxeles, que dificulta el proceso de continuidad temporal. Para ello se utilizan estrategias de similitud de regiones como el test como la chi_cuadrado sobre una diferencia de regiones [8].

- Soporta primer plano estático. Este fenómeno es conocido como efecto de fantasmas o

Figure 2.5: Ejemplo objeto de primer plano estático durante la secuencia de entrenamiento.

efecto de *sleeping person*, un algoritmo capaz de soportarlo detecta si un objeto de primer plano está parado durante el tiempo de inicialización/captura del fondo detectándolo como frente [17]. Éste es una de las características más importantes para una inicialización robusta. Requiere utilizar secuencias de entrenamiento largas (y mucha memoria) pues es necesario guardar cada observación del píxel o región en t como candidato a fondo de escena ya que es posible que el fondo de escena aparezca en un corto periodo de tiempo. La figura 2.5 muestra una secuencia de entrenamiento donde el autobús no pertenece al fondo de la escena pero se muestra estático.

- Output. Si los algoritmos producen como salida un valor de intensidad o una función de densidad de probabilidad (fdp) para representar cada píxel [35]/región [41]. Con una fdp se puede modelar el movimiento del fondo (importante en fondo multimodales) pero un píxel muestra el verdadero valor del mismo si se graba con una cámara en interior y estática.

- Visibilidad fondo. Es una de las características mas importantes, si es necesario que el fondo de escena aparezca durante un mínimo número de imágenes en la secuencia de

Figure 2.6: Ejemplo escasa visibilidad del fondo de escena

entrenamiento. Cuanta menor visibilidad necesite más robusto será pero serás más costoso computacionalmente. La aproximación de Wang et al [30] soporta más del 50% de primer plano mientras que en [31] necesitan que el fondo de escena sea visible más del 50% en cada píxel. La figura 2.6 muestra una secuencia de entrenamiento con escasa visibilidad de fondo de escena.

- Movimiento. Si el algoritmo establece flujo óptico para cada par de imágenes de la secuencia de entrenamiento [34] [35]. Permite predecir el patrón de movimiento de cada píxel ayudando a identificar un píxel como fondo de escena o primer plano Esta característica falla cuando no hay una diferencia evidente entre movimiento de primer y segundo plano o cuando hay excesivo movimiento. En la figura 2.7 se muestra como para un píxel de referencia el flujo óptico puede ser utilizado para detectar objetos de primer plano que ocupan el fondo de escena (figura izquierda) o el caso contrario donde los objetos de primer plano liberan el fondo de escena (figura derecha).

2.4. Modelos de inicialización de fondo

Existen distintos modelos de inicialización de fondo en el estado del arte, dentro de ellos se agrupan los algoritmos existentes [15]. La tabla 2.2 divide los modelos de fondo existentes en tres grandes grupos: básicos, aprendizaje estadístico y aproximaciones espacio temporales.

2.4.1. Modelos básicos

- *Median filtration* [42]. La inicialización del fondo usando el valor de la mediana de la intensidad para cada píxel. Se utiliza en sistemas de monitorización de tráfico basándose

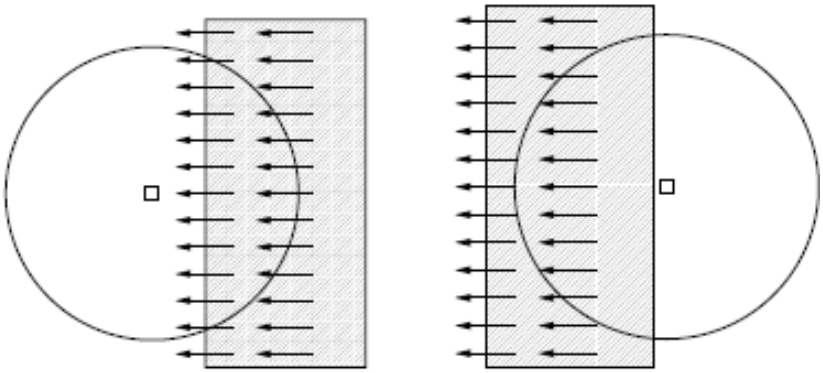


Figure 2.7: Flujo óptico para un píxel de referencia [35]

	Modelos
Básicos	*Median filtration, Stable intensity extraction, Relative constant intensity extraction*
Aprendizaje estadístico	*Mixture of Gaussian Distribution, HMM, Codebook-based, Statistical Aproach, KDE*
Híbridos	*Espacio-temporal*

Table 2.2: Clasificación modelos de inicialización de fondo

en el supuesto de que cada píxel de fondo será visible durante más de un 50% del tiempo en la secuencia de formación. La ventaja de usar la mediana a la media es que evita la mezcla de valores de los píxeles. La media de la intensidad de un píxel puede no corresponder a un valor real del píxel durante un tiempo. En la figura 2.8 se observan los fallos de este tipo de modelos. Cuando el fondo no aparece más de un 50% la imagen de fondo de escena aparece emborronada.

- *Stable Intensity Extraction* [43]. Este método elige el periodo más largo, aplicando la continuidad temporal, como el más probable para representar el fondo. Funciona bien cuando todos los objetos en primer plano se mueven. Si se paran durante un tiempo muchos píxeles se clasificarán incorrectamente como fondo.

- *Relative Costant Itensity Extraction* [35]. Añade al algoritmo anterior una etapa basada en flujo óptico después de formar candidatos temporalmente estable. Evita el problema de objetos de primer plano estáticos. La ventaja del algoritmo es que la decisión en cada píxel es independiente de sus vecinos y que no sólo está basada en los valores observados en el pasado del píxel sino también en la información del movimiento. Es flujo óptico hace que

Figure 2.8: Ejemplo de extracción de fondo de escena con modelos básicos (mediana) [38]

este tipo de algoritmos sean más costosos computacionalmente que algoritmos anteriores. Son sensibles al ruido de cámara.

2.4.2. Modelos de aprendizaje estadístico

Acorde con [44] existen distintos modelos de inicialización de fondo de escena:

- *Background with A Mixture of Gaussian Distributions.* Hay varios métodos disponibles para la construcción de tal modelo de mezcla. Un algoritmo utilizado es la maximización de las expectativas, que utiliza un proceso iterativo para encontrar la mejor mezcla de Gaussianas para un conjunto de datos particular.

- *Background Model Based on Kernel Density Estimation.* En [26] Elgammal et al estiman la función de densidad de la de distribución de píxeles en un momento reciente esperando obtener un descubrimiento susceptible. La función densidad de probabilidad permitirá obtener un valor de intensidad para un píxel en un tiempo t al final de una secuencia de entrenamiento.

- *Modelos ocultos de Markov* [45] [46]. Se puede utilizar HMM para representar el proceso de píxel donde sus estados pueden representar diferentes variaciones que puedan producirse en el procesado de píxeles, tales como fondos de escena, en primer plano, las sombras, iluminación de día y de noche. En [19] proponen un algoritmo de inicialización que integra el nivel de píxel y región. Para el nivel de píxel múltiples hipótesis de fondo de escena son generadas modelando la intensidad con un HMM y también capturando secuencialmente la diferencias de intensidades. En el nivel de región los HMMs resultantes son agrupados

con una nueva medida que permita eliminar los objetos de primer plano de una secuencia y obtener una imagen segmentada.

- *Codebook-Based.* En [47] [48] cuantifican los valores de fondo de cada píxel en grupos de palabras clave que constituyen un libro de código para cada píxel. Al principio, el libro de códigos de un píxel está vacío sin palabras clave. Cuando una nueva muestra para un píxel es encontrada (en el período de formación), si no existen palabras en clave en el libro de códigos para ese píxel, se asume que es una palabra en clave, y su valor de brillo se utiliza para estimar las medidas que representan la palabra en clave. Si hay palabras codificadas en el libro de códigos, esta muestra de píxel se compara con cada palabra clave en el libro de códigos utilizando como medida la distorsión de color y el brillo, si es semejante a una palabra clave, su valor de brillo se utiliza para actualizar las medidas de esta palabra clave, si no hay ninguna coincidencia, se asume que es una palabra en clave nueva, y así sucesivamente hasta el final del período de formación. Posteriormente se elige un valor del *codebook* para cada píxel atendiendo a distintas medidas,

- *Statistical Approach.* La inicialización de un modelo de fondo podría ser abordada estadísticamente como la tarea de obtener una imagen de fondo de escena en secuencias con gran cantidad de objetos en primer plano .Así puede tolerar más del 50% del ruido en los datos, en contraste con los métodos de uso de la mediana los cuales no funcionarán cuando el fondo constituye menos del 50% de los datos de entrenamiento [49]. [41] se basa en la idea de que los valores de un píxel que más ocurren para un tiempo $t-1$ deberían predecir lo que ocurre en un tiempo t. Para ello sugieren que que hay que encontrar el valor de intensidad de un píxel que tuvo máxima probabilidad en ese tiempo $t-1$. [49] [30] introducen un método robusto de inicialización de fondo para superar los problemas inherentes en los métodos basados en la media, mediante el empleo de dos pasos. Primero se encuentran todas las sub-secuencias estables que no se superponen, utilizando una ventana deslizante con una longitud mínima, si la sub-secuencia candidata con la longitud mínima predefinida no se puede encontrar, otro de longitud mínima se utiliza, observando que incluso después de este paso, la sub-secuencia elegida puede contener píxeles de primer plano, fondo sombras, luces, etc. El segundo paso es considerado un paso crucial donde la sub-secuencia más fiable es elegida, ello motivó la definición de la fiabilidad dada por [50], y la utilización del valor medio de cualquier nivel de intensidad de gris o de las intensidades de color sobre la sub-secuencia como el valor de fondo del modelo. Por ejemplo en [30] primero buscan sub-intervalos estables l_k con una ventana deslizante L_w, $x_{l_k}(t)$ es el valor del píxel x en la kth sub-secuencia en el tiempo t.

Figure 2.9: Ejemplo de extracción de fondo modelos de aprendizaje estadístico.

$$(t-1,t) \in l_k, \left\{ \begin{array}{c} |x_{l_k}(t) - x_{l_k}(t-1)| \leq T_f \\ x_{l_k}(t) - \bar{x}_{l_k}(t-1)| \leq T_f \end{array} \right\} \tag{2.1}$$

donde experimentalmente $L_w = 5$ y $T_f = 10$. $\bar{x}_{l_k}(t-1)$ es el valor medio desde el principio de la sub-secuencia l_k hasta tiempo $t-1$, posteriormente elijen la mejor sub-secuencia para representar el fondo de escena como:

$$\hat{l}_k = argmax_k(n_{l_k}/S_{l_k}) \tag{2.2}$$

donde n es el número de muestras en la secuencia kth y S la varianza de las mismas. En la figura 2.9 se observa como este modelo no soporta objetos de primer plano estáticos durante la mayor parte de la secuencia de entrenamiento.

2.4.3. Modelos híbridos

Estos modelos proponen un análisis espacio-temporal donde primeramente se seleccionan candidatos estables temporalmente (agrupaciones, *clusters*) y posteriormente se estudian continuidades espaciales entre ellos y el fondo de escena existente [33] [8] [12] [36].

- En [12] primero hacen un análisis temporal a nivel de región para determinar cual es el bloque (conjunto de píxeles) más estable (semilla/s) sobre el que hacen crecer el fondo mediante una continuidad espacial con el fondo de escena existente teniendo en cuenta a su vez el resultado temporal. Para ello definen las regiones como $B_f(i,j)$ (bloques cuadrados de 16x16 píxeles) donde $f \in [1,...,f]$ imágenes de la secuencia. Para cada posición (i,j) definen un bloque representativo $R(i,j)$ que inicializan a cero. Después para cada bloque

$B_f(i,j)$ si es el primero se introduce en $R(i,j)$, sino se compara con los $R(i,j)$ existentes mediante:

$$\{(r_k(i,j) - \mu_{rk}(i,j))'(b_f(i,j) - \mu_{bf}(i,j))\} / \{\theta_{rk}\theta_{bf}\} > T1 \tag{2.3}$$

y

$$1/N^2 \sum_{n=0}^{N^2-1} |d_{kn}(i,j)| < T2 \tag{2.4}$$

.

Donde $r_k$es un *cluster* temporal existente en $R(i,j)$ convertido a vector donde se promedian los bloques b_k que cumplen lo anterior, donde b_k es la observación de $B_f(i,j)$ convertido a vector. Para un $R(i,j)$ nuevo se le asocia un peso $W = 1$, cuando un b_k coincide con un r_k el peso se actualiza como $W = W + 1$. $T1 = 0.8$ y $T2 = [0.5 : 4]$ son seleccionados empíricamente. μ_{rk}, μ_{bf}, θ_{rk}, θ_{bf} son la media y varianza de r_k y b_f respectivamente. De esta forma obtienen todos los posibles candidatos a ser fondo de escena. Posteriormente tienen una semilla de fondo (el más estable temporalmente, cuyo $R(i,j)$ tenga dimensión uno, es decir todos los bloques se hayan promediado) a partir de la cual hacen crecer el resto. Para ello actúan de la siguiente manera: para una posición (x,y) que aun no tiene fondo y existe el mismo en al menos dos de sus bloques vecinos, forman una nueva imagen con sus 8-vecinos colindantes y el bloque de análisis lo inicializan a cero, aplican la 2dct y los coeficientes resultantes los escriben en C sin la componente continua. Aplican la 2DCT al bloque de análisis y los coeficientes sin la componente continua lo escriben en D. El bloque que representa mejor el fondo es aquel que minimiza la función de coste:

$$cost(k) = (\sum_{v=0}^{M-1} \sum_{u=0}^{M-1} |C(v,u) + D_k(v,u)|)\lambda_k \tag{2.5}$$

donde $\lambda_k = e^{-\alpha w_k}$, $\alpha \in [0,1]$, $w_k = W_k / \sum_{k=1}^{S} W_k$.

En la figura 2.10, se muestra el procedimiento por el cual a partir delas semillas obtenidas (bloques constantes en la secuencia de entrenamiento) se hace crecer el resto del fondo de escena. Se observa como de un bloque temporalmente estable se construye una imagen de fondo atendiendo a la continuidad espacial entre bloques.

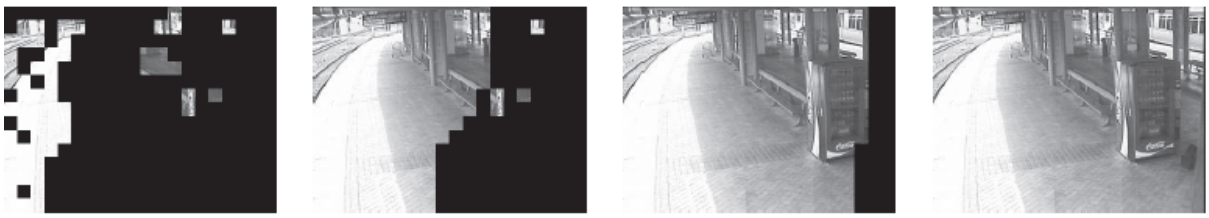


Figure 2.10: Crecimiento de una imagen de fondo a partir de una semilla [12]

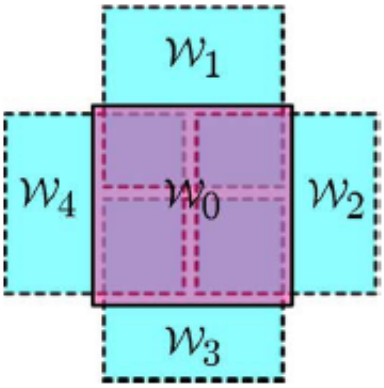


Figure 2.11: Ejemplo continuidad espacial con bloques solapados

- En [8] siguen el esquema de [12] conseguir una imagen de fondo de escena a partir de una semilla encontrada en un análisis temporal. Utilizan regiones pero en este caso solapadas. La figura 2.11 muestra la idea de solapamiento de regiones. Cada bloque W_i es una ventana o bloque. En las zonas moradas hay solape entre dos bloques y es en esas zonas donde aplican una función chi-cuadrado para asegurar la continuidad del fondo de escena. Posteriormente en las zonas turquesas aplican una diferencia de color entre candidatos para asegurar la continuidad con la zona solapada.

La estabilidad temporal la estudian con una función chi-cuadrado que acepta o rechaza el candidato en función de los grados de libertad teniendo el cuenta el ruido de la cámara.

La imagen es dividida en W_i que son bloques de NxN píxeles donde $N = 30$. $v_s = W_i x[1, ..., L]$, es una sub-imagen a lo largo de una secuencia temporal

$$SSD(W, t_1, t_2) = 1/(2\sigma^2) \sum ||v_{x,y,t_1} - v_{x,y,t_2}||^2 \tag{2.6}$$

un bloque en semejante a otro si:

$$SSD(W, t_1, t_2) < x^{-1}_{3N^2}(\alpha) \tag{2.7}$$

donde α es un nivel de confianza, $\alpha = 0.99999$.

Si dos o más bloques cumplen la función anterior forman un *clusters* k sobre un bloque W promediando dichos bloques:

$$U_{x,y,k} = 1/|T_k| \sum_{t \in T_k} v_{x,y,} \in W \tag{2.8}$$

Posteriormente eligen la semilla del fondo de escena (el más estable) y hacen crecer el fondo a su alrededor. En este caso en lugar de la DCT usan el solape de las ventanas con una función chi-cuadrado como continuidad espacial más una etapa de *graph cut* entre candidatos en la parte no solapada.

Para la parte solapada:

$$SSD(W_0 \bigcap W, k_0, k) = 1/(\sigma^2_{k0} + \sigma^2_k) \sum_{W_0 \bigcap W} ||u_{x,y,k_0} - u_{x,y,k}||^2 \tag{2.9}$$

donde W_0 es el bloque que es fondo de escena y W el que se quiere comprobar.

Si cumple la función aceptación chi-cuadrado:

$$SSD(W_0 \bigcap W, k_0, k) < x^{-1}_M(\alpha) \tag{2.10}$$

Donde $M = W_0 \bigcap W$ se comprueba la similitud espacial usando una técnica de *graph cut* no descrita correctamente en el artículo [8]. Los candidatos que cumplen la primera etapa de solapado entran en un *round-robin*. Se parte de una imagen diferencia binarizada entre candidatos donde se evalúan los contornos de la binarización. En la figura 2.12 se observa como de la semilla obtenida en la etapa temporal se construye el fondo a partir de la continuidad espacial explicada anteriormente.

2.5. Evaluación en inicialización de fondo

2.5.1. Métricas

Las técnicas más utilizadas en la literatura existente [30], [12] y [35] para la evaluación de inicialización de fondo son:

- Number of Error pixels (NE). Número de píxeles erróneos. Un píxel erróneo es aquel cuyo valor de intensidad obtenido en la imagen de fondo difiere en δ unidades del valor real del fondo o *ground truth* (GT) en escala de grises.

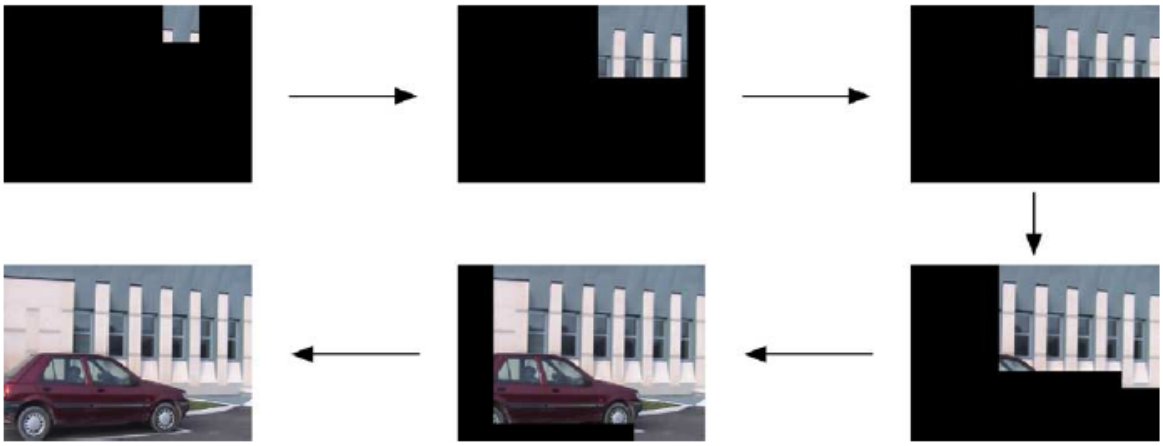


Figure 2.12: Ejemplo crecimiento de una imágen de fondo de escena a partir de semilla [8]

$$Error(x,y) = \left\{ \begin{array}{ll} 1, & si \quad |BG(x,y) - GT(x,y)| > \delta \\ 0, & resto \end{array} \right\} \tag{2.11}$$

$$NE = \sum_{x=1}^{W} \sum_{y=1}^{H} Error \tag{2.12}$$

donde $BG(x,y)$ es la intensidad de un píxel en una posición de la imagen de fondo obtenida, $GT(x,y)$ es la intensidad de un píxel en una posición de la imagen de *ground truth*, W es el ancho de la imagen, H es el alto de la imagen y δ es el umbral de decisión (típicamente $\delta = 20$) en escala de grises.

- *Average gray-level Error* (AE). Error medio del nivel de gris. Es el porcentaje de píxeles erróneos con respecto en la imagen de fondo de escena obtenida.

$$AE = \frac{NE}{W * H} \tag{2.13}$$

- *Number of Clustered error píxels* (NC). Este error se produce cuando los 4 vecinos conectados a un píxel de error también son erróneos. Esta medida es la más relevante pues un error en ella indica que posiblemente se debe a un objeto de primer plano.

$$ErrorCluster(x,y) = \left\{ \begin{array}{ll} 1, & si \quad \sum_{=x-1}^{x+1} \sum_{y=y-1}^{y+1} Error(x,y) == 4 \\ 0, & resto \end{array} \right\} \tag{2.14}$$

$$NC = \sum_{x=1}^{W} \sum_{y=1}^{H} ErrorCluster(x, y) \tag{2.15}$$

Para obtener las medidas primero generan el *Reference Frame* (RF) para cada secuencia de test usando la media de las imágenes que estén libres de objetos de primer plano (manualmente seleccionadas).

Existen otras formas de evaluación:

- En [9] a pesar de ser de mantenimiento de fondo y no de inicialización dan otra manera de evaluar los algoritmos. Consiste en la clasificación en términos de falsos negativos (píxeles de primer plano que se clasifican como fondo de escena) y falsos positivos (píxeles de fondo de escena marcados como primer plano). Para ello es necesario el *Ground truth* donde los objetos de primer plano son marcados a mano y con el cual se comparan los algoritmos. Para poder utilizar esta clasificación haría falta detectar tanto fondo como frente.

2.6. Datasets disponibles

Para esta tarea, la disponibilidad de conjunto de datos para evaluar es reducida.

En [30] hay disponibles dos secuencias semejantes formadas por imágenes de 320x240 píxeles. La primera de ellas está formada por 600 imágenes y la segunda por 900. Son secuencias interiores con personas caminado por una sala. El problema de las secuencias se reduce a obtener un fondo de escena libre de objetos de primer plano. Podemos encontrar las secuencias en http://www.ecse.rpi.edu/~cvrl/humanbody.

En [8] utilizan seis secuencias para la evaluación de los algoritmos. Tres de ellas solventan el problema de gran cantidad de primer plano: una esta formada por 349 imágenes de 320x240 píxeles. En ella aparecen personas andando y unas hojas sintéticas durante más del 90% del tiempo. En los otros dos vídeos de 150x146 píxeles con 333 imágenes y 200x148 píxeles con 400 imágenes respectivamente en las cuales aparece primer plano sintético durante la mayor parte de la secuencia. Encontramos una secuencia de 200x136 píxeles con 257 imágenes donde el problema es que una persona se queda parado más de la mitad del tiempo en la una posición fija de la imagen. En el mismo *dataset* encontramos una secuencia de 200x164 con 227 imágenes donde lo característico son los cambios de iluminación y las sombras al igual que en la última de las secuencias formadas por 462 imágenes de 174x200 píxeles. Las podemos encontrar en http://profs.sci.univr.it/~fusiello/demo/bkg/.

En [9] encontramos la secuencia bootstrap formada por 3000 imágenes de 160x120 píxeles, El principal problema es la cantidad de objetos (personas) de primer plano moviéndose en la

escena así como la aparición del efecto *sleeping person.* Esta secuencia se puede obtener de: http://research.microsoft.com/en-us/um/people/jckrumm/wallflower/testimages.htm

Adicionalmente, existen aproximaciones que utilizan *datasets* genéricos de vídeo-seguridad como CAVIAR [12] o ETISEO [17] que contienen multitud de secuencias con tamaños y problemas variados. No obstante, estos *datasets* no abordan el problema de inicialización de fondo y consecuentemente, no permiten realizar una evaluación exhaustiva de las ventajas y desventajas del algoritmo.

Capítulo 3

Algoritmo propuesto

3.1. Introducción

Después del estudio de los diferentes algoritmos existentes en el estado del arte se concluye que las mejores puntuaciones en algoritmos de inicialización de fondo de escena en entornos poblados se consiguen con algoritmos a nivel de región mediante la fusión de dos etapas, una temporal y una consecutiva espacial [12], [33], [8]. En una primera etapa temporal se detectan los posibles candidatos a fondo de escena (aquellos estables temporalmente) mientras que en una segunda etapa se hace crecer el fondo atendiendo a cierta continuidad espacial con una cierta suposición de fondo inicial.

En este capítulo se describe el algoritmo desarrollado (sección 3.2). Para ello se empezará mostrando un esquema general del algoritmo introduciendo las etapas del mismo y sus parámetros más importantes como son el estudio del tamaño óptimo de la región de análisis (sub-sección 3.2.1), el agrupamiento temporal donde se forman candidatos a fondo de escena mediante la fusión de imágenes en las regiones obtenidas (sub-sección 3.2.2) y la continuidad espacial (sub-sección 3.2.3) donde se introducirá la manera en que se decide cuál de los candidatos formados es el ideal en una determinada región. Por último analizaremos las ventajas de la organización de la aproximación en estas tres etapas (sección 3.3)

3.2. Esquema del algoritmo

Se ha seleccionado una aproximación basada en regiones debido a su superioridad sobre algoritmos a nivel de píxel (ver capítulo 2). Gracias a este tipo de aproximación se podrá resolver tanto el problema de inicialización de fondo de escena en secuencias con gran cantidad de objetos de primer plano como el relacionado con sombras de objetos. El algoritmo propuesto consta de tres partes diferenciadas (figura 3.1): estimación de la región óptima de análisis, agrupamiento temporal y continuidad espacial. En la primera se define el tamaño óptimo de las

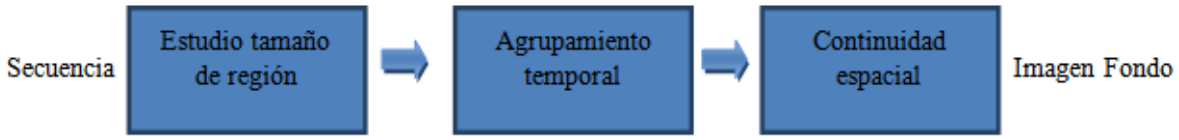


Figure 3.1: Esquema del algoritmo propuesto para la inicialización de fondo

regiones, en la segunda se agrupan las regiones estables temporalmente (*clusters*) para formar candidatos a fondo de escena y en la última se decide que *cluster* es el más apropiado para componer el fondo. La entrada al algoritmo es una secuencia de entrenamiento compuesta de T imágenes en color $I_t \in [I_1, ..., I_T]$ y a la salida se obtiene una imagen de fondo de escena donde cada píxel es representado con un valor. A continuación se describen dichas partes.

3.2.1. Estimación del tamaño de la región de análisis

Esta etapa utiliza la secuencia de entrenamiento completa para estimar el tamaño de las regiones que se analizarán en posteriores etapas. Para cada región, se obtendrá su tamaño óptimo, su posición dentro de la imagen e indicadores de estabilidad (un número máximo y mínimo de sub-secuencias estables) mediante la técnica *frame difference* seleccionando un umbral adaptativo.

Para ello lo primero es definir el tamaño las regiones de análisis, a partir de ahora bloques. Definimos un bloque $B_t^k(i,j)$ como la región k de I_t (imagen t de la secuencia de entrenamiento) delimitada por las coordenadas $i \in [x_k^{min}, x_k^{max}]$ e $j \in [y_k^{min}, y_k^{max}]$ (de los canales RGB para representar el color). Estas regiones serán de tamaño $W_k \times H_k$, siendo el ancho $W_k = x_k^{max} - x_k^{min}$ y el alto $H_k = y_k^{max} - y_k^{min}$. En este trabajo solo se consideran regiones cuadradas ($W_k = H_k$) definidas con distintos tamaños constantes a lo largo del tiempo. La figura 3.2 muestra como una imagen de entrada I_t es inicialmente dividida, según la posición, en bloques B_t^k de igual tamaño.

Dependiendo del tamaño de bloque el algoritmo obtiene resultados variables para una secuencia de entrenamiento determinada. El principal motivo es la cantidad de objetos de primer plano en la secuencia de entrenamiento así como el tamaño de los mismos. Cuanto mayor es el tamaño de bloque hay más probabilidad de que aparezcan objetos de primer plano en dicho bloque y por tanto será más difícil visualizar el fondo. No obstante, el uso de tamaños grandes permite realizar agrupaciones temporales mejor definidas y en menor cantidad con lo que se reduce el coste computacional de etapas posteriores. Sin embargo tamaños demasiado pequeños de bloque permiten visualizar el fondo con mayor facilidad a cambio de aumentar el coste com-

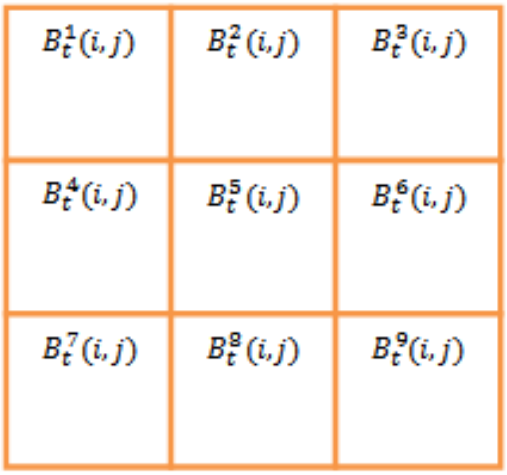


Figure 3.2: División inicial de una imagen de entrada en bloques

putacional del algoritmo. Por ello es importante estimar el tamaño óptimo de cada bloque entre unos valores máximos y mínimos.

3.2.1.1. Tamaño máximo y mínimo de los bloques

Partimos de una secuencia de entrenamiento con T imágenes $I_t(t = 1...T)$ y se buscan los tamaños máximos y mínimos para los bloques $B_t^k(i,j)$. Para poder decidir este tamaño hay que tener en cuenta que el objetivo es que sea el apropiado para la posterior etapa de agrupamiento temporal. Para ello se realizó un estudio previo para encontrar el tamaño máximo y mínimo óptimo (ver capítulo 6). El estudio consistió en medir tanto los tiempos de ejecución en la etapa de agrupamiento como en validar el funcionamiento de las agrupaciones formadas en la siguiente sub-etapa de agrupamiento temporal para distintos tamaños de bloque pues juntos son los parámetros que deben guiar en la elección del tamaño máximo y mínimo. El resultado de dicho estudio dio lugar a la elección de $W_{max} = H_{max} = 40$ y $W_{min} = H_{min} = 5$.

3.2.1.2. Tamaño óptimo mediante Frame difference adaptativo

Una vez han sido definidos el valor máximo y mínimo de bloque, para cada bloque B_t^k se persigue obtener su tamaño optimo y su número de segmentos estables (máximo y mínimo) a lo largo de la secuencia de entrenamiento. El número de segmentos estables es utilizado en la etapa de agrupamiento temporal para formar los candidatos a fondo de escena (capítulo 4). Para ello utilizamos un *frame difference* cuyo umbral es determinado adaptativamente.

Partiendo de la secuencia de entrenamiento $I_{1...T}$, se buscan bloques de tamaño tal que no exista movimiento entre dos imágenes consecutivas. Asumiendo así que donde es capaz de

encontrarlos se podrá formar un candidato estable al menos con esas dos imágenes. Para buscar el movimiento se utiliza técnica de *frame difference* que viene dada por:

$$fd_t = |I_t - I_{t-1}| > \tau \tag{3.1}$$

donde τ es un umbral de decisión que se estima de manera adaptativa para cada fd_t calculada. En cuanto a la obtención del mismo existen en el estado del arte varias opciones, entre las más comunes y las que ofrecen mejores resultados encontramos las aproximaciones de Otsu, Rosin y Kapur (sub-sección 3.2.1.3).

Una vez realizado el *frame difference* para toda la secuencia, se realiza un barrido de la imagen desde la esquina superior izquierda intentando encontrar bloques sin movimiento del mayor tamaño posible entre el valor máximo (40) y mínimo (5) determinados en la fase anterior. Así pues encontramos un tamaño óptimo cuando se cumple:

$$\sum_{i=x_k^{min}}^{x_k^{max}} \sum_{j=y_k^{min}}^{y_k^{max}} fd_t(i,j) = 0 \,\text{para cualquier}\, t \tag{3.2}$$

En el caso afirmativo, almacenamos las coordenadas del bloque k-ésimo B_t^k y se procede a analizar el siguiente bloque $k+1$ (comenzando con el tamaño máximo) hasta cubrir completamente la imagen:

$$\begin{gathered} k \Longrightarrow [x_k^{min}, x_k^{max}, y_k^{min}, y_k^{max}] \\ k = k+1 \end{gathered} \tag{3.3}$$

Si no es capaz de encontrar sub-intervalos estables reduce su tamaño a la mitad ($x_k^{max} = x_k^{max}/2$ e $y_k^{max} = y_k^{max}/2$) y se repite el proceso iterativamente hasta llegar al mínimo tamaño de bloque ($W_{min} = H_{min} = 5$) . Si se llega al tamaño mínimo y no es capaz de encontrar $fd_t(i,j) = 0$ se considera que el bloque no puede ser reconstruido. En los extremos de la imagen, si la misma no es divisible por $W_{max} = H_{max}$, el bloque será del tamaño máximo posible y el procedimiento será el mismo.

En la figura 3.3 se puede observar el resultado tras la etapa de búsqueda del tamaño óptimo de bloques. En el bloque de abajo a la izquierda el *frame difference* no ha encontrado intervalos estables y ha dividido por la mitad el tamaño máximo de W y H dando lugar a las regiones $B_t^7(i,j)$, $B_t^8(i,j)$, $B_t^9(i,j)$ y $B_t^{10}(i,j)$.

Para obtener los valores mínimo y máximo de segmentos estables (*clusters)* para un bloque dado se procede de la siguiente manera: en la secuencia de entrenamiento cada vez que el *frame*

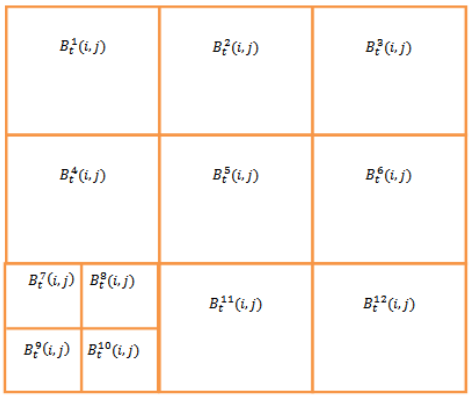


Figure 3.3: Estimación del tamaño de las regiones después del *frame difference.*

difference encuentra una secuencia estable de n imágenes consecutivas, es decir durante esas imágenes $fd_{t_1 \ldots t_n} = 0$, asume que las imágenes son la misma y que por tanto todas formarán el mismo *cluster*. Cada vez que se produzca un fd_t distinto de cero suma uno al número máximo de *clusters* y será la siguiente etapa temporal la encargada de decidir si ese intervalo es o no semejante a algún otro no consecutivo en el tiempo. Si la secuencia tiene T imágenes y se obtienen x intervalos estables donde cada x esta formado por n_x imágenes:

$$N_{cmin} = T - \sum_{i=0}^{x} n_x \tag{3.4}$$

$$N_{cmax} = T - x \tag{3.5}$$

Por tanto al final de esta etapa se obtiene una partición de la imagen en k bloques donde para cada uno de ellos se almacenan sus coordenadas $[x_k^{min}, x_k^{max}, y_k^{min}, y_k^{max}]$ y un número máximo y mínimo de *clusters* (N_{cmax} y N_{cmin}). La figura 3.4 resume el esquema del análisis previamente explicado.

3.2.1.3. Obtención del umbral adaptativo

Para obtener el umbral τ a aplicar en la técnica *frame difference* de manera adaptativa dependiendo de la secuencia de entrada, proponemos utilizar algoritmos clásicos de umbralización automática. Entre los existentes, se han seleccionado inicialmente los algoritmos de Otsu, de Kapur y de Rosin y tras un estudio previo (ver capítulo 6 y apéndice A), se ha seleccionado el algoritmo de Kapur.

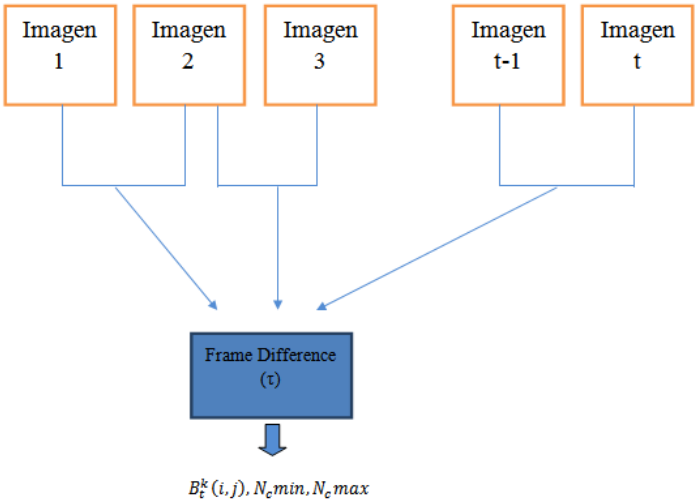


Figure 3.4: Diagrama de bloques del frame difference

La entrada a la sub-etapa es la resta de dos imágenes consecutivas de la secuencia de entrenamiento y la salida el umbral τ. El algoritmo de Kapur utiliza la entropía de la imagen. Considera la imagen diferencia como dos clases de eventos cada uno de ellos caracterizado por una función de densidad de probabilidad (fdp). A continuación el método maximiza la suma de la entropía de las dos fdps para converger a un único valor umbral [51]. En [52] Kapur fue el algoritmo de mejor rendimiento tanto cuantitativa como cualitativamente. Recibió las puntuaciones sustancialmente mayores que los otros, y en la inspección visual fue mejor. La probabilidad de los niveles de grises sobre la clase 1 (parte negra de una imagen diferencia (movimiento)) viene dada por:

$$\frac{p_0}{p_B}, \frac{p_1}{p_B}, ..., \frac{p_s}{p_B} \tag{3.6}$$

Y la probabilidad de la clase 2 (parte blanca de una imagen diferencia (no movimiento)) como:

$$\frac{p_{s+1}}{1-p_B}, \frac{p_{s+2}}{1-p_B}, \frac{p_{n+1}}{1-p_B} \tag{3.7}$$

s es el umbral y $p_i(i = 0, ..., n-1)$ es la probabilidad estadística de los píxeles con nivel de gris en toda la imagen, p_B es la probabilidad de los píxeles con nivel de gris por debajo del umbral s

$$p_B = \sum_{i=0}^{s} p(i) \tag{3.8}$$

la entropía de la parte negra es:

$$H_B^{(s)} = -\sum_{i=0}^{s} \frac{p_i}{p_B} log_2(\frac{p_i}{p_B}) \tag{3.9}$$

la entropía de la parte blanca:

$$H_w^{(s)} = -\sum_{i=s+1}^{n-1} \frac{p_i}{1-p_B} log_2(\frac{p_i}{1-p_B}) \tag{3.10}$$

la entropía total

$$H_T^{(s)} = H_B^{(s)} + H_w^{(s)} \tag{3.11}$$

se busca el umbral s que maximiza la entropía total.

Nuestra aproximación utiliza la técnica de Kapur pues como se puede apreciar en el capítulo 6 o en el apéndice A es el que mejores resultados ofrece.

3.2.2. Agrupamiento temporal

En la etapa de agrupamiento temporal, detallada en el capítulo 4, se busca para un bloque B_t^k agrupar aquellos instantes temporales en la secuencia de entrenamiento $I_{1...T}$ donde los valores de los píxeles son similares para formar C candidatos U_c^k donde $c \in [1, ..., C]$ $(C < T)$. Se busca reducir el número de candidatos a fondo de escena $(B_t^k \Rightarrow U_c^k)$ para disminuir la complejidad (y el coste computacional) de la etapa de continuidad espacial. A su vez se pretende no utilizar umbrales por lo que se elije una técnica común de agrupamiento que nos permite agrupar bloques de modo jerárquico [53]. Posteriormente, se evalúan distintas posibilidades de agrupación de los distintos bloques en un sólo candidato y se decide cual es el óptimo número de agrupaciones (es decir, C) a través de unos índices de validación interna [54]. Finalmente, en la etapa de continuidad espacial se elegirá el que mejor represente el fondo de escena en dicha región considerando los U_c obtenidos y el número de B_t^k que han dado lugar a cada uno. En

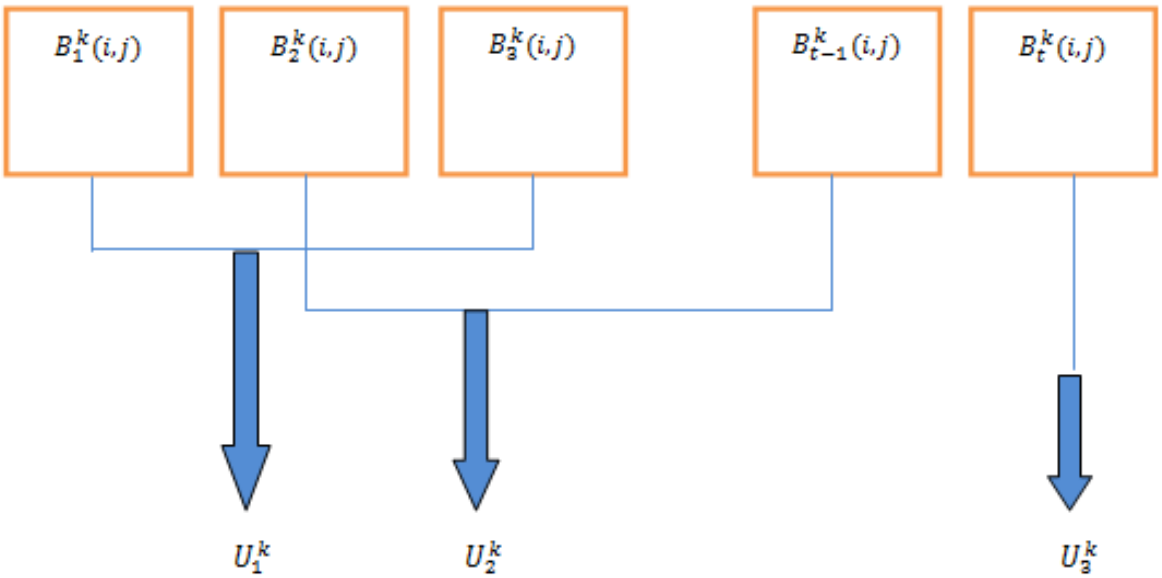


Figure 3.5: Diagrama de bloques del agrupamiento temporal

la figura 3.5 se puede observar como para todos los bloques B_t^k en una posición de la imagen definida por $[x_k^{min}, x_k^{max}, y_k^{min}, y_k^{max}]$ se obtienen los candidatos, los cuales no tienen porqué formarse con bloques sucesivos en el tiempo.

3.2.3. Continuidad espacial

La etapa de continuidad espacial tiene como objetivo calcular una imagen de fondo de escena con los candidatos resultantes de la etapa temporal previa. Es decir, para los C candidatos U_c^k de una posición k se persigue encontrar aquel que mejor represente el fondo de escena (F^k). Para lograrlo el algoritmo se divide en dos sub-etapas:

La primera de ellas consiste única y exclusivamente en aprovechar la información del agrupamiento temporal para obtener una semilla de fondo de escena. Dicha semilla es el candidato U_c^k, en cualquier posición definida por k, que más bloques B_t^k haya utilizado para formarse.

La segunda sub-etapa consiste en extender la semilla inicial mediante la continuidad espacial de dicha semilla con sus candidatos adyacentes (o vecinos) en la imagen para construir el fondo de escena. Este apartado se detallará en el capítulo 5. En resumen, esta fase aplica iterativamente cuatro etapas distintas para medir la continuidad espacial con aquellos bloques que son fondo buscando obtener resultados similares en cada una de ellas:

- La primera es un continuidad de borde donde se calcula la diferencia de los píxeles adyacentes de bloques vecinos de fondo con cada uno de los candidatos a considerar mediante

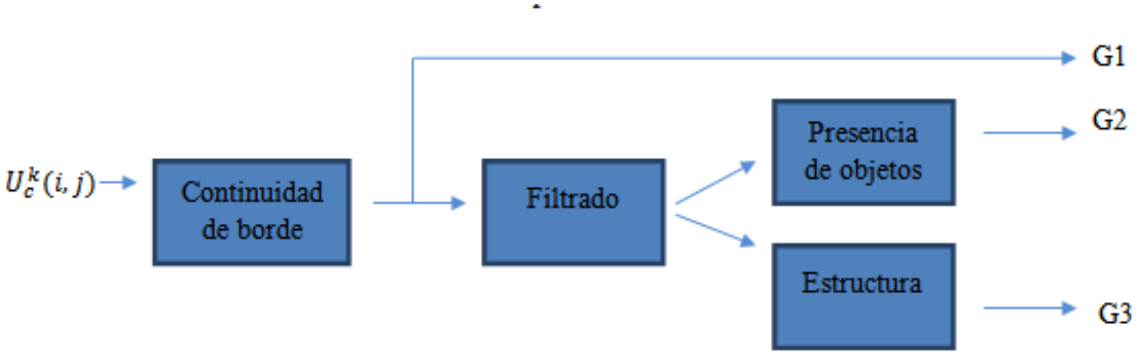


Figure 3.6: Diagrama de bloques de la etapa de continuidad espacial

la técnica [55]. El objetivo es estudiar la continuidad espacial entre sus fronteras. Posteriormente, se ordenan los candidatos de menor a mayor diferencia para indicar su mayor probabilidad a ser fondo. Finalmente, esta lista se filtra eliminando aquellos candidatos con una alta diferencia de fronteras mediante un umbral adaptativo.

- La segunda persigue identificar si hay objetos de primer plano en los candidatos de manera similar a [8]. Partiendo de la lista ordenada (y filtrada), se obtiene la diferencia de color de los candidatos dos-a-dos y se obtiene una máscara binaria con la diferencia. Después, se extraen las componentes conexas de esta máscara y se calcula el contraste de los contornos de los *blobs* [55]. El candidato con menor contraste resulta ganador y se prueba con el siguiente candidato y así sucesivamente

- La tercera persigue comparar las estructuras de los bloques de fondo de escena vecinos (semilla o alguno ya reconstruido) con los candidatos existentes mediante el uso del histograma de gradientes orientados (HOG) [56].

- Si las tres técnicas anteriores no coinciden en la decisión de cuál es el mejor de los candidatos para representar el fondo de escena, aplicamos una condición de suavidad y seleccionamos aquel candidato más simple. Para ello, utilizamos la transformada discreta del coseno (DCT) [12] y analizamos su energía en los componentes de baja frecuencia. Como se detalla en el capítulo 5 se evaluará la suavidad espectral del paso de un bloque candidato a los bloques adyacentes que ya forman parte del fondo de escena.

En la figura 3.6 se muestra el procedimiento a seguir en la etapa espacial. Para cada $B_t^k(i,j)$ de la imagen se tienen unos $U_c^k(i,j)$ procedentes de la etapa temporal, para cada uno de ellos se aplican las características anteriores para encontrar el que mejor representa el fondo de escena en esa posición. Después se repite el proceso para cada k posición de la imagen.

3.3. Ventajas

Las ventajas del algoritmo propuesto son las siguientes:

- Gracias a la primera etapa de estudio del tamaño de la región de análisis no necesitamos ningún valor aleatorio para el tamaño de la misma. Como se ha explicado en la sub-sección 3.2.1, el algoritmo solo usa un tamaño máximo y mínimo que se obtiene de un estudio que se muestra en el capítulo 6. Con este estudio se puede aprovechar las ventajas de tener un tamaño grande de ventana o uno pequeño según las características del vídeo de entrada. Con un tamaño grande mejora el tiempo de proceso del algoritmo en su totalidad, sin embargo se puede llegar a perder en el movimiento de objetos pequeños que entren en la totalidad de un bloque. Con un tamaño pequeño se puede no ser capaz de encontrar movimientos en un bloque si los objetos son grandes.

- Las ventajas de usar la etapa temporal son varias. La primera de ellas es que se reducen los candidatos que pueden llegar a ser fondo de escena. La segunda es que conseguimos hacer que los candidatos temporales estén bien definidos, es decir que cuando haya muchos bloques iguales en distintas imágenes el candidato se forme con la media de todos ellos para dar sólo uno, y a su vez conseguimos que cuando un bloque sólo aparece en dos o tres imágenes también aparezca como tal. Con esta etapa sólo formamos candidatos, no decidimos si hay mejores o peores con lo que se evita objetos de primer plano parados en el fondo de escena y podemos soportar más de un 50% de primer plano. La otra principal ventaja es que para el desarrollo de la misma no se utiliza ningún tipo de umbral ajustado.

- Las ventajas de utilizar esta etapa espacial es la combinación de varias técnicas que sucesivamente filtran candidatos a fondo de escena. Adicionalmente, se mejora la robustez de las técnicas por separado como se demuestra en el capítulo 6. Como veremos la continuidad de borde no es fiable pues no implica que en el resto de la ventana no aparezca primer plano e incluso no es fiable donde coincida un borde justo en la frontera. La identificación de objetos mediante el contraste de los contornos de *blobs* no es fiable cuando coincida que el objeto de primer plano se introduce en un bloque de fondo de escena del mismo color y en al caso de las estructuras no se puede asegurar que haya continuidad entre aquellas que componen el fondo. Sin embargo utilizando todas ellas a la vez minimizamos el número de errores.

En resumen, el algoritmo propuesto basado en etapa temporal más etapa espacial puede soportar gran cantidad de objetos primer plano en la secuencia de inicialización, evitar los objetos de primer plano estáticos en el fondo de escena, el fenómeno del camuflaje y todo ello en cortas secuencias de entrenamiento.

Capítulo 4

Agrupamiento temporal

4.1. Introducción

La etapa de agrupamiento temporal es la encargada de formar *clusters* (agrupamientos) para cada bloque definido en la etapa anterior y formar así uno o más candidatos a ser fondo de escena. De esta manera reducimos el número de candidatos a ser fondo de escena a analizar en la siguiente fase. Es decir, en vez de asumir que cada bloque de cada imagen puede ser el fondo, agrupamos aquellos que son similares como un solo candidato.
Para ello recibe de la etapa anterior los bloques con un tamaño definido y con un máximo y un mínimo de intervalos estables. Esta etapa agrupa bloques en función de esos intervalos estables variando entre el mínimo y el máximo y decide cual de los agrupamientos es óptimo.
Este capítulo empieza con una visión general de las distintas sub-etapas de las que consta el agrupamiento (sección 4.2). Posteriormente se explica cada una de las sub-etapas empezando por el análisis PCA necesario para reducir la dimensionalidad de los datos de entrada al agrupamiento (sección 4.3), siguiendo por el propio agrupamiento temporal el cual se divide en dos etapas: la primera consiste en formar diferentes *clusters* en función de los resultados de la etapa previa (sección 4.4) y la segunda en evaluarlos (sección 4.5).
El objetivo del agrupamiento y posterior validación propuestos es que, a diferencia de otros algoritmos existentes en el estado del arte, no necesita umbrales para decidir que imágenes de la secuencia son estables y por tanto deben formar un único *cluster*. Por otro lado esta etapa no decide que candidato es óptimo para el fondo de escena (solo los agrupa) con lo que se solventa los problemas de primer plano estáticos.

4.2. Esquema global

Esta etapa de agrupamiento pretende reducir el número de bloques candidatos a fondo de escena agrupando los B_t^k (definidos por su posición en la imagen, el tamaño del mismo y tanto el mínimo

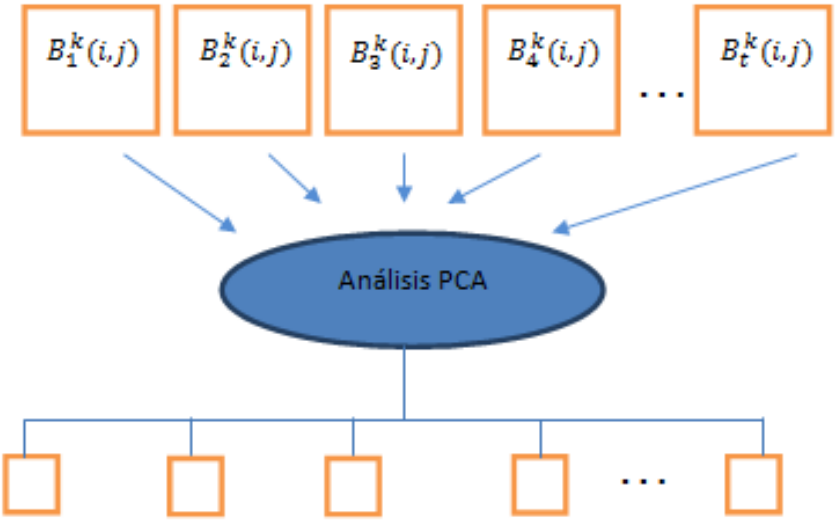


Figure 4.1: Esquema análisis PCA, reducción de datos.

como el máximo número de *clusters*) en candidatos a fondo de escena U_c^k $(c < t, \forall k)$

Para conseguir el objetivo la primera sub-etapa consiste en reducir el tamaño de los datos para conseguir tener una aproximación más eficiente computacionalmente. Se aplicará la conocida técnica de análisis PCA cuyo efecto será el mostrado en la figura 4.1. Se puede observar en ella como los datos de entrada de un bloque B_t^k son reducidos en tamaño.

Como muestra la figura 4.2 posteriormente se aplica un algoritmo de agrupamiento jerárquico. Variando el número de *clusters* entre el mínimo y el máximo calculado en la etapa previa se agrupan los distintos bloques B_t^k, donde $t \in [1, ..., T]$, en función del número de *clusters* evaluado. A la salida se evalúan las diferentes agrupaciones con distintos índices de validación y se selecciona el agrupamiento óptimo que proporcionará los candidatos U_c^k a la siguiente etapa.

Para evaluar el agrupamiento se utilizan los índices de validación: *silhouette y Davies-Bouldin* (sección 4.5). Para cada agrupamiento entre el valor mínimo y el máximo, cada índice devuelve una puntuación (*score*) que se combinan con el objetivo de decidir el agrupamiento óptimo de los bloques B_t^k. Siendo N_c el número óptimo de *cluster* habrá $U_c(i,j)$ candidatos dónde $c \in [1, ..., N_c]$.
Si la entrada es una secuencia de bloques B_t^k, el resultado serán N_c clusters donde cada uno se formará con determinados B_t^k. Por ejemplo, el candidato U_c^k se formarán como la media de sus bloques B_t^k

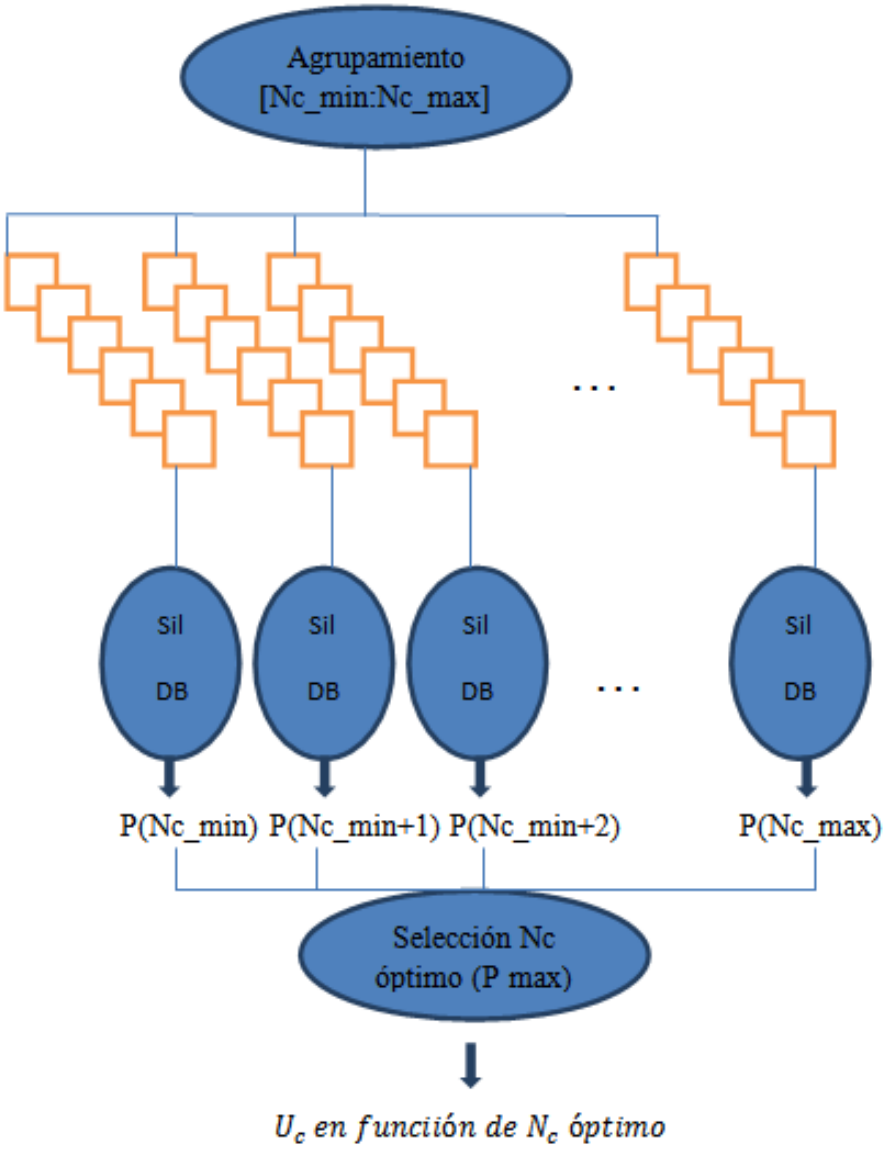


Figure 4.2: Esquema agrupamiento temporal más validación

$$U_c^k = \frac{1}{n} \sum_{t=<c>} B_t^k \tag{4.1}$$

donde n es el número de bloques que dan lugar al candidato y $< c >$ representa los índices (imágenes) de los bloques seleccionados para el *cluster* c. En la figura 4.3 se observa como se forman candidatos promediando las imágenes que tienen el mismo etiquetado en función del agrupamiento óptimo:

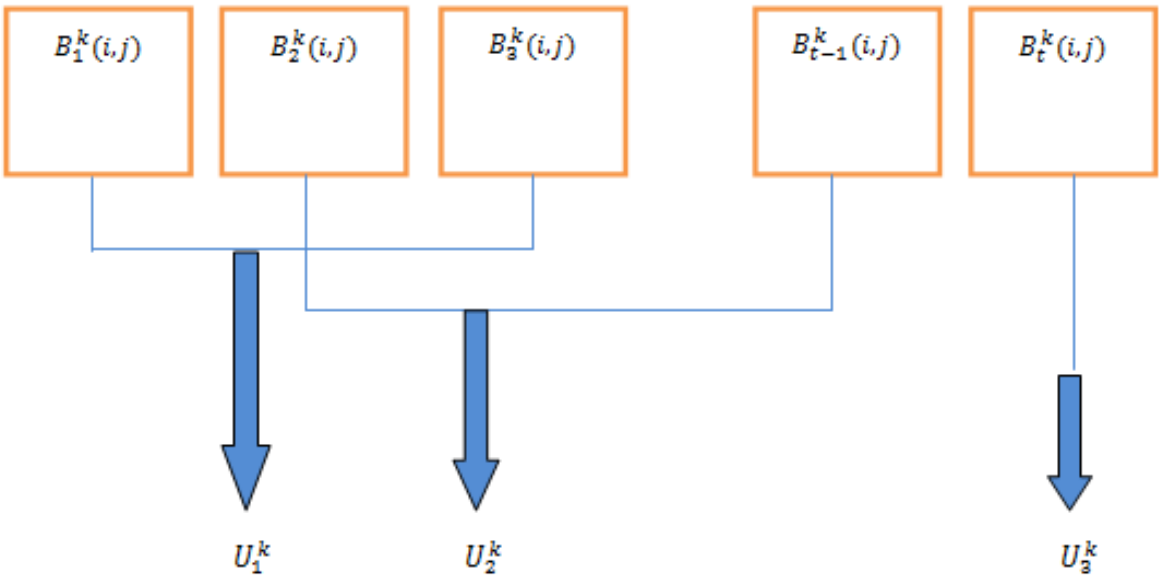


Figure 4.3: Esquema de extracción de candidatos en la etapa de agrupamiento temporal

4.3. Reducción de dimensionalidad

Con el objetivo de reducir la dimensionalidad de los datos de entrada y por tanto reducir el tiempo de ejecución del algoritmo se introdujo una sub-etapa de re-dimensionalización de datos. Para ello se optó por la técnica más común existente en el estado del arte, el análisis de componentes principales PCA.

El análisis de componentes principales, PCA, es una técnica utilizada para reducir la dimensionalidad de un conjunto de datos. La técnica sirve para hallar las causas de la variabilidad de un conjunto de datos, ordenarlas por importancia y proyectar los datos en espacios con una dimensionalidad reducida. Técnicamente, el PCA busca la proyección según la cual los datos queden mejor representados en términos de mínimos cuadrados. [57][58]

Una de las ventajas del PCA para reducir la dimensionalidad de un grupo de datos, es que retiene aquellas características del conjunto de datos que contribuyen más a su varianza, manteniendo un orden de bajo nivel de los componentes principales e ignorando los de alto nivel. El objetivo es que esos componentes de bajo orden a veces contienen el aspecto más importante de esa información. En la inicialización de fondo, los bloques B_t^k suelen presentar movimiento en pequeñas partes de ellos con lo cual, la mayor parte de información (la parte estática) no es relevante para realizar una posible agrupación.

En nuestro algoritmo, utilizamos el método basado en covarianzas, cuyo objetivo es transformar un conjunto de datos o matriz X que representa la secuencia de entrenamiento B_t^k $t \in [1...T]$. Cada fila de X se corresponde con un B_t^k linealizado (es decir, un vector fila de dimensión

Tamaño de bloque	3x3	10x10	20x20	40x40	60x60	80x80	100x100
Dimensión sin pca	27	300	1200	4800	10800	19200	30000
Dimensión con pca	1	1	2	4	6	7	7
Tasa de compresión	27	300	600	1200	1800	2742.85	4285.71

Table 4.1: Tasa de compresión del análisis PCA bloque sin movimiento

	3x3	10x10	20x20	40x40	60x60	80x80	100x100
Dimensión sin pca	27	300	1200	4800	10800	19200	30000
Dimensión con pca	3	2	4	6	8	7	6
Tasa de compresión	9	150	300	800	1350	2742.85	5000

Table 4.2: Tasa de compresión del análisis PCA bloque con movimiento

$WxHx3$). Como resultado, se forma una matriz X con T filas y $WxHx3$ columnas. Se quiere buscar que esas T filas se representen con menos variables (columnas). El objetivo es obtener los datos reducidos $Y = PX$ donde P es la matriz de transformación para proyectar X. Para ello, procedemos de la siguiente manera [59]:

1. Calculamos las desviaciones de la media de X, $Z = X - media(X)$.

2. Calculamos la matriz de covarianzas (o la matriz de dispersión) de Z, S.

3. Diagonalizamos S: $S = VDV^t$, con D matriz diagonal de auto-valores y V matriz ortogonal ($VV^t = I$) formada por los auto-vectores en columnas. Las matrices D y V están ordenadas por orden decreciente de los auto-valores.

4. Seleccionamos los k primeros auto-valores y auto-vectores. Si llamamos V_k a la matriz formada por las k primeras columnas de V, la matriz de transformación será $P = V_k^t$

El análisis PCA permite una reducción de los datos de ejecución tanto en bloques sin movimiento como muestra la tabla 4.1 como en bloques con movimiento 4.2. Como es de esperar la tasa de compresión es mayor en la secuencia sin movimiento pero es igual de efectiva en secuencias con primer plano.

4.4. Agrupamiento jerárquico

El objetivo de esta etapa es agrupar un conjunto de datos, con el objetivo de reducir los candidatos a fondo de escena en la siguiente etapa, de manera que se maximice la similitud dentro de los *clusters* y reduzca al mínimo la similitud entre dos *clusters* diferentes. En esta etapa se

persigue un agrupamiento sin el uso de umbralizaciones. Para cada k, la entrada de la etapa es la matriz Y (B_t^k con dimensionalidad reducida) y la salida son los candidatos U_c^k que pasarán a una etapa posterior en la cual se decidirá que candidato es el óptimo a fondo de escena.

Para llevar a cabo el agrupamiento se ha utilizado el método más popular existente en el estado del arte denominado *agglomerative hierarchical* donde de un conjunto de muestras distintas se agrupan de acuerdo a ciertas métricas (apéndice B). En esta aproximación el algoritmo parte de un conjunto de puntos distintos que son las filas de la matriz Y y realiza el agrupamiento jerárquico obteniendo distintos *clusters* de acuerdo a una métrica hasta llegar al número de cluster que se está evaluando en ese momento. En nuestro algoritmo, el agrupamiento se realiza de la siguiente manera.

Primero se computa la distancia euclídea entre pares de objetos en una matriz de datos Y de $t \times W' \times H'$) donde W' y H' son las dimensiones de la matriz B_t^k reducida. Las filas corresponden a las observaciones (bloques) y las columnas a las variables transformadas (originalmente los píxeles de cada bloque). La distancia entre dos filas de la matriz Y, se define como:

$$d_{ij} = \sqrt{\sum_{r=1}^{M} (y_{ir} - y_{jr})^2}, \tag{4.2}$$

donde y_{ir} corresponde con el elemento r la fila i y M indica el número de elementos que tiene cada fila de Y. Para formar las agrupaciones se usa la métrica *maximun o complete linkage*, la cual calcula la máxima distancia entre los datos (filas) que conforman dos *clusters* U_i^k y U_j^k de la partición de los datos.

$$D(U_i^k, U_j^k) = max(d(b_1, b_2)) : \; b_1 \in U_i^k, b_2 \in U_j^k) \tag{4.3}$$

donde b_1 y b_2 son los datos que forman parte de, respectivamente, los *clusters* U_i^k y U_j^k (es decir, las filasy_{ir}, $r \in [1...M]$ de la matriz Y).

El agrupamiento jerárquico analiza los datos hasta el máximo numero de *clusters* usando la información de la distancia como criterio, es decir calcula N_{max} agrupaciones distintas.

Posteriormente se evaluará cada agrupamiento con índices de validación para para determinar el valor óptimo y formar los candidatos según el mismo.

4.5. Índices validación de agrupamiento

El objetivo de esta sub-etapa es validar los distintos agrupamientos desde el número mínimo de *clusters* hasta el máximo y decidir cual de ellos es el número óptimo. La entrada son las

agrupaciones realizadas en la etapa previa de agrupación y la salida son los candidatos formados con el número óptimo de *clusters* encontrado que formarán los distintos U_c^k.
Con el objetivo de determinar el número óptimo de de *clusters*, se propone utilizar índices de índices de validez [54]. Hay dos tipos de índices de validez: los índices externos e índices internos. Un índice externo es una medida que compara dos agrupamientos de datos donde la primera es a priori conocida (también llamado *ground-truth*), y la segunda son los resultados del algoritmo. Los índices externos más conocidos incluyen *Rand, adjusted Rand, Jaccard, and Fowlkes Mallows* (FM) [60] [61].
Los índices internos se utilizan para medir la bondad de una estructura de la agrupación, sin información externa. Los índices internos evalúan los resultados con las cantidades y características propias del conjunto de datos. El número de *clusters* óptimo de los datos (NC) suele estar determinado mediante el uso de un índice de validez interna (nuestro caso). Los índices internos más populares son: *Silhouette, Davies-Bouldin, Calinski- Harabasz, Dunn, Hubert-Levin (C-index), Krzanowski-Lai and Hartigan* [62] [63] [61] ; *the Root-mean-square standard deviation (RMSSTD), R-squared, Semi-partial R-squared (SPR) and Distance between two clusters* (CD) [60] [64]; *the weighted inter-intra index* [65]; *and the Homogeneity y and Separation* [66].
Al no disponer de un *ground-truth* para evaluar los agrupamientos realizados, se realizó un estudio preliminar de los índices internos para utilizar los de mejor funcionamiento. El resumen de este estudio se puede encontrar en el apéndice C. Para nuestro algoritmo, finalmente utilizamos los índices *Silhouette* y *Davies-Bouldin*.
Silhouette index [62] *:* refleja la solidez y la separación de grupos. Un valor promedio mayor indica una mejor calidad de los resultados de la agrupación, por lo que el óptimo NC es el que da el mayor valor promedio del *Silhouette*.
Para un *cluster* determinado U_c^k $c \in [1, ..., C]$ de un bloque k donde se han agrupado los datos en C clusters, este método asigna a cada muestra (bloque) de U_c^k una medida de calidad $S_i(i = 1, ..., m)$, conocida como *Silhouette width*. El *Silhouette width* es un indicador de confianza del *ith* bloque en el *cluster* U_c que se define como:

$$s(i) = \frac{b(i) - a(i)}{max\left\{a(i), b(i)\right\}'}. \tag{4.4}$$

donde $a(i)$ es la distancia media entre el bloque *ith* y todas los bloques incluidos en U_c^k, y $b(i)$ es la distancia mínima entre la bloque *ith* y todos los bloques agrupados en U_r^k $r \in [1, ..., C]$ $c \neq r$. De esta fórmula se deduce $-1 \leq s(i) \leq 1$.
Para un *cluster* determinado, U_c^k , es posible calcular el índice *Silhouette* mediante el promediado del *Silhouette width* de cada bloque, el cual caracteriza las propiedades de heterogeneidad y aislamiento de este *cluster*:

$$S_c = \frac{1}{m}\sum_{i=1}^{m} s(i) \tag{4.5}$$

donde m es el numero de muestras (bloques) en U_c^k. Finalmente, se obtiene el índice global de *Silhouette* para la agrupación en C clusters como

$$S = \frac{1}{C}\sum_{c=1}^{C} S_c \tag{4.6}$$

donde el óptimo NC es el que da el mayor valor promedio del *Silhouette* en las distintas agrupaciones de C clusters que son validadas (desde N_{min} hasta N_{max}).
Davies-Bouldin index (DB): Una medida de la similitud media entre cada grupo y su más similares, pequeños valores corresponden a grupos que son compactos y tienen centros que están muy lejos unos de otros, por lo tanto, su valor mínimo determina la óptima NC [62]. El índice de *Davies-Bouldin* tiene como objetivo identificar los conjuntos de grupos que son compactos y bien separados. El índice DB se define como:

$$DB = \frac{1}{C}\sum_{i=1}^{C} max_{i\neq j}\left\{\frac{\Delta(U_i)+\Delta(U_j)}{\delta(U_i,U_j)}\right\} \tag{4.7}$$

donde U_i e U_j representan, respectivamente, el *cluster* i y j de una partición data; $\delta(Ui, U_j)$ denota la distancia euclídea entre *clusters* U_i y U_j, $\triangle(U_i)$ representan la distancia *intra-cluster* del *cluster* U_i y C es el número de *clusters* de la partición. Pequeños valores de DB corresponden a grupos que son compactos, y cuyos centros están muy lejos unos de otros. Por lo tanto, la configuración que minimiza DB se toma como el número óptimo de *clusters*.
Después las puntuaciones (*scores*) obtenidas se combinan para decidir el valor óptimo de C. En el caso del *silhouette* el óptimo *score* es el máximo y para el caso del *Davies-Bouldin* el óptimo es el mínimo. Se pondera cada *score* entre el máximo y el mínimo de su valor, es decir en el caso del *silhouette* se asigna una probabilidad 100% al máximo y un 0% al mínimo y para el *Davies-Bouldin* 100% al mínimo y 0% al máximo. Posteriormente se asigna una probabilidad total de forma cada agrupamiento tenga asociada una probabilidad:

$$P_{Sil} = \frac{Sil - min(Sil)}{max(sil) - min(sil)} \tag{4.8}$$

Figure 4.4: Secuencia de 20 imágenes para un bloque dado, B_t^k

$$P_{DB} = \frac{DB - max(DB)}{min(DB) - max(DB)} \tag{4.9}$$

$$P_t = P_{Sil} * w + (1 - w) * P_{DB}. \tag{4.10}$$

donde w controla la contribución de cada índice. Inicialmente $w = 0.5$.
El número de *clusters* óptimo será aquel que tenga mayor probabilidad. Cuando se obtiene el número de *cluster* óptimo se forman los candidatos en función del etiquetado de imágenes para ese número de *clusters*. Las etiquetas corresponden a los diferentes intervalos estables encontrados según el agrupamiento realizado para el número óptimo de clusters. A cada imagen de la secuencia se le asignará una etiqueta.

En la figura 4.4 se observan los distintos bloques que existen para una posición de la imagen, en este caso en una secuencia de 20 imágenes. En la figura 4.5 se puede observar el funcionamiento de los índices. Después de la etapa de estudio del tamaño de la región obtenemos un número mínimo de 2 clusters y un número máximo de 10. Se hace el agrupamiento jerárquico variando entre el mínimo y máximo de uno en uno y se evalúan los resultados. Observamos como en este caso tanto el índice *Silhouette* y *Davies-Bouldin* coinciden por lo que el número óptimo de *clusters* obtenido ponderando ambos índices (marca roja) es, en este caso, también el mismo 8. Observando la secuencia se puede ver que coincide con lo esperado.

La figura 4.6 presenta un ejemplo de los resultados finales de esta etapa de continuidad temporal en la secuencia vídeo 11 detallada en el capítulo 6

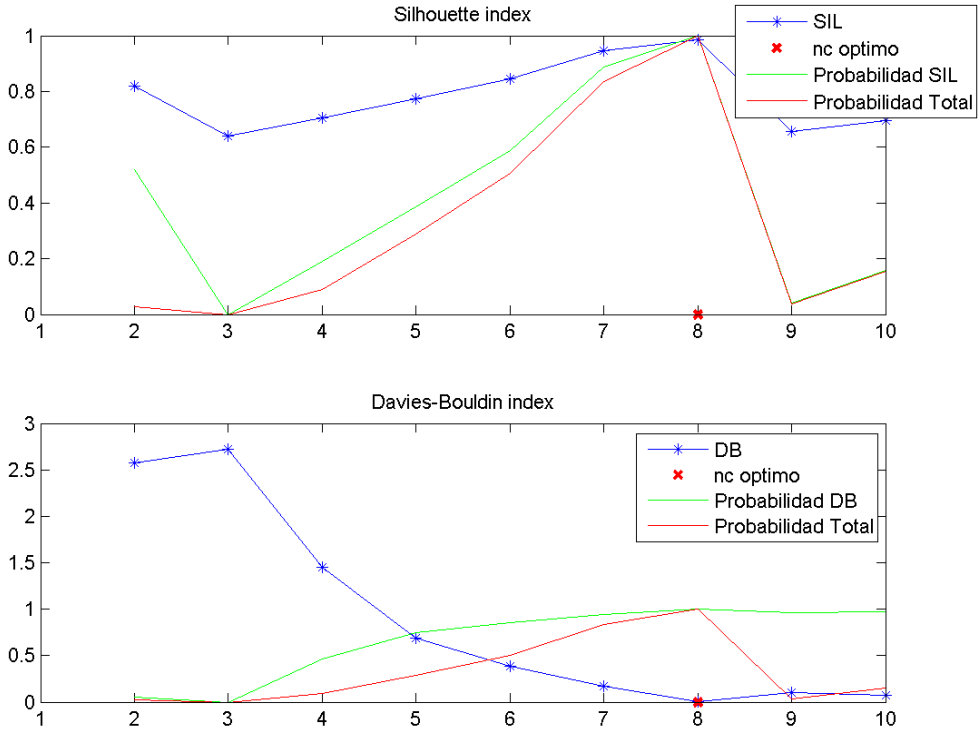


Figure 4.5: Funcionamiento de los índices *Silhouette* y *Dabies Bouldin* para los datos de ejemplo

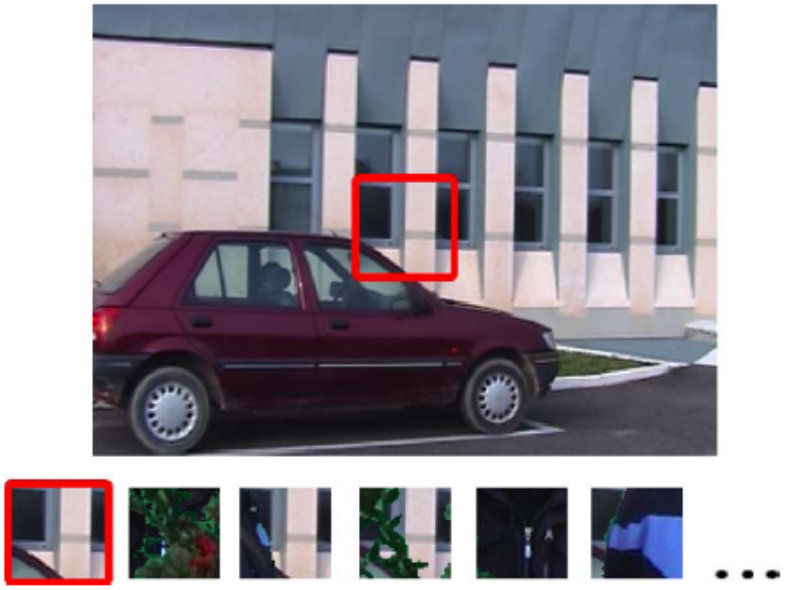


Figure 4.6: Ejemplo de clusters obtenidos en secuencia real [8]

Capítulo 5

Continuidad espacial

5.1. Introducción

Esta parte del algoritmo es la encargada de escoger el mejor candidato U_c^k para cada uno de los bloques B_t^k de la secuencia de entrenamiento. En este capítulo se describen tanto el procedimiento como las técnicas usadas para encontrar una continuidad espacial con el fondo de escena existente y así discernir cual de los candidatos es el óptimo según que bloque.

Partiendo del candidato inicial (semilla), resultante del agrupamiento temporal [8] [12], se intenta aprovechar la continuidad espacial de la imagen de fondo de escena para ir extendiendo dicho candidato con los datos de bloques adyacentes. Para ello siempre se busca un bloque (posición k) que no haya sido procesado y que tenga al menos uno de sus bloques adyacentes con fondo de escena ya determinado. En esos bloques, para todos los candidatos existentes, se aplican varias técnicas que estudian la continuidad espacial de diferentes formas y se decide cuál de todos es el más apropiado: primeramente se analiza la continuidad de borde entre el bloque que se está procesando y los adyacentes donde haya sido fijado el fondo de escena (sección 5.3), la segunda es distinguir entre candidatos observando si hay objetos de primer plano dentro del bloque (sección 5.3.2) [8], la tercera consiste en comparar la estructura de los bloques adyacentes que son fondo de escena con los candidatos que se quiere evaluar (sección 5.3.3) y por último usamos la Transformada discreta del coseno (DCT) para observar suavidad espectral entre bloques cuando las técnicas anteriores no permitan decidir el candidato óptimo (sección 5.3.4) [12].

5.2. Esquema global

De los candidatos procedentes de la etapa temporal se busca obtener el más adecuado, U_c^k óptimo, para representar el fondo de escena en la posición k definida por las coordenadas $[x_k^{min}, x_k^{max}, y_k^{min}, y_k^{max}]$. Se empieza con una suposición de fondo de escena conocido como

Figure 5.1: Ejemplo de una semilla de fondo a partir de la cual se reconstruye un fondo de escena.

bloque semilla, el bloque más estable de la etapa de agrupamiento temporal, el candidato, U_c^k, que más bloques (imágenes), B_t^k, ha utilizado o promediado para formarse. En la figura 5.1 observamos un fondo de escena que no se ha empezado a reconstruir del cual sólo tenemos una semilla, el bloque más estable de la secuencia.

Posteriormente se evalúa la continuidad espacial de cada candidato con el fondo de escena que le rodea mediante las siguientes cuatro características:

- Continuidad de borde (color) entre el candidato y los bloques adyacentes de fondo de escena.
- Estudio de objetos en el interior de los candidatos.
- Similitud estructural del candidato con los bloques adyacentes que son fondo de escena.
- Suavidad espectral de la DCT al pasar de un candidato a un fondo de escena adyacente.

El objetivo es encontrar un candidato óptimo a ser fondo de escena que no dependa en exclusiva de ninguna de las características anteriores pues, aunque todas ellas funcionan correctamente, su efectividad depende de las circunstancias.

En la figura 5.2 se muestra el procedimiento a seguir en la etapa espacial. Para cada B_t^k de la imagen se tienen unos U_c^k procedentes de la etapa temporal. Para cada uno de ellos se aplica la primera característica mencionada anteriormente la cual ofrece un candidato ganador G_1 y un posterior filtrado de candidatos. Después a los candidatos que pasan el filtro, se les aplica la segunda y tercera características las cuales devuelven dos ganadores G_2 y G_3. Para ello, procedemos de la siguiente manera:

1. Se empiezan buscando las posiciones k donde existen bloques de fondo de escena confirmados en alguno de sus vecinos. En ellos se aplican las tres primeras características citadas

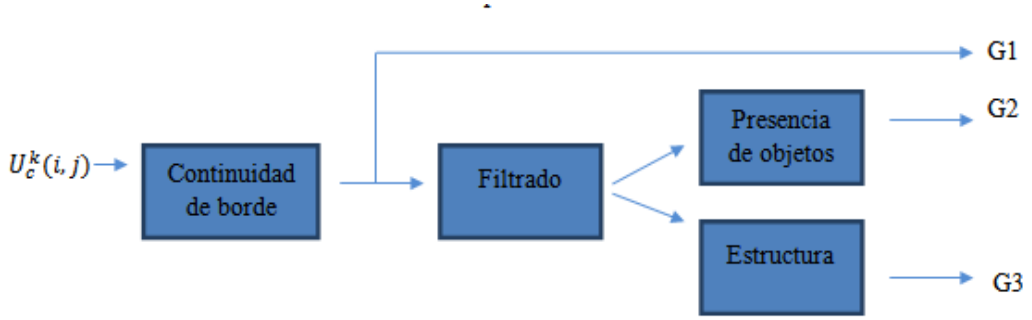


Figure 5.2: Diagrama de bloques de la etapa de continuidad espacial

anteriormente y se obtienen tres candidatos óptimos (G_1, G_2 y G_3). Si todos los candidatos óptimos coinciden, se marca dicho candidato como fondo de escena confirmado/seguro y pasa a formar parte de la imagen de fondo de escena. Se repite el proceso actualizando el fondo de escena con los bloques de fondo de escena seguros hasta que el algoritmo no sea capaz de encontrar ningún otro bloque donde coincidan las tres características o hasta que haya reconstruido el fondo de escena por completo.

2. Cuando no encuentra más coincidencias de los tres candidatos óptimos, se buscan posiciones k donde dos de las tres características coincidan. La búsqueda se comienza por los bloques que tengan más fondo de escena a su alrededor (asegurando así mayor continuidad). Si se encuentran bloques donde uno de los candidatos cumpla lo anterior se introduce en el fondo de escena y se repite el proceso anterior buscando de nuevo coincidencia de las tres características.

3. Si en algún momento no se consigue ninguna coincidencia de las tres o dos de las características se buscan los bloques que tienen más fondo de escena seguro alrededor y se realiza el análisis DCT únicamente de los candidatos óptimos de cada una de las otras técnicas. En este caso el ganador se introduce en la imagen de fondo de escena pero se marca como no seguro, la diferencia es que estos bloques no se tendrán en cuenta en la búsqueda de mayor fondo seguro alrededor de un determinado bloque.

En la figura 5.3 se puede observar la manera de proceder para una bloque determinado. El bloque de Análisis Espacial corresponde con la figura 5.2 mientras que a la salida del mismo puede ocurrir que tres G_s o dos G_s coincidan, donde $s \in [1, 2, 3]$, para una posición k en cuyo caso el U_c^k pasa a ser fondo de escena o por el contrario $G_1 \neq G_2 \neq G_3$ donde se aplica un análisis de la DCT para decidir cual de ellos es el fondo de escena. A continuación se describen cada una de las características empleadas para medir la continuidad.

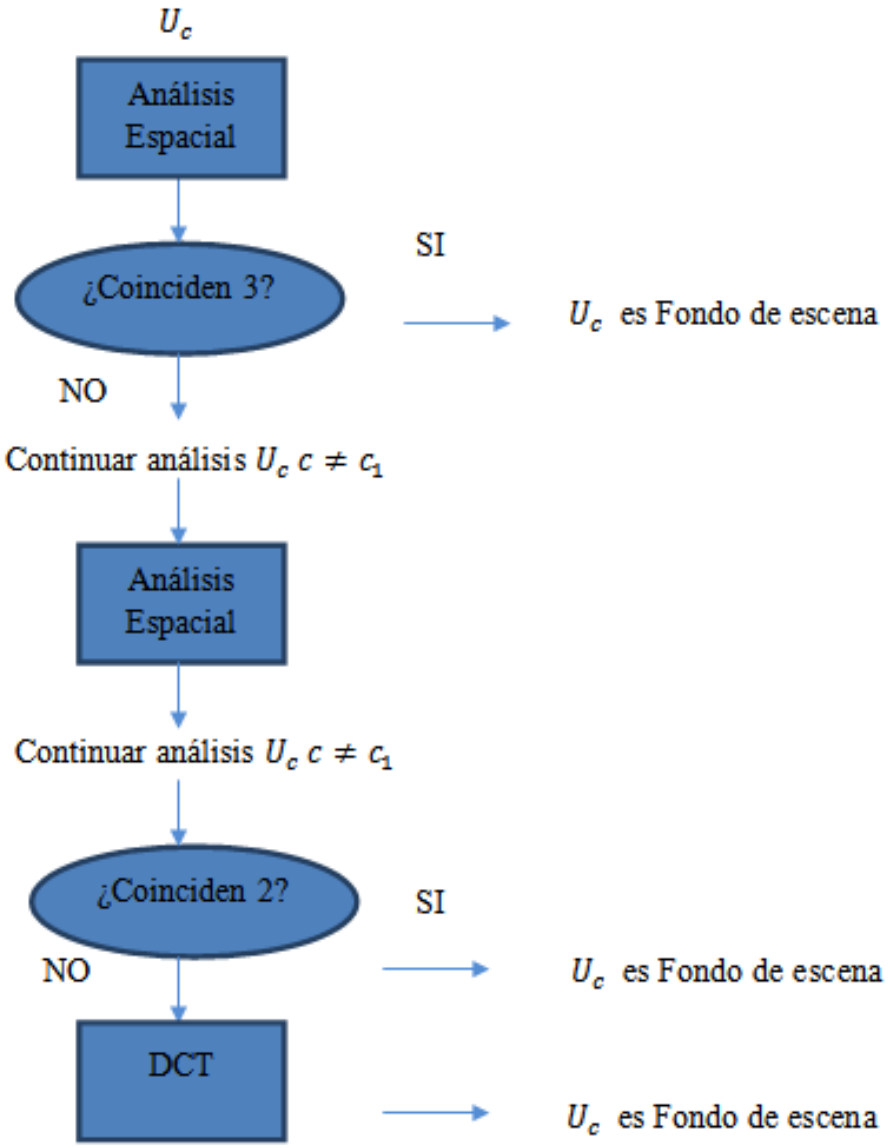


Figure 5.3: Procedimiento por el cual se decide el candidato óptimo en la etapa de continuidad espacial

5.3. Selección de candidatos

5.3.1. Continuidad de borde más filtrado

Para una posición k, el objetivo de esta etapa es evaluar los diferentes U_c^k y organizarlos en función de la continuidad de color con el fondo de escena de alrededor que es conocido. La salida de esta etapa es una lista con los distintos U_c^k ordenados en función de la mínima diferencia de color y un U_c^k óptimo (el de menor diferencia), el primero de la lista, que llamaremos G_1.

Primeramente se buscan bloques que tengan al menos un bloque vecino con fondo confirmado, entendiendo por vecino el bloque de arriba, derecha, izquierda o abajo al que está siendo analizado. Cuando se encuentra se guarda en qué posición o posiciones alrededor del mismo se ha encontrado fondo. Es decir, se busca una posición k para comenzar el análisis.

Una vez que se tiene identificados los vecinos, se trata a cada uno de los posibles candidatos resultantes de la etapa temporal U_c^k, $c \in [1, .., c]$ siendo c el número total de candidatos obtenidos del agrupamiento temporal. Se calcula una diferencia de borde con cada bloque vecino existente, es decir en el mejor caso habrá cuatro diferencias: CB_I (con el bloque de la izquierda), CB_D(derecha), CB_{AR} (arriba), CB_{AB} (abajo). Si no existe fondo de escena en alguna de las posiciones esa diferencia es 0.

$$CB_I = \frac{1}{H} \sum_{y=1}^{H} |I(W, y) - U(1, y)| \tag{5.1}$$

$$CB_D = \frac{1}{H} \sum_{y=1}^{H} |D(1, y) - U(W, y)| \tag{5.2}$$

$$CB_{AR} = \frac{1}{W} \sum_{x=1}^{W} |AR(x, H) - U(x, 1)| \tag{5.3}$$

$$CB_{AB} = \frac{1}{W} \sum_{x=1}^{W} |AB(x, 1) - U(x, H)| \tag{5.4}$$

donde $W = \left|x_k^{max} - x_k^{min}\right|$ es el ancho del bloque y $H = \left|y_k^{max} - y_k^{min}\right|$ es el alto. Por último se calcularía la media dividendo por el número de bloques vecinos que tienen fondo de escena:

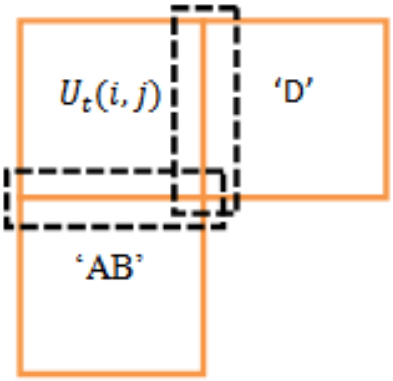


Figure 5.4: Ejemplo extracción de continuidad borde con dos bloques vecinos con fondo de escena

$$CB_T = \frac{(CB_I + CB_D + CB_{AR} + CB_{AB})}{\#vecinos} \tag{5.5}$$

En la 5.4 se muestra un esquema donde se quiere procesar un candidato que tiene fondo de escena en el bloque de abajo y en el de la derecha. Se calcularía la diferencia del bloque de análisis con la frontera derecha, y por otro lado la diferencia con frontera de abajo. Posteriormente se haría la media de ambas y se guardaría el valor.

El resultado será una diferencia de color para cada uno de los candidatos y podremos tener una lista ordenada de los U_c^k acorde con esta característica. El primero de dicha lista será el de menor diferencia de color y así sucesivamente. Los candidatos cuya diferencia media sea superior al umbral obtenido por Kapur en la etapa de estudio de región (*frame difference* adaptativo) son descartados para conformar la lista filtrada que es analizada con las otras técnicas.

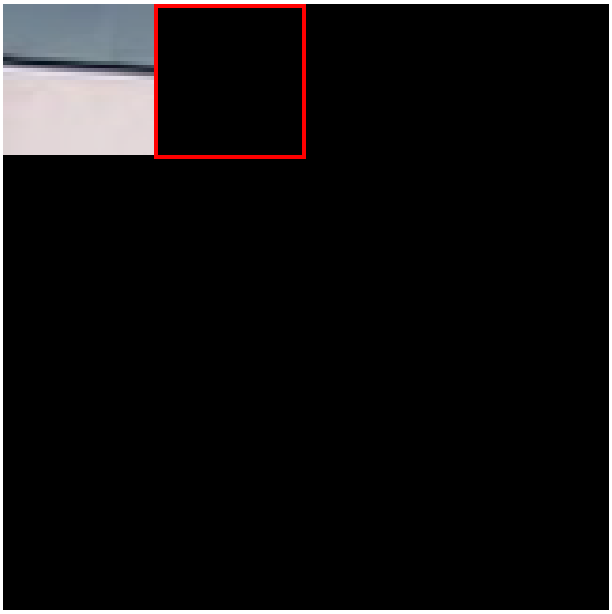
En la figura 5.5 se muestra un ejemplo de la continuidad de borde, en la parte superior se muestra la imagen (parcial) de fondo de escena con un bloque ya confirmado y donde se quieren comparar los distintos (U_c^k) de la segunda fila para la posición marcada en rojo de la derecha del bloque de fondo. En estas condiciones los bloques candidatos se organizarían de menor a mayor diferencia de color siguiendo los resultados presentados en la tabla mostrada en la figura.

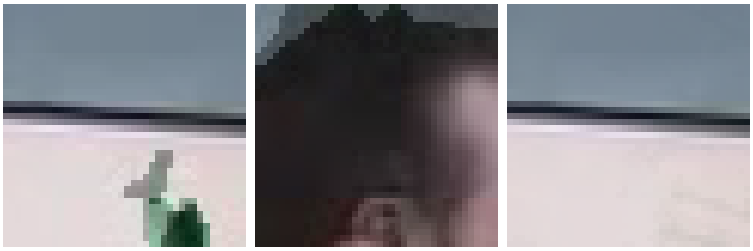
Esta característica ofrece dos problemas:

1. Cuando el borde del bloque coincide con un borde del fondo de escena. El color no será el mismo a uno y otro lado del borde.

2. Cuando el objeto de primer plano no aparece en el borde (el caso del primer candidato).

Para intentar corregir esas deficiencias se utilizaron otras etapas de análisis.





	U_1^k	U_2^k	U_3^k
CB_T	0.41	118.80	0.25

Figure 5.5: Ejemplo real continuidad borde con tres candidatos para bloque recuadrado en rojo

5.3.2. Presencia de objetos

El objetivo de esta etapa es evaluar los diferentes U_c^k procedentes del filtrado de la etapa anterior y encontrar objetos de primer plano dentro de los bloques asumiendo para ello que uno de los U_c^k es el fondo de escena. La entrada son los U_c^k y la salida es un U_c^k óptimo según está característica que llamaremos G_2.

Para los U_c^k $c \in [1, ..., C]$ organizados según la diferencia de color del borde obtenida anteriormente se buscan objetos de frente utilizando para ello los dos primeros candidatos U_1^k y U_2^k. Para ello se utiliza una máscara binaria utilizando los umbrales que devuelve el *frame difference* adaptativo (kapur):

$$f_d = |U_1^k - U_2^k| > T_{kapur} \tag{5.6}$$

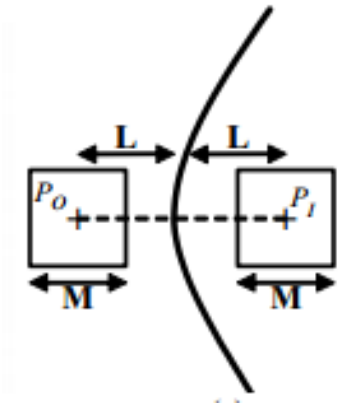


Figure 5.6: Diagrama cálculo diferencias de color alrededor de borde [55]

La máscara estará formada por *blobs* (extraídos mediante un análisis de componentes conexas) que serán objetos de primer plano. Se utiliza dicha máscara para obtener los bordes de los *blobs* que aparecen en ella con el método de *'canny'*. En este caso definimos *blob* como una región conectada de píxeles con tamaño mayor que ξ, donde $\xi = 2$. Posteriormente se evalúa la diferencia de color media a lo largo de cada contorno para todos los *blobs* en U_1^k y U_2^k. De este se obtiene una diferencia de color para cada candidato de la comparación. Para obtener la diferencia, utilizamos la técnica [55] que recorre el contorno y para cada píxel define dos regiones (P_0 y P_1) de tamaño $M \times M$, $M = 3$, en la perpendicular al píxel del contorno (ver figura 5.6):

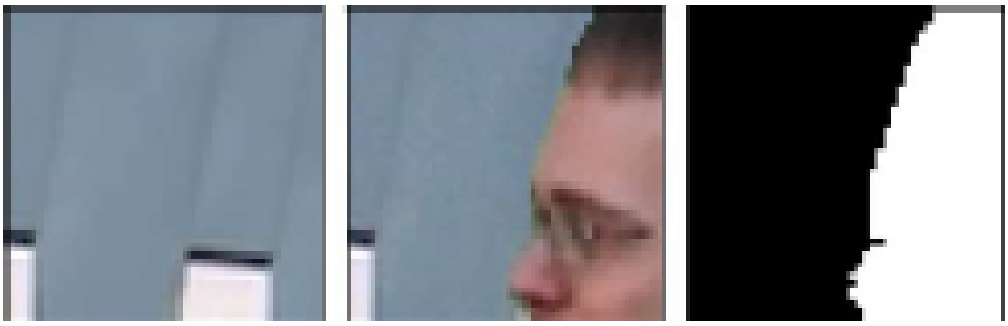
$$ColorP_0 = \frac{1}{M^2} \sum_{x=1}^{M} \sum_{y=1}^{M} I(x,y) \in P_0 \tag{5.7}$$

$$ColorP_1 = \frac{1}{M^2} \sum_{x=1}^{M} \sum_{y=1}^{M} I(x,y) \in P_1 \tag{5.8}$$

$$DIF = \frac{1}{\sqrt{3 * 255 * 255}} \sqrt{(ColorP_0 - ColorP_1)^2} \tag{5.9}$$

donde $I(x,y)$ es el píxel de la imagen a considerar (U_1^k y U_2^k).

El mejor candidato será el que muestre una menor diferencia de color pues los contornos corresponden a objetos de primer plano que han entrado en alguno de los candidatos. En el



	U_1^k	U_2^k
DIF	0.01	0.13

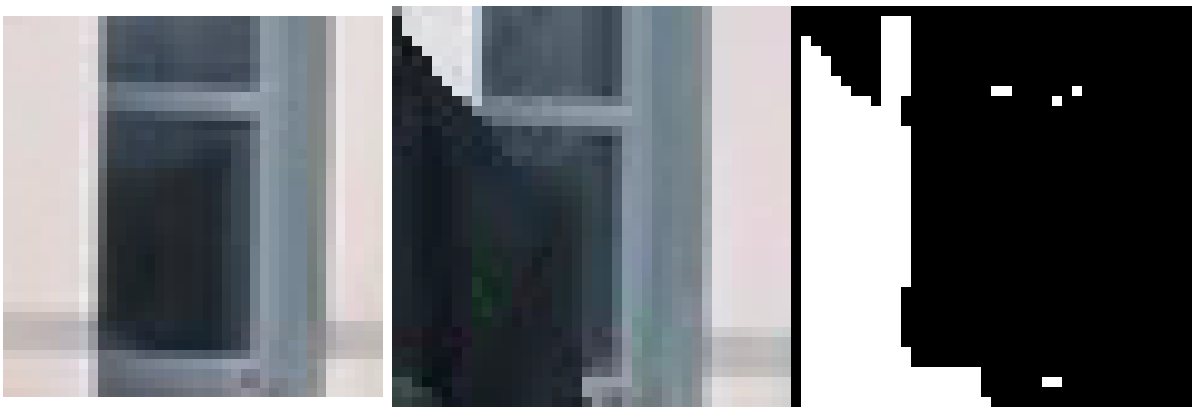
Figure 5.7: Ejemplo evaluación objetos de primer plano en un bloque mediante diferencias de color

candidato que no haya primer plano la diferencia será cero o prácticamente cero cuando no coincidan los bordes del contorno con objetos que si sean parte del fondo de escena.

En la figura 5.7 se observan dos candidatos para un bloque dado y su máscara resultante (primera fila). En la tabla se muestran los resultados de evaluar la diferencia de color en el borde del *blob* en cada candidato.

El problema de utilizar está técnica como única, semejante a [8], es que cuando en el fondo de escena se introduce un objeto de primer plano que se camufla en él y entonces, la máscara resultante que utilizamos para medir el contorno y la diferencia no es la adecuada. Dicho fenómeno se puede observar en la figura 5.8. En este caso habría más de 1 *blobs*. Si se observa el *blob* más grande se puede apreciar como al buscar la diferencia de color en los dos candidatos va a ser menor en el candidato de la segunda columna (a lo largo del contorno extraído de la máscara). Esto es porque justo el contorno extraído del *blob* coincide en el candidato de la primera columna con un cambio de blanco a negro y en el de la segunda el objeto justo es negro con lo que la diferencia siempre será menor. Por ello si no existiesen otras técnicas en paralelo no podría reconstruir el fondo a partir de este punto. Como muestra la tabla la diferencia de color es menor en el candidato erróneo.

Cuando se obtiene el mejor candidato de los dos, se compara de la misma manera con el tercero de la lista obtenida previamente y así hasta el final de la misma por lo que al final se obtiene un candidato óptimo G_2 de esta sub-etapa como muestra la figura 5.9



	U_1^k	U_2^k
DIF	0.24	0.15

Figure 5.8: Ejemplo candidatos con camuflaje, mal funcionamiento de la técnica presencia de objetos

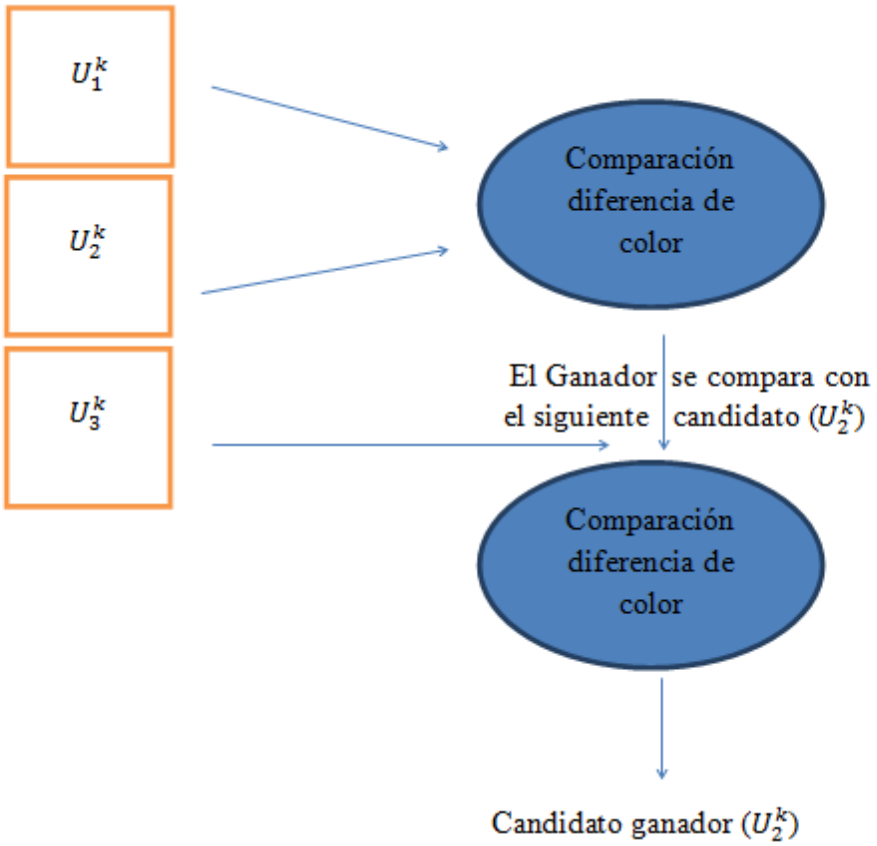


Figure 5.9: Esquema del funcionamiento de la característica: presencia de objetos.

5.3.3. Estructura de bloque

El objetivo es estudiar la diferencia entre estructuras de bloques adyacentes a través del Histograma de Gradientes Orientados (HOG) [56] y corregir los defectos de buscar el candidato óptimo con las técnicas anteriores. La entrada son los U_c^k obtenidos en el filtrado de la etapa de continuidad de borde y la salida es un U_c^k óptimo según está característica que llamaremos G_3.

En el algoritmo partimos de la base de que en teoría bloques próximos tienen que tener un tipo de estructura similar, aunque es cierto que dicha estructura se rompe en ocasiones con objetos que forman parte de un fondo. Con HOG, la apariencia y forma en una imagen puede ser descrita por una distribución de intensidad de gradientes.

La implementación de estos descriptores puede ser alcanzada dividiendo la imagen en pequeñas regiones conectadas, llamados *cells*, y para cada uno de ellos calcular el descriptor HOG. Para mejorar los histogramas locales pueden ser contrastados-normalizados calculando una medida de intensidad alrededor de una región más grande de la imagen llamada macro-bloque, y usar este valor para normalizar todos los *cells* que contiene el macro-bloque. Esta normalización resulta mejor e invariante ante cambios de iluminación y sombras [56].

Como muestra la figura 5.10 el procedimiento consiste en calcular el descriptor (histograma) HOG sobre el bloque de análisis para cada candidato U_c^k y compararlo con el histograma de los bloques vecinos que son fondo de escena. Para comparar ambos histogramas se utiliza la distancia *battacharyya* que proporciona una similitud entre histogramas y consecuentemente, entre la estructura del bloque analizado y el bloque vecino. Se compara cada U_c^k con cada vecino y posteriormente se promedia el porcentaje en función de los vecinos con fondo de escena. Para un bloque adyacente:

$$coef_{bat} = \sum_{i=1}^{nbins} \sqrt{\left(\sum h_U \sum h_{BG}\right)} \tag{5.10}$$

$$dist_{bat} = \sqrt{(1 - coef_{bat})} \tag{5.11}$$

donde h_U y h_{BG} son el histograma HOG de los bloques candidato y fondo confirmado respectivamente. Acorde a esta distancia, el mejor candidato será aquel que ofrezca menor distancia. Por tanto obtendremos un candidato óptimo definido por esta sub-etapa.

La distancia en función del bloque adyacente será $dist_I$ (con el bloque de la izquierda), $dist_D$(derecha), $dist_{AR}$ (arriba), $dist_{AB}$ (abajo).

$$dist_{total=} \frac{dist_I + dist_D + dist_{AR} + dist_{AB}}{\#vecinos} \tag{5.12}$$

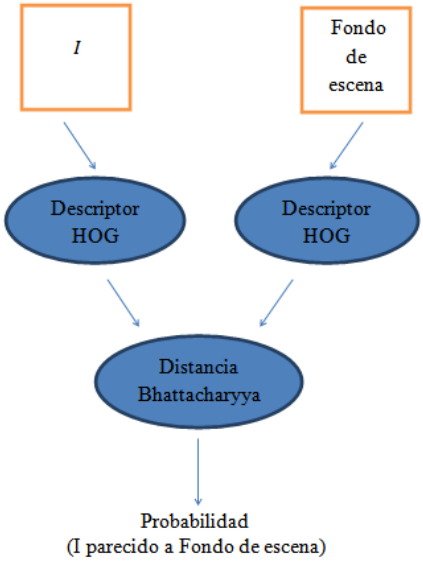


Figure 5.10: Esquema funcionamiento de la característica: semejanza estructural

En la figura 5.11 observamos un ejemplo real de utilización de HOG. En la primera fila hay una imagen (parcial) de fondo de escena con un bloque ya analizado. En la segunda fila se pueden ver los distintos candidatos a fondo de escena en el bloque marcado en rojo mientras que en la tabla se aprecia como el candidato dos es el mejor ya que tiene menor distancia.

El problema vuelve a ser que la estructura de un fondo de escena no tiene porqué ser uniforme y dos bloques distintos pueden tener estructuras diferentes por lo que se decidió utilizar un criterio de desempate como medida de evaluación de las tres técnicas anteriores en el caso en que no coincidiesen.

5.3.4. Criterio de desempate

El objetivo de esta etapa es decidir cual de los candidatos óptimos locales, G_1, G_2, G_3 es el candidato óptimo global (en caso de que no hayan coincidido previamente los tres o posteriormente, solamente dos). Se decidió usar la DCT porque llegados a esta etapa el fondo de escena está casi completado y en el estado del arte, es la técnica que mejor podía aprovechar la continuidad con el objetivo de estudiar si es semejante al fondo de escena que hay a su alrededor. En la figura 5.12 se observa el procedimiento por el cual se evalúan los tres candidatos ganadores de cada una de las etapas anteriores mediante una DCT y se elige un candidato óptimo.

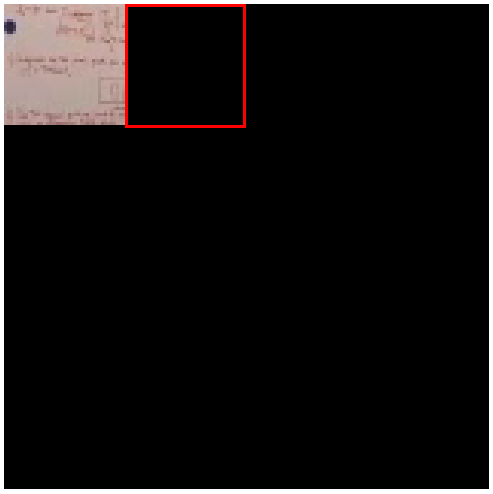
Para realizar esta decisión, se forman bloques conjuntos con el fondo de escena de alrededor. Primeramente, se compone una imagen macro-bloque (con el bloque candidato E y los adyacentes confirmados) (ver figura 5.13). Se calcula la DCT de cada una de los macro-bloques, se elimina la componente continua y se suman en valor absoluto todos los coeficientes de la DCT. Finalmente,

se acumula el resultado de todos los macro-bloques. De esta manera, si E se parece a su alrededor no tiene que haber cambios en la frecuencia por lo que se busca un mínimo.

Al igual que en etapas anteriores se mide la DCT con cada bloque adyacente, dct_I, dct_D, dct_{AR}, dct_{AB}, y por tanto:

$$dct_{total} = \frac{dct_I + dct_D + dct_{AR} + dct_{AB}}{\#vecinos} \tag{5.13}$$

En la figura 5.14 se muestra un ejemplo donde se aprecia el resultado de evaluar dos candidatos a fondo de escena mediante la DCT. La primera fila muestra el fondo de escena inicial (parcial). La segunda fila los dos candidatos a fondo de escena en el bloque marcado, la tercera fila el resultado de formar los macro-bloques, la mitad izquierda de la imagen corresponde al fondo de escena mientras que la mitad derecha corresponde a los candidatos, por último la cuarta fila muestra la tabla con los resultados de aplicar la DCT a dichos bloques. Se observa como el segundo es el óptimo.



	U_1^k	U_2^k
$dist_{bat}$	0.20	0.06

Figure 5.11: Resultado real de comparar dos bloques mediante estructuras.

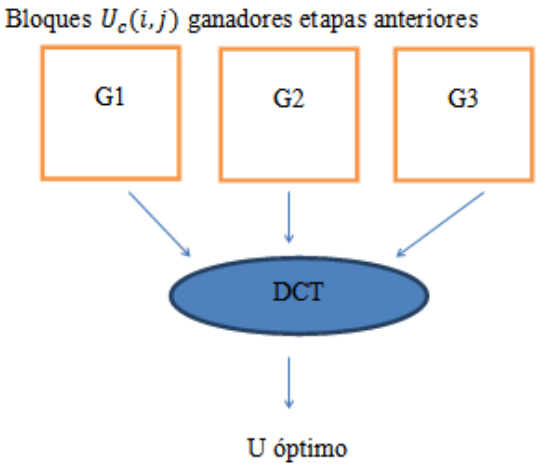


Figure 5.12: Esquema del funcionamiento de la DCT como criterio de desempate

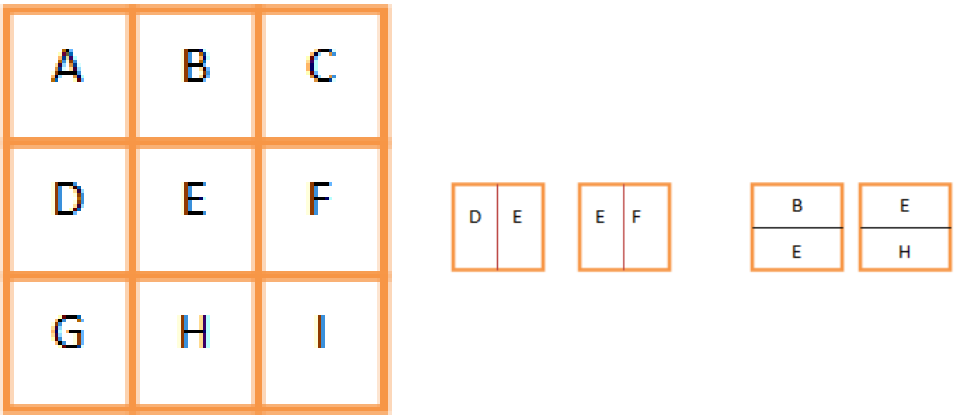
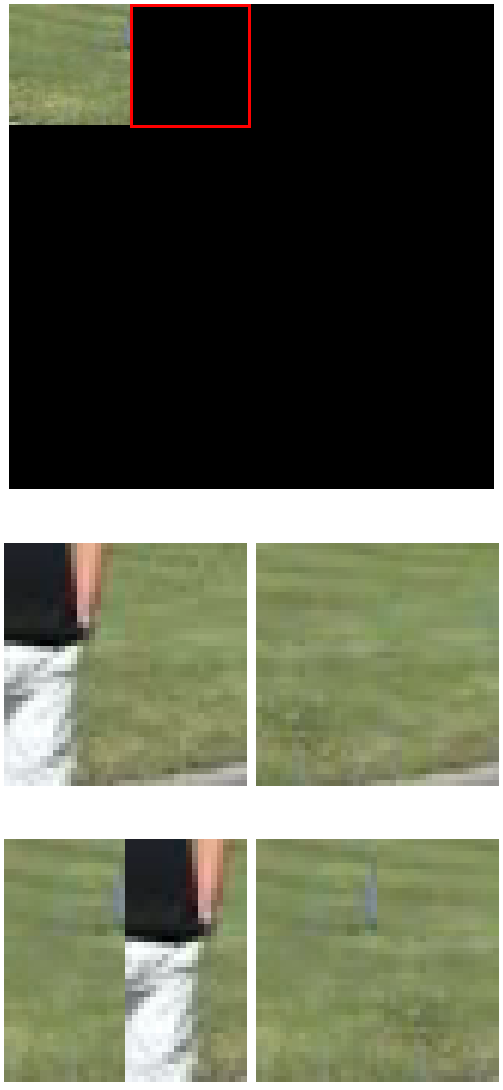


Figure 5.13: Ejemplo disposición de bloques (izquierda) antes de formar los macro-bloques (derecha) para utilizar la DCT como característica espacial.



	U_1^k	U_2^k
$SUM(DCT)$	6.06	0

Figure 5.14: Resultado real de comparar bloques candidatos mediante la DCT

Capítulo 6

Trabajo experimental

6.1. Introducción

En este capítulo se listan los experimentos realizados para analizar el algoritmo propuesto. Primeramente se presentan los resultados de las técnicas comparadas para la evaluación del tamaño adecuado de bloque (Kapur, Otsu y Rosin) (sección 6.2). Posteriormente, la sección 6.3 describe los resultados del estudio del tamaño de bloque y experimentos necesarios para validar el tamaño en función del agrupamiento temporal. Después se estudian tanto los resultados del agrupamiento como el análisis de los distintos índices existentes para la evaluación de dicho agrupamiento. Por último, en la sección 6.4 se evalúan las distintas técnicas de continuidad espacial de manera aislada y combinadas como se indica en el algoritmo propuesto. Adicionalmente se proporciona una comparativa con algoritmos existentes.

El trabajo se ha implementado en matlab 2010. Para la realización del algoritmo se ha utilizado la función que permite obtener el umbral de kapur para el *frame difference* adaptativo cuyo código matlab está disponible en: http://clickdamage.com/sourcecode/index.php. En la segunda etapa, temporal, se hace uso de la función de matlab *pca* para la reducción de datos, *cluster* para el agrupamiento, así como de la función *silhoutte* para la evaluación. En la tercera etapa se ha utilizado código dispuesto por el VPU-Lab donde se estudian diferencias de color a lo largo de contornos [55] así como la función que permite obtener un descriptor HOG para una imagen cuyo código está disponible en: http://www.mathworks.com/matlabcentral/fileexchange/28689-hog-descriptor-for-matlab.

6.2. Análisis de frame difference

Para el análisis del tamaño óptimo de bloque y por consiguiente para el agrupamiento temporal es fundamental el uso de una técnica eficiente para calcular el umbral adaptativo en la etapa de *frame difference*. Su resultado depende del umbral utilizado para calcular la diferencia

Figure 6.1: Ejemplos de imágenes de las secuencias usadas para realizar las pruebas visuales del *frame difference* adaptativo.

entre imágenes. Se han estudiado tres algoritmos de cálculo de umbral adaptativo: Rosin, Otsu y Kapur. A continuación se muestran pruebas visuales con algunas de las secuencias disponibles en la realización de este trabajo.

6.2.1. Conjunto de datos (*dataset*)

Para realizar la comparación de los distintos algoritmos se han realizado pruebas visuales con imágenes seleccionadas de los vídeos disponibles. En concreto, se han seleccionado las siguientes:

- Seq_1 (de la secuencia *ca_vignal* en [8]): 5 imágenes con fondo estático cuyo *frame difference* debe ser nulo.
- Seq_2 (de la secuencia *ca_vignal* en [8]): 5 imágenes del mismo fondo con movimiento identificado.
- Seq_3 (de la secuencia *video11* en [8]): 5 imágenes con mucho movimiento, una persona moviéndose por la imagen así como primer plano sintético al mismo tiempo.

En la figura 6.1 podemos observar imágenes representativas de las secuencias anteriores. Cada columna de la imagen corresponde a cada una de las secuencias descritas en el mismo orden.

6.2.2. Pruebas

En las figuras 6.2, 6.3 y 6.4 observamos el comportamiento de los tres algoritmos analizados en las tres secuencias seleccionadas. En la primera figura se observan los efectos de la umbralización en fondos estáticos donde a simple vista tanto Rosin como Otsu no son adecuados en este tipo de fondos pues generan demasiado primer plano en la máscara binaria donde no existe movimiento alguno. Consecuentemente, nuestro algoritmo formaría más *clusters* de los necesarios. En la segunda y en la tercera figura figuras se muestra el resultado de aplicar los algoritmos a secuencias con primer plano en movimiento. En general Otsu y kapur obtiene resultados semejantes pero Rosin no funciona de manera adecuada. Se identifica a Kapur como el mejor método para

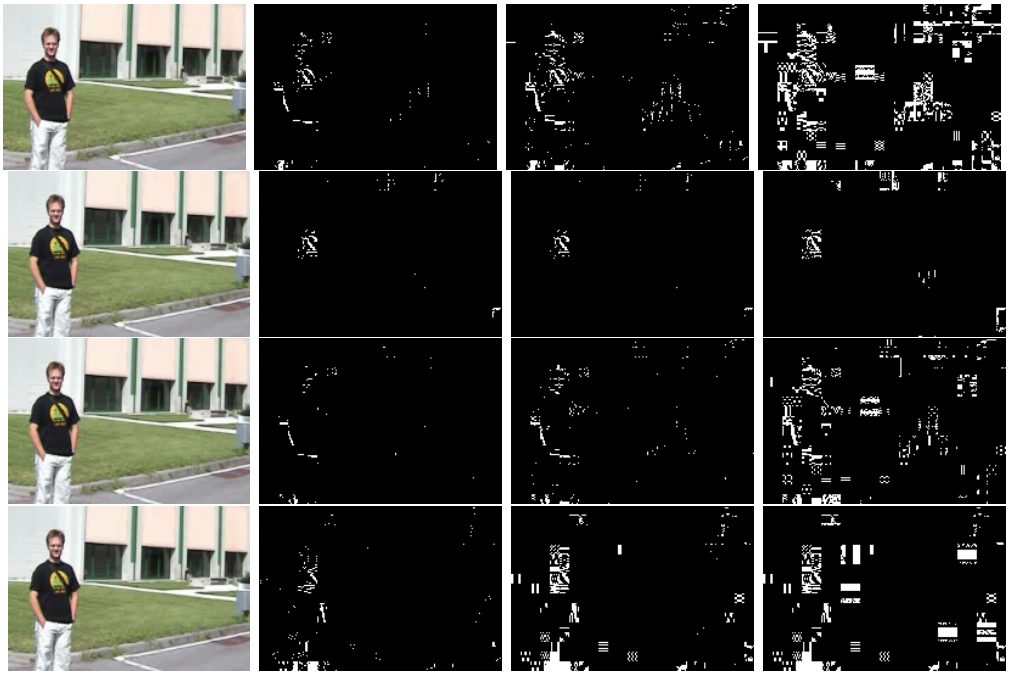


Figure 6.2: Ejemplo de *frame difference* en secuencia sin movimiento. La primera columna muestra la imagen actual y los resultados en la segunda (Kapur), tercera (Otsu) y cuarta (Rosin).

realizar el *frame difference* adaptativo pues funciona mejor en los casos sin y con movimiento de primer plano.

6.3. Agrupamiento temporal

El objetivo de los siguientes experimentos es validar la etapa de agrupamiento para lo cual hay que estudiar el tamaño máximo de bloque, el análisis PCA y el agrupamiento jerárquico con la posterior validación de índices.

6.3.1. Conjunto de datos (*dataset*)

Para la realización de los experimentos se creó un conjunto de datos sintético intentando emular los distintos tipos de escenarios posibles que se pueden encontrar cuando se analiza un bloque a lo largo de la secuencia de entrenamiento. Las principales variantes en un fondo son el ruido de la cámara y la cantidad de primer plano. Por ello sobre dos fondos se generaron datos con distintos niveles de ruido, distintos tamaños y probabilidades de aparición de primer plano.

Se parte de una imagen de dos de las secuencias disponibles, en este caso *foliage* y *video11* como se muestra en la figura 6.5:

- Ambas imágenes se replicaron 200 veces para formar secuencias de vídeo. En ellas se introdujo ruido blanco gaussiano simulando el ruido de cámara con distintos niveles de su

varianza (entre $1 \cdot 10^{-5}$ *y* $3.5 \cdot 10^{-5}$ en incrementos de $0.5 \cdot 10^{-5}$) más la secuencia sin ruido. Por tanto se forman un total de 7 secuencias para cada imagen de partida.

- Se forman imágenes de primer plano consistentes en bloques de distintos tamaños: 3x3, 10x10; 20x20; 30x30; 50x50; 80x80; 100x100. Las imágenes de primer plano son los bloques donde todos los píxeles son 0 menos ciertos píxeles aleatorios (que realmente representan el primer plano). La figura 6.6 muestra figuras de primer plano para distintos tamaños. Para cada tamaño se generan 200 imágenes con valores de píxel aleatorios del 0 al 255, posteriormente para las 10 primeras imágenes se ponen a 0 los valores superiores a 5, de la imagen 10 a la 20 se ponen a 0 los superiores a 10, de la 20 a la 30 por encima de 30, de la 30 a la 50 por encima de 50, de 50 a 80 por encima de 80, de 80 a 150 por encima de 150 y el resto por encima de 200. Así hay distintas imágenes de primer plano.

- Para cada una de las imágenes de las secuencias anteriores se inserta o no una imagen de primer plano con una probabilidad de aparición de primer plano. La probabilidad de primer plano es para cada caso 0.1, 0.25, 0.5, 0.75 ó 0.9.

En la figura 6.7 observamos como para cada una de las imágenes de partida se le ha añadido una imagen de primer plano en el primero de los bloques el cual será objeto del análisis.

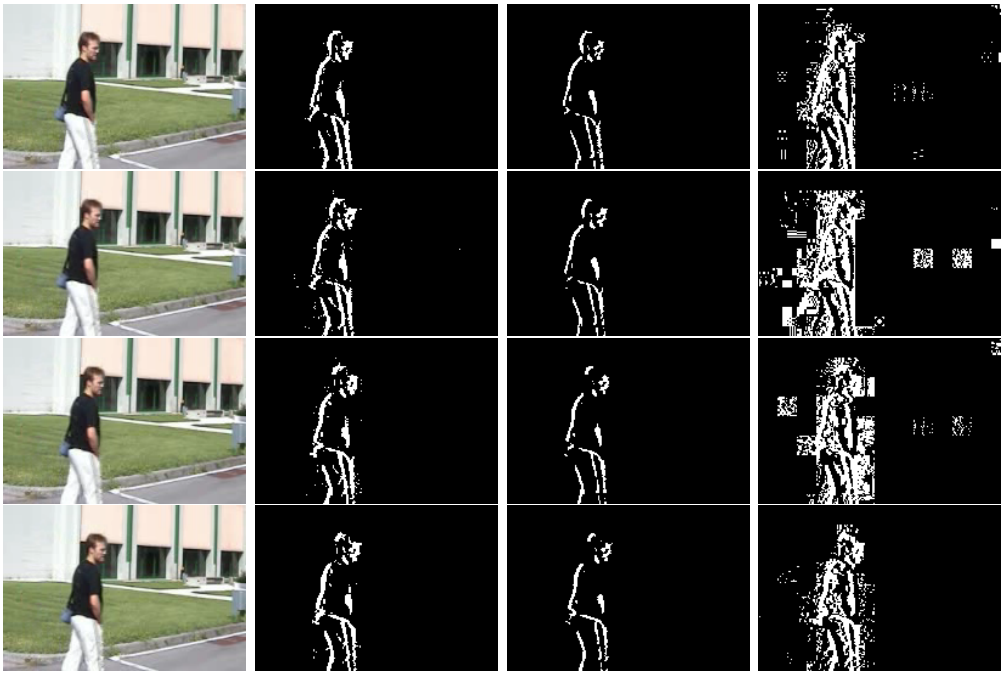


Figure 6.3: Ejemplo de *frame difference* en secuencia con movimiento. La primera columna muestra la imagen actual y los resultados en la segunda (Kapur), tercera (Otsu) y cuarta (Rosin).

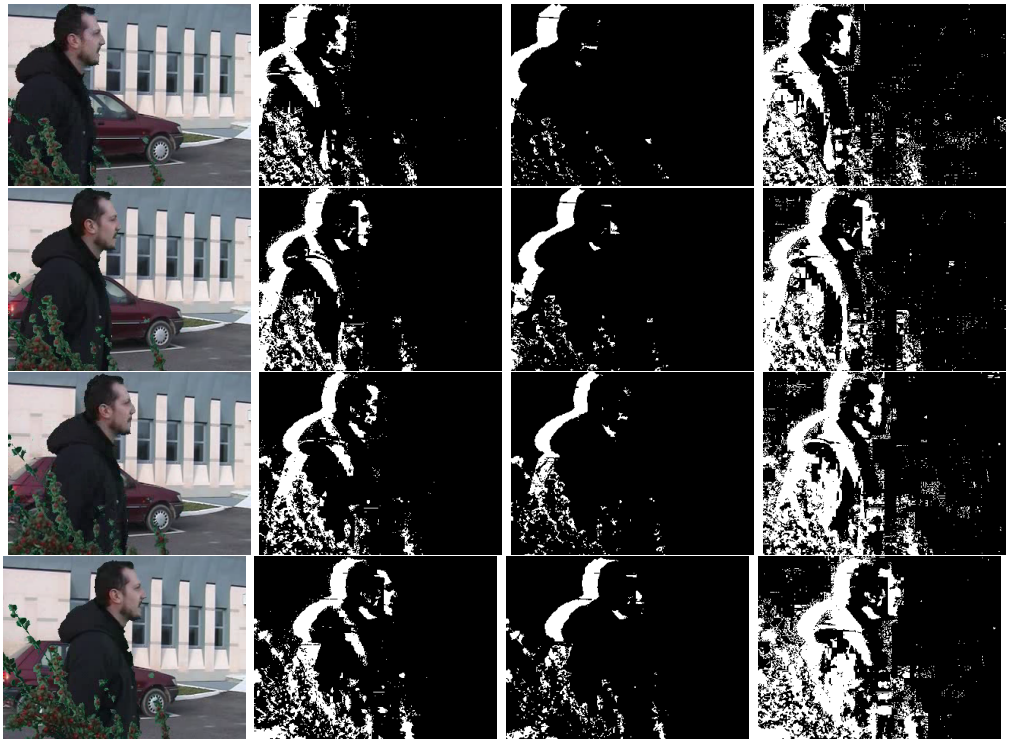


Figure 6.4: Ejemplo de *frame difference* en secuencia con movimiento. La primera columna muestra la imagen actual y los resultados en la segunda (Kapur), tercera (Otsu) y cuarta (Rosin).

6.3.2. Pruebas tamaño máximo de bloque y PCA

Para validar el tamaño máximo es necesario comparar el tiempo de ejecución del agrupamiento así como el propio agrupamiento en función del tamaño máximo del bloque. Para la evaluación de estas pruebas se utiliza la secuencia número uno con ruido $2 * 10^{-5}$ y probabilidad de primer plano de 0.75. Se miden:

1. Tiempo de ejecución.

2. Funcionamiento del agrupamiento. El tamaño del bloque influye en el funcionamiento del algoritmo en tanto en cuanto dependiendo del tamaño elegido los *clusters* se formarán con mayor o menor facilidad. Es decir si se forman *clusters* grandes es más probable que siempre haya algo de primer plano dentro del mismo. Sin embargo si se forman *clusters* pequeños será difícil discernir cuando el bloques es fondo de escena y cuando es primer plano.

La tabla 6.1 muestra como para mayor tamaño de bloque, el algoritmo se muestra más lento pero no ocurre así cuando reducimos la dimensionalidad de los datos mediante PCA como es nuestro caso. Por este motivo se decide procesar los datos de entrada del agrupamiento mediante el análisis PCA.

Figure 6.5: Imágenes semilla del *dataset* sintético

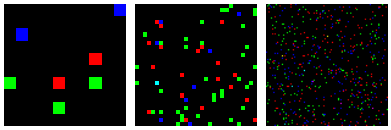


Figure 6.6: Ejemplos de primer plano para formar *dataset* sintético (tamaños 10x10, 30x30, 100x100).

Figure 6.7: Imágenes del *dataset* sintético

Tamaño bloque	3x3	10x10	20x20	30x30	50x50	80x80	100x100
Tiempo ejecución sin pca	1.0279	1.5514	12.5893	81.7115	no memoria	no memoria	no memoria
Tiempo ejecución con pca	No realizable	2.1992	0.5967	0.5758	1.0859	1.2754	1.2270

Table 6.1: Tiempos de ejecución en función del tamaño del bloque y de un análisis previo PCA.

Tamaño bloque	3x3	10x10	20x20	30x30	50x50	80x80	100x100
Sil	7.7145	13.7018	0	6.1425	no memoria	no memoria	no memoria
Db	13.9138	13.7018	16.0997	6.1425	no memoria	no memoria	no memoria
Rmss	7.7145	15.6545	0	0	no memoria	no memoria	no memoria
Spr	3.0069	63.1573	2.7006	93.9485	no memoria	no memoria	no memoria
Cd	37.6125	5.7497	10.9472	93.9485	no memoria	no memoria	no memoria
Total	7.7145	15.6545	0	1.0967	no memoria	no memoria	no memoria

Table 6.2: Análisis del funcionamiento del agrupamiento en función del tamaño de bloque sin PCA (función β).

Tamaño bloque	3x3	10x10	20x20	30x30	50x50	80x80	100x100
Sil	No realizable	0	0	1.1188	24.7814	17.2253	19.5791
Db	No realizable	13.7018	0	6.1425	24.7814	23.4242	19.5791
Rmss	No realizable	2.8135	2.7006	1.0967	2.6177	0	0
Spr	No realizable	2.8135	4.6237	93.9485	19.3841	11.5699	2.6762
Cd	No realizable	2.6274	4.0017	93.9485	2.6177	28.6826	83.9370
Total	No realizable	2.8135	4.6237	1.0967	2.6177	1.1531	17.4504

Table 6.3: Análisis del funcionamiento del agrupamiento en función del tamaño de bloque con PCA (función β).

Posteriormente, analizamos si los *clusters* realizados son similares a los que hemos introducido artificialmente en el dataset. Para ello utilizamos la función β que muestra la diferencia entre el *cluster* que marca el *ground-truth* como óptimo y el elegido a través del método del agrupamiento más validación de índices tras la reducción de PCA.

$$\beta = |n_{gt} - n_{algotitmo}|.$$

La tabla 6.2 muestra la función Beta para la secuencia sin un análisis PCA previo y la tabla 6.3 muestra los resultados para la misma secuencia habiendo hecho un análisis previo. En el conjunto de las tablas se observa que globalmente, nuestro algoritmo funciona mejor con PCA acorde a la función. Se observa como los mejores resultados se obtienen para valores de tamaño de ventana de entre 20 y 50. Como nuestra aproximación reduce el tamaño de bloque si no es capaz encontrar intervalos estables para dicho bloque se decide utilizar un tamaño de 40x40. De este modo en caso de no encontrar estabilidades habrá 3 etapas posteriores correspondientes a 20x20; 10x10; 5x5 que será el mínimo.

6.3.3. Pruebas validación del agrupamiento

El agrupamiento temporal está formado por el agrupamiento jerárquico más una etapa de validación del mismo. La manera de medir la efectividad o no del mismo viene dada por el conjunto de ambas.

El objetivo es que los índices de validación del agrupamiento sean capaces de escoger el mejor de los agrupamientos realizados entre el máximo y mínimo explicados en capítulos anteriores (es decir, el número óptimo de *clusters*). Para ello se proponen dos formas de hacerlo: visualmente y objetivamente.

- Visualmente el ojo humano es capa de discernir cuantas imágenes distintas puede haber en una secuencia corta de entrenamiento.
- Objetivamente consiste en introducir primer plano conocido en cualquier fondo de escena. Para ello el experimento realizado es semejante al estudio del tamaño de bloque.

Con la función β explicada anteriormente y el conjunto de datos, reducido a 20 imágenes por secuencia, se evaluó el agrupamiento temporal con la validación de índices. A continuación en las figuras 6.8 y 6.9 observamos como varía la función β en relación con el ruido de la cámara y la cantidad de primer plano. Para altos niveles de primer plano como el 90% y 75 %, los cuales representan mejor el problema que se quiere resolver, se puede observar como los índices *Silhouette* y *Dabies Bouldin* son los que mejor rendimiento ofrecen ya que se mantienen siempre por debajo de $Beta = 6$. El índice *Silhouette* es el que mejor se comporta para todos las cantidades de primer plano excepto para 0.1 %, cuyo caso no es objeto de análisis en este trabajo. Con el índice *Dabies Bouldin* ocurre algo semejante pero tampoco se comporta de manera adecuada con 0.25 % de probabilidad de primer plano. Tanto el índice *CD* como *SPR* ofrecen los peores resultados posibles con picos de $Beta \geqslant 10$ para altas probabilidades de aparición de primer plano (el caso analizado en este proyecto). El índice *RMSS* se comporta de manera adecuada en la secuencia número dos pero para la primera ofrece un pico de $Beta = 9$ para 0.75% de primer plano. El índice total consiste en ponderar todos los índices del mismo modo que se explica en el capítulo 4, se observa como al ponderar varios índices el resultado global es mejor ya que se reduce la función $Beta$.

En las figuras 6.10 6.11 se observa el funcionamiento de los índices *Silhouette y Dabies Bouldin* (los de mejor funcionamiento para alta probabilidad de primer plano) habiendo realizado un análisis PCA previo. En las figuras se observa como ambos funcionan de la misma manera que sin el análisis PCA.

Posteriormente se observó el funcionamiento con las secuencias reales de nuestro *dataset* para comprobar el funcionamiento. En las siguientes figuras 6.12 6.13 se muestra un ejemplo para el vídeo *foliage*. Para una secuencia de entrada 6.12, después de realizar el agrupamiento jerárquico para distintos números de *clusters* y evaluarlos con los índices estudiados anteriormente se forman los distintos clusters para un número de *clusters* óptimo encontrado mediante los índices evaluados.

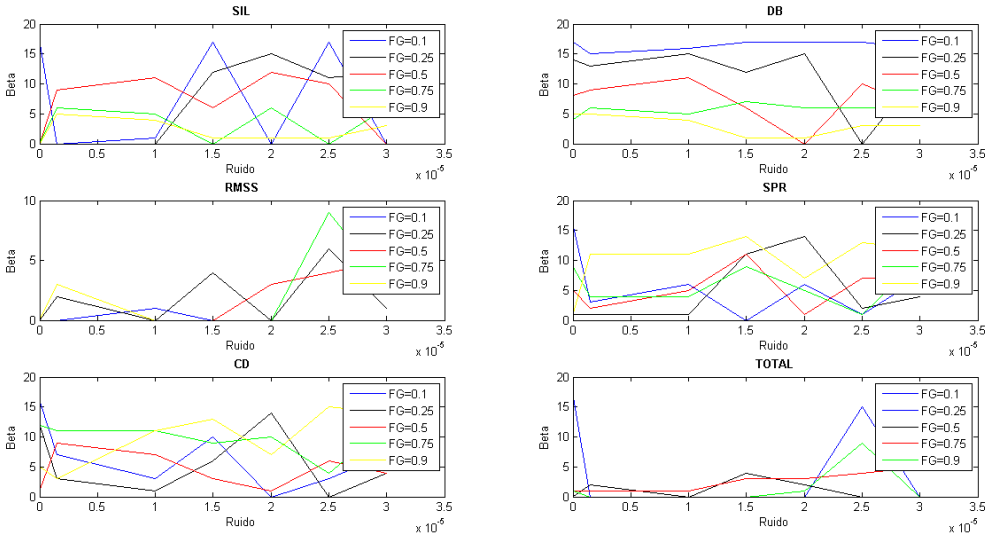


Figure 6.8: Índices de validación para la secuencia 1 sin análisis PCA.

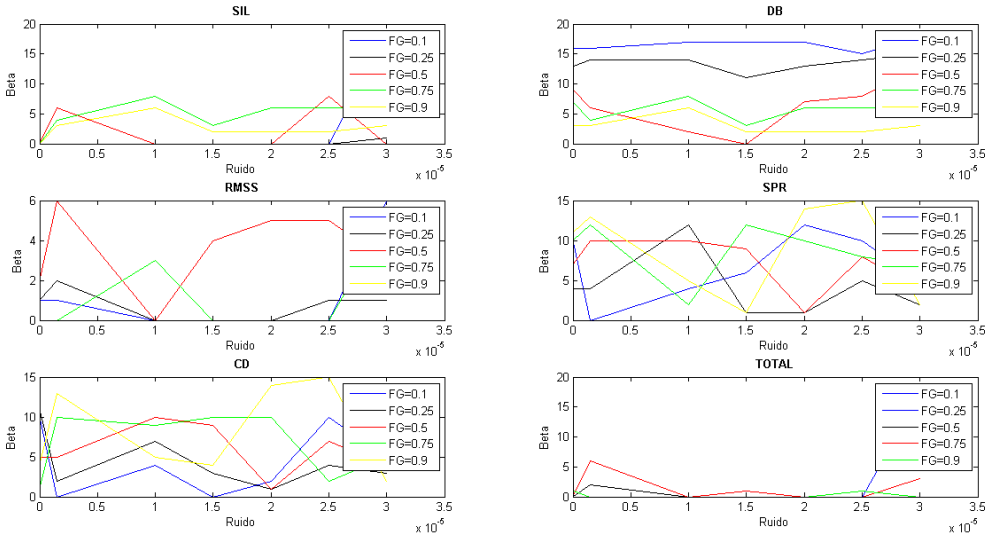


Figure 6.9: Índices de validación para la secuencia 2 sin análisis PCA.

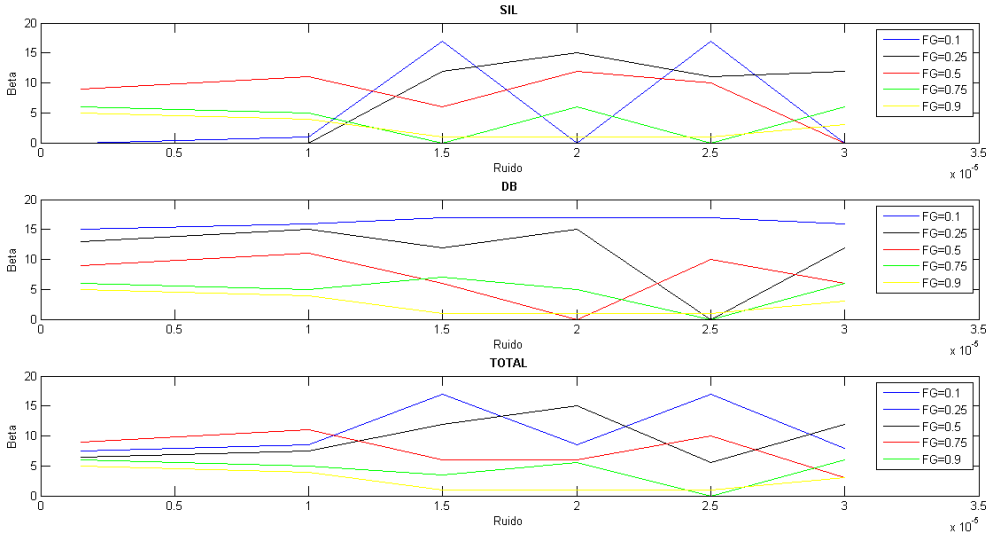


Figure 6.10: Índices de validación (SIL, DB) para la secuencia 2 con análisis PCA.

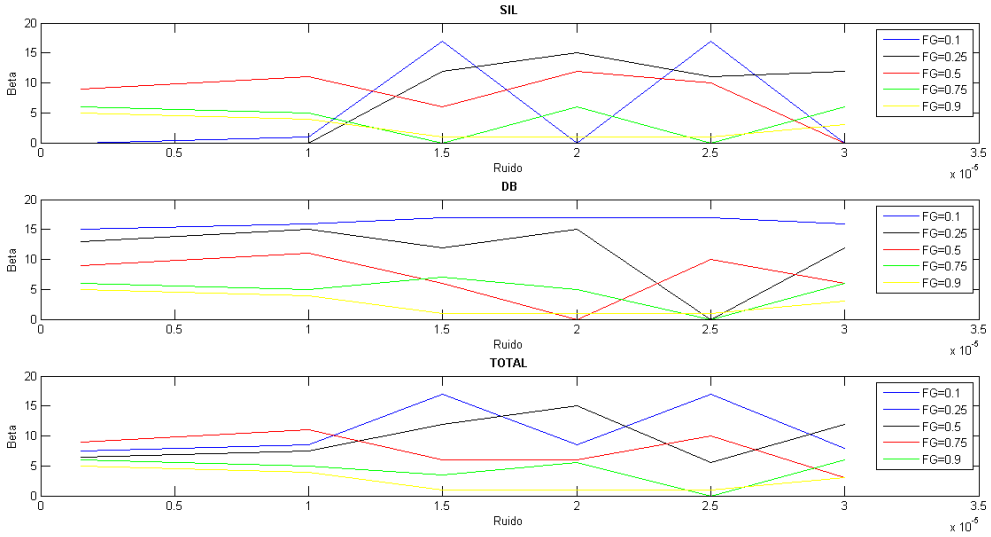


Figure 6.11: Índices de validación (SIL, DB) para la secuencia 1 con análisis PCA.

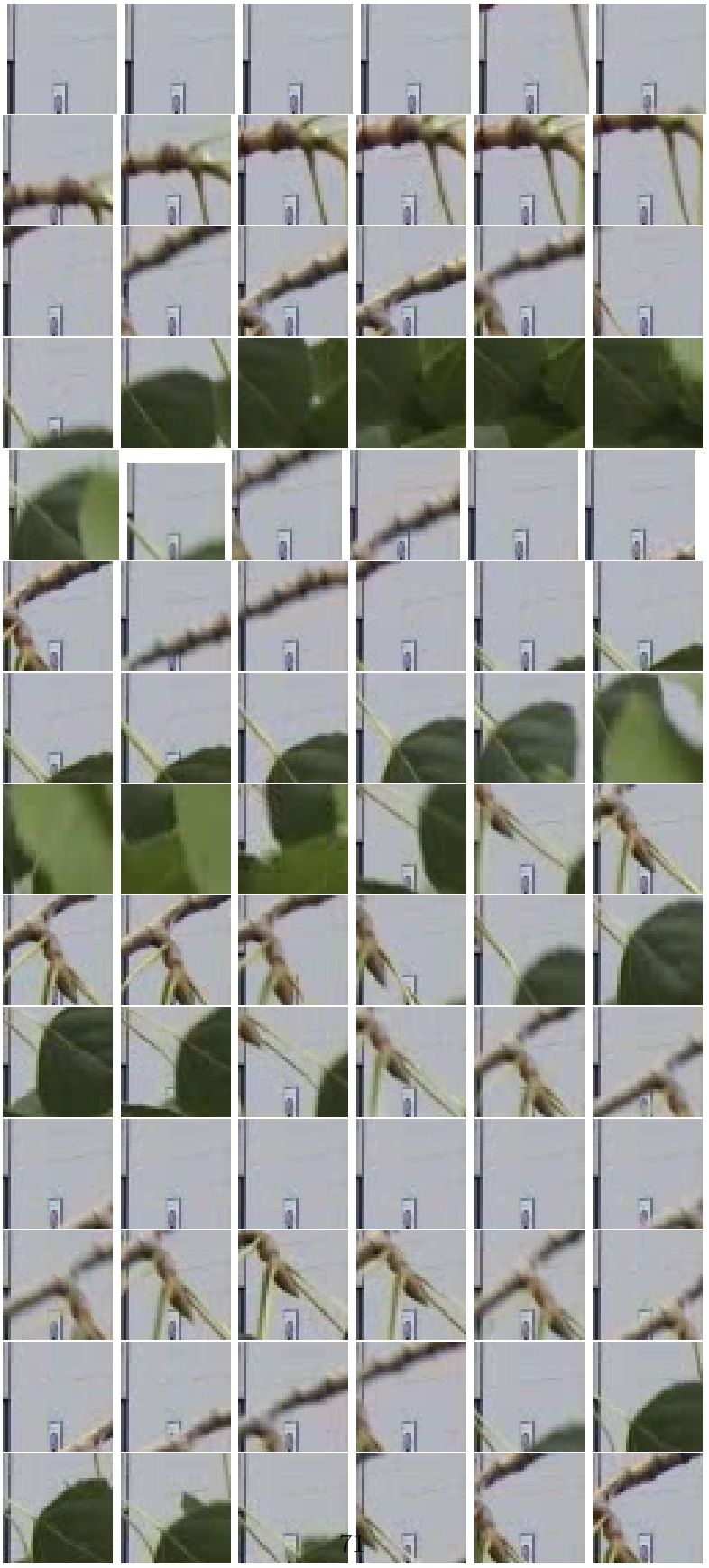


Figure 6.12: Imágenes de una secuencia de entrada a la etapa de agrupamiento

Figure 6.13: Clusters

6.4. Sistema global

6.4.1. Conjunto de datos

Para las pruebas de las distintas características espaciales así como para el algoritmo final se han utilizado las secuencias de la tabla 6.4.

6.4.2. Métricas

Para evaluar tanto las pruebas espaciales como los distintos algoritmos en la sección 6.4.4 hemos utilizado la comparación con un *ground-truth* (imagen de fondo generada manualmente). Conseguimos una imagen de error mediante la diferencia absoluta de las mismas (ground-truth

Vídeo	Frames	Tipo	FG sintético	Problema	Dificultad
video11 [8]	349	exterior	SI	poca visibilidad de BG	alta
board [8]	400	interior	NO	sombras	alta
ca_*vignal [8]*	258	exterior	NO	sleeping person	media
foliage [8]	258	exterior	SI	poca visibilidad de BG	alta
granguardia [8]	463	exterior	NO	sleeping person/sombras	alta
Snellen [8]	334	sintético	SI	poca visibilidad de BG	alta
bootstrap [8]	294	interior	NO	poca visibilidad de BG	alta
camara1 (TRECVID 2010)	500	interior	NO	poca visibilidad de BG	muy alta
camara2 (TRECVID2010)	300	interior	NO	poca visibilidad de BG	muy alta
camara3 (TRECVID2010)	300	interior	NO	poca visibilidad de BG	muy alta
camara4 (TRECVID2010)	300	interior	NO	sleeping person	media
camara5 (TRECVID2010)	350	interior	NO	poca visibilidad de BG	muy alta
vid8 (LIRIS 2012)	315	interior	NO	poca visibilidad de BG/sombras	alta
vid16 (LIRIS 2012)	380	interior	NO	sleeping person/sombras	baja
vid22 (LIRIS 2012)	345	interior	NO	sleeping person	media
vid36 (LIRIS 2012)	128	interior	NO	poca visibilidad de BG	baja
vid44 (LIRIS 2012)	254	interior	NO	poca visibilidad de BG	media
vid62 (LIRIS 2012)	208	interior	NO	poca visibilidad de BG/sleeping person	alta
vid80 (LIRIS 2012)	689	interior	NO	poca visibilidad de BG/sleeping person	alta
sequence1 [30]	400	interior	NO	sleeping person	alta
sequence2 [30]	400	interior	NO	poca visibilidad de BG/sombras	media

Table 6.4: Conjunto de secuencias seleccionadas para experimentos con el sistema completo.

y estimación). Las técnicas utilizadas son :

- *Number of Error pixels (NE).* Número de píxeles erróneos. Un píxel erróneo es aquel cuyo valor de intensidad obtenido de fondo de escena a nivel de gris difiere en 20 del valor real del GT o *ground truth* a nivel de grises.

$$If\ |BG(x,y) - GT(x,y)| > 20,\ \forall p \in (x,y) \rightarrow BG(x,y) = Error$$

- *Average gray-level Error (AE).* Error medio del nivel de gris. Es la diferencia entre los fondos reales y los estimados. Si la diferencia entre píxeles de fondo estimado y real es mayor que un umbral , entonces es un error, este umbral es 20 para asegurar calidad.

$$\{\sum_{w=1}^{w} \sum_{y=1}^{H} BG(x,y)\}/(W * H),\ BG(x,y) \in Error.$$

- *Number of Clustered error pixels* (NC). Número de píxeles conectados erróneos. Este error se produce cuando los 4 vecinos conectados a un píxel de error también son erróneos. Esta medida es la más relevante pues un error en ella indica que posiblemente se debe a un objeto de primer plano.

6.4.3. Evaluación características

El objetivo de la etapa espacial es conseguir decidir que *cluster* es más propicio a ser fondo de escena en un determinado bloque atendiendo a características espaciales. Para ello se han estudiado los resultados de las cuatro características utilizadas: continuidad de color en el borde (C1), objetos en el interior de los candidatos (C2), probabilidad de semejanza estructural (C3) y suavidad espectral de la DCT (C4) (en este caso entendida en el mismo bloque).

Al reconstruir el fondo de manera creciente un error en un bloque acarrea multitud de problemas. Por ello se estudió la combinación de la distintas características mencionadas para minimizar el número de bloques erróneos. Como explica el capítulo 5 se consideró que si todas o dos de las características: continuidad de color en el borde, estudio de objetos en el interior de los candidatos y probabilidad de semejanza estructural coincidían el *cluster* de análisis es considerado fondo de escena. En caso contrario se hicieron distintas pruebas donde en cada una de las mismas se utilizaba uno de los criterios para resolver esos bloques inciertos. Es decir, en los bloques donde no coincidían se probó a insertar en el fondo de escena cada uno de los G_s $s = [1, 2, 3, 4]$.

La tabla 6.5 lista los distintos errores para todas las secuencias disponibles. En la tabla 6.6 se muestra la media de cada error para todas las secuencias. La característica de estructuras se muestra como la mejor en media pero no es fiable pues en ocasiones obtiene un fondo de escena erróneo. Sin embargo es imposible decidir cuál de ellas es mejor pues depende de cada secuencia

	NE				AE				NC			
	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4
video 11	767	17838	9480	488	0,0099	0,2322	0,1234	0,0063	363	15393	7517	238
board	9008	9656	5578	9656	0,2746	0,2943	0,1700	0,2943	8298	6033	3972	6033
ca_vignal	1	0	0	0	0	0	0	0	0	0	0	0
foliage	0	0	0	0	0	0	0	0	0	0	0	0
granguardia	4494	103	192	101	0,1291	0,0029	0,0055	0,0029	2783	6	22	6
Snellen	0	328	1053	1057	0	0,0149	0,0480	0,0482	0	179	711	711
bootstrap	2430	1702	2181	1636	0,1265	0,0886	0,1135	0,0852	1301	985	1286	947
camara1	10438	9819	10561	9819	0,5436	0,5114	0,5500	0,5114	6156	5646	6288	5646
camara2	11845	11931	11993	11993	0,6169	0,6214	0,6246	0,6246	6258	6397	6581	6581
camara3	6068	5153	5981	5155	0,3160	0,2683	0,3115	0,2684	2858	2083	2804	2083
camara4	2184	2168	2202	2203	0,1137	0,1129	0,1146	0,1146	564	554	567	568
camara5	7859	8082	8111	8072	0,4093	0,4209	0,4224	0,4204	4387	4527	4535	4512
vid8	34895	21027	10795	21027	0,3365	0,2028	0,1041	0,2028	30680	18048	7348	18048
vid16	173	427	118	997	0,0016	0,0041	0,0011	0,0096	24	39	9	423
vid22	455	277	846	277	0,0043	0,0026	0,0081	0,0026	121	70	239	70
vid36	11563	11563	11565	11563	0,1115	0,1115	0,1115	0,1115	9028	9028	9028	9028
vid44	246	246	246	246	0,0023	0,0023	0,0023	0,0023	131	131	131	131
vid62	2853	819	1374	4465	0,0275	0,0078	0,0132	0,0430	1858	209	591	3118
vid80	376	9018	229	690	0,0036	0,08697	0,0022	0,0066	124	7527	42	367
sequence1	4415	4910	3764	8575	0,0574	0,0639	0,0490	0,1116	3079	3286	2538	6098
sequence2	49660	49336	49257	49417	0,6466	0,6423	0,6413	0,6434	35567	35259	35166	35384

Table 6.5: Análisis de las características espaciales para el dataset disponible. Las características analizadas son continuidad de color en el borde (C1), objetos en el interior de los candidatos (C2), probabilidad de semejanza estructural (C3) y suavidad espectral de la DCT (C4). En rojo se marca la mejor puntuación (menor error).

	C1	C2	C3	C4
Media NE	7606.2	7828.7	*6453.6*	7020.8
Media AE	0.1777	0.1759	*0.1627*	0.1672
Media NC	5408.6	5495.2	*4256.0*	4.7615

Table 6.6: Errores medios de las características analizadas: continuidad de color en el borde (C1), objetos en el interior de los candidatos (C2), probabilidad de semejanza estructural (C3) y suavidad espectral de la DCT (C4).

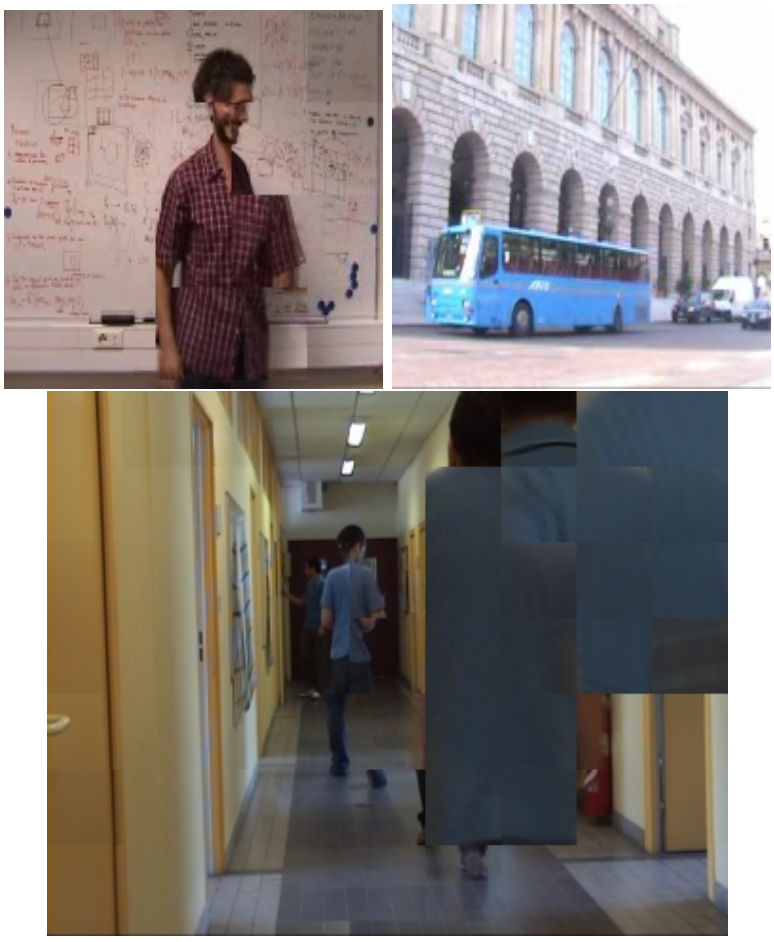


Figure 6.14: Ejemplos fallidos de aplicar continuidad borde como característica única

de entrada. Por ello se opta por una aproximación donde se ponderen de una u otra forma todas y cada una de ellas con el objetivo de minimizar errores.

En la figura 6.14 observamos ejemplos de fallos al utilizar continuidad de borde. De arriba abajo y de izquierda a derecha se muestran los fondos obtenidos para las secuencias *board*, *granguardia* y *vid8*. En la primera y en la tercera imágenes los errores son evidentes y en la segunda el autobús no corresponde al fondo de escena. Cuando por el efecto de una sombra, por ruido o porque un cambio de bloque coincide con un borde se reconstruye erróneamente el error se acarrea al resto de la imagen de fondo.

En la figura 6.15 se muestra que el algoritmo usando las diferencias de color en ocasiones falla por camuflaje. El ejemplo es para la secuencia *vídeo11*. Las tres primeras imágenes muestran el primer bloque donde falla el algoritmo y a partir del cual el resto se construye de manera errónea. Observamos como un objeto de primer plano se inserta en un fondo de escena del mismo color y la diferencia de color pasa a ser mínima en el cluster con primer plano.

Figure 6.15: Ejemplos fallidos de aplicar el estudio de objetos en el interior del bloque como característica única

En la figura 6.16 muestra fallos al usar el HOG como característica. De arriba abajo y de izquierda a derecha se muestran los fondos obtenidos para las secuencias *board*, *bootstrap*, *vídeo11*. Cuando hay fondos texturados no funciona correctamente .

En 6.17 observamos fallos al usar la DCT, el problema de esta característica es que al no tener en cuenta bloques vecinos si en un *cluster* hay un objeto de primer plano que ocupa todo el tamaño de bloque y no tiene cambios de frecuencia puede resultar que tenga menos energía que el propio fondo de escena. De arriba abajo y de izquierda a derecha se muestran los fondos obtenidos para las secuencias *sequence1* y *vid62*.

Se han hecho ligeras pruebas con otras características sin llegar a probar el *dataset* completo debido al escaso éxito alcanzado:

- Utilizando flujo óptico [34] [35]. El objetivo es definir un movimiento de objetos a lo largo de la secuencia para poder determinar en intervalos estables si el objeto que produjo ese movimiento anterior y posterior esta dentro o no del bloque.

- Máquina de estados. El objetivo es crear una máquina de estados en función de los píxeles en movimiento de imágenes anteriores y posteriores a los intervalos estables.

- Histogramas de color. El objetivo era definir utilizar el histograma de color como característica espacial para aumentar el fondo de escena.

La conclusión es que en ambientes muy poblados de primer plano es imposible identificar los movimientos reales del primer plano pues se confunden unos con otros si hay varios objetos moviéndose en varias direcciones distintas y coincidiendo en la misma región.

6.4.4. Sistema propuesto

Como consecuencia de los resultados mostrados en el apartado anterior (ninguna característica es mejor que otra en todo momento) se decidió aplicar el algoritmo propuesto con la DCT como característica discriminatoria. Sin embargo se definió el concepto de macro-bloques explicado en el capítulo 5. Adicionalmente, se comparó el algoritmo propuesto con tres aproximaciones representativas del estado del arte en inicialización de fondo de escena (*Wang* et al [30], *Reddy* et al [12] y *Colombari* et al [8]).

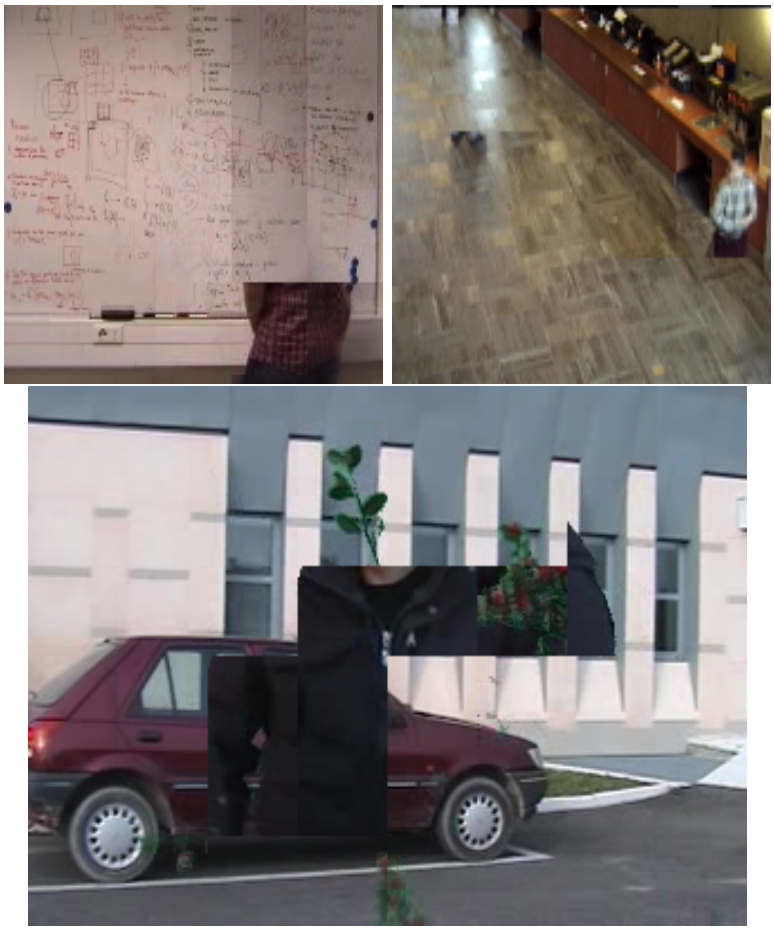


Figure 6.16: Ejemplos fallidos de utilizar la comparativa de estructuras como característica única

Figure 6.17: Ejemplos fallidos por utilizar la DCT como característica única

A continuación se muestran las tablas con los resultados con el dataset descrito anteriormente 6.7. Se observa que la *Wang* no soporta las secuencias *sequence1* y *sequence2* pues no hay memoria suficiente. *Colombari* no puede ser comparada con gran parte del dataset pues no se dispone del código necesario. Adicionalmente los resultados mostrados se obtienen de comparar los vídeos con los fondos disponibles en la misma url indicada en el capítulo 2. En la tabla 6.8 se observa como la *Wang* es la que a priori se comporta mejor pero hay que tener en cuenta que no puede ser ejecutada para dos de las secuencias y como veremos las imágenes de fondo suelen obtenerse emborronadas. *Reddy* necesita de una semilla que sea estable durante toda la secuencia de entrenamiento, lo que es difícil de encontrar y en caso de disponer de la misma necesita que cada bloque que se quiere reconstruir este rodeado de dos o más bloques con fondo de escena. Debido a ello en la mayor parte de las ocasiones no puede terminar de reconstruir el fondo pues necesita mucha estabilidad en la secuencia. Nuestra aproximación resulta ganadora en gran cantidad de las secuencias disponibles. En la tabla 6.9 se ofrece una comparación sobre el tiempo medio que necesita cada algoritmo para obtener una imagen de fondo de escena de una secuencia de entrenamiento de 300 imágenes con un tamaño 320x240. Se observa como *Wang y Reddy* se muestran muy superiores pues tardan escasos minutos, todo lo contrario que *Colombari* que tarda más de 6 horas. Nuestra aproximación está entre medias con 2 horas de media. Todas ellas ejecutadas en el entorno de programación matlab.

En la figura 6.18 observamos los fondos obtenidos en nuestro dataset para la aproximación de Wang et al [30]. Observamos como en general falla para fondos con primer plano estático o fondos donde no aparece el fondo de escena en varias imágenes consecutivas.

En la figura 6.19 observamos los resultados de la aproximación realizada por Reddy et al [12] para las secuencias de *ca_vignal*, *gran_guardia*, *cámara1*, *cámara2*, *cámara3*, *cámara4*, *cámara5*, *vid16* y *vid44*. La aproximación tiene dos graves problemas. El primero de ellos es que necesita encontrar una semilla que sea estable durante toda la secuencia (las secuencias que no mostramos no cumplen esta condición y por tanto la aproximación no puede ejecutarse). El

	NE				AE				NC			
	Wang [30]	*Reddy*[12]	*Colom.*[8]	Propuesto	*Wang* [30]	*Reddy*[12]	*Colom.*[8]	Propuesto	*Wang* [30]	*Reddy*[12]	*Colom.*[8]	Propuesto
video11	21257	76167	9	*488*	0.2767	0.9917	0,0001	*0.0063*	8169	74267	0	*237*
board	2636	30700	5025	*182*	0.0803	0.9993	0,1701	*0,0055*	636	29953	626	*31*
ca_vignal	2147	11522	7379	*0*	0,0789	0,4688	0,2919	*0*	1249	11013	2963	*0*
foliage	3400	26576	11172	*0*	0,1148	0,9612	0,4225	*0*	848	24954	4120	*0*
granguardia	*3934*	9228	13692	4493	*0,1130*	0,3003	0,4197	0,1291	*2311*	8751	4501	2775
Snellen	16227	19324	1843	*1057*	0,7409	0,9319	0,0876	*0,0482*	12722	17539	168	*718*
bootstrap	*769*	17298	ND	2430	*0,0400*	0,9652	ND	0,1265	*375*	16327	ND	1303
camara1	10979	16010	ND	*9819*	0,5718	0,8934	ND	*0,5114*	6273	15196	ND	*5647*
camara2	*11832*	14665	ND	11864	*0,6162*	0,8183	ND	0,61791	*6343*	12676	ND	6408
camara3	*5187*	15645	ND	5264	*0,2701*	0,8730	ND	0,2741	*1759*	14749	ND	2178
camara4	*1907*	1613	ND	2203	*0,0993*	0,0900	ND	0,1147	*274*	328	ND	567
camara5	*7313*	16544	ND	7937	*0,3808*	0,9232	ND	0,4133	*3772*	15851	ND	4450
vid8	*3272*	96290	ND	10804	*0,0315*	0,9498	ND	0,1042	*355*	93443	ND	7330
vid16	4651	2125	ND	*138*	0,0448	0,0209	ND	*0,0013*	1199	1154	ND	*0*
vid22	4235	99948	ND	*437*	0,0408	0,9859	ND	*0,0042*	367	97791	ND	*121*
vid36	*1731*	100660	ND	11565	*0,0166*	0,9929	ND	0,1115	*422*	98667	ND	9030
vid44	467	39858	ND	*246*	0,0045	0,3931	ND	*0,0023*	*112*	38091	ND	131
vid62	*1978*	101197	ND	4465	*0,0190*	0,9982	ND	0,0430	*916*	99646	ND	3119
vid80	2413	50750	ND	*665*	0,0232	0,5006	ND	*0,0064*	404	49707	ND	*374*
sequence1	NM	8984	ND	*4427*	NM	0,1169	ND	*0,0576*	NM	6744	ND	*3059*
sequence2	NM	72947	ND	*49417*	NM	0,9498	ND	*0,6434*	NM	69825	ND	*35352*

Table 6.7: Análisis de las aproximaciones existentes para el dataset disponible. (Clave. ND:No Disponible. NM:No Memoria). En rojo se marca la mejor puntuación (menor error).

	Wang [30]	*Reddy*[12]	*Colombari* [8]	Propuesto
Media NE	5596.57	39431	10097.16	6090.52
Media AE	0.1875	0.7202	0.3022	0.1534
Media NC	2536.94	37936.76	5081.50	3934.33

Table 6.8: Errores medios de las características

	Wang [30]	*Reddy*[12]	*Colombari* [8]	Propuesto
Tiempo de ejecución	8 min	22 min	6h	2h15min

Table 6.9: Comparación tiempos de ejecución (tiempo medio).

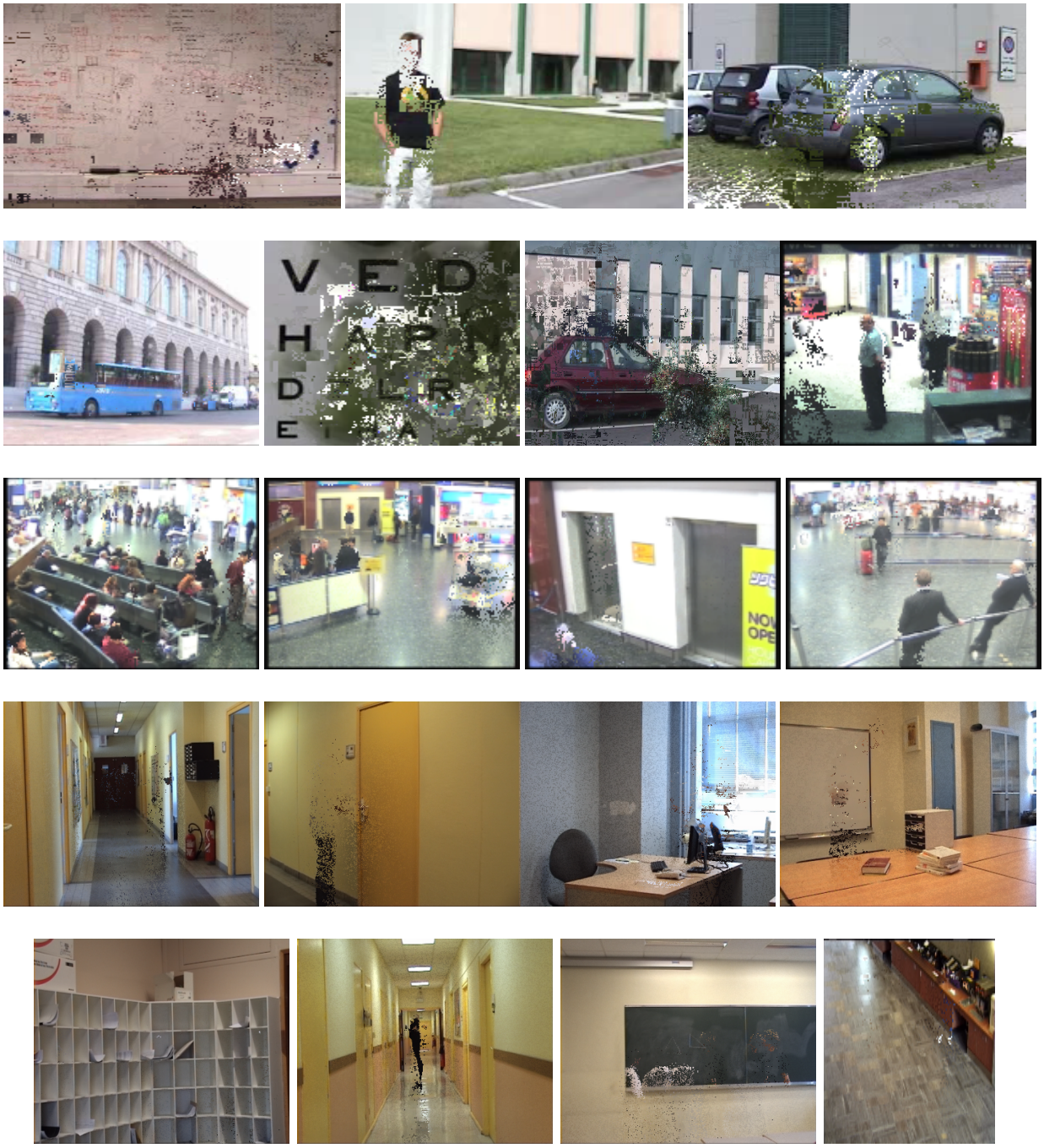


Figure 6.18: Fondos de escena obtenidos mediante Wang 2006

segundo es que para usar la DCT requieren que el bloque de análisis este rodeado de al menos dos de los bloques vecinos con fondo de escena y que estos sean contiguos con lo que el algoritmo concluye sin obtener el fondo en multitud de ocasiones. Por último al tener un tamaño de bloque fijo cuando las imágenes no son divisibles se pierde información.

Figure 6.19: Fondos de escena obtenidos mediante Reddy 2009

En la figura 6.20 se muestran los resultados obtenidos en [8]. Obtienen los mejores resultados visuales para las 6 secuencias disponibles en su dataset. Sin embargo los fondos que obtiene (primera columna) no son del mismo tamaño que la secuencia de entrada (segunda columna) si la imagen de la secuencia de entrada no es divisible por su tamaño de bloque (teniendo en cuenta el solapamiento). Adicionalmente la tablas 6.7 no se corresponden con lo visual dado que han podido comprimir la imagen con pérdidas para disponer de ella en la url descrita en el capítulo 2.

En la figura 6.21 se muestran los fondos obtenidos con nuestra aproximación. Funciona correctamente en nueve de los veintiún vídeos disponibles.

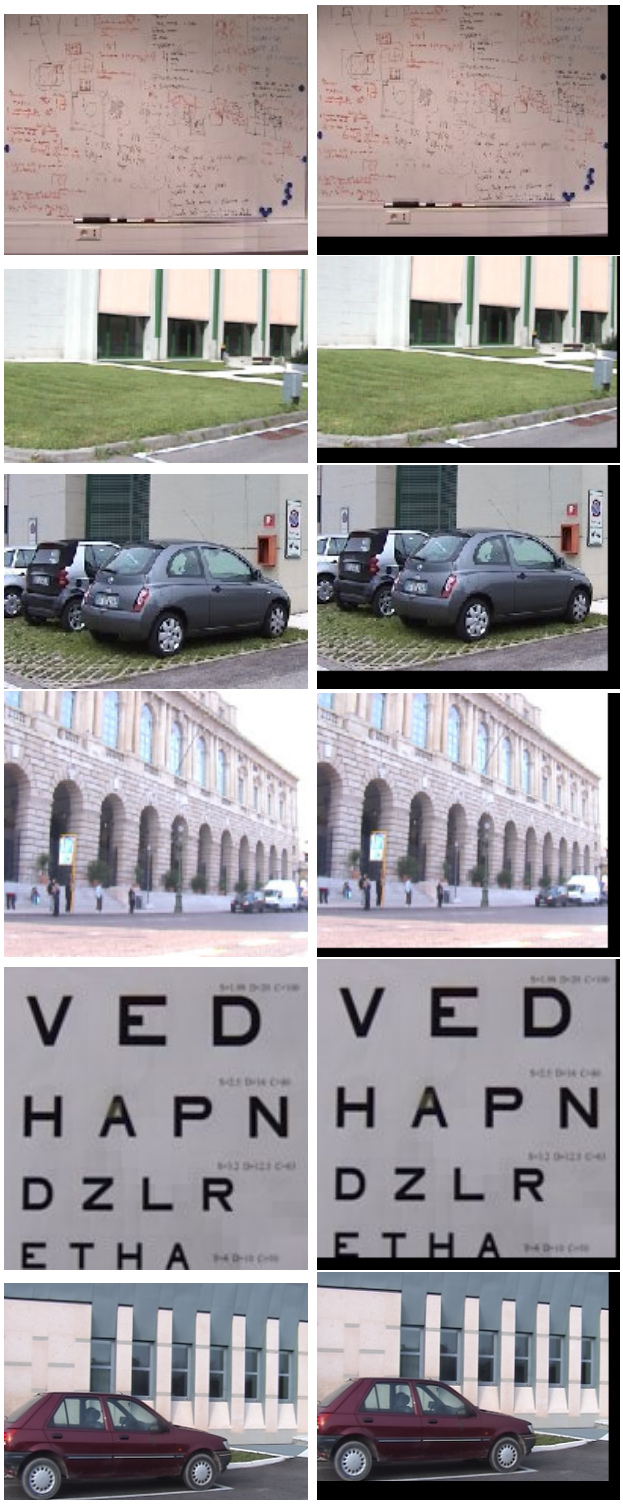


Figure 6.20: Fondos de escena obtenidos en Colombari et al [8]. La primera columna muestra los fondos obtenidos (descargados desde su página web) y la segunda columna muestra el fondo de escena ampliado al tamaño de las imágenes que componen la secuencia.

Figure 6.21: Fondos de escena obtenidos mediante nuestro algoritmo

Capítulo 7

Conclusiones y trabajo futuro

7.1. Resumen del trabajo

En este documento hemos presentado un nuevo algoritmo de inicialización de fondo en secuencias permitiendo que se pueda obtener el mismo en secuencias muy pobladas así como con efectos de camuflaje u objetos de primer plano estáticos en el fondo de escena.

Primero se hizo un extenso estudio del estado del arte lo cual fue necesario para comprender tanto el funcionamiento de un algoritmo de inicialización como las diferentes métricas para evaluar los resultados y los diferentes *datasets* disponibles en este ámbito. Este estudio ha dado lugar al capítulo 2

Después de esto se prestó especial atención a los algoritmos que parecían tener mejores resultados atendiendo a sus características espacio-temporales e intentando observar cuales eran las virtudes y defectos de cada uno. La principal idea era mejorar los algoritmos existentes implementando uno nuevo que fuese capaz de funcionar en secuencias muy pobladas como son las secuencias de vídeo-seguridad.

Con la información disponible del estudio de esos algoritmos y el objetivo de mejorar lo existente se decidió hacer un algoritmo en tres etapas tal como se relata en el capítulo 3. En una primera se estudiaría un tamaño apropiado de regiones de análisis, en una segunda se formarían candidatos temporales para dichas regiones y en la última se elegirían el mejor candidato de los disponibles en función del fondo existente. En la sección 3.2.1 se describe el proceso para obtener los bloques de análisis y las ventajas de usar esta esta etapa. Lo reseñable de la etapa es la utilización del uso de la técnica *frame difference* con un umbral adaptativo y saber que gracias a él somos capaces de operar sin seleccionar ningún umbral.

El siguiente paso del algoritmo es la etapa de agrupamiento donde se forman los candidatos a ser fondo de escena. Se decidió un modelo de agrupamiento jerárquico aglomerativo combinado con índices internos de validación *Silhoutte* y *Dabies-Bouldin* con el objetivo de determinar el agrupamiento óptimo de los datos. En el capítulo aparecen 4 aparecen los detalles de la etapa.

El último paso fue estudiar las diversas maneras de poder escoger para cada bloque el candidato adecuado. Para ello se utiliza la información del fondo de escena confirmado y se intenta que los nuevos bloques se parezca a dicho fondo mediante el uso de la continuidad espacial. En este estudio se analizaron diferentes técnicas entre las cuales están las que finalmente se usaron para la realización del algoritmo: continuidad de borde, análisis de objetos de primer plano en los bloques a través de diferencias de color, estructura de histogramas orientado y DCT's. Todo ello se describe en el capítulo 5.

Por último y tal como se describe en el capítulo 6 se probó el nuevo algoritmo con distintas secuencias de vídeo y se comparó con otros existentes en el estado del arte.

7.2. Conclusiones

Después de la evaluación y comparación de nuestro algoritmo con los existentes en el estado del arte se observó que no se obtuvo un algoritmo óptimo en todas las situaciones. cada algoritmo funciona en unas condiciones determinadas y para resolver algún problema concreto de los algoritmos de inicialización.

En las siguientes secciones, discutimos los resultados obtenidos en base a los diferentes problemas que se pueden encontrar:

- **Cantidad de primer plano** : una de las ventajas de nuestro algoritmo es que soporta mucho más de un 50% de primer plano en la secuencia utilizada para inicializar el fondo. Esto es gracias a la sucesión de etapas y a la no-discriminación de candidatos durante la etapa temporal. Muchos algoritmos no llegan a soportar el cincuenta y el nuestro sólo es comparable a [8].

- **Camuflaje**: nuestro algoritmo es capaz de soportar el camuflaje en la escena gracias a la utilización de varias herramientas de nivel espacial. Si sólo nos basásemos en el estudio de objetos de primer plano a través de diferencias de color, como en [8], tendríamos este problema como se muestra en los capítulos 5 y 6.

- **Primer plano estático**: otra de las características importantes es que es capaz de soportar objetos de primer que plano permanecen gran parte del tiempo parados en el fondo de escena. De hecho en un bloque sólo es necesario que el fondo aparezca durante al menos dos imágenes para que pueda ser reconstruido. El éxito reside una vez más en utilizar la etapa temporal única y exclusivamente para formar candidatos y no para decidir cuáles son fondo y cuáles no.

- **Sombras y reflejos**: nuestro algoritmo es capaz de reconstruir (mejor que el estado del arte relacionado) el fondo en secuencias donde hay sombras y reflejos producidos por

objetos del primer plano. Siempre que la semilla (fondo inicial asumido) sea la adecuada podremos evitar las sombras gracias a la continuidad de borde principalmente.

- **Parámetros de configuración**: nuestro algoritmo es el único existente en el estado del arte que no necesita ningún tipo de parámetro de configuración. Los únicos parámetros que recibe es considerar que, en las diferencias de color entre candidatos, un *blob* sólo se forma con dos o más píxeles contiguos y que las diferencias de color a lo largo del contorno se evalúan en ventanas de 3x3 píxeles alrededor de cada uno. No existe ningún umbral de decisión que deba ser adaptado manualmente a las características de la secuencia de vídeo.

- **Ruido cámara**: el algoritmo que hemos implementado no es capaz de trabajar correctamente si la secuencias es grabada presenta un alto ruido introducido por la cámara. Esto es debido al f*rame difference adaptativo* ya que depende de la técnica para calcular dicho umbral, en nuestro caso, del método de Kapur.

- **Fondos multimodales**: nuestra aproximación no es capaz de soportar fondos multimodales debido a que considera fondos de escena estáticos. Por lo cual, se recomienda utilizar el algoritmo desarrollado en interiores.

7.3. Trabajo Futuro

Los resultados obtenidos en las secciones previas muestran que el algoritmo funciona mejor en determinados ambientes y que es capaz resolver algunos problemas de inicialización mejor que la literatura existente. Esta sección se proponen algunas líneas de investigación que podrían mejorar los resultados obtenidos.

- **Ruido cámara**. la primera línea sería investigar en cómo obtener el ruido de la cámara en una secuencia de vídeo. El problema viene porque a priori no conocemos ninguna región estática donde poder calcular dicho ruido y no podemos diferenciar lo que es ruido de lo que es movimiento de primer plano.

- **Bloques de análisis solapados.** El uso de bloques de análisis solapados espacialmente (como en [8]) permitiría poder utilizar técnicas más robustas para las etapas de análisis temporal y espacial. No obstante, el coste computacional se incrementaría considerablemente.

- **Etapa temporal**. Teniendo bien definido el ruido de la cámara podríamos encontrar otras formas de agrupación temporal más precisas y que a posteriori podrían funcionar con mayor exactitud.

- **Etapa espacial**. Esta línea es la fundamental en este algoritmo. Hay que investigar sobre las distintas características existentes para llegar a un punto donde podamos afirmar que una de las ventanas en este caso es fondo seguro sin tener ningún tipo de incertidumbre
- **Fondo inicial asumido.** Actualmente, el primer bloque de fondo confirmado (semilla) es aquél cuyo agrupamiento temporal tiene más elementos (es decir, mayor número de bloques que dan lugar al candidato confirmado). En esta línea, se deberían investigar distintas técnicas de selección de semilla debido a que se observó una gran dependencia de los resultados respecto a esta asunción inicial.
- **Fondo multimodal**. Nuestro algoritmo no trabaja a nivel de píxel por lo que no es capaz de soportar fondos multimodales. Un hito sería conseguir mezclar los beneficios de los algoritmos nivel de píxel (fondos multimodales) con los beneficios de los algoritmos a nivel de región (mejor detección del fondo). Con ello podríamos encontrar una aproximación que soportase fondos poblados multimodales.
- **Condiciones iniciales.** La última línea de investigación iría en la dirección de desarrollar el algoritmo sin ninguna de las asunciones, es decir, cámaras en movimiento y ambientes no controlados.

Bibliography

[1] J. SanMiguel, J. Bescós, J. Martínez, and A. García, "Diva: A distributed video analysis framework applied to video-surveillance systems," in *Proc. of IEEE Int. Workshop on Image Analysis for Multimedia Interactive Services*, 7-9 May 2008, pp. 207–210. 1

[2] M. Chang, M. Sezna, and A. Telkalp, "An algorithm for simultaneous motion estimation and scene segmentation," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 5, pp. 221–224, April 1994. 1

[3] R. Lienhart, C. Kuhmunch, and W. Effelsberg, "On the detection and recognition of television commercials," in *nternational Conference on Multimedia Computing and Systems*, 1997, pp. 509–516. 1

[4] O. sukmarg and k. Rao, "Fast object detection and segmentation in mpeg compressed domain," *Proc. IEEE TENCON*, vol. 3, pp. 364–368, 2000. 1

[5] A. Dix, *Human-computer interaction*. Prentice hall, 2004. 1

[6] D. Zhang and G. Lu, "Segmentation of moving objects in image sequence: A review," *Circuits, systems, and signal processing*, vol. 20, no. 2, pp. 143–183, 2001. 1, 2

[7] S. Herrero and J. Bescós, "Background subtraction techniques: Systematic evaluation and comparative analysis," in *Proc. of the Advanced Concepts for Intelligent Vision Systems*, Bordeaux (France), 28-2 Sept.-Oct. 2009, pp. 33–42. 1, 4, 7, 8

[8] A. Colombari and A. Fusiello, "Patch-based background initialization in heavily cluttered video," *IEEE Trans. on Image Processing*, vol. 19, no. 4, pp. 926–933, April 2010. 2, 8, 11, 12, 18, 20, 21, 22, 23, 25, 33, 44, 45, 53, 62, 72, 77, 79, 81, 82, 86, 87

[9] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Int. Conf. on Computer Vision*, Redmond, 1999. 2, 23

[10] J. C. SanMiguel and J. M. Martínez, "Robust unattended and stolen object detection by fusing simple algorithms," in *Proc. of IEEE Int. Conf. on Advanced Video and Signal based Surveillance*, Santa Fe (USA), 1-3 Sept. 2008, pp. 18–25. 2, 9

[11] C. R. M. I. T. M. Prati, A., "Analysis and detection of shadows in video streams:a comparative evaluation." in *Conf. Computer Vision Pattern Recognition*, 2001. 2, 100

[12] V. Reddy, C. Sanderson, and B. Lovell, "An efficient and robust sequential algorithm for background estimation in video surveillance," in *Int. Conf. on Image Processing*, Australia, 2009, pp. 1109–1112. 2, 11, 12, 18, 20, 21, 24, 25, 33, 45, 77, 78, 79

[13] A. Colmenarejo, "Segmentación de secuencias de vídeo basada en el modelado de fondo mediante capas," Ph.D. dissertation, Universitad Autonoma de Madrid, Spain, July 2011. 2, 4

[14] K. Rank, M. Lendl, and R. Unbehauen, "Estimation of image noise variance," *IEE Proc. Image Signal Process*, vol. 146, no. 2, 1999. 2

[15] Y. Shireen, M. Khaled, and H. Sumaya, "Moving object detection in spatial domain using background removal techniques - state-of-art," *Recent Patents on Computer Science*, vol. 1, no. 1, pp. 32–54, October 2008. 2, 4, 8, 9, 14

[16] Y. Benezeth, P. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, "Comparative study of background subtraction algorithms," *Journal of Electronic Imaging*, vol. 19, no. 3, p. 11, July 2010. 2, 4

[17] X. X. and T. Huang, "A loopy belief propagation approach for robust background estimation," in *Int. Conf. on Computer Vision and Pattern Recognition*, USA, 2008. 2, 11, 13, 24

[18] S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," *Visual Comm Image Proce*, pp. 881–892, 2004. 3, 7

[19] M. M. V. Cristani, M. Bicego, "Multilevel background initialization using hidden markov models," in *IWVS*, California (USA), 7 november 2003. 8, 12, 16

[20] F. Tiburzi, M. Escudero, J. Bescos, and J. Martinez, "A ground truth for motion-based video-object segmentation," in *Int. Conf. on Image Processing*, oct. 2008, pp. 17 –20. 8

[21] R. RJ, A. S, A.-K. O, and R. B., "Image change detection algorithms: A systematic survey," *IEEE Trans on Image Processing*, vol. 14(3), pp. 294–307, March 2005. 8

[22] D. Wang, T. Feng, H. Shum, and S. Ma, "A novel probability model for background maintenance and subtraction," in *The 15th Int. Conf. on Vision Interface*, 2002, pp. 109–117. 8

[23] M. Piccardi, "Background subtraction techniques: a review," in *The ARC Centre of Excellence for Autonomous Systems*, Sydney, 15 April 2004. 8

[24] T. Kentaro, K. John, B. Barry, and M. Brian, "Practice of background maintenance," in *Conf. on Com Vision (ICCV)*, 1999, pp. 255–261. 9

[25] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Realtimetracking of the human body," *IEEE Trans Pat Anal Mach Intel*, vol. 19, no. 7, pp. 700–785–224, 1997. 9

[26] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *6th European Conf. on CompVision*, 2000, pp. 751–767. 9, 16

[27] I. Haritaoglu, D. Harwood, and L. Davis, "Who? when? where? what? a real time system for detecting and tracking people," in *Third Face and Gesture Recog Conf*, April 1998, pp. 222–227. 9

[28] C. Stauffer, W. Grimson, B. Brumitt, and B. Meyers, "Adaptive background mixture models for real-time tracking," in *Comput Society Conf. on CompVision and Patt Recog (CVPR)*, February 1999, pp. 246–252. 9

[29] A. Cavallaro, O. Steiger, and T. Ebrahimi, "Semantic video analysis for adaptive content delivery and automatic description," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 15, no. 10, pp. 1200–1209, October 2005. 9

[30] H. Wang and D. Suter, "A novel robust statistical method for background initialization and visual surveillance," in *Asian Conference on Computer Vision (ACCV)*, Victoria (australia), 2006. 10, 12, 14, 17, 21, 23, 72, 77, 78, 79

[31] Z. Hou and C. Han, "A background reconstruction algorithm based on pixel intensity classification in remote video surveillance system," in *Proc. of FUSION*, China, 2004. 10, 14

[32] M. Granados, H. seindel, and H. Lensch, "Background estimation from non-time sequence images," in *Graphics Interface Conference 2008*, Ontario (Canada), 28-30 May 2008, pp. 33–40. 10

[33] V. Reddy, C. sanderson, A. sanin, and B. Lovell, "Adaptive patch-based background modelling for improved foreground object segmentation and tracking," in *2010 Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance*, Australia, 2010, pp. 172–179. 11, 18, 25

[34] C. Chen and J. aggarwal, "An adaptive background model initialization algorithm with objects moving at different depth," in *Int. Conf. on Image Processing*, USA, 2008, pp. 2664–2667. 11, 14, 76

[35] D. Gutchess, M. Trajkovic, e. Cohen-Solalz, D. Lyonsz, and a. Jainy, "A background model initialization algorithm for video surveillance," in *ICCV*, Michigan, 2001. 11, 13, 14, 15, 21, 76

[36] F. Baltieri, R. Venazzi, and R. Cucchiara, "Fast background initialization with recursive hadamard transform," in *2010 Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance*, Modena (Italia), 2010, pp. 165–171. 11, 18

[37] H. Lin, T. Liu, and J. Chuang, "A probabilistic svm approach for background scene initialization," in *Int. Conf. on Image Processing 2002*, Taiwan, 2002, pp. 893–896. 11

[38] D. Farin, W. Effelsberg, and P. de with, "Robust background estimation for complex video sequences," in *Int. Conf. on Image Processing*, Germany, 2006. 12, 16

[39] P. Daeyong and B. Hyeran, "Object-wise multilayer background ordering for public area surveillance," in *Advanced Video and Signal Based Surveillance*, seul (Korea), 2009, pp. 484–489. 12

[40] S. Cohen, "Background estimation as a labeling problem," in *IEEE International Conference on Computer Vision (ICCV)*, San Jose, 2005. 12

[41] A. Bevilacqlia, "A novel background initialization method in visual surveillance," in *MVA2002 IAPR Workshop on Machine Vision Applications*, Nara (Japan), 13 november 2002, pp. 614–617. 13, 17

[42] B. Gloyer, H. Aghajan, K. Siu, and T. Kailath, "Video-based freeway monitoring system using recursive vehicle tracking," in *IST-SPIE Symposium on Electronic Imaging*, 1995. 14

[43] w. Long and Y. Yang, "Stationary background generation: An alternative to the difference of two images," in *Patt Recog*, 1990, pp. 1351–1359. 15

[44] K. M. Shireen Y.E and S. H.A, "Moving object detection in spatial domain using background removal techniques," in *Recent Patents on Computer Science*, Cairo (Egypt), 28 August 2007, pp. 32–54. 16

[45] V. Digalakis and L. Neumeyer, "Speaker adaptation using combined transformation and bayesian methods," *IEEE Trans. Speech Audio Process*, vol. 3, pp. 357–366, 1995. 16

[46] Q. Huo and C. Lee, "Online adaptive learning of the correlated continuous density hidden markov models for speech recog," *IEEE Trans. Speech Audio Processing*, vol. 6, pp. 386–397, 1999. 16

[47] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005. 17

[48] k. Kim, "Algorithms and evaluation for object detection and tracking in comp vision," Ph.D. dissertation, University of Maryland, E.U., July 2005. 17

[49] C. Lambert, s. Harrington, C. Harvey, and A. Glodjo, "fficient on-line nonparametric kernel density estimation," in *Aorithmica*, 1999, pp. 37–57. 17

[50] M. Fischler and R. Rolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981. 17

[51] A. Engin and A. Derya, "An expert system based on fuzzy entropy for automatic threshold selection in image processing," *Expert Systems with Applications: An International Journal*, vol. 36, no. 2, pp. 3077–3085, March 2009. 30

[52] E. Rosin, P. ioannidis, "Evaluation of global image thresholding for change detection," in *Pattern Recognition Letters*, Cardiff, January 2003, pp. 2345–2346. 30, 97, 98

[53] G. karypis, E. Han, and V. Kumar, "Chameleon: Hierarchical clustering using dynamic modeling," in *Cover Feature*, Minnesota, August 1999. 31, 101

[54] K. Wang, B. Wang, and L. Peng, "Cvap: Validation for cluster analyses," *Data Science Journal*, vol. 8, no. 1, pp. 88–93, Mayo 2009. 31, 41, 103

[55] J. C. SanMiguel and J. M. Martínez, "On the evaluation of background subtraction algorithms without ground-truth," in *Proc. of the IEEE Int. Conf. on Advanced Video and Signal based Surveillance*, Boston (USA), 1-3 Sept. 2010, pp. 180–187. 33, 52, 61

[56] U. Nunes, "An application to pedestrian detection," in *Conference On Intelligent Transportation Systems*, USA, 2009, pp. 432–437. 33, 55

[57] J. E. Jackson, *A User's Guide to Principal Components*. John Wiley and Sons, 1991. 38

[58] I. T. Jolliffe, *Principal Component Analysis*. Springer, 2002. 38

[59] A. G. Luis F. Lago, Manuel Sanchez-Montanes, "Análisis de componentes pca e ica," in *Métodos Avanzados en Aprendizaje Artificial*, 2010. 39

[60] M. Haldiki, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of Intelligent Information Systems*, vol. 17, no. 2, pp. 107–145, 2001. 41, 103

[61] S. Dudoit and J. Fridlyand, "A prediction-based resampling method for estimating the number of clusters in a dataset," in *Genome Biology*, USA, June 2002. 41, 103

[62] N. Bolshakova and F. Azuaje, "Cluster validation techniques for genome expression data," in *Signal Processing*, UK, 2003. 41, 42, 103, 104

[63] ——, "Estimating the number of clusters in dna microarray data," UK, 2006. 41, 103

[64] M. Haldiki, Y. Batistakis, and M. Vazirgiannis, "Clustering validity checking methods: Part ii," *Sigmod record*, vol. 31, no. 3, pp. 19–27, Septembber 2002. 41, 103

[65] A. Strehl, "Relationship-based clustering and cluster ensembles for high-dimensional data mining," Ph.D. dissertation, The University of Texas at Austin, USA, May 2002. 41, 103, 106

[66] R. Sharan, A. Maron-Katz, and R. Shamir, "a system for clustering and visualizing gene expression data," *Bioinformatics*, vol. 19, no. 14, pp. 1787–1799, January 2003. 41, 103

[67] s. Bryan, "Thresholding," in *SH and B*, January 2000. 97

[68] C. Su and A. Amer, "A real-time adaptive thresholding for video change detection," in *Int. Conf. on Image Processing*, Canada, 2006, pp. 157–160. 97, 98

[69] P. Rosin, "Unimodal thresholding," *Pattern Recognition*, vol. 34, no. 11, pp. 2083–2096, March 2001. 98

[70] C. Olson, "Parallel algorithms for hierarchical clustering," in *Elsevier Science B.V*, NY (USA), january 1995. 102

[71] D. Frossyniotis and C. Pateritsas, "A multi-clustering fusion scheme for data partitioning," Greece, August 2005. 102

[72] M. Haldiki, Y. Batistakis, and M. Vazirgiannis, "Clustering algorithms and validity measures." 102

[73] C. Fraley, "Algorithms for model-based gaussian hierarchical clustering," in *Technical Report*, USA, October 1996. 102

Apéndice A

Métodos de obtención de umbral adaptativo.

Para obtener el umbral τ a aplicar en la ténica *frame difference* de manera adaptativa dependiendo de la secuencia de entrada, proponemos utilizar algoritmos clásicos de umbralización automática. Entre los existentes, se han seleccionado inicialmente los algoritmos de Otsu, de Kapur y de Rosin.

- El algoritmo de Kapur aparece descrito en el capítulo 3.
- El algoritmo Otsu asume dos clases distintas correspondientes a fondo de escena y primer plano y se busca que el umbral obtenido minimice la varianza intra-clase y maximice la varianza inter-clase [67]. Es ampliamente utilizado, sub-umbraliza y no es adecuado para la detección de cambios ya que puede dar una medida poco objetiva y ruido de salida [52] [68]. Para la obtención del umbral primero calculan el histograma y la probabilidad de cada intensidad de pixel, posteriormente se inicializan las clases a cero (w_1 y w_2) y se buscan dividir las dos clases con el umbral de Otsu variando entre $T = 1$ y el máximo posible de intensidad de modo que:

$$w_1(T) = \sum_{0}^{T} p(i) \tag{A.1}$$

donde su media viene dada por:

$$\mu_1(T) = \sum_{0}^{T} p(i)x(i) \tag{A.2}$$

donde donde $x(i)$ es el el valor en el centro del ith bin del histograma (se procede de la misma manera para la segunda clase con los valores superiores a T).
Define una suma de varianzas ponderadas de dos clases como:

$$\theta^2(T) = w_1(T)\theta_1^2(T) + w_2(T)\theta_2^2(T) \tag{A.3}$$

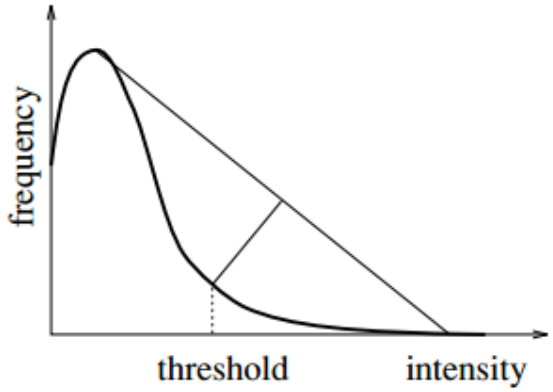


Figure A.1: Ejemplo obtención umbral adaptativo de Rosin [69]

donde T es el umbral de Otsu, w_1 y w_2 son las probabilidades de las clases en las que se divide la imagen separadas por un umbral T, y las θ_i son las varianzas, y la varianza inter-clase como:

$$\theta_b^2(T) = \theta^2 - \theta_w^2(T) = w_1(T)w_2(T)[\mu_1(T) - \mu_1(T)]^2 \tag{A.4}$$

donde μ_i es la media asociada a cada clase.
El umbral T deseado es el maximiza $\theta_b^2(t)$

- En el algoritmo de Rosin suponen que hay una clase dominante que produce un pico en el histograma. La clase secundaria no debe producir un pico o al menos debe ser lo suficientemente pequeño en comparación con el de la clase dominante. Consite en trazar un línea recta desde el pico de la clase dominante hasta el final del histograma (el primer bin vacío). Se selecciona el umbral en el punto en que se maximiza la distancia perpendicular entre la línea y el punto (i, H_i) del mismo modo que en la figura A.1. Sufre de la sombra y el ruido de compresión [52]. Muchos de los algoritmos de umbral tradicionales tienen dificultades con las imágenes que tienen distribuciones de intensidad principalmente unimodal aunque Rosin se muestra superior. [68]

En [52] se hace un análisis de los tres algoritmos en distintas secuencias: Hay tres técnicas para medir los resultados entre un ground-truth y las imágenes resultantes. Están basadas en :

- Verdaderos positivos (TP), número de pixeles de cambio correctamente detectados.
- Falsos positivos (FP), número de pixeles de no cambio detectados como cambio.
- Verdaderos negativos (TN), número de pixeles de no cambio detectados correctamente.
- Falsos negativos (FN), número de pixeles de cambio detectados como no cambio.

		textura uniforme		multi-textura	
	medida	media	mediana	media	mediana
Kapur	PCC	0.9992	0.9992	0.9983	0.9983
	Jaccard	0.3557	0.3432	0.1543	0.1274
	Yule	0.5865	0.5645	0.2292	0.1664
Otsu	PCC	0.9022	0.9040	0.9559	0.9540
	Jaccard	0.0106	0.0063	0.0150	0.0112
	Yule	0.0105	0.0063	0.0149	0.0111
Rosin	PCC	0.9891	0.9892	0.9814	0.9809
	Jaccard	0.0592	0.0523	0.0282	0.0246
	Yule	0.0604	0.0530	0.0283	0.0246

Table A.1: puntuaciones media y mediana

algoritmo	Kapur	Otsu	Rosin
Media	0.9977	0.8264	0.9893
Mediana	0.9984	0.8015	0.9894

Table A.2: PCC

Las técnicas son las siguientes

- Porcentaje correcto de clasificación

$PCC = \frac{TP+TN}{TP+FP+TN+FN}$

- Coeficiente de Jaccard

$\frac{TP}{TP+FP+FN}$

- Coeficiente de Yule

$|\left(\frac{TP}{TP+FP}\right) + \left(\frac{TN}{TN+FN}\right) - 1|$

A continuación en la tabla A.1 se muestran los resultados de los algoritmos para siete secuencias. Los resultados muestran el algoritmo de Kapur como el mejor cuantitativamente.

En la tabla A.2 se presentan los resultados para secuencias donde no hay cambios. Se observa la mayor puntuación en Kapur.

En la tabla A.3 obtenemos los mejores resultados para la técnica de Kapur.

algoritmo	Kapur	Otsu	Rosin
Media	24.42	319.53	42.09
Mediana	22.20	327.30	30.40

Table A.3: Error absoluto

El estudio se ha realizado con el dataset disponible [11]. Consta de 112 imágenes para secuencias indoor las cuales contienen movimientos de personas y disponen del ground-truth con una segmentación manual.

Apéndice B

Agrupamiento jerárquico

El objetivo del agrupamiento jerárquico es agrupar un conjunto de datos de manera que se maximice la similitud dentro de los *clusters* y reduzca al mínimo la similitud entre dos *clusters* diferentes.
La mayoría de algoritmos de agrupamiento encuentran clusters que se ajustan a algún modelo estático (por ejemplo, el número de clusters supuesto). Aunque eficaz en algunos casos estos algoritmos pueden agrupar los datos de forma incorrecta si no se selecciona apropiadamente los parámetros del modelo estático [53].
Un método popular de llevar a cabo la agrupación es la agrupación jerárquica (agglomerative hierarchical) [53]. Este método comienza con un conjunto de puntos distintos cada uno de los cuales se considera un cluster aparte. Los dos clusters que están más cerca de acuerdo con algunas métricas se fusionan de manera iterativa. Esto se repite hasta que todos los puntos pertenecen a un grupo jerárquicamente construido. La estructura final es simplemente un árbol que muestra cómo se fusionan los clusters en cada paso. La figura B.1 muestra el proceso descrito dónde el conjunto de puntos inicial son las distintas letras las cuales se van agrupando de acuerdo a métricas que miden la distancia inter-cluster e intra-cluster.

Las métricas para determinar la distancia entre pares de clusters se pueden dividir en dos clases generales, graph metrics y geometric metrics.
Graph metrics. Consideran la posibilidad de un gráfico completamente conectado, donde los vértices son los puntos que se quieren agrupar y los bordes tienen una función de coste que es la distancia euclídea entre los puntos. Determinan distancias ínter-cluster de acuerdo a las funciones de coste de las aristas entre los puntos en los dos grupos.

- Single link: La distancia entre los dos clusters está dada por el mínimo coste de borde entre puntos de los dos clusters.
- Average link: La distancia entre los dos clusters es el promedio de todos los costes de borde

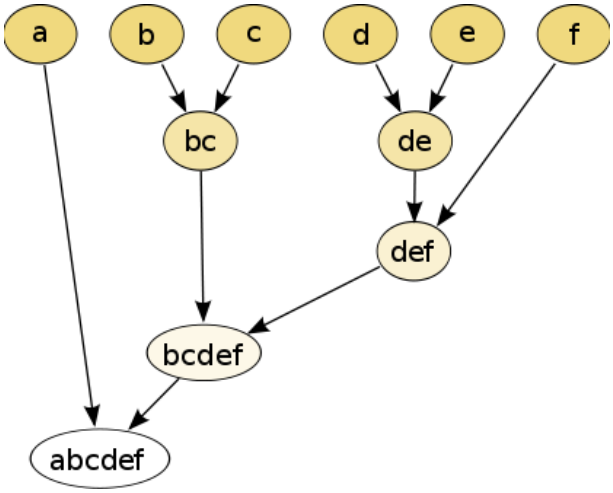


Figure B.1: Agrupamiento jerárquico de muestras

entre puntos de los dos clusters.

- Complete link: La distancia entre los dos clusters está dada por el coste máximo del borde entre puntos de los dos clusters.

Geometric metrics. Estas métricas definen un centro del cluster para cada cluster y usan estos centros de cluster para determinar las distancias entre los clusters:

- Centroid: El centro del cluster es el centro de gravedad de los puntos en el cluster. Se utiliza la distancia euclídea entre los centros de cluster.
- Median: El centro del cluster es el promedio de los centros de los dos clusters que lo forman. Se utiliza la distancia euclídea entre los centros de cluster.
- Minimum variance: El centro del cluster es el centro de gravedad de los puntos en el cluster. La distancia entre los dos clusters es la cantidad de aumento en la suma de los cuadrados de las distancias de cada punto al centro de su grupo. [70]

Diferentes algoritmos de agrupamiento con diferentes propiedades tienden a dar muchas soluciones diferentes, y no existe un método de agrupamiento óptimo para todos los conjuntos de datos posible [71] [72] [73].
Una vez se obtienen los resultados de la agrupación el siguiente paso importante es la evaluación del mismo para determinar la estructura del *cluster* para el conjunto de datos de entrada , por lo general el número de *clusters* (NC).

Apéndice C

Índices de evaluación del agrupamiento jerárquico.

Con el objetivo de validar un agrupamiento de datos para un número determinado de *clusters* se utilizan índices de índices de validez [54]. Hay dos tipos de índices de validez: índices externos e índices internos.

Un índice externo es una medida de acuerdo entre dos particiones donde la primera partición es a priori conocida (también llamado *ground truth*), y la segunda son los resultados del procedimiento de una segunda agrupación. Los índices externos sirven para evaluar los resultados de un algoritmo de agrupamiento basado en una estructura de *clusters* conocidas de un conjunto de datos (o etiquetas de grupo). Los índices externos más conocidos incluyen *Rand, adjusted Rand, Jaccard, and Fowlkes Mallows* (FM) [60] [61].

Los índices internos se utilizan para medir la bondad de una estructura de la agrupación, sin información externa. Los índices internos evalúan los resultados con las cantidades y características propias del conjunto de datos. El número de *clusters* óptimo (NC) suele estar determinado sobre la base de un índice de validez interna (nuestro caso). Los índices internos más populares son: *Silhouette, Davies-Bouldin, Calinski- Harabasz, Dunn, Hubert-Levin (C-index), Krzanowski-Lai and Hartigan* [62] [63] [61] ; *the Root-mean-square standard deviation (RMSSTD), R-squared, Semi-partial R-squared (SPR) and Distance between two clusters* (CD) [60] [64]; *the weighted inter-intra index* [65]; *and the Homogeneit y and Separation indices* [66].

Al no disponer de un *ground truth* para evaluar los agrupamientos realizados para cualquier secuencia de entrada se ha realizado un estudio de los índices internos para utilizar los de mejor funcionamiento de la aproximación:

Silhouette index: Un índice compuesto que refleja la solidez y la separación de grupos, un índice promedio mayor *Silhouette* indica una mejor calidad de los resultados de la agrupación, por lo que el óptimo NC es el que da el mayor valor promedio del *Silhouette* [62] .

Para un *cluster* determinado U_j $j \in [1, ..., c]$ este método asigna a cada muestra de U_j una medida de calidad $S_i(i = 1, ..., m)$ conocida como *Silhouette width*. El *Silhouette width* es un indicador de confianza de la muestra *ith* en el *cluster* U_j. *The Silhouette width* para la muestra *ith* en el *cluster* U_j se define como:

$$s(i) = \frac{b(i) - a(i)}{max\left\{a(i), b(i)\right\}'} \tag{C.1}$$

Donde $a(i)$ es la distancia media entre la muestra *ith* y todas las muestras incluidas en U_j, y $b(i)$ es la distancia mínima entre la muestra ith y todas las muestras agrupadas en U_k $k \in [1, ..., c]$ $k \neq j$. De esta fórmula se deduce insertar formula $-1 \leq s(i) \leq 1$.

Para un *cluster* determinado, $U_j(j = 1, ..., c)$, es posible calcular el índice *Silhouette* U_j , el cual caracteriza las propiedades de heterogeneidad y aislamiento de este *cluster*:

$$S_j = \frac{1}{m} \sum_{i=1}^{m} s(i) \tag{C.2}$$

m es el numero de muestras en U_j. El óptimo NC es el que da el mayor valor promedio del *Silhouette*.

Davies-Bouldin index: Una medida de la similitud media entre cada grupo y su más similares, los pequeños valores corresponden a grupos que son compactos y tienen centros que están muy lejos unos de otros, por lo tanto, su valor mínimo determina el óptimo NC [62]. El índice de *Davies-Bouldin* tiene como objetivo identificar los conjuntos de grupos que son compactos y bien separados. El índice de *Davies-Bouldin*, DB, se define como:

$$DB = \frac{1}{c} \sum_{i=1}^{c} max_{i \neq j} \left\{ \frac{\Delta(U_i) + \Delta(U_j)}{\delta(U_i, U_j)} \right\} \tag{C.3}$$

donde U_i representa el *cluster* i de una partición, U_j representa el *cluster* j, $\delta(Ui, U_j)$ denota la distancia entre *clusters* U_i y U_j, $\triangle(U_k)$ representan la distancia *intra-cluster* del *cluster* U_k y c es el número de *clusters* de la partición de U. Pequeños valores de DB corresponden a grupos que son compactos, y cuyos centros están muy lejos unos de otros. Por lo tanto, la configuración que minimiza DB se toma como el número óptimo de *clusters*.

Si se define SS (*sum of squares*):

$$SS = \sum_{i=1}^{N} (X_i - \overline{X})^2 \tag{C.4}$$

SS_w referido a la suma de cuadrados dentro de un *cluster*. SS_b referido a la suma de cuadrados entre *clusters* . SS_t referido a la suma total de cuadrados, del conjunto de datos:

Hartigan. Para cada número de *clusters* $k \geqslant 1$ se definde el índice como:

$$hart_k = (\frac{trSS_{w_k}}{trSS_{w_{k+1}}} - 1)(n - k - 1) \tag{C.5}$$

El número estimado de *clusters* es el más pequeño $k \geqslant 1$ tal que $hart_k \leqslant 10$ donde tr denota la traza de la matriz, esto es, la suma de la diagonal de entrada. .

Calinski harabasz. Medida entre aislamiento del *cluster* y la coherencia dentro del grupo, su valor máximo determina el óptimo NC. Para cada numero de *clusters* $k \geq 2$ define el índice como:

$$ch_k = \frac{tr(SS_{b_k})/k-1}{tr(SS_{w_k})/n-k} \tag{C.6}$$

El número estimado de *clusters* es $argmax_{k\geqslant 2} ch_k$.

Dunn. Una medida que maximiza las distancias inter-*cluster* mientras que reduce al mínimo las distancias intra-*cluster*, grandes valores indican la presencia de *clusters* compactos y bien separados, por lo que la NC que maximiza el índice se toma como el óptimo NC. Este índice identifica conjuntos de *clusters* que son compactos y bien separados. Para cualquier partición U , el índice de Dunn validación, D, es definido como:

$$D(U) = min_{1\leq i\leq c}\left\{min_{1\leq j\leq c}\left\{\frac{\delta(X_i, X_j)}{max_{1\leq k\leq c}\{\triangle(K_k)\}}\right\}\right\} \tag{C.7}$$

donde $\delta(X_i, X_j)$ denota la distancia entre *clusters* X_i y X_j (distancia inter-*cluster*) , $\triangle(X_k)$ representa la distancia intra-*cluster* del *cluster* X_k y c es el número de *clusters* de la partición de$U$. El principal objetivo de esta medida es maximizar las distancias inter-*cluster* minimizando las distancias intra-*cluster*. Así, los valores grandes de D corresponden a grupos buenos. Por lo tanto, el número de *clusters* que maximiza D se toma como el número óptimo de clusters, c.

Krzanowski-Lai. Para cada numero de *clusters* k≥2 define el índice como:

$$diff_k = (k-1)^{\frac{2}{p}} trSS_{w_{k-1}} - k^{\frac{2}{p}} trSS_{w_k} \tag{C.8}$$

$$Kl_k = \frac{|diff_k|}{|diff_{k+1}|} \tag{C.9}$$

El número de *clusters* estimados es $argmax_{k\geqslant 2} Kl_k$.

Homogeneity and Separation: Definimos la homogeneidad de un *cluster* como la similitud media intra-*cluster* y la separación del agrupamiento como la similitud media entre diferentes *clusters*. Los dos tipos de medidas, homogeneidad *intra-cluster y separación inter-cluster*, son intrínsecamente contradictorios, una mejora en una normalmente corresponden a un empeoramiento de la otra. Un método consiste en fijar el número de clusters y buscar una solución con la máxima homogeneidad. Otro enfoque es el de presentar una curva de homogeneidad frente a la separación en un amplio rango de parámetros para el algoritmo de agrupamiento utilizado.

C index: Para cualquier partición el índice C, se define como:

$$C = \frac{S - S_{min}}{S_{max} - S_{min}} \tag{C.10}$$

donde S, S_{min}, S_{max} se calculan de la siguiente manera. Supongamos que p es el número de todos los pares de muestras donde ambas muestras se encuentran en el mismo *cluster*. Entonces S es la suma de las distancias entre las muestras en esos pares p. Sea P un número de todos los posibles pares de muestras en el conjunto de datos. Ordenando los pares de P por las distancias se pueden seleccionar los pares p con distancias más pequeñas y con más grandes. La suma de

las distancias más pequeñas es igual a S_{min}, mientras que la suma de las mayores es igual a S_{max}. De esta fórmula se deduce que C será pequeño si los pares de muestras tienen pequeña distancia en el mismo cluster. Por lo tanto, los pequeños valores de C corresponden a *clusters* buenos. El número de *clusters* que minimizan C-índice se toma como el número óptimo de clusters, NC.
Rmsstd, R squared (RS), Semi partial Rsquared (SPR), Distance between two clusters (CD): RMSSTD de un nuevo esquema de agrupación definido como un nivel un agrupamiento jerárquico es la raíz cuadrada de la varianza de todas las variables. Este índice mide la homogeneidad de los *clusters* formados en cada paso del algoritmo jerárquico. El objetivo es formar *clusters* homogéneos con el RMSSTD tan pequeño como sea posible. En caso de tener un valor más alto que en el paso anterior nos da la indicación de que el nuevo *cluster* es malo. *SPR* de un *cluster* nuevo es la diferencia entre SS_w del nuevo *cluster* y las suma de todos los SS_w de los *clusters* juntados para obtener el nuevo dividido por el SS_t. Este índice mide la pérdida de homogeneidad después de la fusión de 2 *clusters*. Si el índice es 0 la fusión es perfectamente homogénea si no son heterogéneos. *RS* de un nuevo *cluster* es la proporción de SS_b sobre SS_t, es una medida de diferencia entre *clusters*. $SS_t = SS_b - SS_w$. RS es considerado una medida de disimilitud entre *clusters*. $RS = 0$ indica que no existe diferencia entre *clusters*. $RS = 1$ indica una diferencia significante. El índice *CD* mide la distancia entre 2 *clusters* que se fusionaron en un paso del agrupamiento. La distancia depende de los representativos seleccionados para el agrupamiento jerárquico que representamos.
Weighted inter intra. El objetivo es maximizar la similitud intra-cluster y minimizar la similitud inter- cluster de manera definida en [65].
En la litetatura analizada los índices que mejores resultados obtienen para dataset de imágenes son *Silhouette, Dabies Bouldin, Rmsstd, R squared (RS), Semi partial Rsquared (SPR), Distance between two clusters (CD),* por ello en el capítulo 6 se realizan las pruebas para estos índices.

Apéndice D

Presupuesto

1. **Ejecucion Material**

- Compra de ordenador personal (Software incluido) ..... 2.000 €

  - Alquiler de impresora láser durante 6 meses ..... 260 €
  - Material de oficina ..... 150 €
  - Total de ejecución material ..... 2.400 €

1. **Gastos generales**

  - 16% sobre Ejecucion Material ..... 352 €

2. **Beneficio Industrial**

  - 6% sobre Ejecucion Material ..... 132 €

3. **Honorarios Proyecto**

  - 1800 horas a 15 €/ hora ..... 27.000 €

4. **Material fungible**

  - Gastos de impresión ..... 280 €
  - Encuadernación ..... 200 €

5. **Subtotal del presupuesto**

  - Subtotal Presupuesto ..... 32.774 €

6. **I.V.A. aplicable**

- 18 % Subtotal Presupuesto ..... 5.899,3 €

7. **Total presupuesto**

- Total Presupuesto ..... 38.673,3 €

Madrid, septiembre 2012
El Ingeniero Jefe de Proyecto

Fdo.: Alberto Muñoz García
Ingeniero Superior de Telecomunicación

Apéndice E

Pliego de condiciones

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un "sitema de generación de fondo de escena en secuencias de video-seguridad" para ser visto en pantallas de baja resolución. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.
Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:
Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará

obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.