DPTO. TEORÍA DE LA SEÑAL Y COMUNICACIONES
INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN: SISTEMAS DE TELECOMUNICACIONES

PROYECTO FIN DE CARRERA

TÉCNICAS DE PRE-ANÁLISIS Y PRE-PROCESADO APLICADAS A LA CODIFICACIÓN PERCEPTUAL DE VÍDEO

Autor: Helen Mariela Medina Chanca
Tutor: Manuel de Frutos López

02/11/2011

Agradecimientos

A mis padres que me han apoyado de manera incondicional en mis estudios. Gracias, por todo el cariño y apoyo que me habéis demostrado, no solamente en esta etapa tan importante para mí sino también a lo largo de toda mi vida.

A mi tutor, Manuel de Frutos, por la dedicación y la ayuda ofrecida en la realización de este proyecto.

A alguien que es muy importante para mí, quisiera agradecerle todo su cariño y compresión. Gracias Sergio.

Y un agradecimiento especial a mi tíos y, en general, a toda mi familia.

Resumen

La codificación de vídeo persigue el objetivo de comprimir a tasas bajas una secuencia de vídeo con una reducción mínima de la calidad percibida. Para ello, la codificación debe asignar mayores recursos a regiones en las que la distorsión sea más perceptible, ya sea debido a su textura o por pertenecer a regiones de interés.

Este Proyecto Fin de Carrera propone dos técnicas para ello: pre-análisis y pre-procesado, ambas aplicadas a la codificación de vídeo basada en consideraciones perceptuales. Estas técnicas tienen como objetivo mejorar la calidad de la secuencia codificada sin que ello implique un aumento de la tasa de bits.

La técnica de pre-análisis consiste en el análisis de la estructura de la textura de las secuencias que en combinación con un mapa de prioridad de la región de interés, ajusta el parámetro de cuantificación del codificador.

La técnica de pre-procesado realiza un filtrado selectivo de la secuencia antes de ser codificada. Donde la intensidad del filtrado (filtro bilateral, que tiene en cuenta la textura de la secuencia) de una región dependerá de si esta región pertenece o no a la región de interés.

Finalmente, se incluye un conjunto de pruebas experimentales y subjetivas que se llevaron a cabo para evaluar estas técnicas.

Palabras claves: codificación perceptual, pre-procesado, pre-anásilis, parámetro de cuantificación, filtro bilateral, región de interés.

Abstract

Video encoding aims to compress at low rates a video sequence with a minimal reduction of the perceived quality. To this end, compression should allocate more resources to regions that are most sensitive to distortion, either because of their texture or belonging to the region of interest.

This final project proposes some pre-processing and pre-analysis techniques applied to video coding based on perceptual considerations. These techniques aim to improve the encoded video quality without increasing the bit rate.

The pre-analysis technique analyzes the texture structure of the sequences and in combination with a priority map of the region of interest, adjusts the quantification parameter of the encoder.

The pre-processing technique performs selective filtering before the encoding of the sequence. The intensity of the bilateral filter (a kind of filter that takes into account the texture of the sequence) of a region depends on whether or not this region belongs to the region of interest.

Finally, the present project includes a set of subjective tests that were carried out to evaluate these techniques.

Keywords: perceptual coding, pre-processing, parameter quantification, bilateral filter, region of interest.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción y objetivos

1.1. Introducción

En el ámbito de la codificación de vídeo se llevan a cabo técnicas de compresión que pueden o no introducir pérdidas de calidad y, por tanto, degradar la calidad de la imagen reconstruida. Estas técnicas de compresión se basan en dos principios:

- Ciertas regiones de la secuencia de vídeo pueden pasar inadvertidas para el sistema visual humano.
- Una secuencia de vídeo contiene información redundante que puede ser eliminada.

Para minimizar la percepción de esta degradación, la codificación tiene como reto realizar una asignación apropiada de la tasa de bits con el objetivo de proporcionar una calidad visual óptima. Para ello, se debe realizar un reparto inteligente de los recursos de bits, de manera que se asigne menor cantidad a zonas de la imagen menos sensibles a la distorsión y mayor cantidad a aquellas regiones más sensibles.

Varias técnicas se han desarrollado, basadas en algoritmos de codificación perceptual, que intentan diferenciar información que puede ser o no detectada por el observador, y así eliminar información redundante como información menos significativa perceptualmente. Esto generalmente se basa en las propiedades de enmascaramiento del sistema visual humano, que no es capaz de detectar distorsión en determinadas texturas y, por otra parte, tiende a concentrarse en ciertos detalles de una escena, mientras presta menos atención a otras regiones de la imagen. Por tanto, estas regiones pueden clasificarse en función de su textura o de si trata de una región de interés para el observador.

Regiones de interés. Son aquellas regiones de la imagen que llaman la atención del observador. Este método puede acarrear algunos inconvenientes si la región de interés difiere para algunos observadores.

Textura. Zonas del plano que por su estructura pueden ser más o menos sensibles a la distorsión. Para ello, es imprescindible el estudio de las características del sistema visual humano, ya que determinan la base para delimitar las zonas en las que la degradación de la imagen es o no perceptible.

Basándose en lo anterior, para conseguir aumentar la calidad subjetiva de los vídeos codificados a una tasa de bits igual o menor, la línea de investigación se divide principalmente en dos áreas:

- Ajuste del parámetro de cuantificación del codificador. Este método consiste en la asignación de valores de QP[1] bajos para regiones de interés y valores de QP altos para aquellas regiones que pueden albergar distorsión.
- Técnicas de pre-procesado. Eliminan información redundante que generalmente se encuentra en las altas frecuencias.

1.2. Objetivos

El objetivo principal de nuestro sistema consiste en determinar la región del plano sobre la que el sujeto focaliza su atención o aquellas en que la distorsión será más perceptible (HVS) para poder asignar más recursos a éstas en detrimento de la calidad de áreas que carezcan de interés subjetivo. Esto se puede hacer por dos vías: realizar un pre-análisis de la secuencia de vídeo para ajustar los parámetros de cuantificación del codificador o aplicar una técnica de pre-procesado previo al codificador.

La técnica de pre-análisis constituye el primer objetivo a conseguir. Esta técnica consiste en ajustar el parámetro de cuantificación en función de un mapa de enmascaramiento final, que se obtiene a partir de la combinación del mapa de enmascaramiento, según la región de interés, y del mapa de enmascaramiento basado en texturas. Para obtener éste último, se lleva a cabo el análisis de la imagen y su correspondiente clasificación, en función de su textura.

Debido a los posibles inconvenientes que puede acarrear el ajuste de la QP, se estudian otras técnicas de pre-procesado basadas en el filtrado de la imagen, y que tienen el mismo objetivo que el sistema descrito anteriormente.

Una vez implementadas estas técnicas es necesario evaluar la viabilidad de las mismas, integrando estos algoritmos en un codificador H.264, y, por medio de pruebas subjetivas de codificación, comprobar la calidad visual del resultado final.

1.3. Estructura de la memoria

A continuación se describen las partes que conforman la memoria de este proyecto y se realiza una breve descripción del contenido detallado en cada una de ellas.

El documento consta de 6 capítulos.

En el capítulo 1 se realiza una introducción general del contenido de la memoria y de los objetivos que se pretenden alcanzar.

El capítulo 2 incluye una presentación de las técnicas empleadas en el ámbito de la codificación perceptual y de los conceptos necesarios para el buen entendimiento del proyecto.

En el capítulo 3 se propone una clasificación de texturas alternativa a la clasificación desarrollada por el Grupo Multimedia del Departamento de Teoría

[1] Parámetro de cuantificación.

de la Señal y Comunicaciones de la Universidad Carlos III de Madrid. Asimismo, se realizará una comparación subjetiva de las mismas.

En el capítulo 4 se presentan algunas técnicas de pre-procesado que ofrecen mejores resultados subjetivos que la técnica de pre-análisis de la secuencia de vídeo. Estas técnicas consisten en el filtrado de la imagen teniendo en cuenta la textura y la región de interés. Se llevan a cabo pruebas experimentales para realizar una comparación de estas técnicas de pre-procesado con la técnica de filtrado desarrollada por el Departamento de Multimedia.

El capítulo 5 aporta las conclusiones obtenidas del análisis de resultados correspondientes a las pruebas y se expone algunas ideas que conforman las posibles líneas futuras de este trabajo.

El capítulo 6 detalla el presupuesto estimado de la realización de este proyecto, desglosando los costes asociados al material y a los honorarios.

Por último, se incluye un anexo donde se presentan algunas pruebas subjetivas de las primeras versiones de filtrado implementadas en el capítulo 4. También se muestra la organización de las secuencias incluidas del DVD adjunto.

Capítulo 2

Estado del arte

2.1. Principios básicos de codificación de vídeo

2.1.1. Introducción

La codificación de vídeo es el proceso de comprimir una secuencia de vídeo digital en un pequeño número de bits. La compresión se hace necesaria para el posterior almacenamiento y transmisión de una secuencia de vídeo, ya que el vídeo digital sin comprimir ocupa una enorme cantidad de memoria. La compresión involucra dos sistemas complementarios. Por un lado está el codificador ("encoder"), que convierte los datos originales a una forma comprimida que puede almacenarse o transmitirse. Y por otro lado está el decodificador ("decoder"), que se encarga de convertir los datos comprimidos a su representación original. Este par de sistemas es conocido como CODEC, del que se distinguen dos tipos:

Compresión sin pérdidas ("lossless"): donde los datos a la salida del decodificador son una copia perfecta de los datos originales. La compresión sin pérdidas de un vídeo o imagen sólo alcanza niveles de compresión moderados.

Compresión con pérdidas ("lossy"): los datos a la salida del decodificador no son idénticos a los datos originales, lo que implica que se pueda alcanzar mayor compresión pero a expensas de una pérdida de calidad de la imagen. La compresión con pérdidas está basada en el principio de eliminar información redundante, es decir, elementos de la imagen o secuencias de vídeo que pueden ser desechados sin afectar significativamente a la percepción de la calidad por parte del observador.

Es importante comentar que la mayoría de los codificadores consiguen alcanzar un nivel de compresión mayor a través de la explotación de la redundancia temporal y de la redundancia espacial. La redundancia temporal consiste en la alta correlación o similitud entre planos adyacentes de una secuencia de vídeo (capturados de manera consecutiva), mientras que la redundancia espacial consiste en una alta correlación entre píxeles vecinos de un mismo plano.

2.1.2. CODEC de vídeo

Un CODEC de vídeo codifica una secuencia a una forma comprimida y la decodifica para producir una copia o una aproximación de la secuencia original.

El modelo híbrido de CODEC [1] está compuesto principalmente por tres bloques funcionales: modelo temporal, modelo espacial y un codificador de entropía (véase figura 2.1).

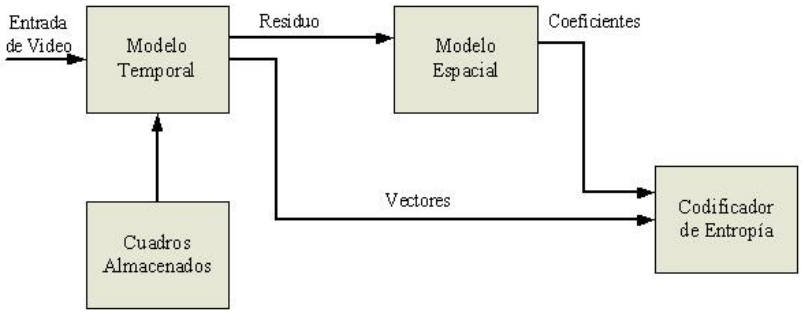


Figura 2.1: Diagrama de bloques de un codificador de vídeo

2.1.2.1. Modelo temporal

Tiene como objetivo eliminar la redundancia temporal de una secuencia de vídeo explotando las similitudes entre planos vecinos en el tiempo. La salida de este proceso es un plano residual y un mapa de vectores de movimiento, que son codificados y enviados al decodificador. De modo que el decodificador usa el vector de movimiento recibido para obtener el bloque de predicción y decodificar el bloque residual. Finalmente se suman ambos bloques, reconstruyendo así la versión original del bloque.

Este proceso está constituido principalmente de dos fases:

Estimación de movimiento (ME): es el proceso mediante el cual se calculan los vectores de movimiento y se elige la región candidata o de referencia (aquella que cumple un cierto criterio). Puesto que la estimación de movimiento requiere de una mayor complejidad en el codificador, se ha venido desarrollando diferentes algoritmos para llevar a cabo el cálculo de los vectores de movimiento, siendo el "block matching" el método de comparación más utilizado en codificación de vídeo.

Compensación de movimiento (MC): Teniendo en cuenta lo anterior, este proceso genera un bloque residual a partir de la diferencia del bloque original y de la región candidata o de referencia. Una vez realizado este proceso, este bloque residual es codificado y transmitido junto con el vector de movimiento que describe la posición de la región de mejor coincidencia respecto a la posición del macrobloque (MB)[1] actual.

Señalar que hay algunas variaciones en el proceso de estimación y compensación de movimiento. La región de referencia puede pertenecer a un plano futuro, previo o a una combinación de ambos. En algunas ocasiones hay cambios muy

[1] Región del plano compuesta por un determinado número de píxeles.

significativos entre el de referencia y el actual (cambio de escena), en los que se considera más eficiente codificar el macrobloque sin compensación de movimiento. Por ello, el codificador puede elegir entre dos opciones: usar la codificación "intraframe" (sin compensación de movimiento) o la codificación "interframe" (con compensación de movimiento) para cada macrobloque.

2.1.2.2. Modelo espacial

Parte del residuo calculado en el modelado temporal y tiene como función eliminar la redundancia espacial. Para conseguirlo, se utiliza la codificación predictiva, DPCM ("Differential Pulse Code Modulation"), que calcula predicciones de muestras a partir de muestras vecinas ya codificadas. También existen otras, la más simple está formada a partir del píxel anterior, y una predicción más exacta puede obtenerse del promedio ponderado de los píxeles vecinos.

Los vectores de movimiento de bloques vecinos son similares, por ello se predice el movimiento en base a vectores previamente calculados. La información que se transmite es el vector de movimiento diferencial, MVD ("Motion Vector Difference"), resultado de la diferencia del vector real con el predicho.

Los modelos espaciales tienen tres componentes principales: transformación, cuantificación y reordenación. El primero desvincula y compacta los datos, el segundo reduce la precisión de los datos transformados, y el tercero organiza los datos de tal forma que los valores importantes queden ordenados.

Transformación Los datos espaciales de la imagen son difíciles de comprimir. Las muestras vecinas están altamente correlacionadas y la energía tiende a estar distribuida en toda la imagen, haciendo difícil descartar datos o reducir la precisión de los datos sin afectar adversamente la calidad de la imagen. Por tanto, trabajando en el dominio de la transformada se pretende reducir la correlación espacial dejando un pequeño número de coeficientes visualmente importantes y un gran número de coeficientes insignificantes que puedan ser descartados. Aplicando este proceso se pretende desvincular las muestras con el fin de conseguir una tasa de compresión mayor.

Muchas transformadas se han propuesto para la compresión de imagen y vídeo. Las más importantes se engloban en dos categorías: transformadas basadas en bloques y basadas en imágenes. Las transformadas basadas en imágenes operan sobre la imagen completa o una porción de ella, destacando la DWT ("Discrete Wavelet Transform"). Las transformadas basadas en bloques incluyen la KLT *(*"Karhunen-Loeve Transform*")* y la DCT ("Discrete Cosine Tranform").

La transformada DCT opera a nivel de bloque (NxN) y es reversible (existe IDCT, "Inverse Discrete Cosine Transform"). Proporciona una matriz de coeficientes que son los pesos de una base de NxN funciones. De todos los coeficientes obtenidos, los más próximos al coeficiente (0,0) suelen disponer de una energía significativa, a diferencia del resto, que por norma general posee una energía despreciable.

Los patrones base para la DCT de tamaño 4x4 y 8x8 están representados en la figura 2.2.

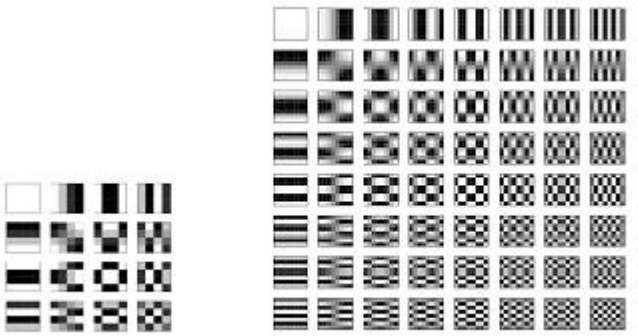


Figura 2.2: Patrones base para una DCT de 4x4 (izquierda) y de 8x8 (derecha)

Cuantificación La cuantificación es el proceso en el que se introducen las pérdidas de información en el sistema, y que permiten una mayor capacidad de compresión. Consiste en reducir el número de niveles de representación de las muestras. La cuantificación más sencilla es la cuantificación uniforme, cuya función de transferencia es la siguiente:

$$FQ = round\left(\frac{X}{QP}\right) \tag{2.1}$$

$$Y = FQ \cdot QP \tag{2.2}$$

donde:

- FQ : es el valor cuantificado.
- X : valor de entrada.
- Y : se corresponde con el valor reconstruido.
- QP : es la distancia entre dos valores cuantificados consecutivos, el escalón de cuantificación. Es el encargado de determinar el grado de compresión alcanzado.

Otro tipo de cuantificación es la cuantificación vectorial, cuyo nombre procede del tratamiento que se realiza de los datos en forma de vector. En este caso, no se cuantifican las muestras individualmente, sino en bloques de muestras o vectores.

Ordenación Antes de realizar una codificación de entropía es necesario reordenar y representar de forma eficiente los coeficientes. A continuación se describe este proceso para la DCT:

Después de realizar la cuantificación, los coeficientes de la DCT para cada bloque son reordenados con el fin de agrupar los coeficientes significativos (no nulos). El camino de reordenación más óptimo depende de la distribución de los coeficientes no-cero. Para un plano típico, un orden de escaneo adecuado es en zig-zag. Este método consiste en ordenar los coeficientes empezando por el coeficiente DC y continuando por los coeficientes restantes en el orden en que

se muestra en la figura 2.3. Los coeficientes se ordenan realizando un escaneo en zig-zag del macrobloque, que varía según se trate de un vídeo progresivo o uno entrelazado.

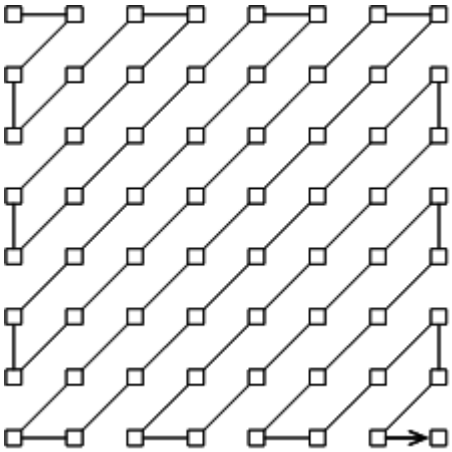


Figura 2.3: Escaneo en zig-zag

2.1.2.3. Codificador entrópico

Una vez realizado el modelo temporal y el modelo espacial, se obtiene un conjunto de datos que presentan algunos patrones estadísticos y que constituyen todo el flujo de información que se va a transmitir o almacenar. El codificador entrópico tiene como objetivo reducir las redundancias estadísticas presentes en esos datos para lograr una mayor compresión del resultado final.

Existen dos tipos de códigos utilizados en el estándar H.264, denominados CAVLC ("Context-Based Adaptive Variable Length Coding") y CABAC ("Context-based Adaptive Binary Arithmetic Coding"), con las siguientes características:

CAVLC El algoritmo de codificación Huffman se propuso como una forma sencilla y óptima de mapear cada símbolo de un alfabeto con un código ("codeword"). De esta forma, para comprimir cada símbolo de la cadena, simplemente debemos usar el código que se ha calculado mediante Huffman. Para conseguir que esta asignación sea eficiente, los símbolos se representan con códigos cuya longitud es inversamente proporcional a la probabilidad del símbolo.

El proceso de asignación de códigos se lleva a cabo mediante la construcción de un árbol binario, desde las hojas hacia la raíz, de manera que los nodos hoja son los símbolos del alfabeto. En la construcción del árbol, los nodos menos probables se unen sucesivamente para formar otro nodo de mayor probabilidad, de forma que cada uno de los enlaces añade un bit al código de los símbolos que estamos juntando. Este proceso termina cuando sólo se dispone de un nodo, de forma que éste representa la raíz del árbol.

Un inconveniente del proceso de decodificación Huffman es que es necesario disponer del árbol a partir del que se codifican los datos. Por lo tanto, no es suficiente con almacenar la cadena final, sino que también hay que comunicar al decodificador las probabilidades de la fuente, de forma que el decodificador

sea capaz de reconstruir el árbol (otra alternativa sería transmitir directamente la tabla de "codewords").

CABAC

En primer lugar, se realiza una binarización de los datos (datos binarios), después se selecciona un modelo de contexto, es decir, un modelo de probabilidad para uno o más bins de los símbolos binarizados. Este modelo se puede elegir en función de las estadísticas de los símbolos codificados previamente. Para terminar, el codificador aritmético codifica cada bin según el modelo de probabilidad seleccionado y actualiza la probabilidad del modelo basado en el valor actual codificado.

2.2. Codificación con consideraciones perceptuales

Los algoritmos basados en codificación perceptual intentan discernir qué componentes de una señal son detectadas por los mecanismos de percepción humanos y cuáles no, para eliminar información redundante que perceptualmente es poco significativa.

Las dos ideas principales en codificación de vídeo son: "esconder" la distorsión de codificación por debajo de los umbrales de detección del ser humano y eliminar información perceptualmente irrelevante. Teniendo presente estas dos ideas, en [2] se utiliza un método práctico en la codificación perceptual de vídeo y que está basado en el concepto de JND ("Just Noticeable Distortion"). Pero existen otros métodos basados en el enmascaramiento visual, que tratan de determinar las zonas donde el ojo humano percibirá menos los errores. De esta manera, es necesario realizar un estudio general de las características más comunes del sistema visual humano (HVS), que serán aprovechadas para introducir distorsiones en secuencias de vídeo sin que el observador las perciba.

2.2.1. Texturas

Se han desarrollado varias técnicas que analizan la segmentación y la clasificación de las texturas.

Este apartado se centra en el enmascaramiento basado en texturas atendiendo a la sensibilidad del HVS a percibir distorsión en ciertas regiones de la imagen en función de su estructura.

Primero, es conveniente mencionar algunos artículos que han realizado estudios sobre la clasificación de texturas.

La referencia [4] realiza una clasificación basada en las siguientes propiedades del HVS:

- "Texture masking": debido a que el HVS es menos sensible a los detalles en zonas con mucha textura (grandes y rápidas variaciones de intensidad, sin patrones estructurados) puede introducirse más ruido sin que pueda ser percibido.
- "Intensity contrast masking": las zonas más oscuras o más brillantes pueden albergar más ruido sin que éste sea percibido.

- "Spatial frequency sensitivity": el HVS actúa como un filtro paso banda en términos frecuenciales, luego las zonas con mayor frecuencia espacial serán idóneas para esconder distorsiones.
- "Preservation of objects boundaries": el HVS también es muy sensible a las distorsiones causadas por el desalineamiento de los bordes de objetos rígidos. En consecuencia, se deberá prestar especial cuidado a dichas zonas.

En base a lo anterior, realiza una clasificación de los macrobloques en 6 categorías:

- "Textured": macrobloques sin patrones estructurados y rápidas fluctuaciones de intensidad. Este tipo de áreas pueden ser codificadas con gran cantidad de ruido sin que éste llegue a ser percibido por el sistema visual humano.
- "Dark contrast": zonas de la imagen cuya intensidad es mucho más baja que la intensidad de las áreas de su entorno.
- "Smooth": áreas con intensidad relativamente constante.
- "Edge": indica que hay un borde en ese macrobloque. Este tipo de macrobloque perceptualmente muestra más distorsión si la línea del borde a través del macrobloque no se conserva.
- "Detailed": zonas con detalles finos que presentan coeficientes dominantes en la DCT.
- "Normal": se trata de aquellos macrobloques que no se encuentran en ninguna de las categorías anteriores.

Teniendo en cuenta la clasificación anterior, algunas investigaciones como [5] han desarrollado un método basado en la DCT, cuyos coeficientes funcionan como indicadores de nivel de textura de un macrobloque. El diagrama de bloques del método se muestra en la figura 2.4:

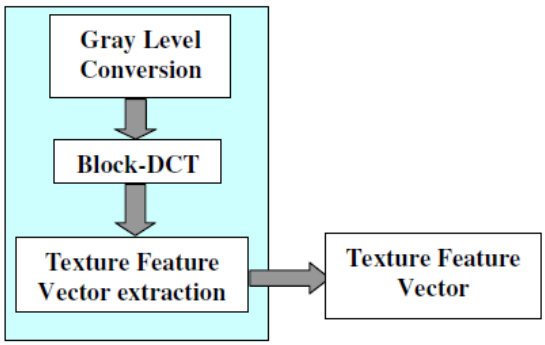


Figura 2.4: Diagrama de bloques del método de extracción de características de textura

Como se puede apreciar en la figura, se convierte una imagen RGB en una imagen en niveles de gris. Posteriormente, se realiza la transformación DCT basada en bloques de la imagen que previamente ha sido dividida en bloques de tamaño NxN. Finalmente, para la extracción de características de textura se utilizan los coeficientes en el dominio transformado como vectores de características en la que cada bloque contiene un coeficiente DC y coeficientes AC.

Este método se basa en las propiedades del HVS y dado que una de sus propiedades es ser menos sensible a percibir errores en los coeficientes frecuenciales altos que en las frecuencias bajas, el método tiene como objetivo extraer un conjunto de características de 9 componentes del vector. Donde el primer componente es el coeficiente DC que representa el promedio de la energía o intensidad del bloque, y los otros 8 coeficientes AC representan diferentes patrones de variación de la imagen e información direccional de la textura. Por esta razón, al desplazarnos en vertical y horizontal por la matriz de coeficientes, los valores representan frecuencias cada vez mayores. Por ejemplo, los coeficientes de las regiones más altas y los de la región más a la izquierda en el dominio transformado DCT representan información vertical y horizontal de un borde respectivamente, como se muestra en la figura 2.5. De igual manera, al desplazarnos en vertical y horizontal por la matriz de coeficientes, los valores representan frecuencias cada vez mayores.

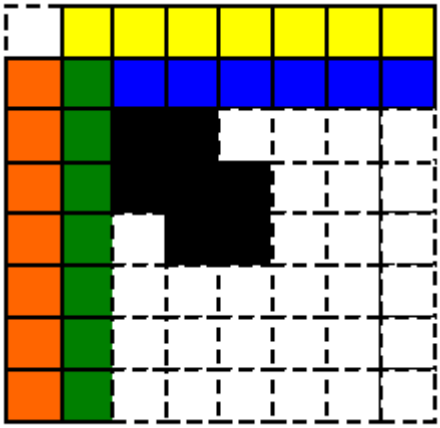


Figura 2.5: Elementos del vector de información direccional

De manera similar, [17] incluye un modelo perceptual basado en propiedades de enmascaramiento en texturas y luminancia pero destinado a un codificador JPEG.

En él se recurre también a la energía AC de los coeficientes DCT de un bloque para medir la actividad de textura. Sin embargo, el algoritmo de clasificación consiste en la comprobación de unas condiciones entre la suma absoluta de valores de unos coeficientes concretos y unos umbrales.

Se distinguen cuatro coeficientes:

- DC: componente continua.
- L: coeficientes de baja frecuencia.

- H: coeficientes de alta frecuencia.
- E: coeficientes de borde.

También se han desarrollado otros métodos para analizar texturas como los descriptores de color, los filtros de Gabor y filtros destinados a la detección de bordes. Estos métodos tienen como objetivo extraer la información relevante con la finalidad de reducir la tasa de bits. La aplicación de estos métodos se puede ver en el artículo [3] donde se desarrolla la técnica "Bit Allocation"[2], también basada en las propiedades del HVS. Este algoritmo propuesto se puede incorporar en procesos de control de tasa en la codificación de vídeo con el objetivo de reducir la tasa de bits sin perjudicar la calidad de la imagen percibida. Para ello, se basa en la existencia de dos tipos de estructuras:

- Textura aleatoria: contiene pequeños bordes con diferentes orientaciones.
- Textura estructurada: se compone de bordes más consistentes y de mayor tamaño.

Esta técnica al igual que las anteriores explota el hecho de que el sistema visual humano es menos sensible a las distorsiones en texturas aleatorias y más sensible en regiones estructuradas.

El modelo se divide en los siguientes pasos:

1. Se recurre a un operador para detectar los bordes de cada plano.
2. Se aplica un promediado de las intensidades de borde y de la distribución de densidad de píxeles del borde de cada macrobloque.
3. Se procede a la extracción de las características locales.

Los dos tipos de detectores de borde estudiados son Sobel [8] y Canny [7]. En primer lugar, el detector de Canny se utiliza con 2 objetivos finales en mente: reducir la tasa de error durante la detección del borde y obtener la localización de los bordes. Sin embargo existe el inconveniente de que los mapas de bordes de Canny funcionan adecuadamente para las regiones "smooth" o estructuradas, pero no para las texturas aleatorias.

El detector de Sobel soluciona el problema anterior pero tiene el inconveniente de que detecta regiones "smooth" como regiones texturadas, por lo que finalmente se opta por utilizar el detector de Canny para realizar una primera clasificación entre regiones texturadas y regiones "smooth" y utilizar el detector de Sobel para diferenciar las regiones aleatorias y las texturas estructuradas.

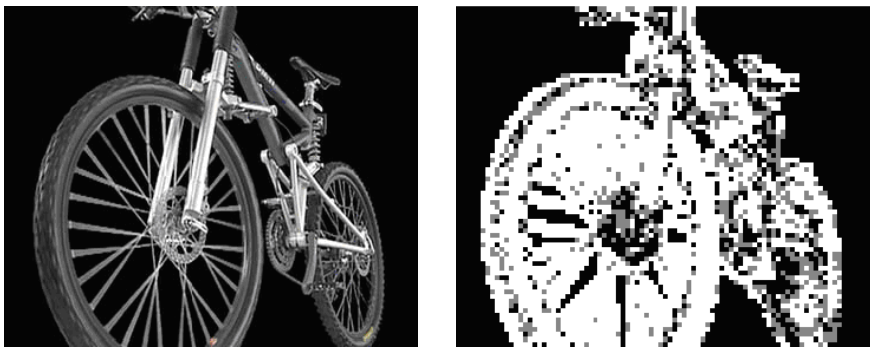
Asimismo, en [16] se desarrolla un algoritmo de "Bit Allocation" basado en consideraciones perceptuales, consistente en medir la cantidad de movimiento entre dos planos y actualizar el multiplicador de Lagrange teniendo en cuenta las consideraciones perceptuales del vídeo. Dicho método también se fundamenta en que el HVS es menos sensible a errores a lo largo de un borde prominente, que impide la percepción de otras variaciones de contraste más bajas en ese mismo bloque. Por lo tanto, se asigna mayor tasa de bits a los bordes dominantes y menos a otros patrones de textura.

Los pasos a seguir del algoritmo son:

[2] Asignación de bits.

1. Cálculo de los operadores de Sobel.
2. Cálculo de la media de los gradientes al cuadrado.
3. Cálculo de la coherencia del gradiente obtenido.
4. Clasificación de regiones como “Edge”, “Texture” o “Background”.
5. Actualización del multiplicador del Lagrange en función de la clasificación anterior.

La figura 2.6 muestra un ejemplo de segmentación de la secuencia “Bike”.



(a) Imagen original (b) Mapa de importancia

Figura 2.6: Mapa de importancia de la secuencia "Bike"

2.2.2. Movimiento y región de interés (ROI)

La codificación de vídeo con consideraciones perceptuales requiere del conocimiento de aquellas regiones de la imagen que son de mayor interés para el espectador respecto a otras que no lo son. Una vez conocidas estas regiones, se les puede dar un mejor tratamiento que aquellas zonas que no son de interés.

Teniendo presente esta idea, muchos investigadores han dirigido su línea de investigación hacia este modelo de atención visual. En el campo de la codificación de vídeo, se han desarrollado técnicas de “eye-tracking” eficaces para el seguimiento de objetos en movimiento, con el objetivo de encontrar zonas vulnerables a la distorsión. En [3], se basan en dos procesos para estudiar este comportamiento de atención visual: “top-down” y “bottom-up”. El primero se refiere a la atención que presta el ser humano de manera intencionada con el fin de ejecutar una tarea. El segundo se produce cuando objetos del entorno llaman la atención del ser humano. Un modelo computacional que también simula este proceso se puede encontrar en [11].

De manera similar, en [12] se asume que los objetos presentes en el primer plano o que están en movimiento llaman la atención del observador. Por ello, se desarrolla un codificador de vídeo que busca el primer plano y el fondo para cada plano y se recurre a un método de compensación de movimiento global-local encargado de analizar el movimiento de la escena, obteniendo una clasificación del movimiento.

Al igual que la codificación perceptual basada en movimiento, existe otra vertiente importante, la codificación perceptual basada en las regiones de interés (ROI). Las ROI son aquellas regiones de la imagen que llaman la atención del espectador. Estas zonas deben ser detectadas para que sean codificadas adecuadamente.

La detección de las regiones de interés consiste generalmente en estimar un mapa de importancia de los elementos de la escena a partir de métodos de segmentación. Este método se destina a escenarios muy concretos donde las regiones de interés están claramente predeterminadas o son fácilmente definibles. Por ejemplo, algunas aplicaciones como videoconferencias donde la región de interés es el rostro de la persona que habla o en lo correspondiente a deportes, donde el centro de atención suele ser los jugadores. Por lo tanto, su uso generalizado no parece recomendable, puesto que enfrentarse a un vídeo de mayor complejidad supondría una técnica de detección complicada.

A continuación se incluyen otros ejemplos acerca de métodos desarrollados sobre detección de la ROI recogidos en la bibliografía. En primer lugar, [10] presenta un estándar de codificación JPEG 2000, cuya codificación perceptual está basada en el escalado de los coeficientes "Wavelet". El principio de escalado consiste en desplazar los bits de los coeficientes hacia planos de bit más significativos. Durante el algoritmo de codificación, estos bits son colocados en el flujo de bits antes que los que no se encuentran en las regiones de interés de la imagen. Por lo que la región de interés es decodificada antes que el resto de la imagen. Este método es implementado en JPEG 2000 [9] como sigue :

1. Calcular la transformada "Wavelet".
2. Si se ha seleccionado una región de interés, se aplica una máscara indicando que estos coeficientes serán requeridos para una reconstrucción sin pérdidas.
3. Cuantificación de los coeficientes "Wavelet".
4. El codificador entrópico codifica los coeficientes resultantes progresivamente, primero codificando los planos de bits más significativos.

La referencia [13] presenta un "Bit Allocation" basado en la ROI con consideraciones perceptuales. Su principal objetivo es encontrar o asegurar una calidad objetivo para el primer plano de la imagen ("foreground"). Esta estrategia se compone de dos pasos:

- Optimización de los parámetros de cuantificación para el "foreground".
- Los parámetros de cuantificación del fondo de la imagen ("background") son obtenidos de modo tal que la calidad se degrade gradualmente a medida que nos alejamos del "foreground". La tasa de caída de la PSNR de fondo se elige de manera que se respete la tasa de bits en la medida de lo posible.

Por otra parte, [14] propone un esquema de compresión de vídeo que tiene como finalidad dividir la imagen en varias regiones y codificarlas en función del tipo al que pertenecen. Este método se basa en la no uniformidad de la distribución de los fotorreceptores en la retina humana, por lo que sólo una pequeña región (fóvea) de 2^{o} - 5^{o} del total del ángulo visual alrededor del punto donde fijamos

la mirada es visualizada con mayor resolución. Por lo tanto, debido a que sólo una porción de la imagen se encuentra dentro del ángulo visual, no es necesario codificar toda la imagen con calidad uniforme, y por ello las demás regiones toleran una distorsión mayor.

Para la selección de dichas regiones se proponen dos métodos:

- "Eye tracking": realiza un seguimiento del ojo del observador, determinando las zonas más destacadas para el usuario, para posteriormente codificarlas con mayor fidelidad, mientras el resto del plano queda degradado. Los resultados obtenidos son eficientes pero está limitado a un único observador.

- Modelo de atención: este método asume la existencia de varios espectadores con distintas localizaciones y distancias respecto a la pantalla, por lo que el mecanismo anteriormente citado carece de validez debido al alto coste computacional que requeriría. Es conveniente utilizar algoritmos que detecten de manera automática las regiones de interés para disminuir el coste computacional. Este mapa de ROI se establece teniendo en cuenta características visuales de bajo nivel tales como:

  - Contraste de color.
  - Contraste de intensidad.
  - 4 orientaciones: 0^{o} , 45^{o} , 90^{o} , 135^{o}.
  - 4 energías de movimiento orientados arriba, abajo, izquierda derecha.

Con estas características se determinará la saliencia de cada zona de la imagen. Un mapa de saliencia es mostrado en la figura 2.7.

Como se puede observar las imágenes de entrada se descomponen en varias características visuales de bajo nivel, y posteriormente son combinadas en un único mapa de saliencia. Este mapa determina la prioridad de codificación, es decir, dará mayor prioridad a los píxeles más salientes y a la zona central de la fóvea.

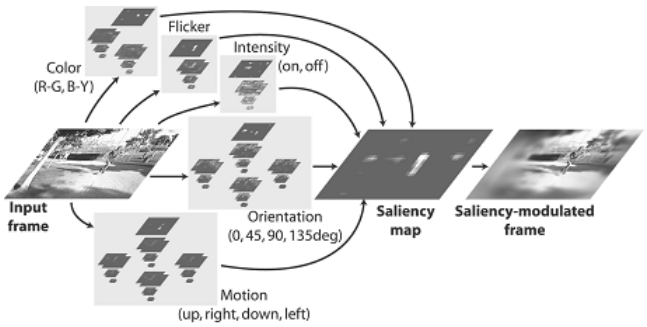


Figura 2.7: Visión general del modelo

Para finalizar con este apartado se describe un algoritmo de control de tasa basado en regiones donde se utiliza un método de segmentación para detectar las ROI, consideradas como aquellas regiones de un plano que presentan movimiento respecto al plano anterior [15], con la finalidad de introducir mayor distorsión en

las zonas que no presentan movimiento. El método empleado para detectar las zonas con movimiento se basa en la diferencia de píxeles entre planos. De manera que si una unidad fundamental presenta una cantidad de píxeles que se han desplazado respecto el plano anterior, se decreta que dicha unidad fundamental pertenece a una región con movimiento. De esta manera se lleva a cabo una clasificación de la imagen en zonas con movimiento, a las que se les asignará mayor cantidad de bits, y zonas estáticas con una menor asignación.

2.3. Detección de bordes

Se define borde de una imagen digital como la transición entre dos regiones de niveles de gris significativamente distintos.

Se han desarrollado una gran variedad de algoritmos que detectan bordes. Dado que se trata de un tema bastante amplio nos centraremos exclusivamente en los filtros detectores de bordes, es decir, métodos basados en el gradiente que detectan bordes en base a las derivadas espaciales de la imagen que se calculan mediante operadores de convolución. Dicha operación consiste en que la máscara se sitúa en cada uno de los píxeles de la imagen y se calcula la suma de los productos de cada uno de los coeficientes de la máscara con el nivel de intensidad de cada píxel perteneciente a la región englobada por dicha máscara (véase la ecuación 2.3), siendo la suma de todos sus coeficientes igual a uno.

$$R = \sum_{i=1}^{N} w_i I_i \tag{2.3}$$

donde:

- w: coeficiente de la máscara.
- I: intensidad del píxel.
- i: posición del píxel.
- N: número de coeficientes de la máscara.

La mayoría de las técnicas para detectar bordes emplean algún tipo de primera y segunda derivada (véase la figura 2.8), aplicado normalmente a un vecindario "pequeño".

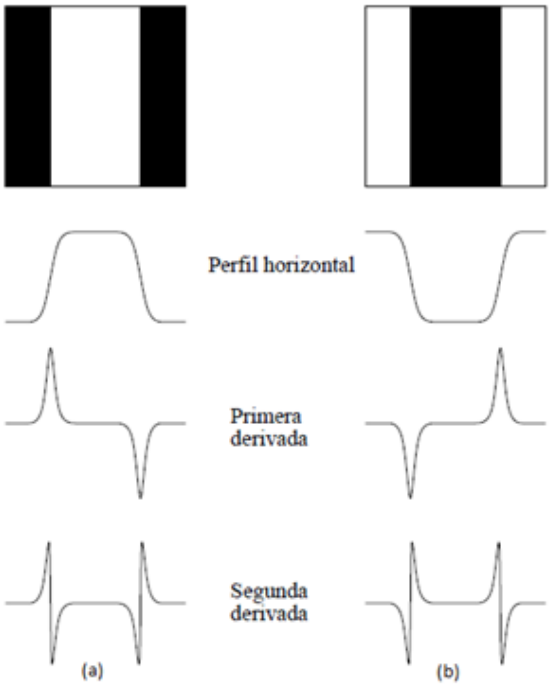


Figura 2.8: Detección de bordes empleando operadores de derivación
(a) Franja de luz sobre un fondo oscuro (b) Franja oscuro sobre un fondo claro

Como puede observarse en la figura, la primera derivada es positiva para un cambio de nivel de gris a uno más claro, negativa en caso contrario y cero en aquellas zonas con nivel de gris uniforme. La segunda derivada presenta valor positivo en la zona oscura de cada borde, valor negativo en la zona clara de cada borde y valor cero en las zonas de nivel de gris constante y en la posición de los bordes.

El valor de la magnitud de la primera derivada nos sirve para detectar la presencia de bordes. La segunda derivada nos indica si el pixel pertenece a la zona clara o a la zona oscura. Dado que la segunda derivada no aporta información relevante para el desarrollo del algoritmo de este proyecto nos centraremos sólo en la primera derivada.

La primera derivada de la imagen nos proporciona las variaciones locales con respecto a la variable x o y, de manera que el valor de la derivada es mayor cuanto más rápidas son estas variaciones, y su vector gradiente siempre apunta en la máxima variación de la imagen (I) en el punto (x,y).

El gradiente de una imagen I en la posición (x,y) viene dado por el vector:

$$\nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} \partial I/\partial x \\ \partial I/\partial y \end{bmatrix} \tag{2.4}$$

El módulo del gradiente de la imagen viene dado por la siguiente ecuación:

$$\nabla I = ||\nabla I|| = \sqrt{I_x^2 + I_y^2} \tag{2.5}$$

donde:

$||\,||$: operador módulo.

En general, para obtener una mejor eficiencia computacional, se suele calcular la amplitud del gradiente como se muestra en la ecuación 2.6, resultado de realizar una aproximación de la ecuación 2.5. En la figura 2.9 se muestra el diagrama de bloques.

$$\nabla I \approx |I_x| + |I_y| \tag{2.6}$$

La dirección del vector gradiente también es otra característica importante a tener en cuenta y se calcula como:

$$\alpha\,(x,y) = \arctan \frac{I_y\,(x,y)}{I_x\,(x,y)} \tag{2.7}$$

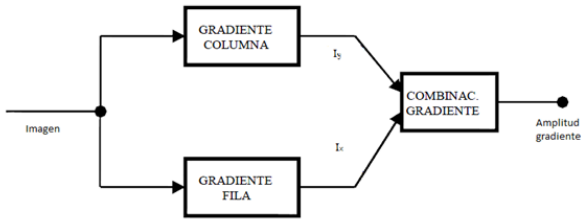


Figura 2.9: Diagrama de bloques de la ecuación 2.6

En definitiva, el objetivo de un algoritmo de detección de bordes es obtener una salida que muestre píxeles con un valor de gradiente alto en zonas de la imagen donde el nivel de intensidad fluctúe bruscamente. Por consiguiente, un borde tendrá valores mayores de gradiente cuanto más rápido se produzca el cambio de intensidad.

2.3.1. Filtros

Según la bibliografía consultada [18] y [19], el método más aceptado en la detección de bordes consiste en filtros de derivada y filtros que combinan suavizado y de tipo de derivada, y así evitar el efecto de amplificación de ruido. Estas técnicas basadas en el gradiente son:

- Operador de Frei-Chen.
- Operador de Sobel.
- Operador de Prewitt.
- Operador de Roberts.

Estas técnicas se basan en dos características importantes del borde:

- Intensidad del borde: es igual a la magnitud del gradiente.
- Dirección del borde: es igual al ángulo del gradiente.

Filtro de Frei-Chen Operador isotrópico que intenta llegar a un equilibrio entre los operadores que detectan bien los bordes verticales y horizontales y los operadores que realizan una buena detección de bordes diagonales. Las máscaras de tamaño 3x3 son:

- Gradiente fila, h_x:

$$\frac{1}{2+\sqrt{2}} \begin{pmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{pmatrix} \tag{2.8}$$

- Gradiente columna, h_y:

$$\frac{1}{2+\sqrt{2}} \begin{pmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{pmatrix} \tag{2.9}$$

Filtro de Sobel Los operadores de gradiente tienen la desventaja de aumentar el ruido en la imagen. Sin embargo, existen otro tipo de operadores, que además de realizar un proceso derivativo, también aplican un suavizado al resultado. De esta manera reducen el efecto de ampliación del ruido y minimizan la aparición de falsos contornos. Un ejemplo de este tipo de filtros es el de Sobel que es rápido y efectivo, y cuya máscara da un mayor peso al píxel central. Tiene la ventaja de ser un buen detector de bordes diagonales, pero no de bordes verticales ni horizontales. Las máscaras de tamaño 3x3 de este filtro son:

- Gradiente fila, h_x:

$$\frac{1}{4} \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \tag{2.10}$$

- Gradiente columna, h_y:

$$\frac{1}{4} \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \tag{2.11}$$

Máscaras (sin normalizar) de tamaño 5x5 son:

- Gradiente fila, h_x:

$$\begin{pmatrix} 1 & 2 & 0 & -2 & -1 \\ 4 & 8 & 0 & -8 & -4 \\ 6 & 12 & 0 & -12 & -6 \\ 4 & 8 & 0 & -8 & -4 \\ 1 & 2 & 0 & -2 & -1 \end{pmatrix} \tag{2.12}$$

- Gradiente columna, h_y:

$$\begin{pmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix} \tag{2.13}$$

Filtro de Prewitt El operador de Prewitt es similar al de Sobel con la diferencia de que no enfatiza el píxel central de la máscara. Proporciona una buena detección de bordes verticales y horizontales.

- Gradiente fila, h_x:

$$\frac{1}{3}\begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix} \tag{2.14}$$

- Gradiente columna, h_y:

$$\frac{1}{3}\begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \tag{2.15}$$

Filtro de Roberts Es el operador de gradiente más simple. Obtiene una buena respuesta ante bordes diagonales, pero tiene el gran inconveniente de que es muy sensible al ruido y por ello tiene pobres cualidades de detección. Sin embargo, este filtro se puede tener en cuenta si se trabaja con vídeos sin ruido.

- Gradiente fila, h_x:

$$\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{2.16}$$

- Gradiente columna, h_y:

$$\begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{2.17}$$

2.3.2. Histogramas de gradientes orientados

En el área de procesamiento de imágenes, la detección de los bordes de una imagen es de suma importancia y utilidad, pues facilita la realización de muchas tareas, entre ellas, el reconocimiento de objetos y la segmentación de regiones. En este proyecto nos centraremos en su aportación al algoritmo Histogramas de Orientación del Gradiente (HOG), que se utiliza para la extracción de características del objeto.

La idea principal del Histograma de Orientación del Gradiente (HOG, en adelante) [37] se fundamenta en que la apariencia local del objeto y su forma

pueden ser caracterizadas por la distribución de los gradientes de intensidad locales o direcciones de los bordes, incluso sin un conocimiento preciso de la posición de los bordes correspondientes.

Este algoritmo evalúa histogramas locales de orientaciones de gradiente en un conjunto de regiones de la imagen. Los pasos seguidos para implementar este algoritmo son:

1. Leer la imagen.
2. Pasar la imagen a niveles de intensidad.
3. Calcular el gradiente mediante máscaras (véase apartado 2.3.1).
4. Dividir la imagen en bloques.
5. Calcular las direcciones de los gradientes para cada bloque y su magnitud.
6. Almacenar de manera acumulativa las magnitudes del gradiente en cada "bin" correspondiente del histograma en función de las direcciones del gradiente.
7. Normalizar el histograma.

2.4. Filtrado de imagen

2.4.1. Introducción

Debido a la existencia de una gran variedad de filtros se han realizado diversas clasificaciones: según a la familia a la que pertenecen, según sus características, etc. En este apartado se realizará una clasificación de los filtros en función de si el filtro se aplica en el dominio espacial o en el dominio frecuencial.

2.4.1.1. Filtros en el dominio espacial

Se denomina filtrado espacial al empleo de máscaras espaciales para el procesamiento de imágenes, habitualmente para técnicas de eliminación de ruido. Sin embargo, hay que tener en cuenta que los filtros definidos en el domino espacial tienen repercusión en el dominio de la frecuencia.

Dentro de esta categoría destacamos dos tipos de filtros: filtros locales y filtros no locales.

Filtros locales Filtros que realizan una transformación local, es decir, el valor filtrado correspondiente a un píxel depende de la vecindad "local" de ese píxel. Esta clasificación se divide a su vez en otras dos: filtros basados en la proximidad espacial y filtros que tienen en cuenta la proximidad en los valores de niveles de gris que se encuentran detallados en [38].

Existen varios filtros que pertenecen a esta familia, entre los que podemos destacar:

- **Filtros de suavizado**

El más simple de todos es el filtro de media, que consiste en una máscara de tamaño NxN con todos sus coeficientes iguales y ponderados para que la suma de todos ellos sea igual a la unidad. Sin embargo, este filtro tiene el inconveniente de realizar una mala atenuación de las frecuencias altas debido a que su respuesta en frecuencia es de tipo sinc. Por consiguiente, no son los más adecuados para eliminar información redundante que se encuentran en las altas frecuencias.

También existe otra familia de filtros dentro de este apartado, los filtros Butterworth. Son filtros paso bajo más completos y con ellos se pueden configurar parámetros como la frecuencia de corte y el orden del filtro que regula la caída más o menos abrupta entre las bandas de paso y eliminada. No obstante, estos filtros acarrean una gran complejidad en el diseño de los mismos. Otro aspecto a tener en cuenta es la transición entre la banda de paso y la banda atenuada, ya que una transición abrupta genera un efecto indeseado denominado "ringing", consistente en la aparición de estelas alrededor de los bordes. Por tanto, debemos elegir caídas suficientemente suaves en las transiciones de la banda de paso-banda atenuada y que reduzcan considerablemente las altas frecuencias. Existe otra familia de filtros más sencillos de diseñar que los filtros Butterworth, los filtros gaussianos que se explicarán detalladamente en el apartado 2.4.2.

- **Filtros anisotrópicos**

Introducen un coeficiente de difusión que varía espacialmente, fomenta y fortalece un suavizado dentro de una misma región (intra), en lugar de entre regiones (inter). De esta manera, el filtro anisotrópico (AF) intenta evitar el efecto del blurring (ocasionado por la gaussiana) convolucionando la imagen solamente en una dirección ortogonal al gradiente por píxel en la imagen, de manera que se preservan los bordes. Sin embargo, la preservación del borde que beneficia a estos filtros conlleva la aparición de nuevas estructuras de ruido, además de realizar un peor filtrado que el filtro gaussiano en zonas homogéneas (aparece una cierta degradación).

- **Variación total**

Su principal objetivo consiste en la preservación de bordes. Su criterio de filtrado se basa en obtener como solución la minimización del problema, tal como indica la ecuación 2.18:

$$TVF_{\lambda}(v) = argminTV\omega(u) + \lambda \int |v(x) - u(x)|^2 dx \tag{2.18}$$

Donde:

- $TV\omega(u) = \int |Du|$: donde Du es una medida de Radon.
- v: imagen sin filtrar.
- u: imagen filtrada.
- λ: multiplicador de Lagrange.

Este filtro se encuentra sujeto a algunas limitaciones que se imponen a través de un parámetro λ que indica el grado de filtrado. Pero al igual que el anterior, introduce distorsión como cambios graduales de contraste en zonas homogéneas y especialmente cerca de los bordes.

Filtros "neighborhood". Realizan el filtrado de un píxel mediante un promediado de niveles de intensidad similares de los píxeles vecinos. Se puede citar el filtro bilateral, que se detallará más en profundidad en el apartado 2.4.3.

Filtro no local El filtro no local (NL-means), descrito en [20], es un filtro basado en el promediado no local de todos los píxeles de la imagen. El valor final del píxel dependerá de la similitud de las regiones que se comparan.

El valor estimado para el píxel i se calcula mediante un promediado de todos los píxeles en la imagen:

$$v(i') = \sum_{j \in T} w(i,j)v(j) \tag{2.19}$$

Donde:

- $v(j)$: intensidad del píxel j.
- $w(i,j)$: peso de similitud ente el pixel i y j.
- T: número de regiones.

Los pesos se obtienen aplicando esta fórmula:

$$w(i,j) = \frac{1}{Z(i)} \exp\left(-\frac{||v(N_j) - v(N_i)||}{h^2}\right) \tag{2.20}$$

Con $Z(i)$ como:

$$Z(i) = \sum_{j} \exp\left(-\frac{||v(N_j) - v(N_i)||}{h^2}\right) \tag{2.21}$$

Donde:

- $v(N_j)$: representa vectores de intensidad que contienen una región de tamaño fijo y centrada en el píxel j.
- h: regula el grado de filtrado.

Este filtro es más robusto que los filtros "neighborhood", ya que a diferencia de los filtros "neighborhood" (compara píxeles), compara regiones de la imagen.

2.4.1.2. Filtros en el dominio frecuencial

Se aplican a la imagen en el dominio de la frecuencia, esto es, después de aplicarles una transformada. Estos filtros actúan independientemente sobre cada coeficiente transformado para finalmente obtener la imagen filtrada después de realizar la transformada inversa de los nuevos coeficientes.

En este sentido se han llevado a cabo muchas investigaciones sobre los filtros de Gabor (ver apartado 2.4.4) y sobre filtros basados en "Wavelet thresholding".

"Wavelet thresholding": este método basa su desarrollo en que a menor tamaño de coeficiente (obtenido realizando la transformada "Wavelet"), mayor probabilidad de que el mismo sea ruido y que las características importantes de la imagen coincidan con los coeficientes altos. Por tanto, teniendo en cuenta lo anterior se aplica un umbral a cada coeficiente de

tal manera que si el coeficiente es más pequeño que dicho umbral, se sustituye por cero y en caso contrario el valor del coeficiente se mantiene o se modifica. La imagen filtrada se obtiene realizando la transformada inversa "Wavelet" de los coeficientes estimados.

2.4.2. Filtros gaussianos

El filtro gaussiano es un filtro paso bajo que convoluciona la imagen con una gaussiana con el objetivo de eliminar o resaltar información. El valor máximo aparece en el píxel central y la anchura de la campana de Gauss está regulada por el parámetro σ. El inconveniente de este filtro es que lleva a cabo un filtrado pobre (destruye importantes características) en zonas que están presentes texturas y bordes (discontinuidades en las intensidades) consiguiendo que toda la imagen se vuelva borrosa y difusa. Por estas razones, este método no es adecuado para la mayoría de aplicaciones de procesamiento de imágenes.

Posteriormente, en [38] se presenta una modificación del anterior filtro consiguiendo uno que detecta y preserva bordes destacados en la imagen, e introduciendo un coeficiente de difusión que varía espacialmente de tal manera que fomenta o fortalece un suavizado dentro de una misma región (intra) en lugar de entre regiones (inter). Sin embargo, la preservación de borde conlleva la creación de nuevas estructuras de ruido o incluso realizan peor filtrado que el filtro gaussiano en zonas homogéneas, lo cual puede ser perjudicial.

2.4.3. Filtrado bilateral

Técnica de filtrado espacial que suaviza zonas homogéneas de una imagen preservando los bordes pronunciados por medio de una combinación lineal de los valores de los píxeles de la imagen.

Este filtro combina los niveles de grises en función de dos criterios: la cercanía geométrica, del mismo modo que se emplea en casi todos los filtrados espaciales anteriormente definidos, y la similitud fotométrica de los píxeles vecinos al píxel central. De esta manera, se tiende a dar más importancia a píxeles con valores cercanos al actual en ambos dominios. Aunque se definirán sus detalles más importantes a continuación, para más información se remite a [21].

Se podría decir que el filtro está compuesto, a su vez, de otros dos filtros:

Filtro de dominio o filtro gaussiano Filtro lineal que utiliza una máscara constante para todo el dominio y que tiene en cuenta la distancia euclídea de los píxeles vecinos al píxel central.

$$d(x, y) = \exp\left(\frac{-(x^2 + y^2)}{2\sigma_d^2}\right) \tag{2.22}$$

Donde:

- σ_d: desviación típica del filtro gaussiano.

Filtro de rango Mientras que el filtro gaussiano se encarga de realizar un proceso de suavizado de la imagen, el filtro de rango es el encargado de destacar las discontinuidades presentes en las intensidades [22]. Para ello se utiliza una

máscara no lineal que mide las variaciones de intensidad de los píxeles de la vecindad con respecto al píxel central.

$$r(a_i) = \exp\left(\frac{-(f(a_i) - f(a_0))^2}{2\sigma_r^2}\right) \tag{2.23}$$

donde:

- σ_r: desviación típica del filtro de rango.
- $f(a_i)$: intensidad del píxel.
- a_0: píxel central.

Por lo que la máscara final que se aplica a la imagen se obtiene a partir de la multiplicación elemento a elemento de la máscara de proximidad de distancia y la máscara de proximidad de intensidades, obtenidas a partir de los anteriores filtros, tal y como se indica a continuación (véase la figura 2.10):

$$M = M_{distancia} \cdot M_{intensidades} \tag{2.24}$$

donde:

- M: máscara bilateral.
- $M_{distancia}$: máscara del filtro gaussiano.
- $M_{intensidades}$: máscara del filtro de rango.

En conclusión, en el filtrado bilateral se lleva a cabo un promedio de los pesos de una vecindad, donde los pesos más altos corresponden a aquellos píxeles que están próximos al píxel central tanto geométricamente como en sus valores de intensidad [23].

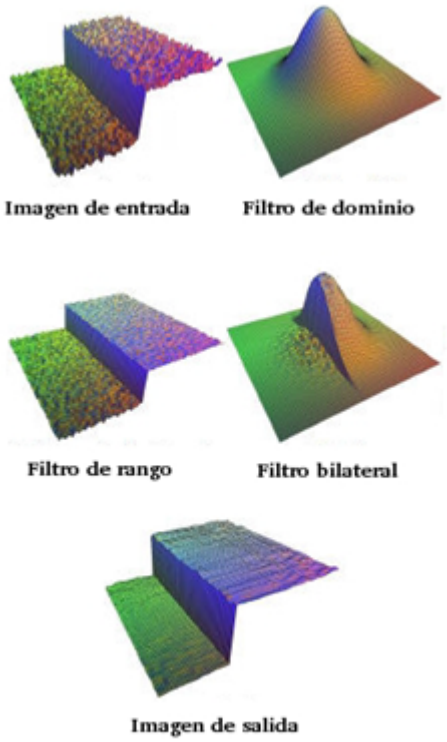


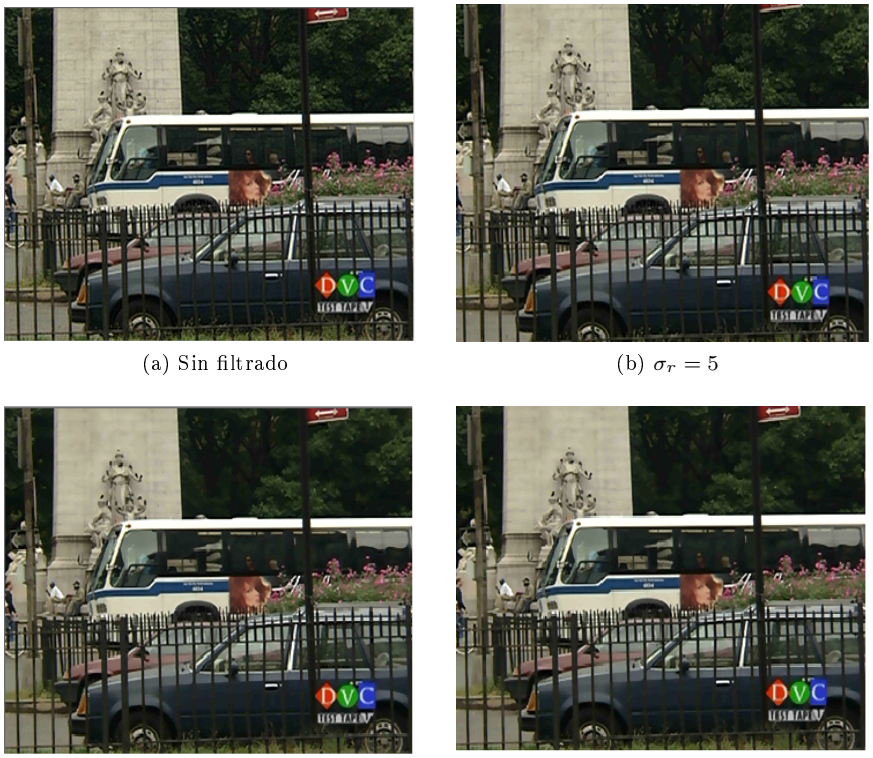
Figura 2.10: Filtro bilateral.

Como podemos observar en la figura 2.10, el filtro bilateral se obtiene a partir de la combinación del filtro de dominio y del filtro de rango aplicado sobre la imagen de entrada. Finalmente, se obtiene la imagen de salida después de filtrar la imagen de entrada con el filtro bilateral obtenido.

2.4.3.1. Filtrado para distintas configuraciones

A continuación se incluyen algunas imágenes filtradas con el método del filtro bilateral.

Bus

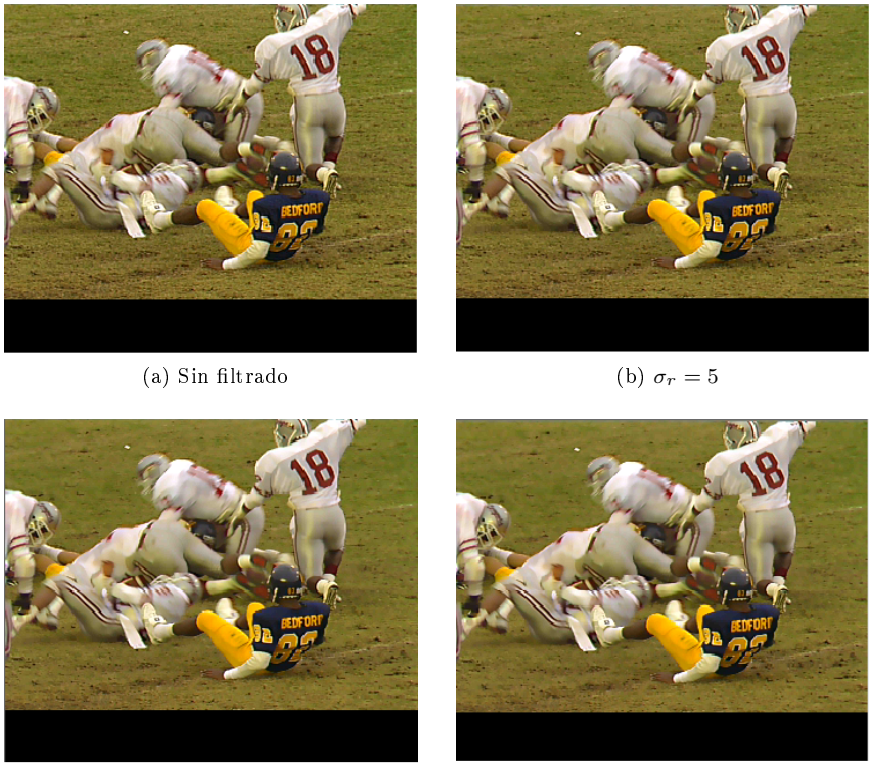


(a) Sin filtrado (b) $\sigma_r = 5$

(c) $\sigma_r = 10$ (d) $\sigma_r = 14$

Figura 2.11: Filtrado bilateral de la secuencia "Bus"

Football



(a) Sin filtrado (b) $\sigma_r = 5$

(c) $\sigma_r = 10$ (d) $\sigma_r = 14$

Figura 2.12: Filtrado bilateral de la secuencia "Football"

2.4.4. Filtros de Gabor

Generalmente, es un buen enfoque llevar a cabo la simplificación en el dominio frecuencial, ya que como se comentó anteriormente, los detalles irrelevantes y texturas complejas se encuentran en la información de alta frecuencia. Sin embargo, la información de bordes y otras características de la imagen se extienden sobre un rango de frecuencias alrededor de una frecuencia crítica, complicando la distinción de estructuras de alto nivel de las estructuras de bajo nivel.

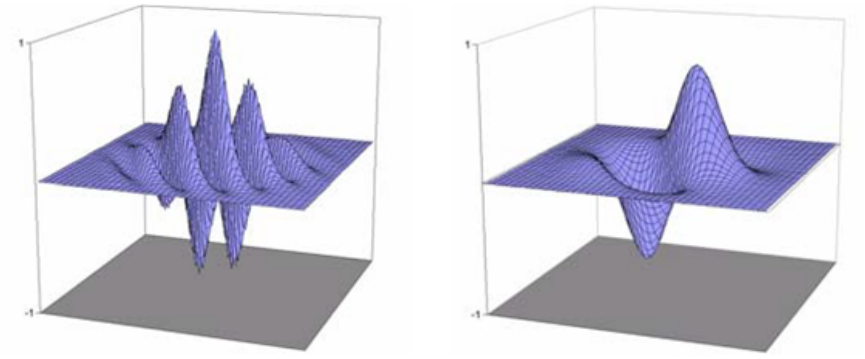
Con el fin de percibir texturas en las escenas, el sistema visual humano utiliza un conjunto de filtros paso-banda que varían en frecuencia y en orientación [24]. De igual manera que las funciones de Gabor se pueden realizar filtros en el dominio de la frecuencia para extraer características de texturas específicas de cada frecuencia y orientación, lo que convierte a los filtros de Gabor en una elección tradicional si se quiere obtener información frecuencial de manera similar a como lo hace el sistema visual humano. Dada la complejidad de estas funciones, profundizaremos más en detalle sobre los filtros de Gabor [25].

Los filtros de Gabor se definen en el espacio como el producto de una gaussiana localizada por una exponencial compleja:

$$g(x, y, \theta, \phi) = \exp\left(-\frac{x^2 + y^2}{\sigma^2}\right) \cdot \exp\left(2\pi\theta i(x\cos\phi + y\sin\phi)\right) \tag{2.25}$$

donde:

- $g(x, y, \theta, \phi)$: función que define un filtro de Gabor centrado en el origen.
- θ: frecuencia espacial del filtro.
- ϕ: orientación del filtro.
- σ: desviación estándar del núcleo gaussiano (depende de la frecuencia espacial θ).



(a) Parte real (b) Parte imaginaria

Figura 2.13: Filtro de Gabor

Los filtros de Gabor se utilizan en el procesamiento digital de imágenes debido a ciertas peculiaridades del mismo, de las que podemos destacar dos aspectos importantes:

- La fase de los coeficientes de Gabor lleva información de la estructura perceptible de la imagen (en caso de que exista un borde, los coeficientes son ortogonales al mismo y tienen fases coherentes), lo cual es importante ya que se ha demostrado que regiones cuyas fases están alineadas coinciden con características perceptibles de las mismas, tales como bordes y líneas [27].
- Como la convolución en el dominio espacial es el producto en el dominio frecuencial, entonces un conjunto de filtros de Gabor trabaja como filtros paso-banda en el dominio frecuencial [26].

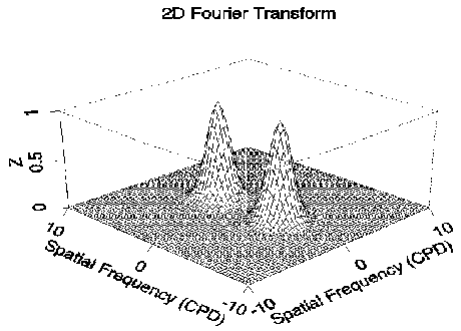


Figura 2.14: La transformada de Fourier de una función de Gabor en el dominio espacial es una gaussiana desplazada en el dominio frecuencial.

Teniendo en cuenta todo lo anterior, podemos obtener la respuesta de un filtro de Gabor aplicado a una imagen a través de una operación de convolución [25]:

$$g(x, y, \theta, \phi) = \iint I(p, q) g(x - p, y - q, \theta, \phi) dpdq \tag{2.26}$$

donde:

- $g(x, y, \theta, \phi)$: respuesta de un filtro de Gabor con una frecuencia θ y una orientación ϕ aplicada a una imagen en el punto (x,y).
- $I(x, y)$: valor de la imagen en el punto (x,y).

2.4.4.1. Banco de filtros de Gabor

Un banco de filtros de Gabor es un conjunto de filtros paso-banda distribuidos en el dominio de la frecuencia y con múltiples orientaciones que de manera aproximada imitan el comportamiento del sistema visual humano.

Diseño de un banco de filtros de Gabor El diseño del banco de filtros debe tener en cuenta ciertos aspectos:

- Debe cubrir el plano de frecuencia uniformemente. Vendrá regulada por la variable escala, s.
- Debe detectar las características en todas las direcciones. Este objetivo dependerá de la variable orientación, ϕ.

Como puede apreciarse en la figura 2.15, se ha de elegir el valor adecuado de ambas variables para que cumplan las anteriores especificaciones.

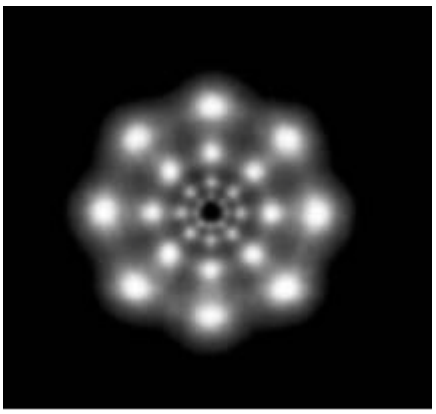


Figura 2.15: Orientaciones y frecuencias radiales de un banco de filtros de Gabor para una escala 4 y con 4 orientaciones [28]

Resaltar, que estos filtros tienen dos principales limitaciones:

- El ancho de banda máxima de un filtro de Gabor está limitado aproximadamente a un octavo.
- No son la elección más óptima si se busca una amplia información espectral con la máxima localización espacial.

2.4.4.2. Log-Gabor

Los filtros log-Gabor [44] solucionan los inconvenientes de los filtros de Gabor, ya que tiene una componente DC (componente continua) igual a cero, por lo que el ancho de banda del filtro no está limitado a un octavo. De esta manera, se puede usar un número menor de filtros para cubrir un espectro deseado. Estos filtros son un derivado de los filtros de Gabor y se usan ampliamente en procesamiento de imágenes. La respuesta en frecuencia de los filtros de log-Gabor es una gaussiana con escala frecuencial logarítmica, en contraposición al filtro de Gabor, cuya respuesta en frecuencia se trata de una escala lineal. La respuesta en frecuencia de una log-Gabor [44] es descrita como:

$$G(w) = \exp\left(-\frac{(\log(\frac{w}{w_0}))^2}{2(\log(\frac{k}{w_0}))^2}\right) \tag{2.27}$$

donde:

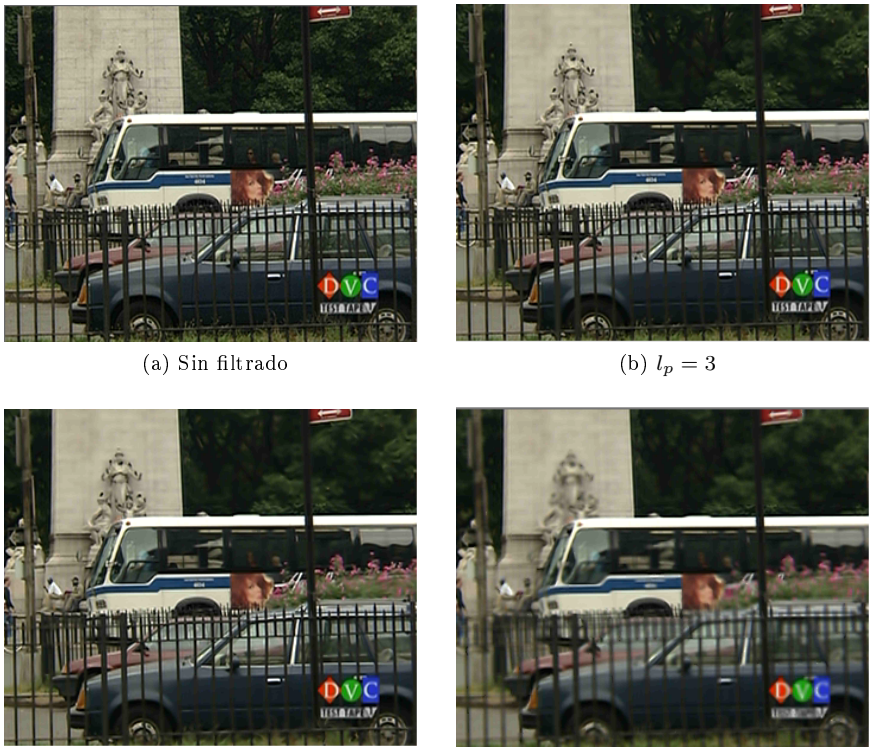
- w_0: frecuencia central del filtro.
- $\frac{k}{w_0}$: ancho de banda.

Señalar que este tipo de filtros presentan las desventajas de ineficiencia computacional y el requerimiento de un gran número de canales para cubrir de forma aproximada el plano de frecuencias en su totalidad.

2.4.4.3. Filtrado para distintas configuraciones

A continuación se incluye algunas imágenes filtradas con un banco de filtros "log-gabor".

Bus

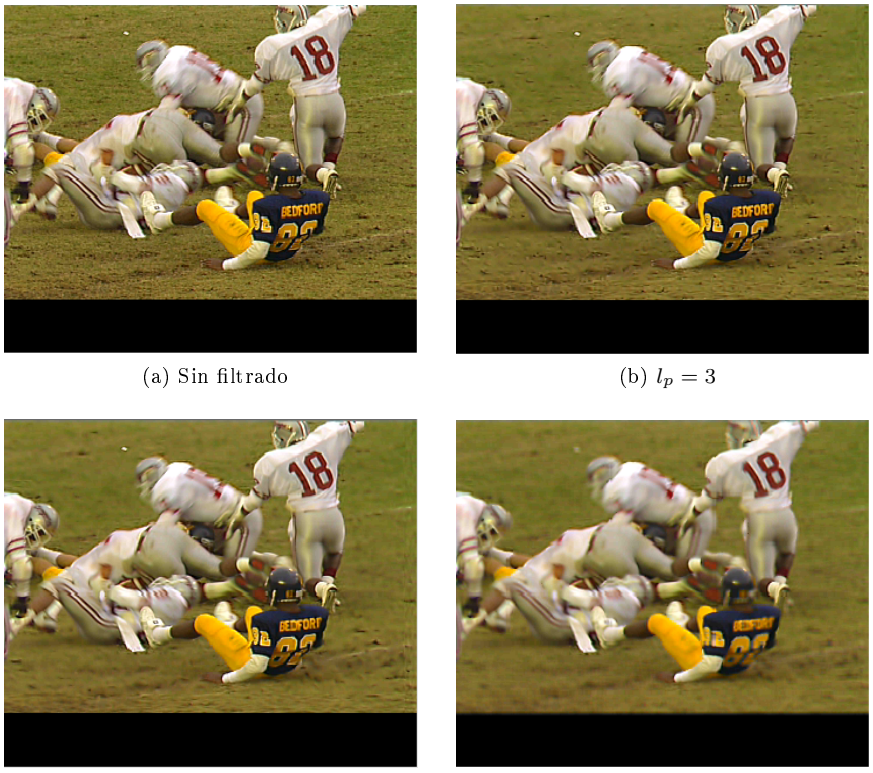


(a) Sin filtrado (b) $l_p = 3$

(c) $l_p = 5$ (d) $l_p = 8$

Figura 2.16: Filtrado de Gabor de la secuencia “Bus”

Football



(a) Sin filtrado (b) $l_p = 3$

(c) $l_p = 5$ (d) $l_p = 8$

Figura 2.17: Filtrado de Gabor de la secuencia "Football"

2.4.5. Filtros tridimensionales

Los filtros tridimensionales calculan el valor final del píxel a partir de un estudio temporal y uno espacial. Algunos artículos realizan una operación de promediado de los pesos obtenidos en el dominio temporal y espacial, y por otro lado, otros realizan un menor o mayor filtrado espacial en función de la información que se obtiene del dominio temporal.

Por ejemplo, en [29] se realiza un procesamiento temporal independiente de un procesamiento espacial para posteriormente combinar ambos resultados y conseguir un plano sin ruido.

Este método aplica un filtro de Kalman 1-D para obtener la redundancia temporal y un filtro Wiener que obtiene la redundancia espacial y preserva los bordes. Como se asume que ambos resultados no están correlacionados, se combinan ambos estimadores de 2 formas:

- Promediado de los resultados obtenidos del filtro Wiener y Kalman para obtener el plano filtrado.
- Empleo de una versión filtrada del plano actual y previo para estimar el vector de movimiento. A continuación, realiza una media de los resultados

de los filtros de Kalman y Wiener.

En [30] el filtrado espacial se lleva a cabo realizando una media de los píxeles (ventana de 3x3), mientras que el proceso de estimación de movimiento del codificador H.264 se reutiliza para obtener el filtrado temporal.

Como el codificador de vídeo H.264 permite el uso de múltiples referencias para predecir un bloque o macrobloques, se utiliza esta estrategia para obtener varias posibles predicciones temporales del píxel actual (píxel filtrado previamente) y se realiza un promediado de estas predicciones para finalmente obtener el plano filtrado. A diferencia del método anterior, [31] utiliza planos pasados, futuros y el plano actual para el filtrado temporal, y para el filtrado espacial utiliza dos filtros espaciales ("Wavelet" y Wiener).

Y con una idea similar al de los anteriores se realiza un promediado de las hipótesis obtenidas del filtrado espacial y temporal. De manera que:

$$f'(x,y,t) = w_{sp1} f'_{sp1}(x,y,t) + w_{sp2} f'_{sp2}(x,y,t) + \sum_{k=-N}^{N} w_k f'_T(x,y,t+k) \quad (2.28)$$

siendo:

- $f'_{sp1}(x,y,t)$: hipótesis espacial obtenida al utilizar una descomposición "Wavelet".
- $f'_{sp2}(x,y,t)$: hipótesis espacial obtenida al utilizar un filtrado Wiener.
- w_{sp1}, w_{sp2} : pesos obtenidos al aplicar una SAD ("Sum of Absolute Differences") a las hipótesis espaciales.
- $f'_T(x,y,t)$: hipótesis temporal obtenida al aplicar una compensación de movimiento basado en bloques.
- w_k: pesos de las hipótesis temporales al aplicar una SAD a las mismas.
- N: número de planos.
- K: distinto de cero.

A continuación se presentan artículos que realizan un filtrado espacial a partir de la información temporal.

Por ejemplo, en [33] se controla la intensidad del filtrado bilateral en función de un mapa de prioridad. Su estrategia se divide en tres pasos:

1. El vídeo original de entrada se convierte al formato QCIF.
2. La combinación de un filtro bilateral adaptado según el movimiento preserva bordes en la región de interés (ROI) y realiza un suavizado en la región de no interés. Para ello, dispone de un conjunto de parámetros cuyo valor dependerá de la cantidad de movimiento (mapa de movimiento). Dispone a su vez de dos funciones de peso que dependerán del valor de los parámetros anteriores. Por tanto, el peso total del filtro bilateral se obtiene de la multiplicación de las dos funciones de peso y la imagen f'.

3. Después de la codificación puede ser necesario un filtrado para una mejora de la calidad visual. Se aplica un filtrado "high-boost" solamente a las regiones con bordes (resalto del borde).

De manera similar en [34] y [35] se ajusta la intensidad del filtrado espacial (bilateral) modificando los parámetros locales del filtro en función de otras medidas. En [34] se modifican estos parámetros a partir de una variable auxiliar L de la manera siguiente:

- Se define L en el rango [0 - L_{max}], L de filtrado nulo a filtrado máximo.
- Para cada plano codificado se decide el nivel de filtrado:
  - Si se trata del primer frame se le da un valor inicial L_{init}.
  - Para los planos restantes, L toma un valor bajo si se asigna un número de bits significativos al frame anterior. Y toma valores altos si por el contrario, se le asignan pocos bits al frame anterior.

Sin embargo, [35] modifica los parámetros del filtro en función de un mapa de control, que a su vez depende de un mapa de atención y del ancho de banda del canal (véase ecuación 2.29).

$$C(x,t) = A(x,t)\frac{B_0}{B(t)} \tag{2.29}$$

siendo:

- B_0: ancho de banda aceptable del canal.
- $C(x,t)$: mapa de control.
- $A(x,t)$: mapa de atención.

De manera que si el mapa de control tiende a cero, se realiza un mayor filtrado y si tiende a 1, se realiza poco filtrado o nada.

Otra forma de proceder la encontramos en las referencias [32] y [36].

En la primera se defiende que los planos adyacentes de una señal de vídeo tienen fuertes correlaciones y que por tanto, es recomendable tener en cuenta los coeficientes temporales vecinos.

El método propuesto se divide en los siguientes pasos:

1. Se realiza una estimación de movimiento de los planos pasados y futuros.
2. Los resultados de la estimación se utilizan para la compensación de movimiento.
3. La transformada "Wavelet" se aplica al plano actual y al resultado de la compensación de movimiento de los planos pasados y futuros.
4. A los vectores de coeficientes "Wavelet" (formados por vecinos espacio-temporales) se les aplica un método de eliminación de ruido ST-GSM ("Spatio-Temporal Gaussian Scale Mixture"). Este modelo captura simultáneamente correlaciones locales de los coeficientes "Wavelet" en espacio y tiempo.

5. Se realiza la transformada inversa de "Wavelet" para obtener el plano filtrado.

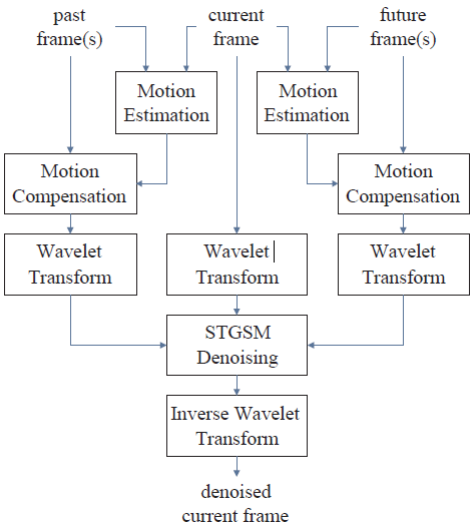


Figura 2.18: Diagrama de bloque del algoritmo

Por otro lado, los autores de [36] proponen un control de pre-filtrado que sigue los pasos siguientes:

1. Se genera un frame predictivo a partir de la estimación de movimiento.
2. Se genera un plano residuo obtenido de la diferencia del plano actual y el predictivo. Y se calcula la varianza de error predictivo y los coeficientes de correlación .
3. Se estima la degradación de la codificación sin prefiltrado y con prefiltrado. De manera, que si este último se obtiene una mejora significativa se realiza un filtrado más intenso de la imagen y de no serlo se realiza poco o nada de filtrado.

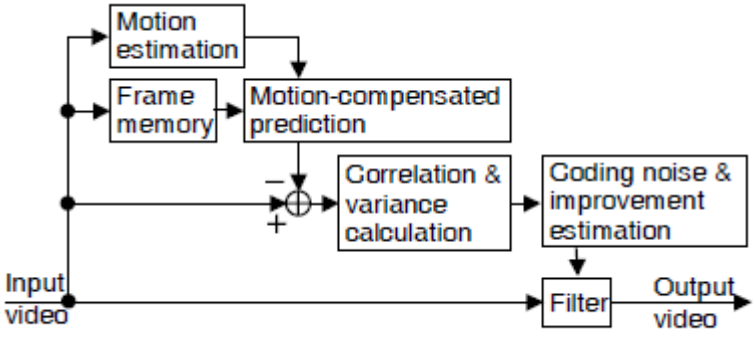


Figura 2.19: Esquema del control del prefiltrado

2.5. Redes Neuronales

Una red neuronal artificial (RNA o artificial neural network (ANN)) es un sistema de procesamiento de información que tiene ciertas aptitudes en común con las redes neuronales biológicas:

- El procesamiento de información ocurre en muchos elementos simples llamados neuronas.
- Las señales son transferidas entre neuronas a través de enlaces de conexión.
- Cada conexión tiene un peso asociado, el cual, típicamente, multiplica la señal transmitida.
- Cada neurona aplica una función de activación (usualmente no lineal) a su entrada de red (suma de entradas pesadas) para determinar su salida.

2.5.1. Estructura de la red Neuronal

La distribución de neuronas dentro de la red [39] se realiza formando niveles o capas con un número determinado de dichas neuronas en cada una de ellas. A partir de su situación dentro de la red, se pueden distinguir tres tipos de capas (véase figura 2.20):

- De entrada: es la capa que recibe directamente la información proveniente de las fuentes externas de la red.
- Ocultas: son internas a la red y no tienen contacto directo con el entorno exterior. El número de niveles ocultos puede estar entre cero y un número elevado. Las neuronas de las capas ocultas pueden estar interconectadas de distintas maneras, lo que determina, junto con su número, las distintas topologías de redes neuronales.
- De salidas: transfieren información de la red hacia el exterior.

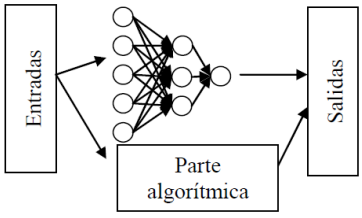


Figura 2.20: Sistema Neuronal Artificial

Cada entrada tiene su propio peso relativo que proporciona la importancia de la entrada dentro de la función de agregación de la neurona. Estos pesos pueden adaptarse dentro de la red y ser modificados en respuesta de los ejemplos de entrenamiento de acuerdo a la topología específica o debido a las reglas de entrenamiento.

Las entradas y los pesos pueden ser combinados de diferentes maneras mediante una función o regla de propagación. Esta regla permite obtener, a partir de las entradas y los pesos, el valor del potencial postsináptico h_i de la neurona.

Algunas de las funciones de propagación más comúnmente utilizadas y conocidas son:

1. Sumatoria de las entradas pesadas: es la suma de todos los valores de entrada a la neurona, multiplicados por sus correspondientes pesos. Este proceso consiste en la suma de las entradas ponderadas con sus pesos sinápticos correspondientes.

$$H_i(t) = \sum_{j=1}^{N} X_j W_j \tag{2.30}$$

   siendo:

   - $H_i(t)$: potencial sináptico de la neurona i en el momento t.
   - W_j: el peso sináptico asociado a la entrada X_j.
   - X_j: la entrada de datos procedentes de la información j.

2. Producto de las entradas pesadas: es el producto de todos los valores de entrada a la red multiplicados por sus correspondientes pesos.

3. Máximo de las entradas pesadas: solamente toma en consideración el valor de entrada más fuerte, previamente multiplicada por su peso correspondiente.

El resultado de la función de propagación es transformado mediante un proceso algorítmico conocido como función de activación.

La función activación calcula el estado de actividad de una neurona, transformando la entrada global en un valor (estado) de activación, cuyo rango normalmente va de (0 a 1) o de (–1 a 1).

Las funciones de activación más comúnmente utilizadas se muestran en la tabla [39]:

Función	Fórmula	Rango
Identidad	$y = x$	$[-\infty, \infty]$
Escalón	$y = \begin{cases} +1 & si\, x \geq 0 \\ 0 & si\, x < 0 \end{cases}$	$[0, 1]$
	$y = \begin{cases} +1 & si\, x \geq 0 \\ -1 & si\, x < 0 \end{cases}$	$[-1, 1]$
Lineal a tramos	$y = \begin{cases} x & si\ -1 \leq x \leq 1 \\ +1 & si\, x > 0 \\ -1 & si\, x < -1 \end{cases}$	$[-1, 1]$
Sigmoidea	$y = \frac{1}{1+\exp^{-x}}$	$[-0, 1]$
	$y = tanh(x)$	$[-1, 1]$
Sinusoidal	$y = sen\,(\omega x + phi)$	$[-1, 1]$

Tabla 2.1: Funciones de activación

Otra función importante es la función de salida que proporciona el valor de salida de la neurona en base al estado de activación de la neurona. La salida puede ser binaria o continua. Generalmente la primera utiliza los valores {0,1} o {-1,1} y las otras admiten valores dentro de un determinado rango, que en general suele definirse como [-1, 1] o [0, 1].

2.5.1.1. Mecanismos de aprendizaje.

La topología de la red y las diferentes funciones de cada neurona (entrada, activación y salida) no pueden cambiar durante el aprendizaje, pero los pesos sobre cada una de las conexiones si pueden hacerlo; esto implica que durante el proceso de aprendizaje los pesos de las conexiones de la red sufran modificaciones. Por lo tanto, se puede afirmar que este proceso ha terminado (la red ha aprendido) cuando los valores de los pesos permanecen estables.

Hay dos métodos de aprendizaje importantes que pueden distinguirse:

- Aprendizaje no supervisado.

  Las redes con aprendizaje no supervisado no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta.

- Aprendizaje supervisado.

  El aprendizaje supervisado se caracteriza porque el proceso de aprendizaje se realiza mediante un entrenamiento controlado por un agente externo que

determina la respuesta que debería generar la red a partir de una entrada determinada. El supervisor controla la salida de la red y en caso de que ésta no coincida con la deseada, se procederá a modificar los pesos de las conexiones con el fin de conseguir que la salida obtenida se aproxime a la deseada. De esta manera se propagan los errores hacia la capa de entrada a través de la red neuronal ajustando los pesos de las capas ocultas. Por lo que el cambio de los pesos en las conexiones de las neuronas, además de influir sobre la entrada global, influye en la activación y, por consiguiente, en la salida de una neurona.

2.5.1.2. Topología de las redes neuronales.

Según [40] la topología o arquitectura de una red neuronal consiste en la organización y disposición de las neuronas en la misma, formando capas o agrupaciones de neuronas más o menos alejadas de la entrada y salida de dicha red. En este sentido, los parámetros fundamentales de la red son: el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de conexiones entre neuronas.

Redes monocapa

Red formada por una única capa de neuronas. Dichas neuronas cumplen la función de neuronas de entrada y salida simultáneamente.

Redes multicapa

Las redes multicapas son aquellas que disponen de un conjunto de neuronas agrupadas en varios niveles o capas. En estos casos, una forma para distinguir la capa a la que pertenece una neurona consistiría en fijarse en el origen de las señales que recibe a la entrada y el destino de la señal de salida. Normalmente, todas las neuronas de una capa reciben señales de entrada desde otra capa anterior (la cual está más cerca a la entrada de la red) y envían señales de salida a una capa posterior (que está más cerca de la salida de la red). A estas conexiones se las denomina conexiones hacia adelante o "feedforward". Sin embargo, también existe la posibilidad de conectar la salida de las neuronas de capas posteriores a la entrada de capas anteriores en un gran número de estas redes; a estas conexiones se las denomina conexiones hacia atrás o "feedback".

Estas dos posibilidades permiten distinguir entre dos tipos de redes con múltiples capas: las redes con conexiones hacia adelante o redes "feedforward", y las redes que disponen de conexiones tanto hacia adelante como hacia atrás o redes "feedforward/feedback".

Las redes neuronales con conexión hacia adelante son las más utilizadas, obteniéndose muy buenos resultados fundamentalmente como clasificadores de patrones y estimadores de funciones. Dentro de este grupo de redes neuronales encontramos al perceptrón multicapa. El perceptrón multicapa es una extensión del perceptrón simple. Su estructura consiste en un número de capas ocultas que realizan el procesamiento complejo sobre las entradas. El uso de neuronas con función de activación no lineal, como lo es la sigmoide, permite a la red aprender las discontinuidades del entorno.

El algoritmo que permite entrenar a redes de muchas capas se denomina "Back Propagation" (para más información ver [43]) o Propagación hacia atrás.

En este algoritmo se calcula el valor promedio del error cuadrado con la ecuación 2.31:

$$e = \frac{1}{N} \sum_{n=1}^{N} e(n)^2 \tag{2.31}$$

donde:

- N: denota el número total de muestras contenidas en el conjunto de entrenamiento.

El perceptrón multicapa posee en general una buena capacidad de generalización, de la que hablaremos a continuación.

2.5.2. Generalización

Una vez finalizada la fase de aprendizaje, la red puede ser utilizada para realizar la tarea para la que fue entrenada. Una de las principales ventajas que posee este modelo es que la red aprende la relación existente entre los datos, adquiriendo la capacidad de generalizar conceptos.

Cuando se evalúa una red neuronal no sólo es importante evaluar si la red ha sido capaz de aprender los patrones de entrenamiento, sino que también es imprescindible evaluar el comportamiento de la red ante patrones nunca vistos antes. Esta característica se conoce como capacidad de generalización y es adquirida durante la fase de entrenamiento.

Es necesario que durante el proceso de aprendizaje la red extraiga las características de las muestras para poder luego responder correctamente a nuevos patrones.

De lo dicho anteriormente surge la necesidad de evaluar durante la fase de entrenamiento dos tipos de errores. El error de aprendizaje que indica la calidad de la respuesta de la red a los patrones de entrenamiento, y el error de generalización, que indica la calidad de la respuesta de la red a patrones nunca antes vistos.

Para poder obtener una medida de ambos errores es necesario dividir el conjunto de datos disponible en dos, en el conjunto de datos de entrenamiento, y en el de validación. El primero se utiliza durante la fase de entrenamiento para que la red pueda extraer las características de los mismos y, mediante el ajuste de sus pesos sinápticos, la red logre una representación interna de la función. El conjunto de evaluación se emplea para evaluar la capacidad de generalización de la red.

También es importante comentar que la causa más común de la pérdida de capacidad de generalización es el sobreaprendizaje. Esto sucede cuando la cantidad de ciclos de entrenamientos tiende a ser muy alta. Se observa que la respuesta de la red a los patrones de entrenamiento es muy buena mientras que la respuesta a nuevos patrones tiende a ser muy pobre. Al aumentar el número de ciclos la red tiende a sobreajustar la respuesta a los patrones de entrenamiento a expensas de una menor capacidad de generalización.

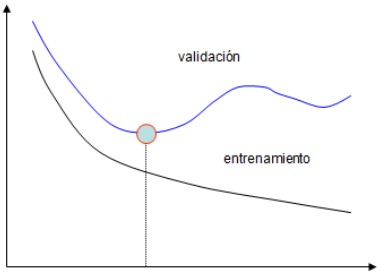


Figura 2.21: Generalización

Como se observa en la figura 2.21, el error de entrenamiento de un clasificador decrece monótonamente durante la fase de entrenamiento, mientras que el error sobre el conjunto de validación decrece hasta un punto a partir del cual crece, lo que indica que a partir del mismo el clasificador realiza un sobreajuste sobre los datos de entrenamiento. Por ello, el proceso de entrenamiento debe finalizar cuando se alcance el primer mínimo de la función de error de validación.

En ocasiones la pérdida de capacidad de generalización se produce por el uso excesivo de neuronas ocultas en la red neuronal. Esto hace que la red tienda a ajustar con mucha exactitud los patrones de entrenamiento, evitando que la red extraiga las características del conjunto.

Capítulo 3

Codificación perceptual por texturas

3.1. Introducción

El enmascaramiento es una de las principales características del sistema visual humano que se emplea en tareas de codificación perceptual de vídeo e imagen. Como ya sabemos, el enmascaramiento basado en texturas consiste en encontrar aquellas regiones de la imagen que, por su estructura, pueden albergar distorsión de forma menos notoria para el sistema visual humano (HVS).

Se propone detectar los macrobloques de textura caótica ("caotic") y textura detallada ("detailed") y, a continuación, aplicar sobre los primeros una cuantificación más elevada y sobre los segundos un valor de cuantificación menor, con el objetivo de conseguir una reducción de la tasa binaria en el proceso de codificación, sin que ello implique una disminución de la calidad de la secuencia.

A continuación se realiza una introducción de un método de enmascaramiento por texturas basado en la DCT, desarrollado por el Grupo Multimedia del Departamento de Teoría de la Señal y Comunicaciones de la Universidad Carlos III de Madrid, seguido de un método alternativo, desarrollado para este proyecto, consistente en el enmascaramiento de texturas basado en el Histograma de Gradientes Orientados.

3.2. Enmascaramiento por texturas basado en la DCT

El objetivo de este trabajo consiste en obtener un mapa de enmascarabilidad que determine qué regiones (dependiendo del tipo de textura que posean) son susceptibles de minimizar o maximizar la distorsión percibida por el HVS. Para conseguirlo, detecta alguna dirección predominante atendiendo a la distribución de los coeficientes AC resultantes de efectuar la DCT sobre un bloque (véase la figura 3.1).

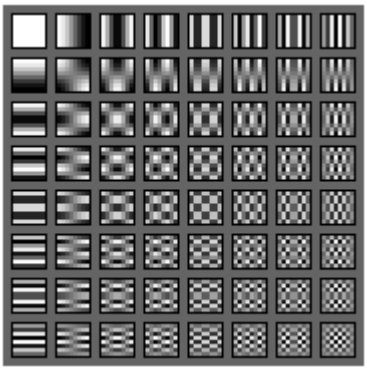


Figura 3.1: Funciones base de la DCT 8x8

Los coeficientes de la primera fila poseen información significativa ante un borde vertical, mientras los coeficientes de la primera columna serán importantes ante un borde horizontal.

Este método consiste en multiplicar 7 máscaras (equivalen a 7 direcciones previamente definidas) por la matriz de coeficientes, obtenidos después de realizar la DCT del macrobloque, con el objetivo de obtener un diagrama de radiación en el que se aprecie la energía que posee cada una de las 7 direcciones definidas, por lo que un valor alto de la energía obtenida es indicativo de la existencia de un borde en esa misma dirección. Para obtener la energía del macrobloque en una dirección, se multiplica la máscara definida en esa dirección por el macrobloque original y se suman todos los valores obtenidos de la matriz resultante. De modo que, bloques "caotic" presentan idealmente un espectro plano y para bloques "detailed", el espectro idealmente presenta una componente muy destacada respecto a las demás (véase las figuras 3.2 y 3.3).

De igual manera que otros trabajos realizan su propia clasificación de texturas, esta línea de investigación realiza la suya, siendo mostrada a continuación:

"Caotic": MB sin patrones estructurados, es decir, los coeficientes de alterna (AC) de la DCT poseen valores similares, sin ninguna frecuencia dominante. Por lo tanto, este tipo de texturas pueden admitir distorsión.

"Detailed": áreas que presentan bordes prominentes y diferenciados. Poseen coeficientes dominantes en la DCT (se corresponden con la dirección del borde). Este tipo de texturas no admite distorsión.

"Smooth": regiones completamente homogéneas o lisas, cuya componente de continua es prácticamente toda la energía de dicha región. Estas regiones pueden admitir distorsión y no ser percibidas por el HVS.

"Dark": zonas de la imagen con una intensidad notablemente menor que la intensidad media de todo el plano. Estas regiones son tratadas como bloques "smooth" y también admiten distorsión.

"Bright": zonas de la imagen con una intensidad notablemente mayor que la intensidad media de todo el plano. Al igual que en el caso anterior, admiten distorsión.

Figura 3.2: Secuencia “Coastguard”

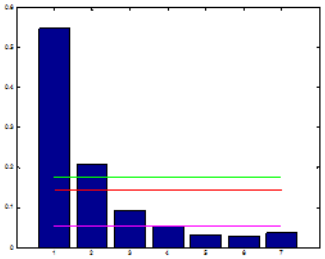
"Detailed"	
	• Línea verde: desviación típica del espectro. • Línea magenta: mediana del espectro. • Línea roja: media del espectro.

Table 3.1: Datos extraídos de una región "detailed" de la figura 3.2

Figura 3.3: Secuencia “LOTR1”

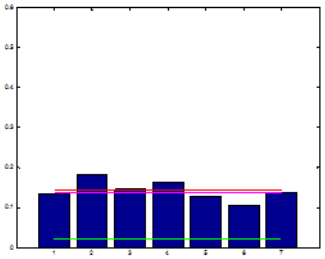
"Caotic"	
	• Línea verde: desviación típica del espectro. • Línea magenta: mediana del espectro. • Línea roja: media del espectro.

Table 3.2: Datos extraídos de una región "caotic" de la figura 3.3

3.3. Enmascaramiento de texturas basado en el Histograma de Gradientes Orientados

Atendiendo a la clasificación realizada en el apartado anterior, esta parte del proyecto se centra en la detección de texturas detalladas y caóticas, empleando para caracterizarlas un algoritmo del tipo HOG como los descritos en la sección 2.3.2 en lugar de los histogramas de direcciones dominantes extraídos del espectro de la DCT.

3.3.1. Clasificación por umbralización

Teniendo claro los conceptos de gradiente y borde, y las definiciones de macrobloque “caotic” y “detailed”, se concluye que los píxeles de los bloques “detailed” tendrán magnitudes de gradiente similar. Asimismo la dirección del gradiente asociado a los píxeles de un mismo borde también será aproximadamente la misma.

Los píxeles de los bloques “caotic” suelen tener gradientes con magnitudes y direcciones diferentes, por consiguiente, tendrán idealmente un histograma plano. Por el contrario, los bloques “detailed” tendrán un histograma con una dirección o varias direcciones predominantes.

3.3.1.1. Elección del filtro

Para el cálculo de los Histogramas de Gradientes Orientados, es necesario utilizar un filtro que nos aporte la magnitud del gradiente y la dirección del gradiente de los píxeles, por lo que se realizó un estudio para encontrar el filtro más adecuado para el desarrollo del algoritmo. Las máscaras que se utilizaron en las pruebas se encuentran descritas en el apartado 2.3.1.

Esta prueba consiste en analizar zonas claramente “detailed” y “caotic”, con el fin de extraer datos estadísticos que se consideren relevantes. En primer lugar, se muestra el efecto de los filtros (estudiados) sobre la imagen de la figura 3.4, que nos ayudará a la elección del filtro.

La figura 3.5a y 3.5b se corresponden al filtrado de la imagen con los filtros Frei-Chen, Sobel y Prewitt, siendo los resultados obtenidos muy similares.

Por otra parte, la figura 3.6 corresponde al filtro Roberts. Como se observa, se distinguen ligeras diferencias entre el resultado proporcionado por el filtro de Roberts respecto a los filtros restantes.

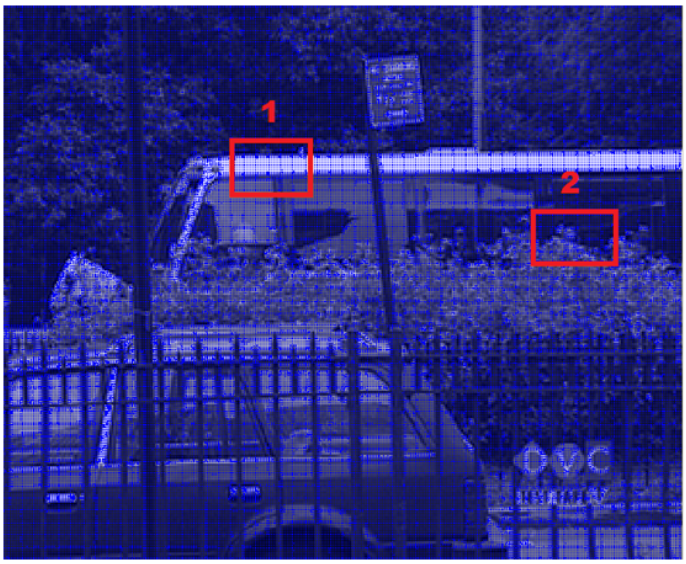
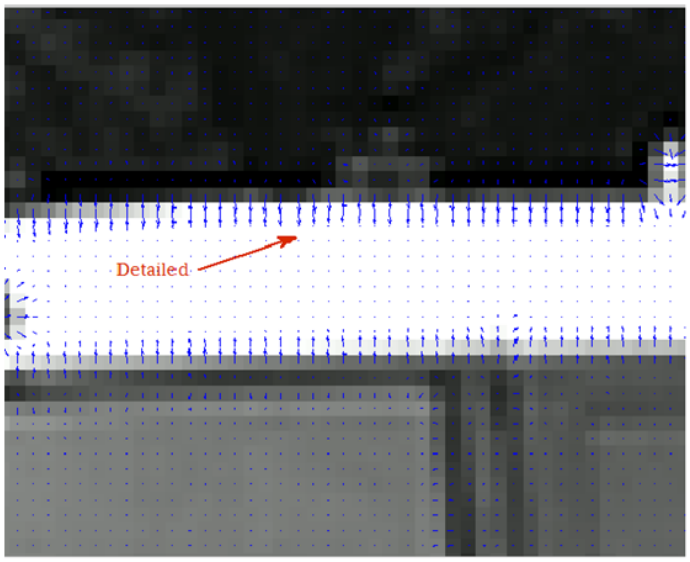
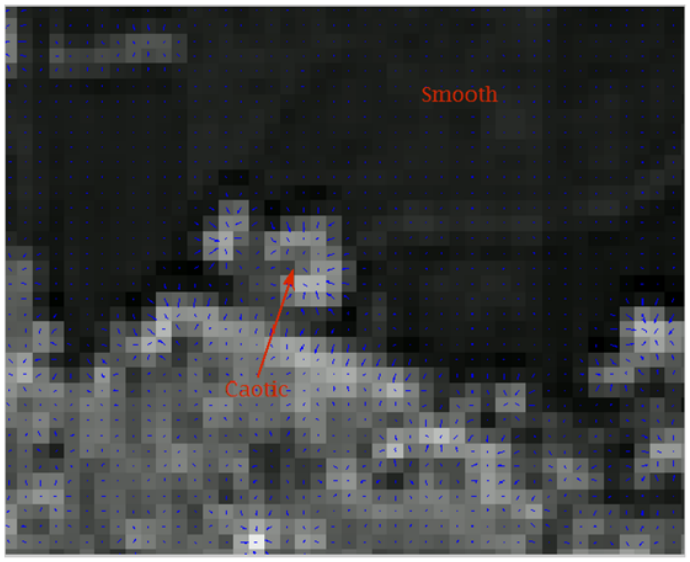


Figura 3.4: Zonas recuadradas de las que se realizan las pruebas



(a) Región 1 - Detailed

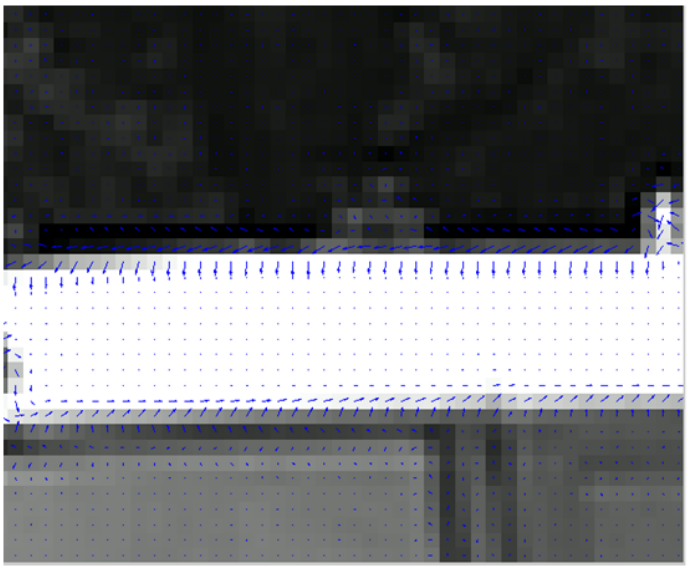


(b) Región 2 - Caotic y smooth

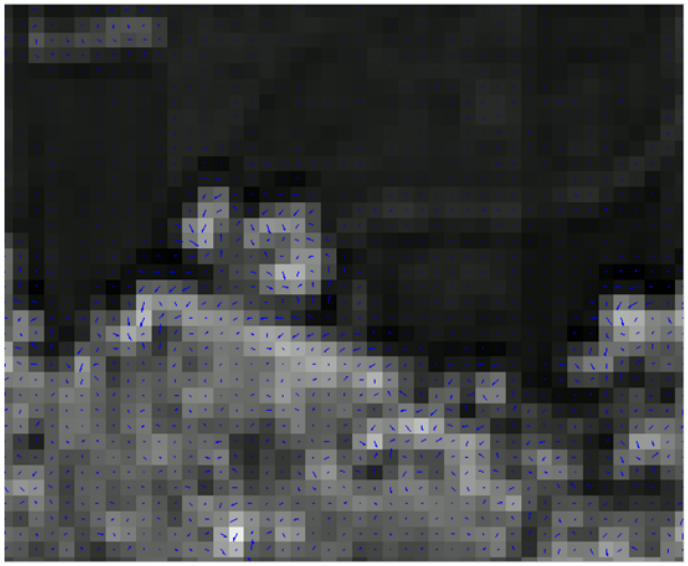
Figura 3.5: Regiones recuadradas.

Como puede observarse en la figura 3.5a, la zona superior del autobús se corresponde claramente con un borde definido ("detailed"), y por tanto, los módulos de los gradientes de los píxeles tienen un magnitud significativa, además de estar orientados en la misma dirección. Sin embargo, en la figura 3.5b se observan píxeles orientados en distintas direcciones con magnitudes variables que se corresponden con una zona "caotic". También se ve una región lisa, donde las magnitudes de los píxeles son casi cero. Esta región se corresponde con una región "smooth".

A diferencia de los filtros anteriores, la salida del filtro de Roberts muestra direcciones de gradiente de los píxeles no bien definidas como puede observarse en la parte superior de la figura 3.6a. Hay que tener en cuenta que estas direcciones más variables pueden acarrear problemas en la detección de un borde. Señalar que no se aprecian diferencias importantes en las zonas caotic.



(a) Región 1 - Detailed



(b) Región 2 - Caotic y smooth

Figura 3.6: Filtro de Roberts

A continuación, se muestran dos imágenes de la secuencia “Bus” en las que

se recuadran zonas “detailed” (3.7a) y zonas “caotic” (3.7b):

(a) Zona "detailed" (b) Zona "caotic"

Figura 3.7: Zonas de extracción de datos estadísticos

La forma de presentar los resultados es la siguiente:

- Columna 1: histograma del macrobloque.
- Columna 2: se presentan estadísticos de la distribución obtenida que se consideran como los más relevantes:
  - Desviación típica.
  - Máximo.
  - Media.
  - Mediana.

Pruebas realizadas con el filtro Frei-Chen

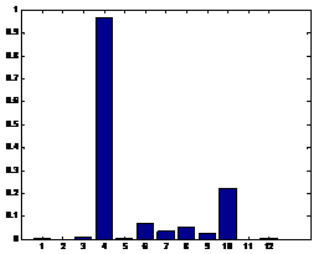
"Detailed"	
	Desviación típica = 0.2750 Máximo = 0.9690 Media = 0.1183 Mediana = 0.0191

Table 3.3: Datos extraídos (bloque 1) para el filtro Frei-Chen de la figura 3.7a

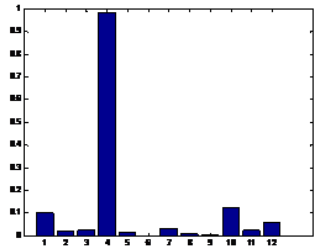
"Detailed"	
	Desviación típica = 0.2756 Máximo = 0.9837 Media = 0.1171 Mediana = 0.0239

Table 3.4: Datos extraídos (bloque 2) para el filtro Frei-Chen de la figura 3.7a

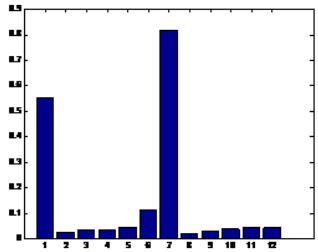
"Detailed"	
	Desviación típica = 0.2577 Máximo = 0.8181 Media = 0.1498 Mediana = 0.0410

Table 3.5: Datos extraídos (bloque 3) para el filtro Frei-Chen de la figura 3.7a

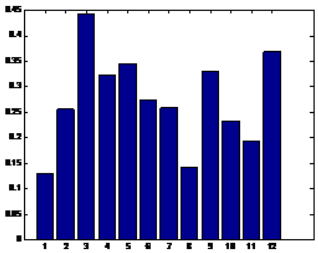
"Caotic"	
	Desviación típica = 0.0932 Máximo = 0.4424 Media = 0.2745 Mediana = 0.2658

Table 3.6: Datos extraídos (bloque 1) para el filtro Frei-Chen de la figura 3.7b

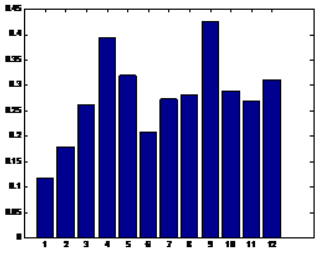
"Caotic"	
	Desviación típica = 0.0849 Máximo = 0.4254 Media = 0.2770 Mediana = 0.2775

Table 3.7: Datos extraídos (bloque 2) para el filtro Frei-Chen de la figura 3.7b

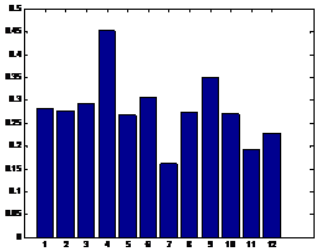
"Caotic"	
	Desviación típica = 0.0740 Máximo = 0.4531 Media = 0.2798 Mediana = 0.2756

Table 3.8: Datos extraídos (bloque 3) para el filtro Frei-Chen de la figura 3.7b

Pruebas realizadas con el filtro Sobel

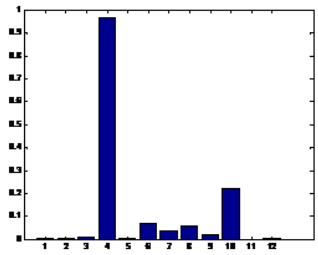
"Detailed"	
	Desviación típica = 0.2752 Máximo = 0.9691 Media = 0.1179 Mediana = 0.0143

Table 3.9: Datos extraídos (bloque 1) para el filtro Sobel de la figura 3.7a

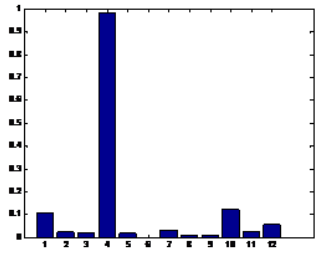
"Detailed"	
	Desviación típica = 0.2755 Máximo = 0.9836 Media = 0.1174 Mediana = 0.0254

Table 3.10: Datos extraídos (bloque 2) para el filtro Sobel de la figura 3.7a

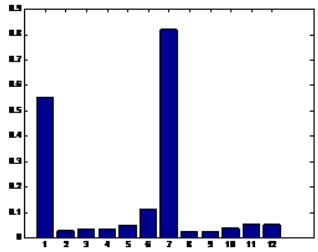
"Detailed"	
	Desviación típica = 0.2569 Máximo = 0.8185 Media = 0.1511 Mediana = 0.0426

Table 3.11: Datos extraídos (bloque 3) para el filtro Sobel de la figura 3.7a

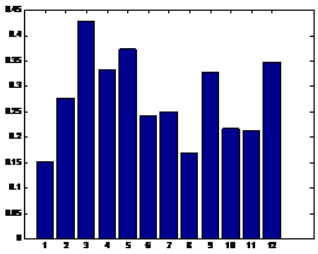
"Caotic"	
	Desviación típica = 0.0852 Máximo = 0.4265 Media = 0.2769 Mediana = 0.2628

Table 3.12: Datos extraídos (bloque 1) para el filtro Sobel de la figura 3.7b

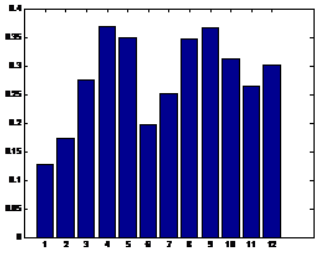
"Caotic"	
	Desviación típica = 0.0789 Máximo = 0.3689 Media = 0.2786 Mediana = 0.2894

Table 3.13: Datos extraídos (bloque 2) para el filtro Sobel de la figura 3.7b

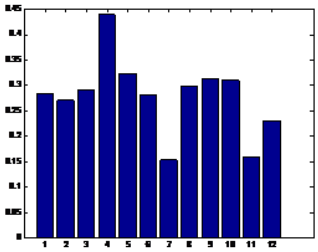
"Caotic"	
	Desviación típica = 0.0756 Máximo = 0.4391 Media = 0.2794 Mediana = 0.2873

Table 3.14: Datos extraídos (bloque 3) para el filtro Sobel de la figura 3.7b

Pruebas realizadas con el filtro Prewitt

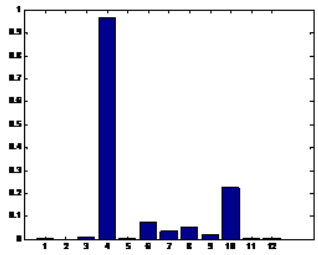
"Detailed"	
	Desviación típica = 0.2749 Máximo = 0.9687 Media = 0.1187 Mediana = 0.0184

Table 3.15: Datos extraídos (bloque 1) para el filtro Prewitt de la figura 3.7a

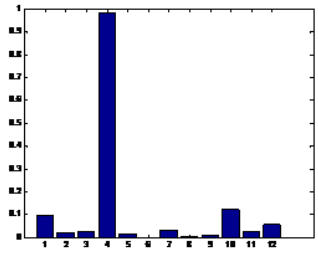
"Detailed"	
	Desviación típica = 0.2756 Máximo = 0.9840 Media = 0.1170 Mediana = 0.0249

Table 3.16: Datos extraídos (bloque 2) para el filtro Prewitt de la figura 3.7a

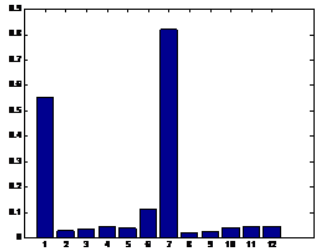
"Detailed"	
	Desviación típica = 0.2578 Máximo = 0.8199 Media = 0.1498 Mediana = 0.0416

Table 3.17: Datos extraídos (bloque 3) para el filtro Prewitt de la figura 3.7a

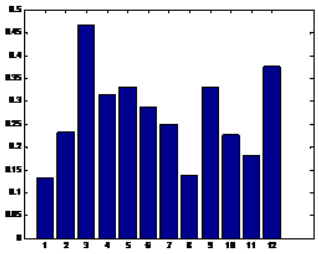
"Caotic"	
	Desviación típica = 0.0986 Máximo = 0.4665 Media = 0.2728 Mediana = 0.2684

Table 3.18: Datos extraídos (bloque 1) para el filtro Prewitt de la figura 3.7b

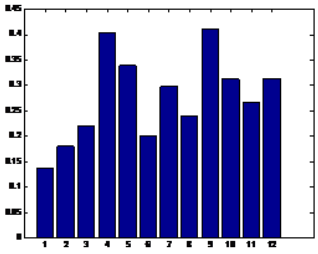
"Caotic"	
	Desviación típica = 0.0855 Máximo = 0.4116 Media = 0.2768 Mediana = 0.2820

Table 3.19: Datos extraídos (bloque 2) para el filtro Prewitt de la figura 3.7b

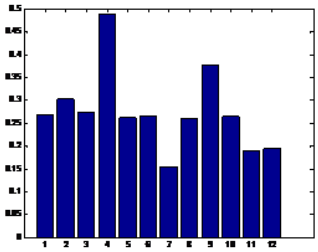
"Caotic"	
	Desviación típica = 0.0882 Máximo = 0.4897 Media = 0.2760 Mediana = 0.2659

Table 3.20: Datos extraídos (bloque 3) para el filtro Prewitt de la figura 3.7b

Pruebas realizadas con el filtro Roberts

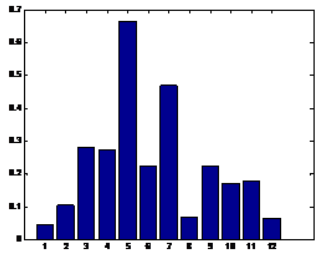
"Detailed"	
	Desviación típica = 0.1812 Máximo = 0.6657 Media = 0.2307 Mediana = 0.2006

Table 3.21: Datos extraídos (bloque 1) para el filtro Roberts de la figura 3.7a

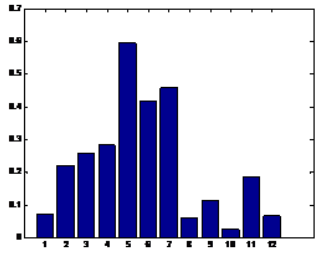
"Detailed"	
	Desviación típica = 0.1816 Máximo = 0.5972 Media = 0.2304 Mediana = 0.2039

Table 3.22: Datos extraídos (bloque 2) para el filtro Roberts de la figura 3.7a

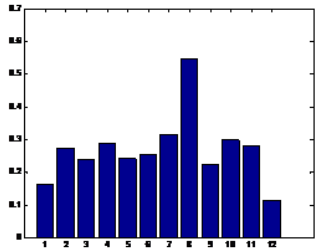
"Detailed"	
	Desviación típica = 0.1045 Máximo = 0.5478 Media = 0.2708 Mediana = 0.2658

Table 3.23: Datos extraídos (bloque 3) para el filtro Roberts de la figura 3.7a

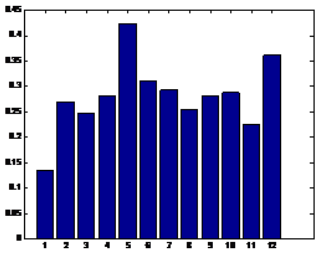
"Caotic"	
	Desviación típica = 0.0702 Máximo = 0.4232 Media = 0.2807 Mediana = 0.2820

Table 3.24: Datos extraídos (bloque 1) para el filtro Roberts de la figura 3.7b

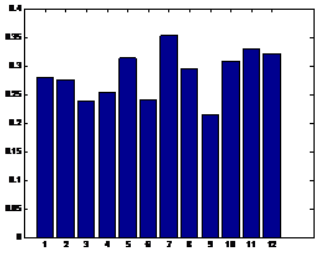
"Caotic"	
	Desviación típica = 0.0423 Máximo = 0.3543 Media = 0.2858 Mediana = 0.2883

Table 3.25: Datos extraídos (bloque 2) para el filtro Roberts de la figura 3.7b

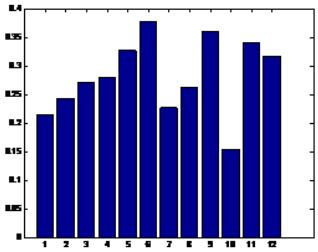
"Caotic"	
	Desviación típica = 0.0658 Máximo = 0.3774 Media = 0.2817 Mediana = 0.2763

Table 3.26: Datos extraídos (bloque 3) para el filtro Roberts de la figura 3.7b

Conclusiones El filtro de Roberts es el que peor funciona, porque los vectores de dirección del gradiente los píxeles no están bien definidos y ello favorece al etiquetado de textura caótica, pero perjudica al etiquetado de textura "detailed". Este filtro se descarta, porque los vectores de gradiente de sus píxeles no están bien definidos y, por tanto, perjudica a la clasificación de los bloques "detailed".

Señalar que los resultados de Prewitt, Sobel y Frei-Chen son muy similares, así que en principio se podría optar por cualquiera de estos filtros. Generalmente funcionan bien ante los dos tipos de texturas expuestas, sin embargo:

- El filtro de Prewitt se trata de un filtro sensible al ruido y aún dando buenos resultados para los bordes verticales y horizontales, no es así para los bordes diagonales.
- El filtro de Frei-Chen se trata de un filtro isotrópico que intenta llegar a un equilibrio entre el filtro de Prewitt y el de Sobel. Aporta buenos resultados.
- El filtro de Sobel parte de los operadores de Prewitt adicionando ciertos pesos en la máscara, y proporciona un suavizamiento Gaussiano. Esto reduce el efecto de amplificación del ruido que es característico de los ope-

radores derivativos, siendo éste el principal motivo por el que generalmente se prefiere al operador de Sobel.

En definitiva, el filtro empleado es el filtro de Sobel.

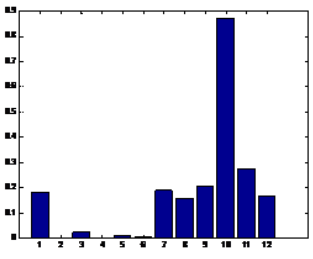
3.3.1.2. Elección de las dimensiones del macrobloque

Las pruebas realizadas con macrobloques 8x8 mostraban que al reducir el tamaño del MB se perdía la estructura general del mismo, es decir, aquellos macrobloques que contenían algún borde no eran detectados como "detailed" debido a que la región era demasiado pequeña y, por tanto, su histograma no mostraba ninguna dirección predominante. Por esta razón, dividir el MB en bloques 8x8 no es admisible debido a que otorgan poca información de la estructura real del MB y aumenta la probabilidad de error de detección.

A continuación se realiza un estudio de una región "detailed" (véase figura 3.8), donde se comprueba este hecho. La forma de presentar la información es similar a la que se presenta en la página 63.

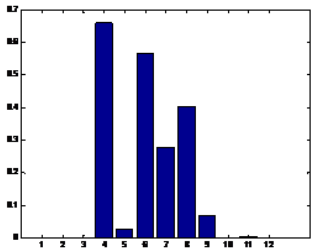
Datos extraídos de la figura 3.8 para macrobloques de tamaño 8x8.

Figura 3.8: Estudio de región 1



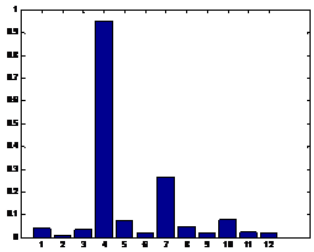
Desviación típica = 0.2403
Máximo = 0.8719
Media = 0.1744
Mediana = 0.1626

Table 3.27: Estudio del bloque 1 de la región 3.8



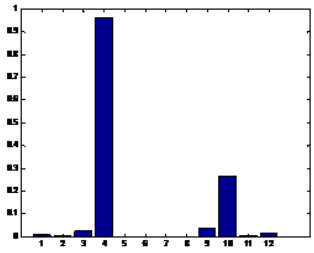
Desviación típica = 0.2460
Máximo = 0.6594
Media = 0.1669
Mediana = 0.0139

Table 3.28: Estudio del bloque 2 de la región 3.8



Desviación típica = 0.2673
Máximo = 0.9531
Media = 0.1336
Mediana = 0.0391

Table 3.29: Estudio del bloque 3 de la región 3.8



Desviación típica = 0.2784
Máximo = 0.9623
Media = 0.1109
Mediana = 0.0070

Table 3.30: Estudio del bloque 4 de la región 3.8

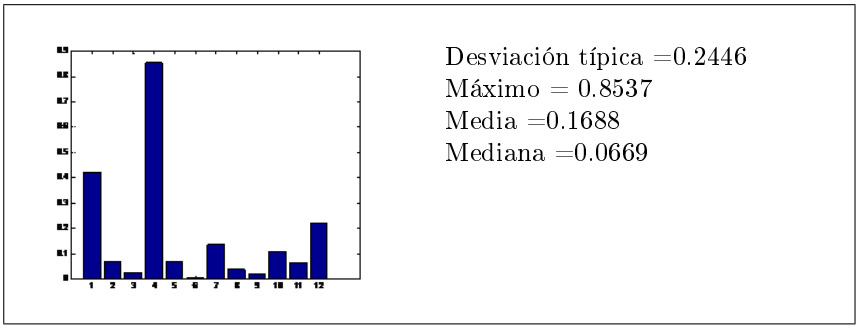


Table 3.31: Estudio del bloque 5 de la región 3.8

Como se puede comprobar, las regiones claramente "detailed" no son bien detectadas, ya que varían los parámetros que pueden afectar a una correcta clasificación. Asimismo, no es recomendable trabajar con macrobloques de tamaño mayor porque se corre el peligro de que la región englobe varios bordes, lo que se traduce en los resultados como la detección de una textura caótica al existir más de una dirección predominante y de esta manera afectar negativamente a la extracción de los datos estadísticos. Por lo tanto, después de realizar pruebas experimentales se concluyó que lo idóneo era trabajar con MBs 16x16, dado que ofrecían mejores resultados. Al mismo tiempo, coincide con el tamaño de macrobloque de H.264, lo cual puede tener ventajas de cara al procesamiento.

3.3.1.3. Número de bins del histograma

Para conocer el número de bins[1] más apropiado, se realizó un estudio de varios bloques pertenecientes a distintas secuencias, que consistía en obtener los gradientes de aquellos píxeles pertenecientes a un mismo borde. Las direcciones de estos gradientes deberían ser iguales para todos ellos con un cierto margen de error. Para calcular este margen de error, se recurre a la siguiente expresión:

$$M = \frac{1}{N_b}\left[\sum_{N_b}(desv_{píxeles})\right] \tag{3.1}$$

donde:

- $desv_{píxeles}$: desviación típica de los píxeles de borde.
- N_b: número de muestras que contienen bordes de distintas direcciones.
- M: margen de "error".

La matriz de la tabla 3.32 es un muestra de la dirección del gradiente asociado a los píxeles de un borde.

[1] Contenedores o barras verticales en los que se divide un histograma.

146,701	-35,0041	-45,6255	2,9961	50,3534	64,5803	138,6447	161,7734	104,3313	42,0649	81,7832	108,2969	117,8803	-178,5241	-109,3325	-42,3138
-45	-17,1257	-16,3249	135,3865	80,8848	69,5528	97,0075	153,724	-90	-38,0826	165,6687	154,9996	-2,7393	-45	-121,5196	-42,8899
-43,7897	-28,028	148,3996	162,1877	-177,1236	36,7704	57,5344	81,0298	-17,9821	-45	174,6393	-176,9713	-4,5175	9,7357	151,325	45
-31,5644	-42,4995	-171,3794	-170,097	-154,2269	-112,4993	4,5176	47,5661	80,2644	95,3607	135	132,2605	46,9654	69,9132	106,1244	84,0879
38,1532	84,4699	128,1532	166,8534	152,1393	83,8251	43,1259	104,1683	81,5361	80,8992	137,7395	135	41,4394	54,7356	126,4569	101,7009
89,0775	75,0592	84,2609	155,8139	173,8249	54,7359	51,9171	167,2358	-69,7261	-61,3253	-71,2959	-101,8708	-86,3052	-70,5287	-78,0369	-68,8179
45	-16,3246	112,5001	138,0289	152,7642	112,4993	-29,207	-78,4326	-145,799	96,1749	-14,8907	-80,4591	-123,6901	-48,1932	-49,4827	-62,7484
-28,675	-22,4993	155,6256	135	-135	-57,4729	-69,0407	-145,6783	-49,0765	-79,6278	-109,6473	138,671	136,6859	-15,0408	-15,5596	-16,7571
7,2795	-99,7354	-114,4652	-94,9761	-88,2843	-42,2662	-97,8533	-80,1468	-48,1103	-82,4951	-126,6377	-131,3299	-64,2608	-61,5557	36,2797	66,3934
41,7269	86,7269	81,3009	90	95,3606	25,3149	120,4425	-34,7104	-32,8978	-76,4747	-142,7256	-149,7681	-21,9225	-168,4949	111,3875	76,8441
59,6387	84,6392	82,0066	90	93,7535	61,4476	100,5219	-5,0027	-6,3031	-33,9612	-173,6049	-176,117	50,616	117,2082	147,1552	62,5256
-90,7178	-88,5721	-88,2983	-89,6679	-88,7178	-89,3605	-91,9447	-32,3732	-15,9474	-34,9725	-168,7213	-155,918	-84,9898	-88,2583	-92,2243	-90
-90,3723	-89,2555	-89,0694	-89,8139	-89,2555	-89,8139	-91,1167	-63,9655	-40,69	-69,3217	-143,1501	-121,0319	-88,6973	-87,9534	-89,8139	-90
-90,4157	-89,1662	-88,9272	-89,782	-89,1069	-89,7754	-91,3364	-68,9423	-46,7725	-45	-135	-113,3476	-88,2608	-87,1319	-89,7232	-90
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabla 3.32: Dirección del gradiente asociado a los píxeles de un macrobloque de tamaño 16x16 de la figura 3.9

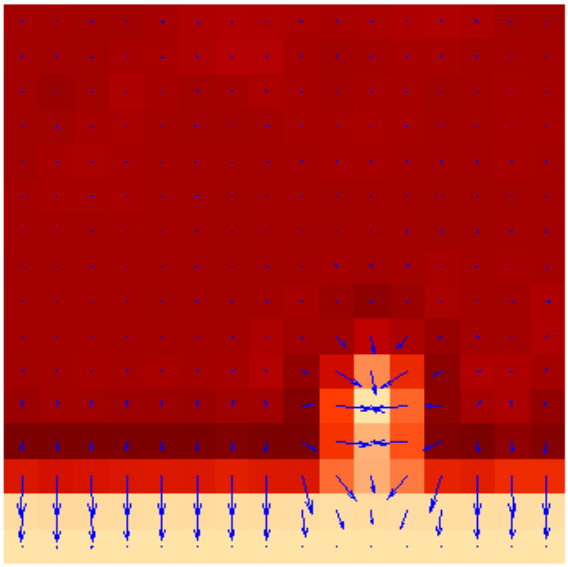


Figura 3.9: Región de una imagen que contiene un borde

El rango de variación de los vectores de gradiente de los píxeles pertenecientes a un mismo borde podría variar. De esta manera, un bin de 30 grados puede contener a los píxeles cuya dirección del gradiente se encuentren dentro de ese intervalo. Para cubrir los 360 grados se necesitan 12 bins, como se muestra a continuación:

Bin del histograma	Intervalo en el que se encuentran los vectores de gradiente del píxel
Bin1	(15,45)º
Bin2	(45, 75)º
Bin3	(75, 105)º
Bin4	(105, 135)º
Bin5	(135, 165)º
Bin6	(165, 195)º
Bin7	(195, 225)º
Bin8	(255, 255)º
Bin9	(255, 285)º
Bin10	(285, 315)º
Bin11	(315, 345)º
Bin12	(345, 15)º

Tabla 3.33: Definición de los intervalos de dirección

3.3.1.4. Selección de variables

A lo largo del desarrollo del algoritmo, se estudiaron las siguientes variables por considerarse de relativa importancia en el desarrollo del algoritmo:

- Mediana del histograma.
- Máximo del histograma.
- Desviación típica del histograma.
- Porcentaje de energía DC del bloque.
- Desviación típica de los píxeles.

Se descartaron algunas variables que no aportaban información adicional a las variables prioritarias o que no eran lo suficientemente importantes para tenerlas en cuenta.

Variables descartadas

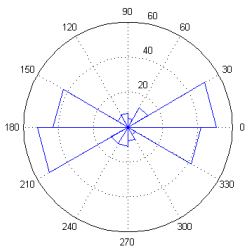
- Mediana del histograma: no era una variable fiable para la discriminación de los dos tipos de bloques. Sólo resultaba útil para casos extremos, es decir, casos donde el histograma es casi plano (bloque "caotic") o con una dirección predominante (bloque "detailed"). Sin embargo, esta distinción nos la aporta las variables media y desviación típica del histograma.
- Máximo del histograma: es de mayor utilidad que la mediana. Puede utilizarse como variable adicional, ya que normalmente alcanza valores altos en bloques "detailed" y valores bajos para bloques "caotic". Se descartó porque existen otras variables, como la desviación típica, que aportan información más relevante.

Variables empleadas

- Media del histograma: es de vital importancia para una primera clasificación del bloque, pero necesita la ayuda de variables adicionales para una segunda clasificación.
- Desviación típica del histograma: es de vital importancia en la clasificación de los bloques. Los bloques "detailed" tienen una mayor desviación típica y los bloques "caotic" tienen una desviación típica más pequeña.
- Porcentaje de energía DC del bloque: aporta información relevante, pero es insuficiente si se trabaja sólo con la media y la desviación típica. Será necesaria la búsqueda de una o varias variables que ayuden en la clasificación.
- Desviación típica de los píxeles: esta variable se calcula como la desviación típica del número de píxeles contenidos en cada bin del histograma. Se utiliza debido a la existencia de bordes poco definidos, bordes con magnitudes de gradiente bajos (un ejemplo de este caso se muestra en la figura 3.10). Señalar que esta variable es insuficiente si se trabaja sólo con la media y la desviación típica. Sin embargo, la combinación de estas cuatro variables da mejores resultados que las anteriores, por lo que finalmente también se añade en el algoritmo.

Figura 3.10: Secuencia “Bus”

"Detailed"



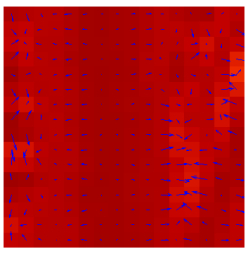


Table 3.34: Gráfica e imagen extraídas de la figura 3.10

La gráfica (histograma de ángulos) de la tabla 3.34 muestra la distribución de los píxeles cuyos vectores de dirección de gradiente se encuentran agrupados en un intervalo de dirección.

Vemos que hay un borde poco definido con magnitudes de gradiente bastante bajos, sin embargo en la gráfica se puede apreciar una gran cantidad de píxeles con direcciones de gradiente apuntando aproximadamente en la dirección 0^{o} y 180^{o}.

3.3.1.5. Experimentos para la justificar la selección de umbrales

Una vez ya conseguidos resultados estadísticos, y realizadas varias pruebas, se lleva a cabo el diseño de un algoritmo que nos permita distinguir entre bloques “caotic” (los histogramas presentan idealmente un aspecto plano) y “detailed” (la forma del histograma presenta idealmente un componente muy destacado respecto a las demás).

Hemos de trabajar con los estadísticos, cuya selección se realizó siguiendo las pautas de la sección anterior, que se pueden extraer del histograma y, teniendo

en cuenta la clasificación que haría un ojo entrenado, establecer una serie de umbrales para clasificar un macrobloque según los estadísticos que consideremos más relevantes. Para ello, se empleó una batería de vídeos que nos ayudó a tal fin.

Figura 3.11: Secuencia “Bridge far”

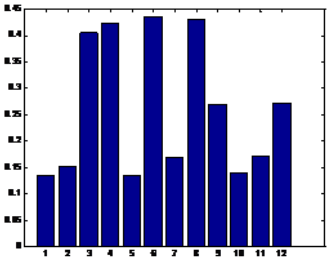
"Caotic"	
	Media bloque = 0.2610 Desviación píxeles = 24.5258 Desviación típica = 0.1287 EDC porcentaje = 93.4762

Table 3.35: Datos extraídos de una región "caotic" de la figura 3.11

Figura 3.12: Secuencia “Coastguard”

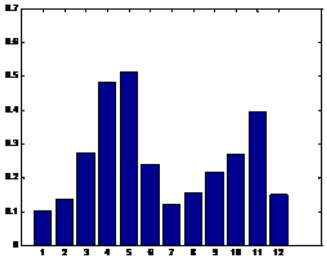
"Caotic"	
	Media bloque = 0.2556 Desviación píxeles = 12.2425 Desviación típica = 0.1402 EDC porcentaje = 97.4889

Table 3.36: Datos extraídos de una región "caotic" de la figura 3.12

Figura 3.13: Secuencia “Stefan”

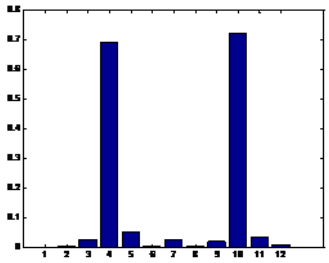
"Detailed"	
	Media bloque = 0.1327 Desviación píxeles = 29.0590 Desviación típica = 0.2678 EDC porcentaje = 98.0645

Table 3.37: Datos extraídos de una región "detailed" de la figura 3.13

Figura 3.14: Secuencia “News”

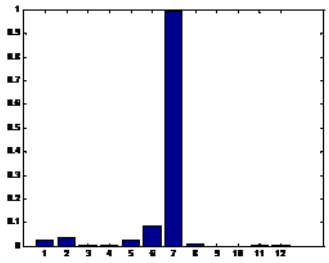
"Detailed"	
	Media bloque = 0.1008 Desviación píxeles = 32.0691 Desviación típica = 0.2825 EDC porcentaje = 56.4908

Table 3.38: Datos extraídos de una región "detailed" de la figura 3.14

Figura 3.15: Secuencia “News”

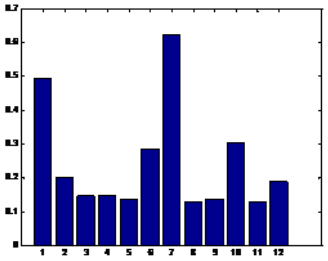
"Caotic"	
	Media bloque = 0.2439 Desviación píxeles = 8.9273 Desviación típica = 0.1612 EDC porcentaje = 71.5270

Table 3.39: Datos extraídos de una región "caotic" de la figura 3.15

Figura 3.16: Secuencia “Paris”

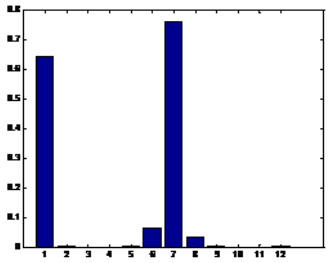
"Detailed"	
	Media bloque = 0.1272 Desviación píxeles = 37.0683 Desviación típica = 0.2707 EDC porcentaje = 89.8992

Table 3.40: Datos extraídos de una región "detailed" de la figura 3.16

Figura 3.17: Secuencia “Paris”

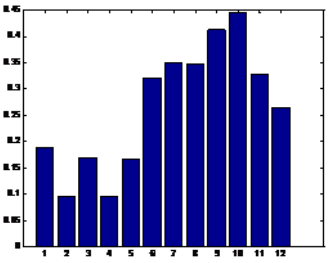
"Caotic"	
	Media bloque = 0.2647 Desviación píxeles = 5.8205 Desviación típica = 0.1204 EDC porcentaje =79.4577

Table 3.41: Datos extraídos de una región "caotic" de la figura 3.17

Figura 3.18: Secuencia “Stefan”

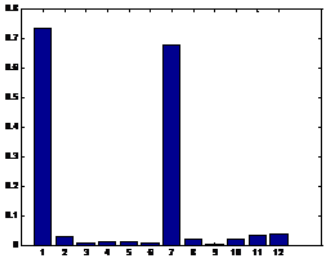
"Detailed"	
	Media bloque = 0.1340 Desviación píxeles = 20.4687 Desviación típica = 0.2671 EDC porcentaje = 95.0215

Table 3.42: Datos extraídos de una región "detailed" de la figura 3.18

Figura 3.19: Secuencia “Waterfall”

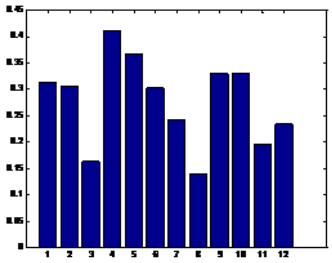
"Caotic"	
	Media bloque = 0.2775 Desviación píxeles = 6.6378 Desviación típica = 0.0830 EDC porcentaje = 97.6723

Table 3.43: Datos extraídos de una región "caotic" de la figura 3.19

3.3.1.6. Desarrollo del algoritmo

En primer lugar, se pasa la imagen a niveles de intensidad y se realiza la clasificación después de dividir la imagen en macrobloques de 16x16. A continuación, se extraen los siguientes estadísticos de cada macrobloque.

- Cálculo de porcentaje de la energía respecto a la energía total del bloque ($EDC_{porcentaje}$). Observamos experimentalmente que los bloques "smooth" presentan unas componentes de porcentaje de energía muy altas.

$$E_{DC/MB} = \frac{\left(\sum_{i,j} \frac{p_{ij}}{Tam_{MB}}\right)^2}{\left(\sum_{i,j} \frac{p_{ij}^2}{Tam_{MB}}\right)} \tag{3.2}$$

  - $\left(\sum_{i,j} \frac{p_{ij}}{Tam_{MB}}\right)^2$: energía coeficiente DC.
  - p_{ij} : píxeles del macrobloque a procesar.
  - Tam_{MB}: tamaño del macrobloque (16x16 píxeles).

También se empleará para clasificar entre "detailed" y "caotic", ya que los bloques caóticos tendrán un valor de porcentaje de la energía DC más alto que los "detailed".

- Media de los histogramas (μ)

  Es una medida muy importante, como puede verse en los experimentos preliminares de la sección anterior.. Se comprueba experimentalmente que los MB que tienen una media baja pueden ser bloques "detailed", aunque también puede ser característico de bloques "normales" (no pertenece a la categoría de bloque "detailed" ni "caotic"). Un valor muy bajo es un indicativo de que se trata de una región "smooth" (la magnitud del gradiente tiende a cero, por ser homogéneos o lisos) o "normal". Sin embargo, esta medida no se utiliza para detectar regiones "smooth", ya que la variable anterior es suficiente para la correcta clasificación de los bloques "smooth".

- Desviación típica de los histogramas (σ)

  La desviación típica se utiliza como complementaria a la media. Se observa experimentalmente que los bloques caóticos tienen una desviación típica baja. Es más propio de los bloques "detailed" tener una desviación típica más alta.

- Desviación de los píxeles ($Desv_{píxeles}$)

  Esta variable se utiliza para detectar bordes poco definidos. Cuando nos encontramos con un borde poco definido, la suma de las magnitudes de los gradientes de los píxeles es un valor bajo. Esta situación lleva a detectar bloques "detailed" como bloques "caotic".

Clasificación de bloques por umbrales Se definen previamente unos umbrales y condiciones que se utilizan para presentar el árbol de decisión del algoritmo. Estos son:

- Condiciones:
  - Condición 1: $(U\_EDC\_1 < EDC_{porcentaje} < U\_EDC\_2)$ & $(Desv_{píxeles} < U\_desv\_p1)$
  - Condición 2: $(Desv_{píxeles} > U\_desv\_p2)$
  - Condición 3:$(EDC_{porcentaje} > U\_EDC\_1)$
  - Condición 4: $(EDC_{porcentaje} < U\_EDC\_2)$
- Umbrales:
  - *U_media*: es el umbral de la variable media del histograma, μ.
  - *U_desv1, U_desv2:* son los umbrales de la desviación típica, σ.
  - *U_EDC_1,U_EDC_2*: son los umbrales del porcentaje del bloque, $EDC_{porcentaje}$.
  - *U_desv_p1, U_desv_p2*: son los umbrales de la desviación típica de los píxeles, $Desv_{píxeles}$.

A continuación se presenta el árbol de decisión y su correspondiente clasificación de los bloques.

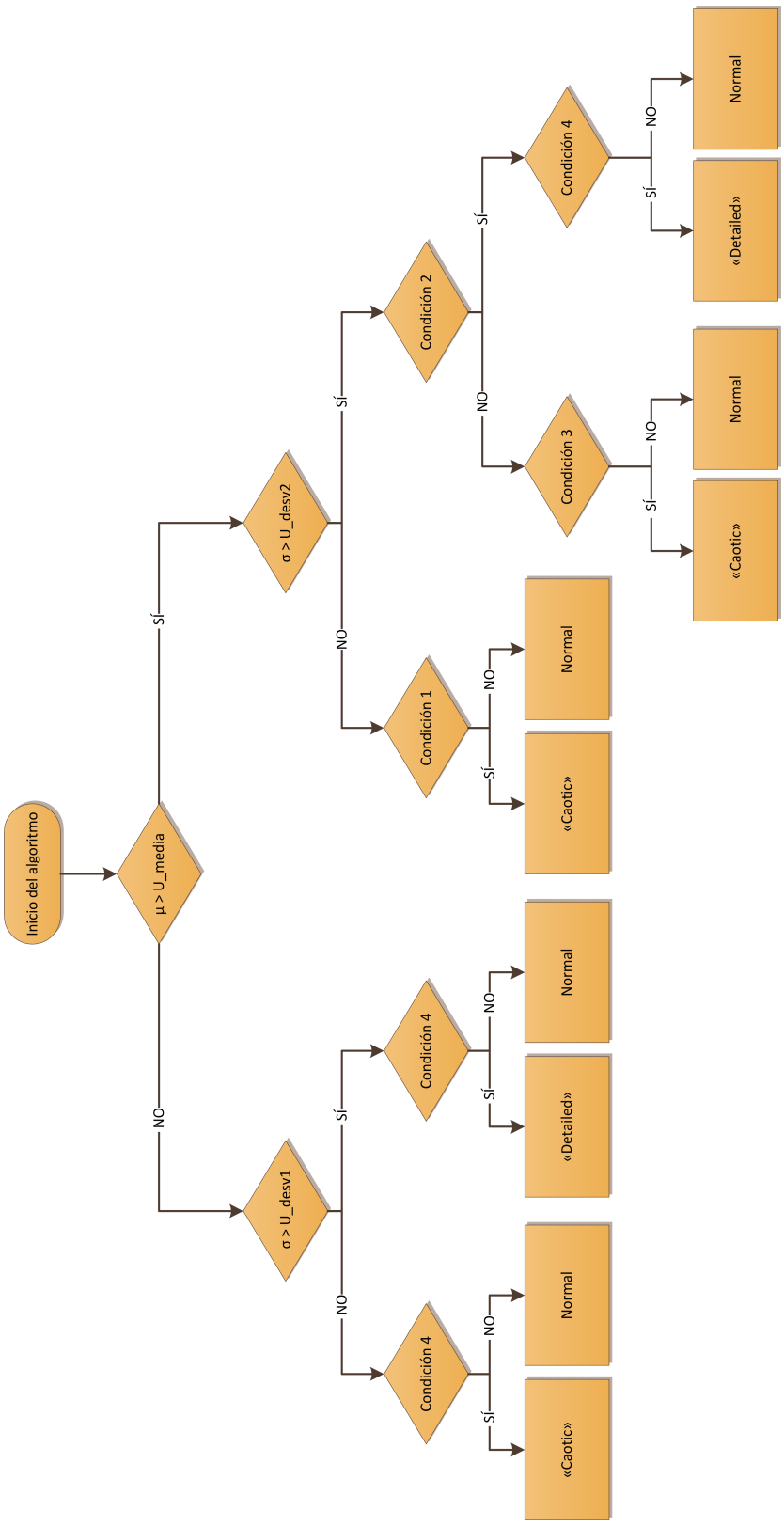


Figura 3.20: Árbol de decisión del algoritmo

Clasificación de los bloques “caotic” y “detailed”:

- Es un bloque "Caotic":

  Si (μ < *U_media*), siempre que
  ($EDC_{porcentaje}$ < *U_EDC_2*) & (σ < *U_desv1*)
  Y Si (μ > *U_media*), y además
  Si (σ < *U_desv2*) debe cumplir

  (*U_EDC_1* < ($EDC_{porcentaje}$ <*U_EDC_2*) & ($Desv_{píxeles}$ < *U_desv_p1*)
  O si σ > *U_desv2* debe cumplir

  ($Desv_{píxeles}$ < *U_desv_p2*) & ($EDC_{porcentaje}$ > *U_EDC_1*)

- Es un bloque "detailed":

  Si (μ > *U_media*), siempre que
  (σ > *U_desv_2*) cumpla que

  ($Desv_{píxeles}$ > *U_desv_p2*) & ($EDC_{porcentaje}$ < *U_EDC_2*)
  o si (μ < *U_media*), siempre que
  Si (σ > *U_desv1*) cumpla
  ($EDC_{porcentaje}$ < *U_EDC_2*)

3.3.1.7. Mejoras introducidas.

Después de aplicar el algoritmo diseñado, se encontraron algunos problemas que dificultaban la correcta clasificación del algoritmo. Por ello, se estudiaron algunas propuestas adicionales al método implementado para mejorar la eficiencia de la clasificación de texturas.

- Zonas en las que existen varios bordes con distintas orientaciones pueden dar lugar a confusión entre bloques “detailed” y “caotic” debido al parecido de sus histogramas. Para solucionar este problema, se añadió otra nueva variable que consistía en lo siguiente:

  - Ordenar los bins de menor a mayor y dividirlos en dos grupos: el primero de ellos formado por los 4 primeros bins más altos y el segundo formado por los bins restantes.
  - Calcular la diferencia entre las medias de los dos grupos. Esta diferencia es alta si se trata de un bloque “detailed” y baja si es un bloque “caotic”.

  Esta variable solucionaba el problema de bloques con varios bordes, sin embargo resultó perjudicial para la correcta clasificación de bloques “detailed” con una sola dirección predominante. Por lo que finalmente fue descartada.

- Zonas con bordes pocos definidos. Se ajusta la igualación de histograma para conseguir una imagen con un histograma de valores de luminancia más uniforme. Esta uniformidad conlleva disponer de una misma frecuencia de aparición en cada uno de los niveles de gris del histograma, ampliando así el rango de luminancia de la imagen y la consiguiente mejora en el contraste. Aunque se consigue un refuerzo del contraste en los bordes, esta etapa puede provocar algunos inconvenientes:
  - Pérdida de la información: los píxeles se distribuyen más ampliamente por todo el rango de valores (de 0 a 255) y en la imagen ecualizada se resaltarán detalles que antes no eran evidentes.
  - Realce de algún error indeseado (bloque caótico): el valor de las magnitudes de los gradientes de bloques caóticos son también reforzados, lo que a su vez, puede provocar un aumento de la media y de la desviación típica.

(a) Imagen original (b) Imagen igualada

Figure 3.21: Comparación de imágenes

- Extremos del plano. El algoritmo en algunos casos detectará bordes en los extremos del plano. Aún siendo precisamente los extremos, bordes, ello puede llegar a falsear las variables con las que se está trabajando. Para solucionar el problema se realiza una extensión de los píxeles que se encuentran en el borde del plano. De esta manera la magnitud de los extremos tendrá un valor cero y no influirá en los datos extraídos.

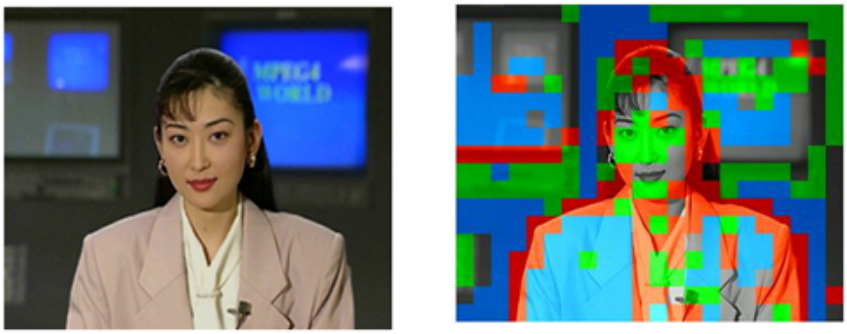
3.3.1.8. Pruebas de umbralización

A continuación se incluyen una serie de capturas de pantalla de las secuencias de vídeo en formato "cif" y "cst", indicando la clasificación realizada por texturas. Es importante comentar que para obtener esta clasificación se tuvieron que realizar algunos estudios y modificaciones adicionales en el algoritmo presentado en este documento.

El código de colores empleado es el siguiente:

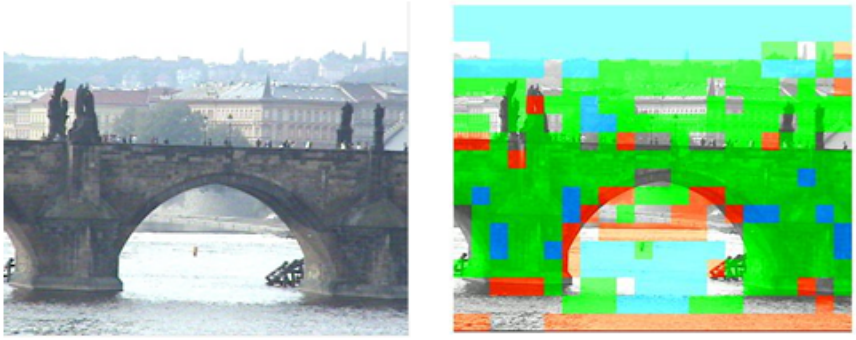
- "Smooth": azul.
- "Detailed": rojo.
- "Caotic": verde.

Formato CIF



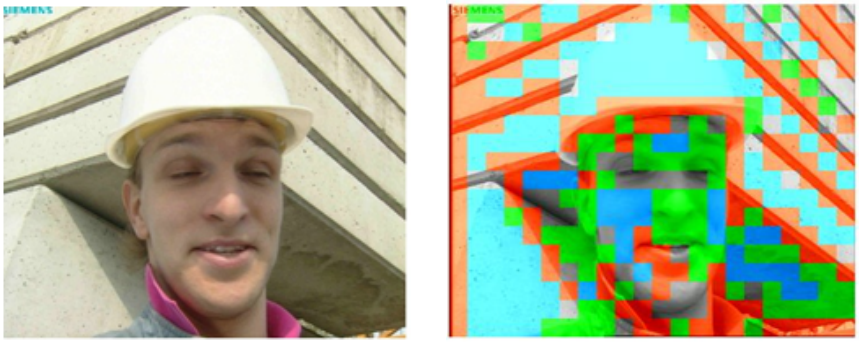
(a) Secuencia original (b) Mapa de enmascaramiento

Figure 3.22: "Akiyo"



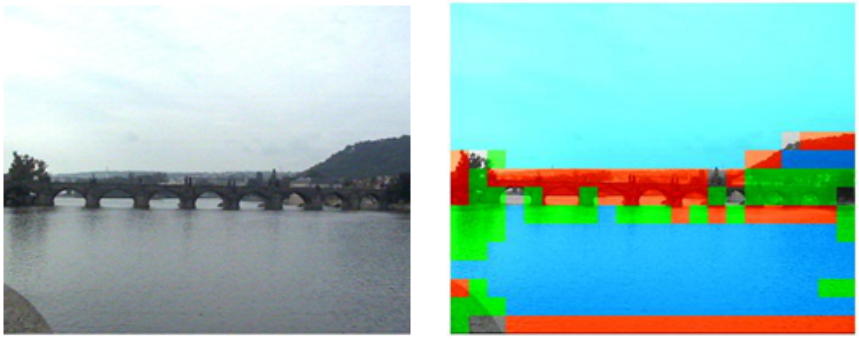
(a) Secuencia original (b) Mapa de enmascaramiento

Figure 3.23: "Bridge close"



(a) Secuencia original (b) Mapa de enmascaramiento

Figure 3.24: "Foreman"

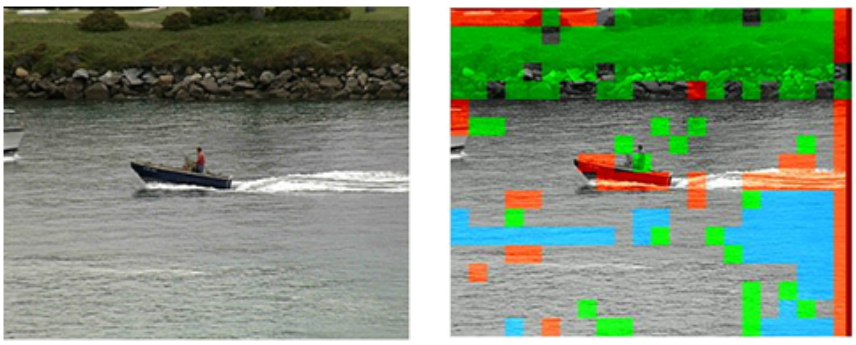


(a) Secuencia original (b) Mapa de enmascaramiento

Figure 3.25: "Bridge far"

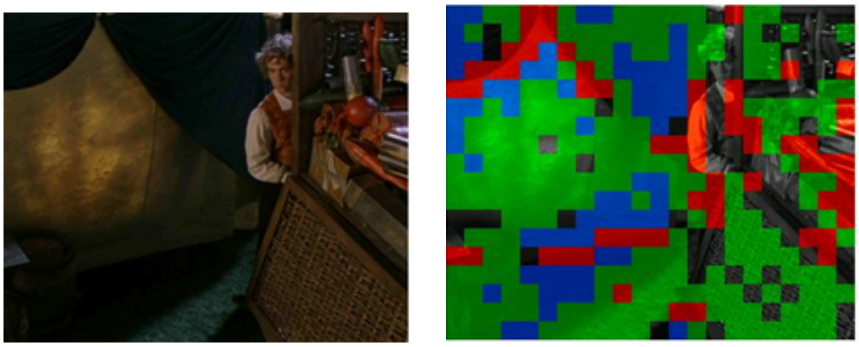
(a) Secuencia original (b) Mapa de enmascaramiento

Figure 3.26: "Bus"



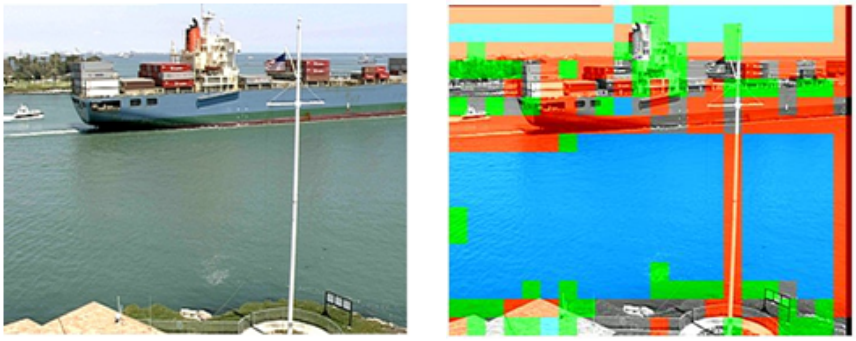
(a) Secuencia original (b) Mapa de enmascaramiento

Figure 3.27: "Coastguard"



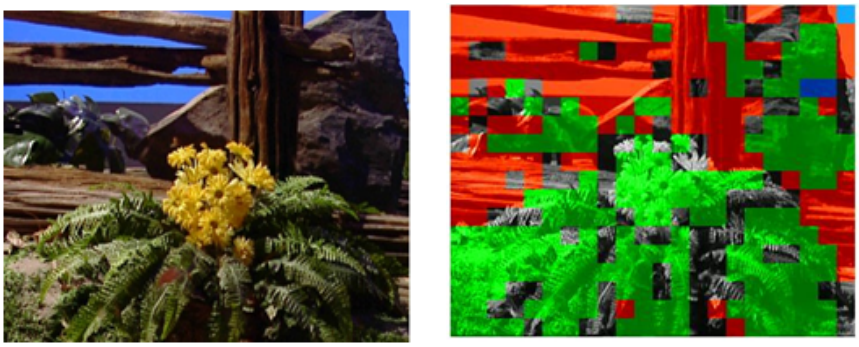
(a) Secuencia original (b) Mapa de enmascaramiento

Figure 3.28: "LOTR1"



(a) Secuencia original (b) Mapa de enmascaramiento

Figure 3.29: "Container"

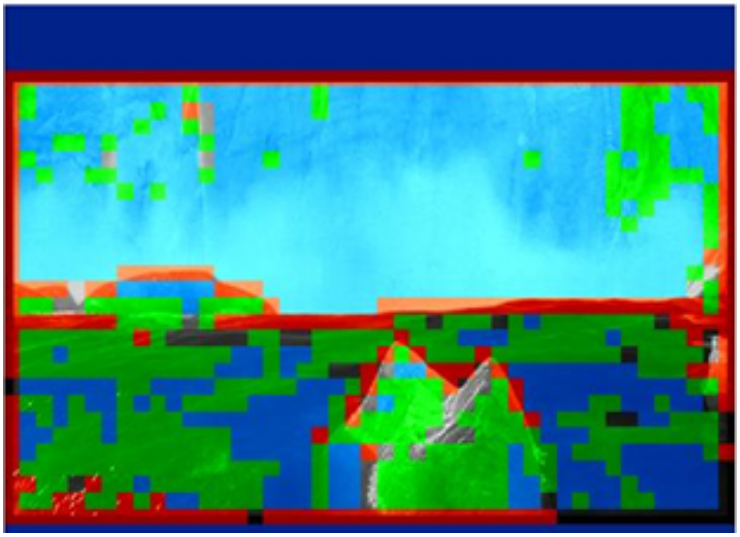


(a) Secuencia original (b) Mapa de enmascaramiento

Figure 3.30: "Tempete"

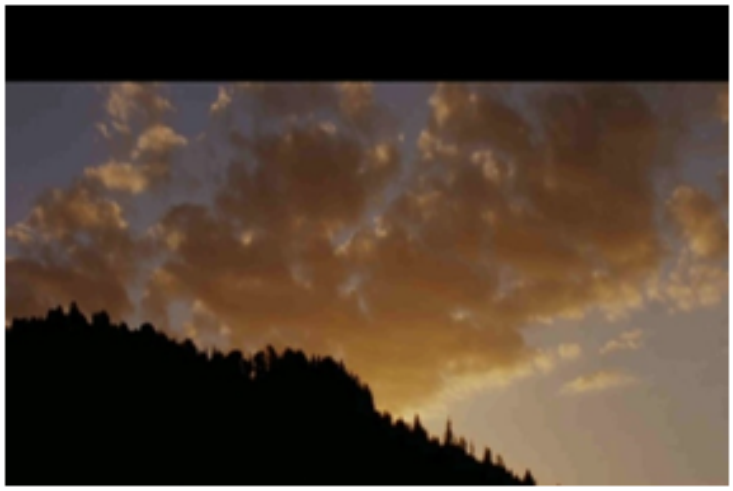
Formato CST

(a) Secuencia original

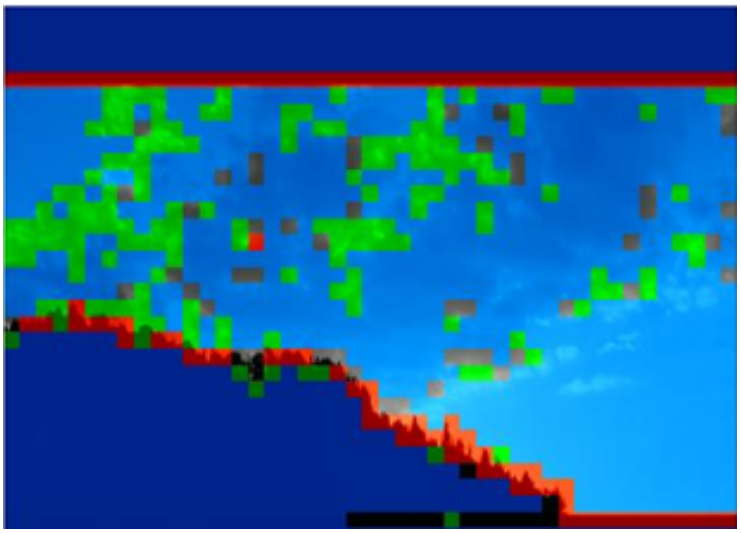


(b) Mapa de enmascaramiento

Figure 3.31: "James Bond"



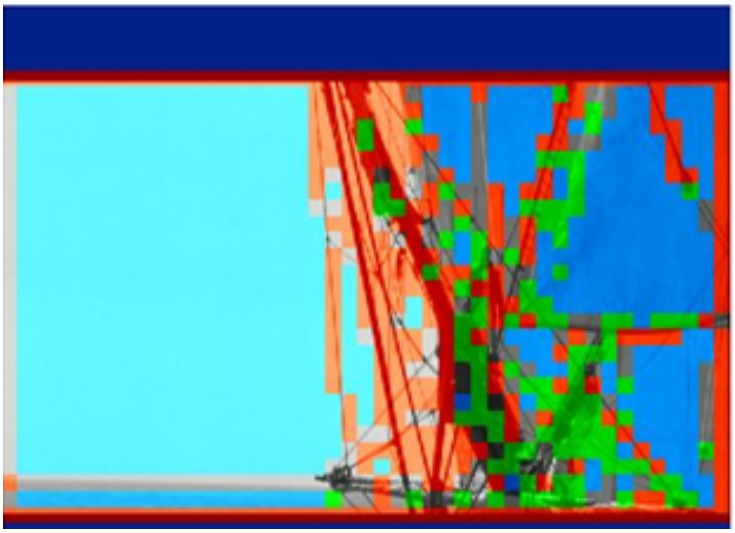
(a) Secuencia original



(b) Mapa de enmascaramiento

Figure 3.32: "Último samurai"

(a) Secuencia original



(b) Mapa de enmascaramiento

Figure 3.33: "Master and Commander"

3.3.2. Clasificación por Redes Neuronales

Una vez se ha concluido la fase de diseño e implementación del algoritmo propuesto, nos encontramos con un clasificador de texturas basado en umbrales que nos provee una salida dura (determina si un macrobloque es "caotic", "detailed" o ninguno de estos dos tipos).

No obstante, dicho mecanismo puede no ser adecuado debido a que ofrece una clasificación que únicamente nos dice el tipo de macrobloque que es, pero no dice el grado de "caotic" o "detailed" del mismo. Es decir, sería mucho más apropiado obtener un clasificador que indicara el grado de enmascarabilidad de cada macrobloque, indicando dicho grado con una salida continua entre -1 y 1.

Por ello, se propone un sistema tal que, ante una entrada, nos proporcione una salida que indique el grado de enmascarabilidad del macrobloque en función de su textura.

El sistema adoptado es una red neuronal que nos proporciona una salida continua para llevar a cabo una codificación de los macrobloques de una imagen en función de su textura. Valores cercanos a -1 son etiquetados como textura caótica y valores cercanos a 1 corresponden a una textura "detailed".

Para tal fin se han de recolectar las variables de entrada de cada macrobloque, que se corresponden exactamente con los estadísticos que se utilizaron para llevar a cabo la clasificación dura:

- σ: desviación típica del histograma.
- μ: media del histograma.
- E_{HF}/E: relación de la energía de alta frecuencia respecto a la total del macrobloque.
- σ píxeles: desviación del número de píxeles contenidos en cada bin del histograma.

Cada MB tendrá un vector de entrada de 4 componentes y una salida, que será el etiquetado del MB ("caotic", "detailed" o neutro). Una vez que hemos recolectado datos de una gran variedad de vídeos de diferente naturaleza, se selecciona el tipo de red neuronal que entrenaremos para llevar a cabo la clasificación.

Basándome en el apartado 2.5.1, se descartan las redes monocapa porque su salida es binaria: (-1,+1) o (0, 1). Nuestras salidas son 3 tipos de MB: "detailed", "caotic" y normal. Por ello, nos vemos obligados a trabajar con un perceptrón multicapa para así poder tener una clasificación acorde con nuestro problema (3 salidas diferentes). Este modelo de red neuronal resuelve de forma eficiente problemas de clasificación y reconocimiento de patrones. Su topología está definida por un conjunto de capas ocultas (en nuestro caso una capa oculta de 12 neuronas), una capa de entrada (4 neuronas de entrada) y una de salida. Siendo la función de activación la sigmoidea, dado que posee la característica de que ella y su derivada son continuas y generalmente funcionan bastante bien.

Después de probar varios esquemas, aquél que obtuvo los mejores resultados fue el siguiente:

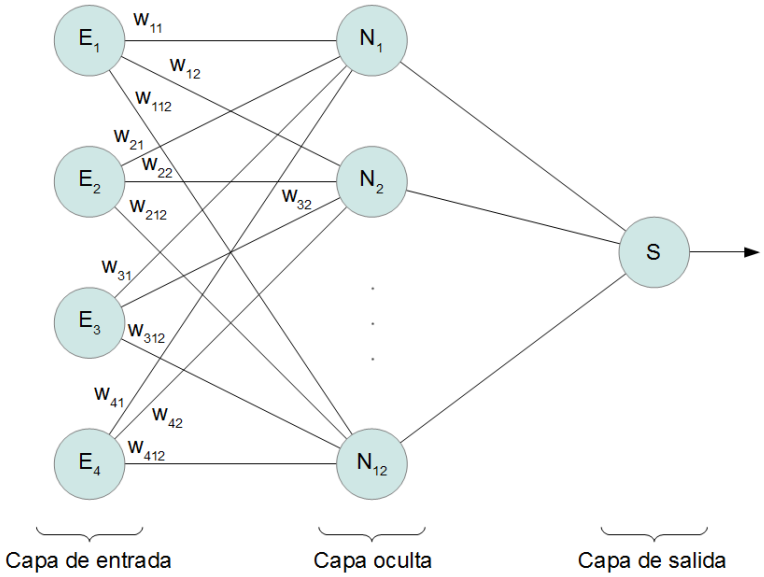


Figure 3.34: 12 neuronas

Una vez establecido el modelo de red neuronal a emplear, se continúa con el entrenamiento de la misma mediante el algoritmo "backpropagation" (retropropagación) hasta la convergencia del mismo. Para que la generalización de la red sea correcta, el orden de los datos de entrada es aleatorizado, y se han de normalizar los datos de entrada restando la media y dividiendo por la desviación típica de cada columna de datos.

En la gráfica 3.35 se muestra el entrenamiento de red para distintos números de neuronas en la capa oculta: 6, 10, 12 y 15 neuronas. Una vez terminado el proceso de aprendizaje y calculados los pesos para cada una de estas versiones de la red neuronal, es importante comprobar la calidad de cada uno de estos modelos. Una medida estándar de error utilizada es RMSE (Root Mean Square Error):

$$Raíz\,del\,error\,cuadrático\,medio\,(RMSE) = \sqrt{\frac{\sum_{k=1}^{n}\left(salida_{deseada} - salida_{red}\right)^2}{n}} \tag{3.3}$$

Los errores de test, de validación y de entrenamiento se encuentran detallados en la tabla 3.44.

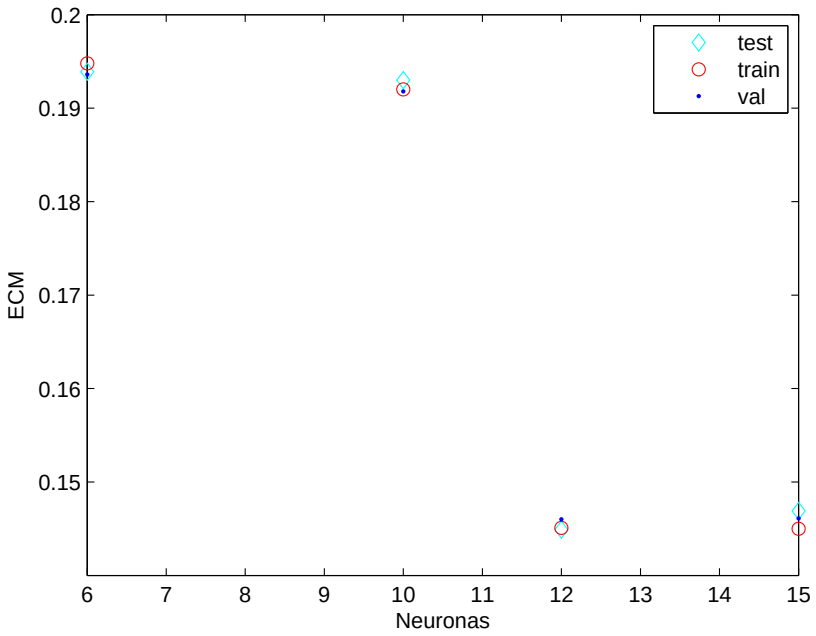


Figure 3.35: Entrenamiento de red para distintos números de neuronas en la capa oculta

La configuración idónea es de 12 neuronas en la capa oculta, siendo el error para esta configuración bastante aceptable. También hay que tener presente que mayor número de neuronas implica un mayor coste computacional del sistema. Por los dos motivos anteriores descartamos las configuraciones restantes.

6 neuronas	10 neuronas	12 neuronas	15 neuronas
error_t = 0.1939 val = 0.1936 train = 0.1948	error_t = 0.1930 val = 0.1918 train = 0.1920	error_t = 0.1449 val = 0.1465 train = 0.1451	error_t = 0.1469 val = 0.1461 train = 0.1453

Tabla 3.44: Errores de entrenamiento, validación y de test

En la figura 3.36 se observa cómo el perceptrón multicapa converge rápidamente para la red neuronal con 12 neuronas en la capa oculta.

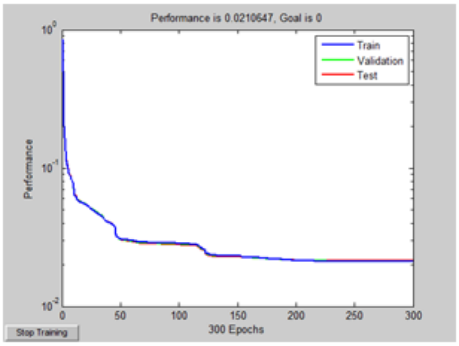


Figure 3.36: Convergencia del sistema.

Al final de todo el proceso, por cada nuevo MB que entre en el perceptrón multicapa, tendremos una salida cuyo valor estará comprendido entre y = -1 e y = +1, denotándose con $y = -1$ un MB con grado máximo de enmascarabilidad ("caotic") y con $y = +1$ un MB con grado mínimo de enmascarabilidad ("detailed"). Un MB con $y = 0$ corresponde a un MB neutro.

3.3.2.1. Post-procesado

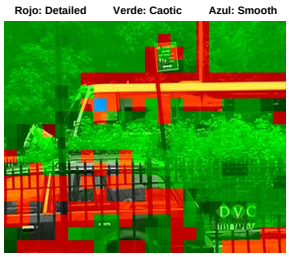
Debido a que el perceptrón multicapa nos provee de una salida continua que indica el grado de enmascarabilidad de cada MB, podemos llevar a cabo un filtrado del mapa de enmascarabilidad conseguido para hacerlo más consistente y suave. Por tanto, se realiza un filtrado de tipo paso bajo a aquellos MB que poseen una enmascarabilidad negativa, bloques claramente "caotic". En las zonas declaradas como degradables conseguimos un mapa más suave, lo que contribuye a que el efecto de degradado sea más agradable para el espectador. Este filtrado no se realiza a los bloques "detailed", ya que un borde puede estar rodeado de bloques neutros o bloques "caotic".

3.3.3. Pruebas Experimentales

Se incluyen algunas pruebas de la clasificación realizada por el algoritmo basado en la DCT y el algoritmo basado en el algoritmo HOG.

3.3.3.1. Formato CIF

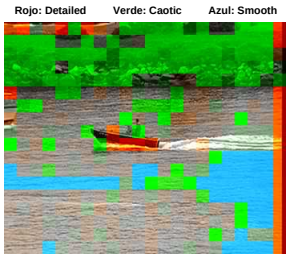
(a) Algoritmo HOG



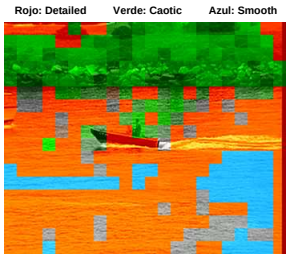
(b) Algoritmo DCT

Figure 3.37: Mapa de enmascaramiento de la secuencia “Bus”

Como se puede observar, la clasificación DCT es mejor que la de HOG. Esto es debido a que regiones como la verja son bordes que no son detectados como "detailed", se tratan de MBs que contienen bordes de más de una dirección, lo que conlleva a una disminución de la desviación típica del MB. Sin embargo, el algoritmo HOG clasifica mejor zonas como los bordes del panel de información.



(a) Algoritmo HOG



(b) Algoritmo DCT

Figure 3.38: Mapa de enmascaramiento de la secuencia “Coastguard”

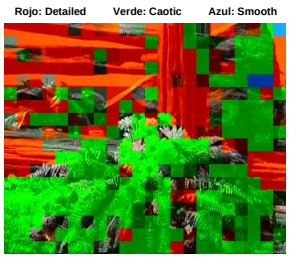
El algoritmo DCT realiza una peor clasificación. Detecta zonas pertenecientes al mar como "detailed", zonas que no deberían ser preservadas.

(a) Algoritmo HOG

(b) Algoritmo DCT

Figure 3.39: Mapa de enmascaramiento de la secuencia “Football”

El algoritmo HOG realiza una mejor clasificación que el algoritmo DCT, ya que realiza un mejor tratamiento de la imagen y, por lo tanto, no degrada regiones consideradas de interés para el observador (jugadores).

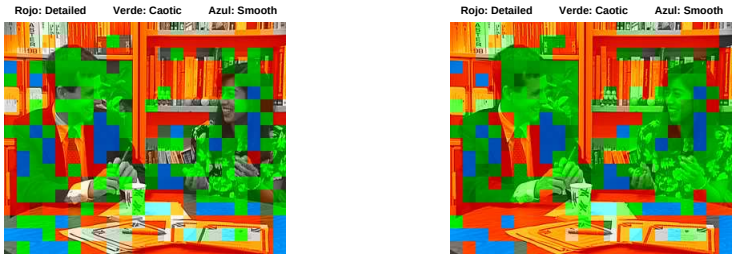


(a) Algoritmo HOG

(b) Algoritmo DCT

Figure 3.40: Mapa de enmascaramiento de la secuencia “Tempete”

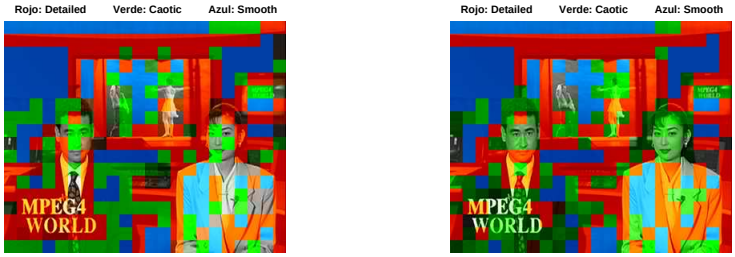
La elección del mejor resultado es más bien subjetiva. El algoritmo detecta zonas de la planta como "detailed", por existir una estructura bastante bien definida. Sin embargo, se podría considerar que dichas zonas son las idóneas para introducir distorsión.



(a) Algoritmo HOG (b) Algoritmo DCT

Figure 3.41: Mapa de enmascaramiento de la secuencia "Paris"

Los dos algoritmos realizan una clasificación similar, sin embargo, el algoritmo HOG preserva mejor las zonas correspondientes a rostros.

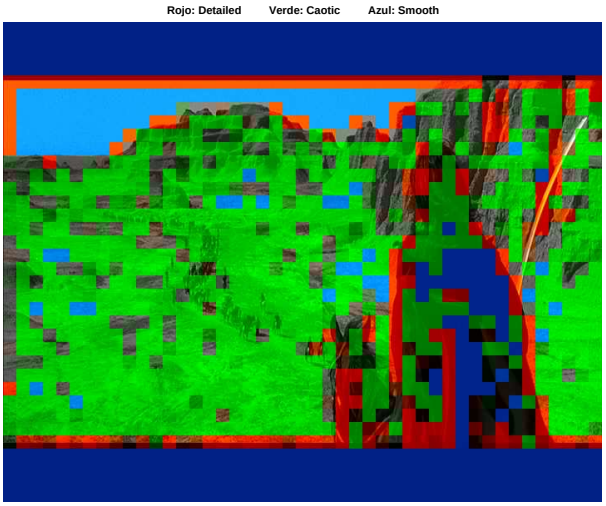


(a) Algoritmo HOG (b) Algoritmo DCT

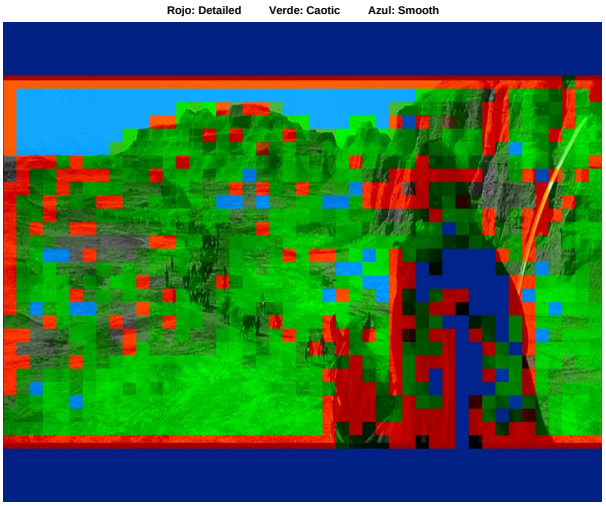
Figure 3.42: Mapa de enmascaramiento de la secuencia "News"

Los dos algoritmos realizan una clasificación similar. La diferencia radica en la mejor preservación de las letras.

3.3.3.2. Formato CST

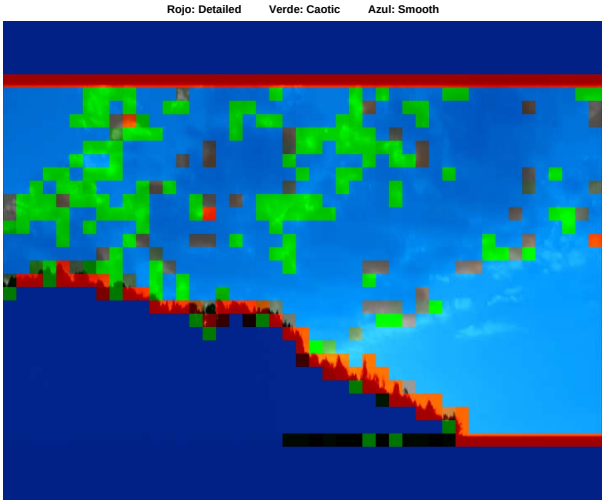


(a) Algoritmo HOG

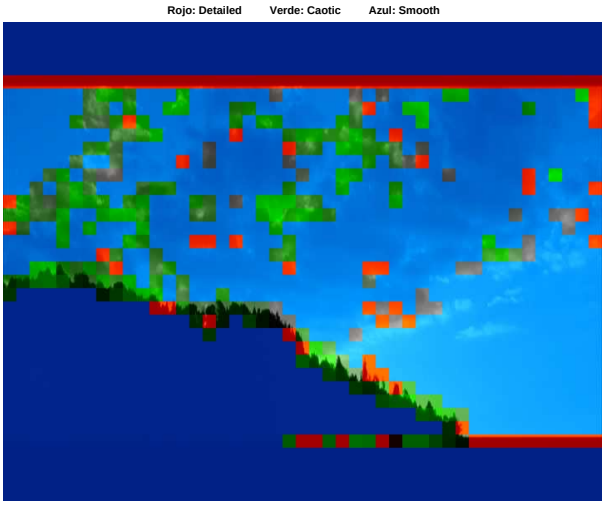


(b) Algoritmo DCT

Figure 3.43: Mapa de enmascaramiento de la secuencia “Tigre y dragón”

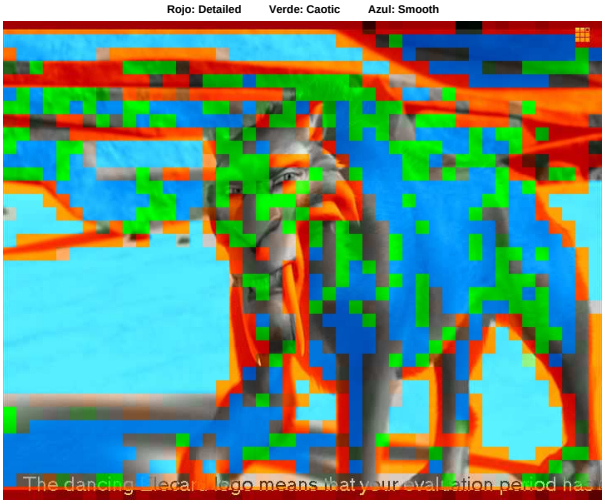


(a) Algoritmo HOG

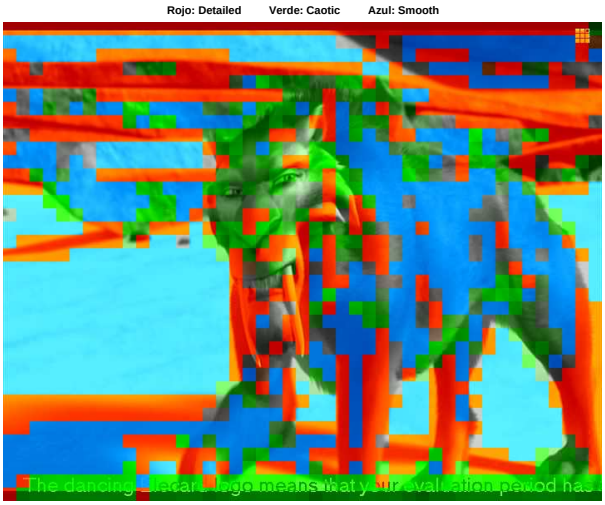


(b) Algoritmo DCT

Figure 3.44: Mapa de enmascaramiento de la secuencia "Último Samurai"



(a) Algoritmo HOG



(b) Algoritmo DCT

Figure 3.45: Mapa de enmascaramiento de la secuencia “Ice Age”

En general, se observa que el algoritmo HOG realiza un mejor tratamiento de aquellas zonas que contienen bordes de diferente curvatura, como pueden ser las letras o zonas específicas del rostro. También reduce el error de clasificación de aquellas zonas clasificadas como bloques “caotic” o normal. Sin embargo, usando la DCT en secuencias como “Coastguard” y “Último Samurai”, las regiones “caotic” son etiquetadas como "detailed". Por otro lado, el algoritmo DCT realiza una mejor clasificación de las regiones que realmente son “caotic” que el algoritmo HOG. Esto se puede apreciar en secuencias como “Tempete” y “Tigre

y dragón", donde el algoritmo HOG realiza el etiquetado de estas regiones como bloques normales.

Capítulo 4

Codificación perceptual por ROI y texturas

4.1. Introducción

Como ya se ha comentado, hay diversas técnicas en el ámbito de la codificación perceptual con el objetivo de reducir la tasa de bits manteniendo una buena calidad visual subjetiva.

En este proyecto nos centraremos en una técnica que analiza el movimiento y la textura de una secuencia de vídeo para conseguir una mejor asignación de los bits. De esta manera se asigna mayor cantidad de bits a zonas que pertenecen a la región de interés a costa de reducir el número de bits de aquellas zonas que no pertenecen a la ROI. Esto es posible debido a que el observador no será capaz de percibir la pérdida de calidad de estas regiones, porque no prestará atención a dichas zonas o porque no será capaz de detectar la distorsión introducida en ellas.

4.2. Enmascaramiento por movimiento y Región de Interés

Para conseguir el enmascaramiento por movimiento y Región de Interés, [41] propone un algoritmo de estimación de movimiento jerárquico que, a su vez, tiene en cuenta el movimiento de cámara (en adelante EMROI), consistente en obtener un mapa de enmascarabilidad que dependerá del grado de movimiento que presente la región y de si esa región pertenece o no a la región de interés. Esta idea se basa en que algunas zonas con movimiento no tienen por qué ser del interés del espectador. Por ejemplo, en el caso de vídeos con movimiento de cámara se puede aplicar distorsión a los MBs que pertenecen al fondo, pues presentan movimiento y no forman parte de la región de interés que conviene preservar; pero, en el caso de vídeos con fondo estático, será considerado de interés todo aquello que presente un movimiento significativo. Teniendo en cuenta lo anterior, se distinguen cuatro regiones en función del grado de movimiento y la región de interés:

- Zona 1: se tratan de regiones que son de interés para el espectador y además presentan poco movimiento, por lo que deben ser protegidas de cualquier distorsión.
- Zona 2: al contrario que en el caso anterior, estas zonas pertenecen al fondo y tienen un movimiento considerable, por lo que a estas zonas se les aplicará toda la distorsión que sea posible introducir.
- Zona 3: estas zonas presentan un grado de movimiento alto y pertenecen a la ROI, por lo que sólo se les introducirá una ligera distorsión.
- Zona 4: esta región presenta poco movimiento y pertenece al fondo, por lo que se puede introducir distorsión.

Para poder detectar estas regiones, el algoritmo EMJ (Estimación de Movimiento Jerárquico) genera un mapa de vectores y lleva a cabo una etapa de estimación y compensación del movimiento de cámara, tomando como base el mapa de vectores anterior, consistente en localizar el movimiento de cámara característico de la secuencia de vídeo y realizar una segmentación de las regiones de interés. Esta etapa consiste en restar los vectores de movimiento de cámara obtenidos a los vectores del algoritmo EMJ, obteniendo, finalmente, un mapa de vectores cuyas magnitudes son evaluadas mediante un umbral β. Por tanto, aquellos MBs con vectores compensados de módulos pequeños se corresponden con el fondo de la imagen, mientras el resto es clasificado como zona de interés.

Un aspecto importante a considerar es que los vectores de movimiento calculados por el algoritmo pueden anularse en zonas "smooth" debido a que se trata de una textura homogénea y, por tanto, los vectores asociados a dichos MBs son nulos y no sirven para estimar el movimiento de la cámara. Para solucionar este problema se opta por eliminar este tipo de bloques durante la estimación del movimiento.

4.3. Enmascaramiento basado en texturas y la ROI

4.3.1. Introducción (EA)

Existen diversos métodos para asignar una mayor tasa de bits a estas zonas durante la codificación. Un método muy estudiado es el de ajustar directamente la QP ("Quantization Parameter") del codificador. Este método consiste en utilizar un valor menor para la QP en regiones que son consideradas de interés o en regiones que son más sensibles a la distorsión. Y por el contrario, utilizar un valor mayor de la QP para las zonas restantes.

La aplicación de este método la podemos apreciar en [3] y [13]. El primero estudia cómo variar la QP del codificador dependiendo del grado de enmascarabilidad de la región. Este grado de enmascarabilidad es obtenido a partir de la combinación de dos índices obtenidos del análisis del movimiento y de la textura del vídeo.

Asimismo, [13] presenta una estrategia "Bit Allocation" basada en la codificación perceptual, que consiste en asignar una cantidad de bits objetivo para los objetos en primer plano, y los bits restantes se asignan al fondo ("background").

A diferencia del anterior, la degradación del fondo se realiza de forma gradual según se aleja del primer plano. Para ello, se calcula el paso de cuantificación a partir de la distorsión del "background" D_{BG} de la siguiente manera:

$$Q_i = \sqrt{\frac{12 N_b D_{BG} \sigma_i}{E \sum_{i=1}^{N_b} \sigma_i}} \tag{4.1}$$

donde:

- N_b: número total de macrobloques que pertenecen al fondo de un plano.
- σ_i: desviación estándar del macrobloque.
- E: relación entre la distorsión verdadera y el modelo de distorsión.

4.3.2. Ajuste del parámetro de cuantificación.

El propósito inicial perseguido en el proyecto desarrollado dentro del Grupo Multimedia era el de dirigir la línea de investigación hacia la elaboración de un mapa de enmascaramiento final basado en la combinación del mapa de enmascaramiento por textura y del obtenido por la región de interés. Finalmente, este proceso fue desestimado porque regiones de interés que contenían textura caótica eran enmascaradas y por tanto degradadas. De modo que el estudio del ajuste de la QP se basa exclusivamente en el mapa de enmascaramiento de la ROI.

El ajuste de la QP se realiza de la siguiente forma:

- Si se trata de regiones enmascarables:

$$QP_{final} = QP + \Delta QP \tag{4.2}$$

- Si se trata de regiones de interés:

$$QP_{final} = QP - \Delta QP \tag{4.3}$$

donde:

- ΔQP: variación máxima de la QP. Siempre es positivo.
- QP: QP global del plano que asigna el "rate control".

Únicamente se degradan aquellas zonas que el algoritmo determine como enmascarables, mientras que las regiones de interés se les asignan valores menores de la QP, consiguiendo así un aumento de la calidad de estas zonas.

Debido a que se aumenta el valor de la QP para aquellas regiones enmascarables, el algoritmo "rate control" se encuentra con un mayor ahorro de bits y por tanto puede bajar la QP y dar un mejor tratamiento al resto de las regiones.

Del estudio realizado con este método se concluyó que a tasas altas conviene emplear un deltaQP más bajo, reservando un deltaQP mayor para tasas más bajas.

Conclusiones Es necesario comentar que la codificación por regiones basada en la diferencia de cuantificación puede acarrear algunos problemas.

- Aunque se han realizado progresos significativos en la segmentación de la imagen, esto continúa siendo un reto.
- Un mismo macrobloque puede contener zonas de la imagen que pertenecen, a la vez, a regiones de interés y a regiones enmascarables.
- El ajuste de la QP puede resultar perjudicial si se realiza un cálculo del enmascaramiento erróneo de algunas regiones de la imagen.
- Esta técnica no es independiente del codificador, por ello si se realizan cambios en el codificador, esta técnica ha de ser adaptada.
- Por último, es necesario comentar que la distorsión de la variación de la QP es incluso más aparente que el "blurring"[1] o el parpadeo de la imagen.

A continuación se muestran algunas imágenes que presentan las distorsiones antes comentadas.

Figura 4.1: Secuencia "Bohemia" codificada con $\Delta QP = 8$ y tasa = 256 kbits

[1] Desenfocado de la imagen.

Figura 4.2: Secuencia "Bohemia" codificada con $\Delta QP = 2$ y tasa = 512 kbits

Como puede observarse, la variación de la QP para valores diferentes de la ΔQP degrada la imagen reconstruida.

Señalar que en el apartado 4.4.6.3 se incluyen unas pruebas de comparación entre las secuencias reconstruidas por la técnica de la ΔQP y las secuencias reconstruidas por la técnica de filtrado.

4.4. Codificación perceptual mediante prefiltrado

Existen otros métodos alternativos a la variación de la QP menos agresivos e independientes del codificador. Estos métodos se dividen en:

- Post-procesado: comúnmente son propuestos para la detección de efectos de bloque y suavizado de bloques en el dominio temporal y frecuencial.
- Pre-procesado: los algoritmos de pre-procesado mejoran la eficiencia de codificación y reducen la probabilidad de degradar la imagen en el proceso de codificación de vídeo eliminando información de alta frecuencia antes de la codificación. Este será el método en el que nos centraremos.

Los motivos para eliminar información de alta frecuencia son:

- En la codificación de una secuencia de vídeo que contiene una gran cantidad de texturas, sobre todo información de alta frecuencia, el codificador de vídeo tiende a asignarles más cantidad de bits. De tal forma que se

reparten menos bits a la ROI (zonas que deberían consumir más bits o ser preservadas).

- Los algoritmos de codificación de vídeo basados en BDCT ("block discrete cosine transform"), como son las familias H.26x y MPEG, sufren efectos de bloque y "ringing" a tasas bajas, ya que los coeficientes de la DCT de alta frecuencia tienden a ser cuantificados a cero debido a parámetros de cuantización altos.

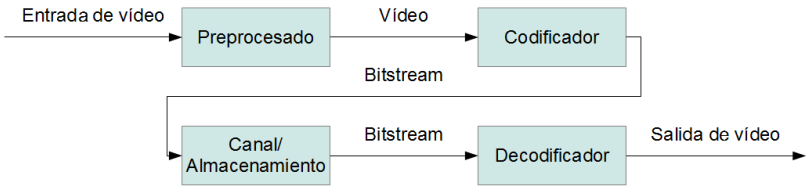


Figura 4.3: Diseño del sistema de pre-procesado de vídeo

4.4.1. Filtrado espacial

Para eliminar los "detalles irrelevantes", que generalmente se encuentran en las altas frecuencias, se implementaron y pusieron a prueba diversas opciones de filtrado. En un principio, para la tarea de eliminar altas frecuencias, parece ser adecuado el filtro gaussiano, pero este filtro tiene dos inconvenientes: el primero de ellos, produce un desenfocado de la imagen (para más información remitirse al apartado 2.4.2) y, el segundo, no preserva la estructura de alto nivel (destruye bordes). Por estos motivos, se estudian los filtros de Gabor (véase apartado 2.4.4). Los resultados de este algoritmo son bastantes satisfactorios, sin embargo, tiene un par de inconvenientes: el coste computacional de filtrar una secuencia, y el no preservar algunas regiones de interés como los rostros (algunas regiones son muy pequeñas y el algoritmo puede confundirlo con "detalles irrelevantes"), esto último supondría utilizar un detector de cara, lo que incrementaría aún más el coste computacional del algoritmo.

Después de los estudios realizados, se decide emplear el filtro bilateral como filtro espacial. Su coste computacional es bastante menor que los de Gabor y, a diferencia del filtro gaussiano, este filtro tiene en cuenta la textura de la imagen y preserva la estructura de alto nivel de la secuencia.

Se recuerda que la máscara final que se aplica a la imagen se obtiene a partir de la multiplicación elemento a elemento de la máscara de proximidad de distancia y la máscara de proximidad de intensidades (para más información véase 2.4.3)

$$M_{bilateral} = M_{distancia} \cdot M_{intensidades} \tag{4.4}$$

donde:

- $M_{bilateral}$: máscara bilateral.

- $M_{distancia}$: máscara del filtro gaussiano.
- $M_{intensidades}$: máscara del filtro de rango.

4.4.1.1. Análisis de las variables

El filtro bilateral dispone de dos parámetros locales (σ_d y σ_r) que regulan el grado de filtrado del píxel en función de la cercanía y la similitud fotométricas de los píxeles vecinos. Aparte de estas variables, hay otro parámetro a tener en cuenta, la ventana, ya que el filtrado no se realiza por bloques sino que se recorre la imagen con un tamaño de ventana previamente definida.

Debido a que el filtrado bilateral cuenta con una serie de parámetros configurables, es necesario determinar estos valores de forma eficaz para conseguir un compromiso entre el coste computacional y la calidad del filtrado. Se efectuaron algunas pruebas de los parámetros del filtro, σ_d y σ_r. Donde σ_d es la desviación típica del filtro gaussiano y σ_r es la desviación típica del filtro de rango, ambos se muestran a continuación:

$$d(x,y) = \exp\left(\frac{-(x^2+y^2)}{2\sigma_d^2}\right) \tag{4.5}$$

- $d(x,y)$: distancia euclídea de los píxeles vecinos al píxel central.
- x: posición horizontal del píxel en la imagen.
- y: posición vertical del píxel en la imagen.

$$r(I_i) = \exp\left(\frac{-(I_i - I_0)^2}{2\sigma_r^2}\right) \tag{4.6}$$

- $r(I_i)$: distancia del nivel de intensidad del píxel vecino i al píxel central i_0.
- I_i :intensidad del píxel i.

Estudio de la variable σ_d En este apartado se pretende analizar el efecto y la importancia de la variable σ_d. Se varía el parámetro a analizar y se fijan los valores de los parámetros restantes, considerando ésta una configuración adecuada para comprobar el efecto de la variable, sin que ello implique un coste computacional importante.

A continuación se describen las pruebas realizadas, y se incluyen algunas capturas de pantalla de la secuencia "París".

La configuración utilizada es:

- Ventana: 5
- σ_d: 1, 5, 15
- σ_r: 5

(a) Secuencia original

(b) Secuencia filtrada con $\sigma_d = 1$

(c) Secuencia filtrada con $\sigma_d = 5$

(d) Secuencia filtrada con $\sigma_d = 15$

Figura 4.4: Filtrado de la secuencia "París" para distintos valores de σ_d

Como puede observarse, las diferencias del filtrado son muy pequeñas, a pesar de la distancia considerable entre los valores de σ_d. Esto se debe a que la influencia de este parámetro está limitada por el tamaño de la ventana, w. Y como no se puede decidir subjetivamente la importancia de esta variable se realizaron algunas pruebas de codificación de la secuencia "París", con distintos valores de σ_d para 50 planos y a QP constante (20). Donde, efectivamente, se comprueba que hay una reducción de tasa para valores de σ_d mayores, por ejemplo, para un valor de σ_d igual a 0.8 se obtiene una tasa de 1300656 bits y para una σ_d igual a 15 se obtiene una tasa de 1283896 bits, siendo esta diferencia de, 16760 bits. Esta reducción de tasa no es muy significativa, por ello elegimos trabajar con un σ_d de valor 3 y así, reducir el grado de filtrado, dado que para valores mayores de esta variable se genera un efecto de desenfoque de la imagen.

Estudio de la variable σ_r De igual manera que en el caso anterior, se realizan unas pruebas para ver el funcionamiento del filtro para distintos valores de la variable σ_r y, asimismo, comprobar el efecto producido para los distintos valores de este parámetro.

La configuración utilizada es:

- Ventana: 5
- σ_d: 3

- σ_r: 3, 10, 20

(a) Secuencia original

(b) Secuencia filtrada con $\sigma_r = 3$

(c) Secuencia filtrada con $\sigma_r = 10$

(d) Secuencia filtrada con $\sigma_r = 20$

Figura 4.5: Filtrado de la secuencia "París" para distintos valores de σ_r

A comparación de la variable σ_d, se aprecia un cambio significativo en los resultados obtenidos, por tanto, se comprueba que σ_r es una variable importante en el resultado final del filtrado de la imagen. Y para comprobar el efecto de reducción de tasa, se realiza algunas pruebas de codificación de la secuencia con distintos valores de σ_r para 50 planos y a QP constante, 20.

Las configuraciones utilizadas en ambas pruebas de codificación son:

Para la prueba de σ_d

- Ventana: 5
- σ_r: 5
- σ_d : 0.8, 15

Para la prueba de σ_r

- Ventana: 5
- σ_d : 3
- σ_r: 0.8, 15

Para σ_r igual 0.8 se obtiene una tasa 1339776 bits y para σ_r igual a 15 se obtiene una tasa de 1138476 bits. En esta última, observamos una reducción de 201300 bits que es mucho más considerable que la anterior. Por ello, decidimos establecer un valor fijo para σ_d y variar el valor de σ_r.

De las pruebas realizadas se obtiene las siguientes conclusiones:

- El efecto visual de cada una de las variables sobre la imagen es diferente, ya que la primera realiza un promediado en función de la distancia y la segunda un promediado en función de la similitud de intensidades del píxel.
- A mayores valores de σ_r, la imagen pierde calidad pero también aumenta en gran medida el ahorro de bits, por lo que se intentará llegar a un compromiso entre la tasa y la PSNR generada.

Estudio de la ventana La ventana también juega un papel importante en el resultado final del filtrado. Un tamaño de ventana mayor implica un aumento de la simplificación de la imagen y también un incremento del coste computacional, por lo que se considera necesario realizar un estudio de esta variable.

La configuración utilizada en las pruebas es:

- Ventana: 1, 5, 10
- σ_d: 3
- σ_r: 15

(a) Secuencia original

(b) Secuencia filtrada con $w = 1$

(c) Secuencia filtrada con $w = 5$

(d) Secuencia filtrada con $w = 10$

Figura 4.6: Filtrado de la secuencia "París" para distintos valores de w

Observamos que esta variable también influye en la intensidad del filtrado. De estos resultados se concluye fijar dos parámetros y variar sólo uno de ellos para un mejor control de filtro.

Una vez realizados los análisis de estas variables del filtro bilateral, podemos concluir que con el uso de este filtro espacial se consigue un ahorro de bits, pero el resultado visual subjetivo no es adecuado debido a la existencia de regiones en movimiento o de rostros no preservados. Esto ocurre porque el objetivo del filtro espacial es eliminar información de alta frecuencia, sin embargo, hay regiones de la imagen que también contienen información de alta frecuencia y que no conviene que sean filtradas, por ser de interés para el observador. Al codificar estas secuencias se aprecia una pérdida de la calidad de la imagen, por ello, es necesario detectar estas regiones de interés para realizar un filtrado selectivo de la imagen.

Para obtener esta información se recurre a los filtros temporales.

4.4.2. Filtrado temporal

Un estudio temporal puede aportar igual o mayor información que un estudio espacial debido a la alta correlación temporal de los planos consecutivos.

Este algoritmo está basado en el filtro "non-local" que calcula el valor final del píxel central a partir del promediado de los píxeles "vecinos" en el dominio

temporal. Este filtrado temporal del píxel i se calcula a partir de la siguiente fórmula:

$$I'_i = \sum_{j \in T} w(i,j) I_j \tag{4.7}$$

donde:

- I'_i: valor de intensidad final del píxel i.
- i: píxel del plano actual.
- I_j: intensidad del píxel j perteneciente a planos cosituados.
- $w(i,j)$: peso de similitud ente el pixel i y j.
- T: número de regiones. Cada región tiene su posición central en el píxel i, en planos cosituados.

Donde los pesos del análisis temporal se calculan como:

$$w(i,j) = \frac{1}{Z(i)} \exp\left(-\frac{||v(N_j) - v(N_i)||}{h^2}\right) \tag{4.8}$$

donde:

- $v(N_i)$: representa vectores de intensidad de la región N_i del plano actual.
- $v(N_j)$: representa vectores de intensidad de la región N_j (con la misma posición que la región N_i) perteneciente a planos cosituados y con centro en el píxel j.
- h: regula el grado de filtrado.
- $Z(i)$: se utiliza para normalizar los pesos. Para más información véase el apartado 2.4.1.1.

El filtrado temporal se puede considerar como una medida de similitud entre dos regiones centradas en el mismo píxel en distintos instantes de tiempo. Es decir, el filtrado del píxel dependerá de la similitud de las intensidades de ambas regiones. Píxeles que pertenecen a regiones estáticas tendrán un peso mayor y píxeles que pertenecen a regiones en movimiento tendrán un peso menor sobre el valor final de la intensidad del píxel actual.

Para realizar el filtrado de un plano se tienen en cuenta planos posteriores y/o anteriores al "frame" actual (planos cosituados), pero no se tiene en cuenta el plano actual. De esta manera se reduce el coste computacional, dado que la comparación de un plano consigo mismo aporta un peso de valor 1 (ver ecuación 4.8). Señalar que los pesos de similitud se encuentran en el rango [0 - 1].

Se realizaron experimentos de codificación a QP fija y se obtuvieron las siguientes conclusiones:

- Reducción de la tasa a costa de una reducción de la PSNR de la imagen. Siendo esta reducción poco significativa y proporcional al número de planos utilizados.

- Un aumento del número de planos implica un incremento del coste computacional.

Por estos motivos, nuestro estudio se orienta a los filtros tridimensionales, que tienen en cuenta la información temporal y la espacial.

4.4.3. Filtrado tridimensional

Para la implementación de los filtros tridimensionales se elige el filtro bilateral como el filtro espacial porque se trata de un filtro que tiene en cuenta la textura de la imagen.

La metodología de este filtro tridimensional (en adelante trilateral) consiste en realizar un filtrado secuencial, es decir, aplicar un filtrado bilateral seguido del filtrado temporal, ambos aplicados independientemente, del píxel actual. Para este cálculo se utiliza la siguiente ecuación:

$$I_{i,t} = \sum_{T=-N}^{N} w(i_t, i_{(t+T)}).I'_{i,(t+T)} \tag{4.9}$$

- $I_{i,t}$: valor final del píxel i en el instante t después de realizar un filtrado trilateral.
- $I'_{i,(t+T)}$: píxel filtrado espacialmente en el instante $(t+T)$.
- w: corresponde a los pesos calculados a partir del filtrado temporal (ver apartado 4.4.2).
- N: número de planos que se utilizan en el filtrado temporal.

Resumiendo, el filtro trilateral consiste en calcular una serie de pesos mediante la comparación de regiones cosituadas en planos anteriores y posteriores al actual (análisis temporal). Una vez calculados estos pesos, se realiza un filtrado temporal sobre los píxeles (pertenecientes a planos cosituados) filtrados espacialmente.

De las pruebas de codificación que se realizaron de las secuencias a QP constante, se comprueba que este filtro consigue preservar, en la medida de lo posible, la textura y movimientos significativos presentes en la imagen. También se observa una reducción de la tasa de bits, pero a costa de disminuir la PSNR y la calidad de la secuencia. Hay que señalar que al igual que los filtros temporales, el coste computacional de este filtro aumenta con el número de planos que se utilizan en el filtrado.

4.4.4. Filtro con consideraciones temporales

Para mejorar la detección de regiones en movimiento se decide realizar un filtrado con consideraciones temporales que consiste en variar la intensidad del filtrado espacial a partir del control de sus parámetros locales. Por tanto, este filtro decide qué regiones deben filtrarse espacialmente más o menos a partir de medidas temporales. De manera que si se detecta movimiento se aplicará poco o nada de filtrado y si no lo detecta aplicará un filtrado más agresivo.

Para conseguirlo, se propuso modificar directamente los valores de la matriz de pesos del filtrado bilateral (ver ecuación 4.4) a partir de los pesos de una matriz temporal de la siguiente manera:

$$M_{filtro-consideraciones} = M_{bilateral}.W_{consideraciones-temporal} \tag{4.10}$$

siendo:

- $M_{filtro-consideraciones}$: corresponde a los pesos del filtro bilateral con consideraciones temporales.
- Pesos del filtrado bilateral

$$M_{bilateral} = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{00} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} \tag{4.11}$$

- Pesos con consideraciones temporales

$$W_{consideraciones-temporal} = \begin{pmatrix} w & w & w \\ w & 1 & w \\ w & w & w \end{pmatrix} \tag{4.12}$$

donde:

- b_{ii}: peso del píxel vecino ii al píxel central b_{00}.
- w: peso con consideraciones temporales, que varían en el rango [0 - 1]. Estos pesos temporales se calculan a partir de la ecuación 4.13.

Los pesos del filtro con consideraciones temporales afectan únicamente a los píxeles vecinos del píxel central, pero no modifica el peso que le corresponde al mismo. De modo que, en función del grado de movimiento, w tendrá valores cercanos a uno si se trata de regiones estáticas, y tendrá valores cercanos a cero si se trata de una región en movimiento. De esta manera no se filtra regiones que presenten movimientos significativos.

El cálculo de los pesos se realiza de la manera siguiente:

$$w = \exp^{\frac{-(1-d_{norm})^2}{\sigma}} \tag{4.13}$$

donde:

- d_{norm}: es una distancia normalizada obtenida a partir del análisis temporal. Si la región es estática su valor será uno y si la región está en movimiento su valor será cercano a cero.
- σ: valor experimental.

El valor de d_{norm} se obtiene a partir de esta ecuación:

$$d_{norm} = \frac{d_{mínimo}}{d} \tag{4.14}$$

Siendo:

$$d = \sum_i \frac{w_0}{w_i} \tag{4.15}$$

donde:

- d_{minimo} :valor mínimo de d, que coincide con el número de "frames" cosituados al "frame" actual. Éstos se utilizan para obtener los pesos con consideraciones temporales (ecuación 4.13).
- w_i: pesos de los píxeles pertenecientes a los "frames" cosituados, obtenidos del análisis temporal (ver ecuación 4.8). Estos pesos son siempre igual o menores que el peso del "frame" actual, w_0.

Hay que tener en cuenta que este método modifica los pesos del filtrado espacial de forma brusca, lo que puede ser muy perjudicial si no se regula adecuadamente el parámetro auxiliar σ de la ecuación 4.13. Para ello, se realiza una modificación de esta versión que consiste en modificar un parámetro del filtro espacial que regule el grado de filtrado.

Esta variable, σ'_r, modifica indirectamente la matriz de pesos del filtrado bilateral, por lo que a partir de esta versión se podría obtener la anterior, con la ventaja de que este filtrado es menos agresivo.

$$\sigma'_r = \sigma_r \exp^{\left(-\frac{(1-d_{norm})}{\alpha}\right)^2} \tag{4.16}$$

donde:

- d_{norm}: distancia normalizada obtenida de medidas temporales. Si la región es estática su valor será uno y si la región está en movimiento su valor será cercano a cero.
- α: valor experimental.
- σ_r: desviación típica del filtro de rango, (véase ecuación 4.6).
- σ'_r: es una versión modificada de la desviación típica del filtro de rango.

Con la implementación de esta última versión se obtuvieron buenos resultados para secuencias como "París", "News" y "Football", en los que se consigue preservar el movimiento de los objetos de interés. El inconveniente de este filtro es que preserva todo tipo de movimiento, lo cual no es adecuado, sobre todo en secuencias con movimiento de cámara, donde se aprecia un movimiento significativo del fondo.

Se muestra una captura de pantalla de la secuencia "Bus" para un valor de σ_r igual a 20.

Figura 4.7: Filtrado con consideraciones temporales

Como ya se comentó, se considera ROI aquellas zonas que presentan movimiento, pero hay que tener presente que aquellas zonas que presentan movimiento podrían no pertenecer a la ROI. Así, en el caso de vídeos con movimiento de cámara, ésta sigue la trayectoria del objeto (u objetos) de interés, donde el objeto de interés presenta movimiento nulo respecto a la cámara. Sin embargo, el fondo presenta movimiento respecto a la cámara, por lo que en este caso es importante filtrar zonas que presentan movimiento (fondo) y por el contrario preservar aquello que no lo presente.

En el caso de vídeos con fondo estático o sin movimiento de cámara, será considerado de interés todo aquello que presente un movimiento significativo, por tanto, no es conveniente introducir distorsión en zonas con movimiento respecto a la cámara, porque es precisamente en ellas donde el usuario centra su atención. Como consecuencia, surge la necesidad de conocer el movimiento de la cámara y utilizarlo en nuestra clasificación para que se encargue de determinar qué zonas son de interés y cuáles no (se asume que objetos, cuyo movimiento difiera del que realiza la cámara, se consideran relevantes o importantes desde el punto de vista subjetivo.), para poder asignar un nivel de distorsión adecuado a cada región sin que implique una reducción de la calidad de la secuencia.

El objetivo es detectar qué regiones son de interés, y, para conseguirlo, se incluye en el algoritmo un proceso de detección del movimiento de la cámara para relativizar el movimiento percibido con respecto al de cámara y averiguar qué agente causa la sensación de movimiento de la escena.

Se propone a atacar este problema por dos vías. Una de ellas es utilizar un mapa de enmascarabilidad (véase apartado 4.2), que indique la intensidad de filtrado y la otra es detectar el movimiento de cámara y aplicarlo a esta última versión para que el parámetro σ'_r se calcule, a partir de la comparación con la región cosituada, mediante la observación de aquella región que corresponda una vez compensado el movimiento de cámara, con la idea de detectar qué regiones se mueven por sí mismas de manera no coherente con la cámara.

4.4.5. Filtrado espacial a partir de la estimación de movimiento jerárquico

Este tipo de filtrado se basa en la idea de un filtrado espacial con consideraciones temporales que consiste en controlar la intensidad del filtrado a partir de un mapa de enmascarabilidad, el cual se obtiene a partir de la Estimación de Movimiento Jerárquico y que ya tiene compensado el posible movimiento de cámara de las secuencias. Con este mapa se pretende ajustar el /los parámetros locales del filtro bilateral, de manera que a partir de dicho mapa se lleve un filtrado más agresivo a las regiones que no son de interés y, por el contrario, no filtrar (o poco filtrado) aquellas regiones que pertenecen a la ROI.

Por tanto, se propone realizar un filtrado selectivo con el objetivo de conseguir una reducción de la tasa binaria generada tras la codificación, sin que ello implique una disminución de la calidad de la secuencia.

4.4.5.1. Funcionamiento

Para regular el grado de intensidad del filtrado se modifica la variable σ_r a partir de la variable M, índice de enmascarabilidad que varía en el rango [0 - 1]. Donde valores del índice de enmascarabilidad del MB cercanos a "0" significa que dicho MB no es enmascarado y valores cercanos a "1" indica un mayor grado de enmascaramiento del MB.

La variable σ_r es ajustada por el índice de enmascarabilidad en la ecuación 4.17:

$$\sigma'_r = \left(\sigma_r . M^2 + \varepsilon\right) \tag{4.17}$$

donde:

ε: se añade para evitar que σ'_r sea cero y, así, evitar la indeterminación de la ecuación 4.6.

σ_r: desviación típica del filtro de rango que representa aproximadamente el máximo filtrado bilateral que se puede realizar.

A continuación se presenta el diagrama de bloques del algoritmo.

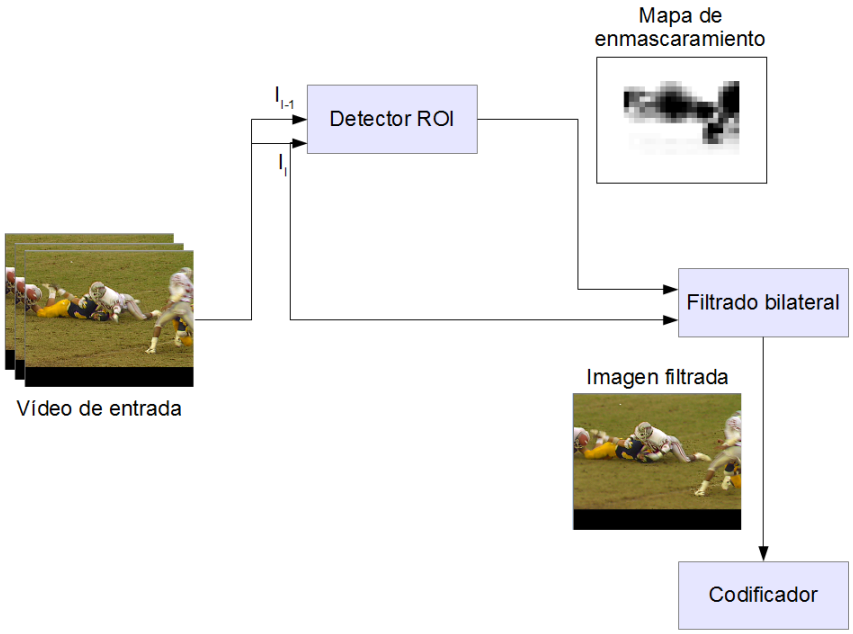


Figura 4.8: Diagrama de bloques basado en el algoritmo EMROI

Vemos en el diagrama de bloques que la imagen se filtra a partir de un mapa de enmascaramiento obtenido del algoritmo EMROI (ver apartado 4.2). Este mapa de enmascaramiento nos indica cuánto una región de la imagen debe ser filtrada.

4.4.5.2. Pruebas experimentales

Estas pruebas experimentales se dividen en dos. La primera de ellas pretende comprobar el efecto del mapa de enmascarabilidad sobre el filtro y la segunda pretende comprobar subjetivamente el funcionamiento adecuado del algoritmo.

Pruebas de umbralización Una vez estudiadas las distintas variables del filtro espacial, se realizan pruebas que muestran la eficacia del filtro bilateral con consideraciones perceptuales. Estas pruebas consisten en unos mapas de colores que indican la intensidad del filtrado que debe aplicarse a la secuencia. Se eligen secuencias, con y sin movimiento de cámara, que pueden ser las más representativas del funcionamiento del algoritmo.

El parámetro local del filtro bilateral adaptado, σ'_r varía en el rango [0 - σ_r]. Por tanto, colores más cercanos al color rojo indican valores pequeños de la variable σ'_r , o lo que es lo mismo, regiones poco filtradas. Y por el contrario, el color verde indica que la variable σ'_r toma valores más cercanos a la σ_r, y por consiguiente, estas regiones se filtran más intensamente. Otros colores indican un grado de filtrado intermedio.

Las secuencias sobre las que se realizan las pruebas son: “Foreman”,” Football”,” Bus”,” Bridge far”.

Foreman

Figura 4.9: Mapa de colores que indica el grado de filtrado de la secuencia "Foreman"

Bus

Figura 4.10: Mapa de colores que indica el grado de filtrado de la secuencia "Bus"

Football

Figura 4.11: Mapa de colores que indica el grado de filtrado de la secuencia "Football"

Bridge far

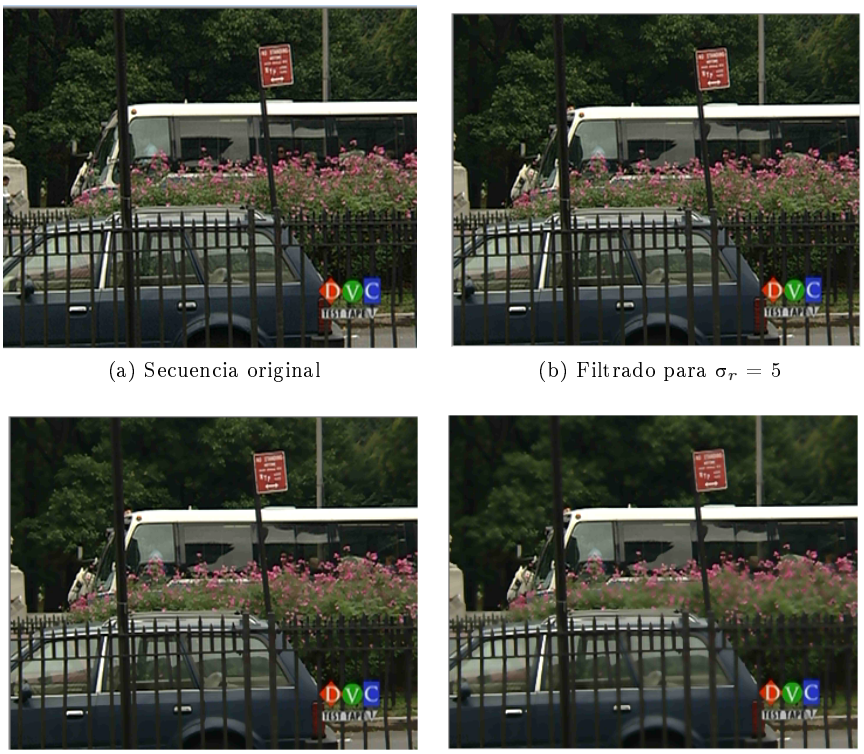
Figura 4.12: Mapa de colores que indica el grado de filtrado de la secuencia "Bridge far"

Pruebas de filtrado Pruebas realizadas para distintos valores de σ_r. El objetivo de esta prueba es comprobar el funcionamiento correcto del algoritmo.

La configuración utilizada es:

- σ_d: 3
- σ_r: 5, 15, 35
- Ventana: 5

Bus



(a) Secuencia original

(b) Filtrado para $\sigma_r = 5$

(c) Filtrado para $\sigma_r = 15$

(d) Filtrado para $\sigma_r = 35$

Figura 4.13: Filtrado basado en el algoritmo EMROI de la secuencia "Bus"

Football

(a) Secuencia original

(b) Filtrado para $\sigma_r = 5$

(c) Filtrado para $\sigma_r = 15$

(d) Filtrado para $\sigma_r = 35$

Figura 4.14: Filtrado basado en el algoritmo EMROI de la secuencia "Football"

Foreman

(a) Secuencia original

(b) Filtrado para $\sigma_r = 5$

(c) Filtrado para $\sigma_r = 15$

(d) Filtrado para $\sigma_r = 35$

Figura 4.15: Filtrado basado en el algoritmo EMROI de la secuencia "Foreman"

Bridge far

(a) Secuencia original

(b) Filtrado para $\sigma_r = 5$

(c) Filtrado para $\sigma_r = 15$

(d) Filtrado para $\sigma_r = 35$

Figura 4.16: Filtrado basado en el algoritmo EMROI de la secuencia "Bridge far"

Con este método de filtrado se soluciona el problema que se tenía con el filtro con consideraciones temporales (ver apartado 4.4.4). Por ejemplo, las secuencias como "Foreman" y "Bus" presentan movimientos de cámara tanto en la vertical como en la horizontal. En estas secuencias se filtra el fondo y se preserva la región de interés, Foreman y el autobús, sin embargo, con la versión antigua del filtro con consideraciones temporales se preservaba toda la imagen en secuencias con movimiento de cámara o incluso se llegaba a filtrar el objeto de interés debido a que no tenía movimiento en relación a la cámara..

Por otro lado, en la secuencia "Football" se consiguen resultados similares al filtro con consideraciones temporales, ya que hay movimiento sólo por parte de los jugadores, los cuales pertenecen a la región de interés.

En el caso de "Bridge far" se trata de una secuencia estática donde toda la imagen es filtrada, incluso el puente, siendo éste objeto de interés. Esto se debe a que el mapa de enmascaramiento tiene valores muy altos para estas regiones, lo que conlleva a un filtrado más intenso. A pesar de ello, el filtrado que se realiza no es muy perjudicial, ya que la distorsión introducida es difícil de ser detectada porque se trabaja con el filtro bilateral que se encarga de preservar la

estructura de alto nivel. No obstante, para filtrados más agresivos si se puede apreciar la pérdida de calidad de esta secuencia.

En conclusión, con este nuevo algoritmo se consigue preservar la ROI y obtener una reducción de la tasa de la secuencia codificada. Esto último no se podía conseguir en secuencias con movimiento de cámara si se aplica únicamente el filtro con consideraciones temporales.

4.4.5.3. Conclusiones

El algoritmo de filtrado basado en la EMJ con estimación de movimiento de cámara generalmente funciona bien, pero se decide buscar otro método alternativo por los siguientes motivos:

- Los índices de enmascaramiento se obtienen por bloques de tamaño 16x16, lo cual es un problema, ya que un macrobloque puede contener regiones que son y no de interés. Esto ocurre en "Football", donde algunos macrobloques contienen zonas del césped y partes del cuerpo del jugador.
- Este proceso exige un alto coste computacional, ya que el generar la Estimación de Movimiento Jerárquico consume muchos recursos.
- Al eliminar bloques "smooth" del algoritmo, se corre el peligro de realizar una estimación del movimiento jerárquica errónea debido a que se ignoran bloques que pueden entrenar el modelo adecuadamente. Se muestra un ejemplo del mapa de enmascaramiento de la secuencia "París", donde el índice de enmascarabilidad tiene valores altos para algunas regiones de interés, como el rostro del presentador, estos valores altos del índice conllevaría a enmascarar zonas que, en principio, deberían ser preservadas.

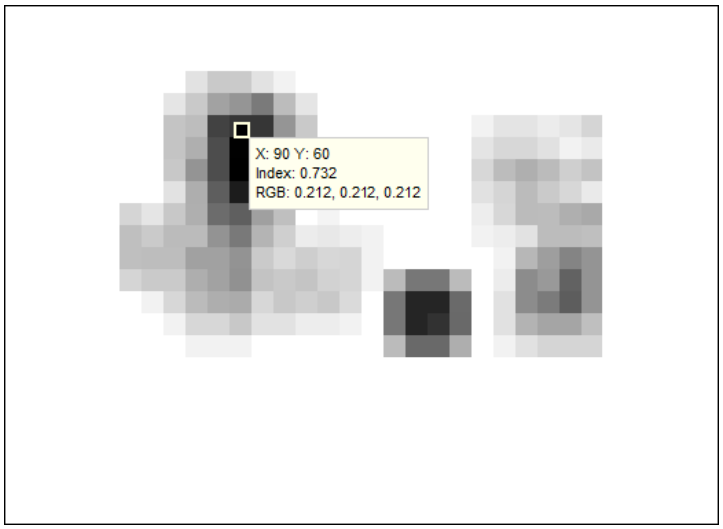


Figura 4.17: Enmascarabilidad por EMROI de la secuencia "París"

- Por otro lado, se preservan algunas regiones que deberían ser enmascaradas. Por ejemplo, en la secuencia "Coastguard", las regiones que perte-

necen al fondo, la zona de hierba, tienen valores muy bajos del índice de enmascarabilidad, es decir, estas regiones no son enmascaradas.

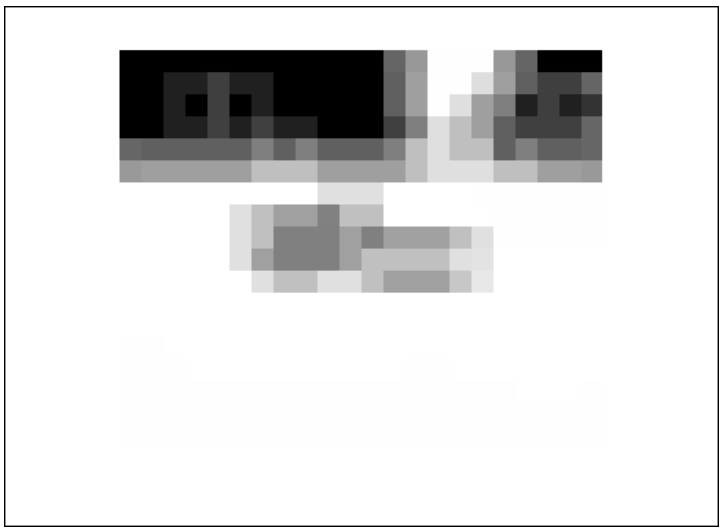


Figura 4.18: Enmascarabilidad por EMROI de la secuencia "Coastguard"

4.4.6. Filtrado espacial basado en la Estimación de Movimiento de Cámara

Debido a los inconvenientes antes mencionados del algoritmo EMJ, se estudia otro tipo de filtrado que solucione, en la medida de lo posible, estos inconvenientes.

En este apartado se pretende alcanzar el mismo objetivo que el algoritmo EMROI, solucionar el problema del filtrado con consideraciones temporales en secuencias con movimiento de cámara. Para ello, se ha reutilizado la Estimación de Movimiento de Cámara (en adelante EMC), obtenido del algoritmo EMROI y descrito más detalladamente en [41].

Señalar que en este proyecto se ha utilizado la EMC, pero podrían haberse empleado otros métodos para estimar el mapa de vectores del movimiento de cámara (MVc) de la secuencia de vídeo.

4.4.6.1. Funcionamiento del algoritmo

A continuación se presenta el diagrama de bloques del algoritmo.

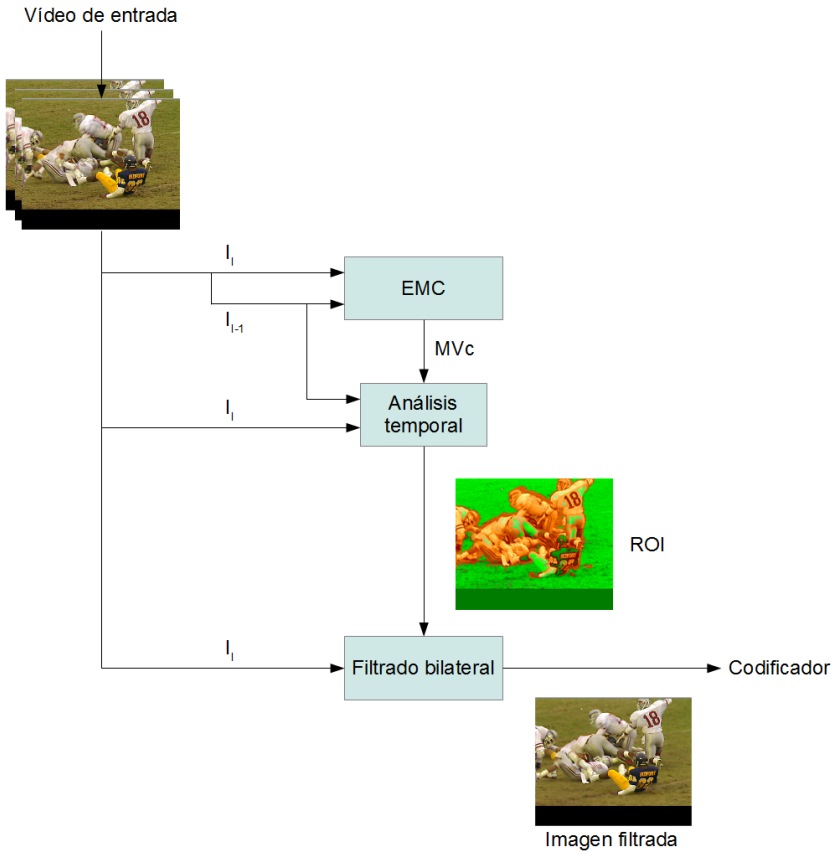


Figura 4.19: Diagrama de bloques del algoritmo basado en la Estimación de Movimiento de Cámara

Este proceso consiste en comparar la región actual con la región de la imagen anterior donde la región actual estaría de no haberse producido un movimiento de cámara. Por tanto, al comparar el plano actual (con la nueva posición calculada) con el plano anterior, las regiones que pertenecen al fondo (regiones en movimiento) se consideran regiones estáticas y aquellas regiones que presentaban movimiento nulo respecto a la cámara, ahora se consideran regiones en movimiento.

4.4.6.2. Análisis de variables

El algoritmo cuenta con una serie de parámetros configurables, cuyo valor es necesario predeterminar de forma eficaz para conseguir un compromiso entre el coste computacional y la calidad del filtrado.

Los parámetros experimentales y variables locales establecidas no son adecuadas para el nuevo algoritmo implementado. Estas variables a analizar, son tres: σ_r (parámetro local del filtro bilateral que indica el máximo filtrado), α (variable experimental) que se utiliza para calcular los pesos con consideraciones

temporales (ver ecuación 4.19), h (controla la caída de la función exponencial) que se utilizaba para realizar el análisis temporal 4.18.

$$w(i,j) = \frac{1}{Z(i)} \exp\left(-\frac{||v(N_j) - v(N_i)||}{h^2}\right) \tag{4.18}$$

$$w = \exp^{\frac{-(1-d_{norm})^2}{\sigma}} \tag{4.19}$$

Encontrar un rango de valores para un nivel de filtrado aceptable en esta nueva versión de filtrado, manteniendo la calidad visual subjetiva, conlleva realizar una búsqueda exhaustiva de dichos parámetros. Se sigue el mismo criterio que en el apartado anterior: se modifica la variable a analizar y se fijan las restantes.

Estudio elección α Esta prueba se compone de dos secuencia de vídeo bastante representativas de la batería de pruebas que se utilizó para determinar el valor de estas variables.

Una de las secuencias es con fondo estático (véase "Football.cif" a modo de ejemplo), donde la cámara no se mueve pero el objeto de interés sí, y la otra se trata de vídeos con el fondo en movimiento (véase "Bus.cif"), en los que la cámara sigue al objeto de interés, en este caso el autobús, de modo que el objeto de interés presenta muy poco movimiento respecto a la cámara, y el fondo presenta una grado de movimiento mayor.

El color rojo indica zonas con un nivel de filtrado poco intenso y el color verde indica zonas con un grado de filtrado mayor. La intensidad del color varía según el grado del filtrado.

- Ventana: 2
- σ_r : 10
- h: 6
- σ_d: 3
- α: 0.3, 0.7, 0.9

(a) Cantidad de filtrado para "Football" (b) Cantidad de filtrado para "Bus"

Figura 4.20: Análisis de la variable $\alpha = 0.3$

(a) Cantidad de filtrado para "Football" (b) Cantidad de filtrado para "Bus"

Figura 4.21: Análisis de la variable $\alpha = 0.7$

(a) Cantidad de filtrado para "Football" (b) Cantidad de filtrado para "Bus"

Figura 4.22: Análisis de la variable $\alpha = 0.9$

Comprobamos que a medida que se aumenta la variable α, sólo se preservan zonas que presentan movimientos más importantes. Esta variable controla el grado de filtrado en función del nivel de movimiento, por tanto aquellas regiones que tengan poco movimiento, serán filtradas espacialmente. Por lo que hay que tener especial cuidado en secuencias como "París", donde la región de interés presenta movimientos poco significativos.

De las pruebas concluimos que el rango de valores aceptables para la variable alfa es:

α={0.55 - 0.95}

Estudio elección h

- Ventana: 2
- σ_r: 10
- α: 0.6
- σ_d: 3

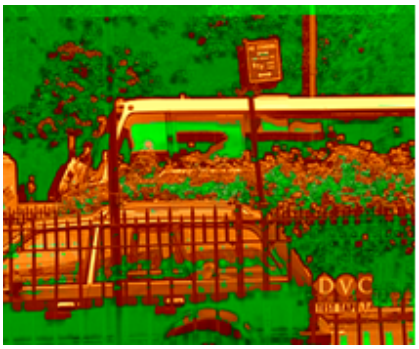
- h: 5, 9, 12

(a) Cantidad de filtrado para "Football"

(b) Cantidad de filtrado para "Bus"

Figura 4.23: Análisis de la variable $h = 5$

(a) Cantidad de filtrado para "Football"



(b) Cantidad de filtrado para "Bus"

Figura 4.24: Análisis de la variable $h = 9$

(a) Cantidad de filtrado para "Football"

(b) Cantidad de filtrado para "Bus"

Figura 4.25: Análisis de la variable $h = 12$

Se observa que la variable h tiene más o menos el mismo comportamiento que la variable α, lo cual es lógico, ya que ambas variables regulan el grado de filtrado en función del grado de movimiento que presentan las regiones. Por consiguiente, para valores pequeños de h, se preservan movimientos menos importantes y, por el contrario, para valores mayores de h, se intensifica el movimiento, es decir, se preservan sólo movimientos significativos.

El rango de valores aceptable para h es:

h={6 - 12}

Visto el comportamiento de ambas variables se decide fijar una de ellas y variar la otra. Se decide fijar el valor de α a 0.6, ya que la variable que se utiliza para el cálculo de los pesos de los píxeles vecinos en el tiempo es h.

Señalar que se pueden emplear otros valores para esta variable, donde esta decisión depende de la cantidad de filtrado y de la reducción de tasa que se desee alcanzar. En conclusión, para controlar la intensidad del filtrado se modificarán las variables h y σ_r.

Estudio del efecto del parámetro σ_r

- Ventana: 2
- σ_r: 5, 15, 35
- α : 0.6
- σ_d: 3
- h: 9.5

Después de realizar varias pruebas se elige una configuración de filtrado que no degrade, en la medida de los posible, la imagen.

Football

(a) Secuencia original

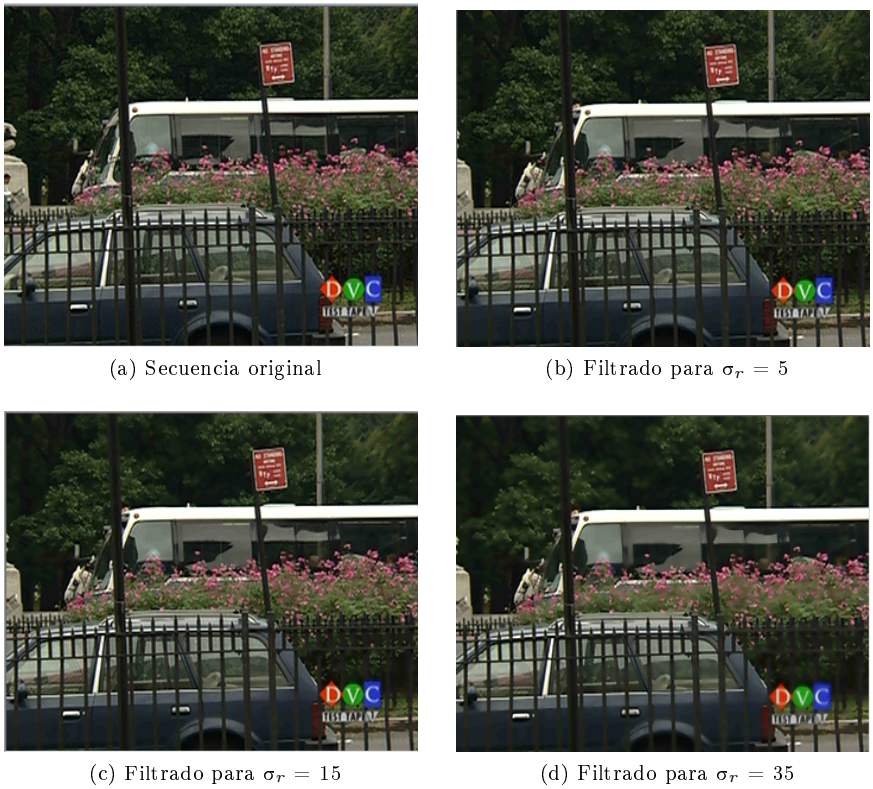
(b) Filtrado para $\sigma_r = 5$

(c) Filtrado para $\sigma_r = 15$

(d) Filtrado para $\sigma_r = 35$

Figura 4.26: Filtrado de la secuencia "Football" para distintos valores de σ_r

Bus



(a) Secuencia original

(b) Filtrado para $\sigma_r = 5$

(c) Filtrado para $\sigma_r = 15$

(d) Filtrado para $\sigma_r = 35$

Figura 4.27: Filtrado de la secuencia "Bus" para distintos valores de σ_r

Foreman

(a) Secuencia original

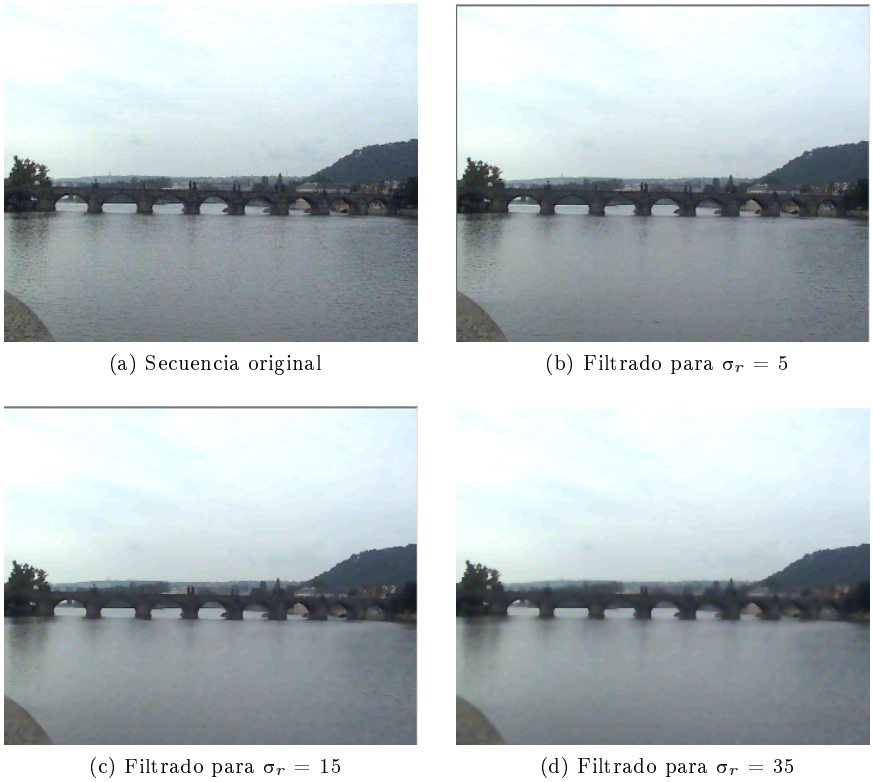
(b) Filtrado para $\sigma_r = 5$

(c) Filtrado para $\sigma_r = 15$

(d) Filtrado para $\sigma_r = 35$

Figura 4.28: Filtrado de la secuencia "Foreman" para distintos valores de σ_r

Bridge far



(a) Secuencia original

(b) Filtrado para $\sigma_r = 5$

(c) Filtrado para $\sigma_r = 15$

(d) Filtrado para $\sigma_r = 35$

Figura 4.29: Filtrado de la secuencia "Bridge far" para distintos valores de σ_r

4.4.6.3. Pruebas experimentales

Estas pruebas tienen como objetivo comparar las versiones de filtrado, implementadas en este proyecto, con el filtro gaussiano con consideraciones perceptuales, desarrollado previamente en el grupo Multimedia. Este filtro gaussiano con consideraciones perceptuales consiste en variar la intensidad del filtro gaussiano (máscara espacial 3x3) cuyos pesos están controlados por una σ (desviación típica del filtro gaussiano, véase apartado 2.4.2) que depende directamente del mapa de enmascarabilidad del algoritmo EMROI, para que se filtren más aquellas regiones de cada plano que no pertenezcan a la ROI.

Se muestran gráficas de PSNR/"Bitrate" y gráficas de SSIM/"Bitrate", donde la SSIM se utiliza como un método para medir la similitud entre dos imágenes [42].

Las configuraciones de filtrado que se comparan son:

- Filtro bilateral
  - Parámetro σ_r: 5, 10
  - Ventana: 2

  - σ_d: 3

- Filtrado bilateral basado en la EMROI
  - Parámetro σ_r: 5, 10
  - Ventana: 2
  - σ_d: 3
- Filtrado bilateral basado en la EMC
  - Parámetro σ_r: 5, 10
  - Ventana: 2
  - α : 0.6
  - σ_d: 3
  - h: 9.5
- Filtro gaussiano con consideraciones perceptuales.

La configuración del codificador es:

- QP constante (Rate control desactivado): 20, 24, 28, 32, 36, 40
- Algoritmo VBR
- Frame rate: 26 fps
- Número de planos: 100

Gráficas

- Gráficas PSNR/"Bitrate"

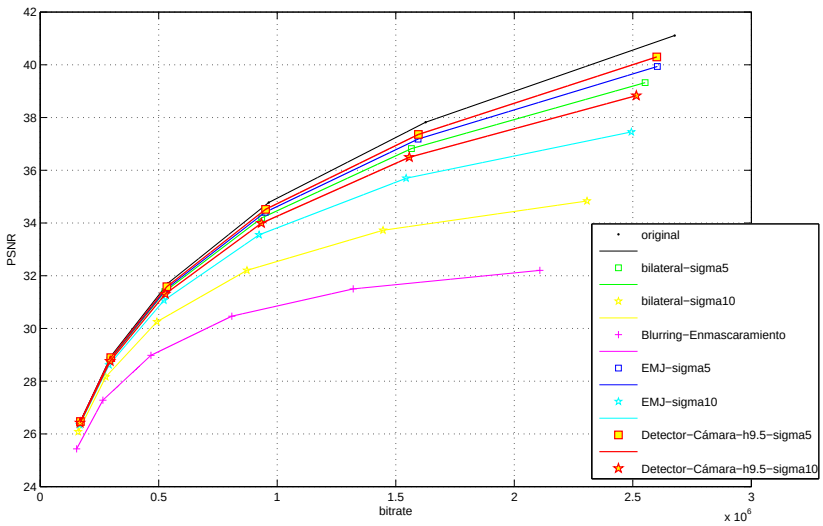


Figura 4.30: Gráfica PSNR/"Bitrate" de la secuencia "Bus"

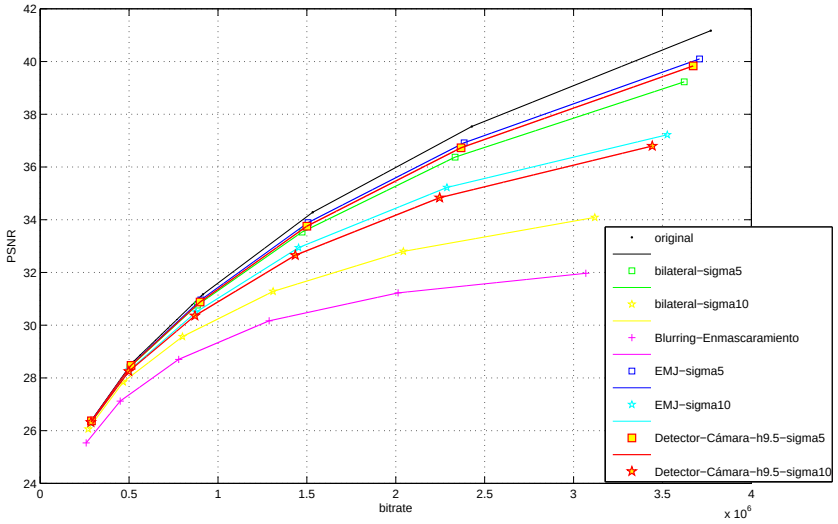


Figura 4.31: Gráfica PSNR/"Bitrate" de la secuencia "Football"

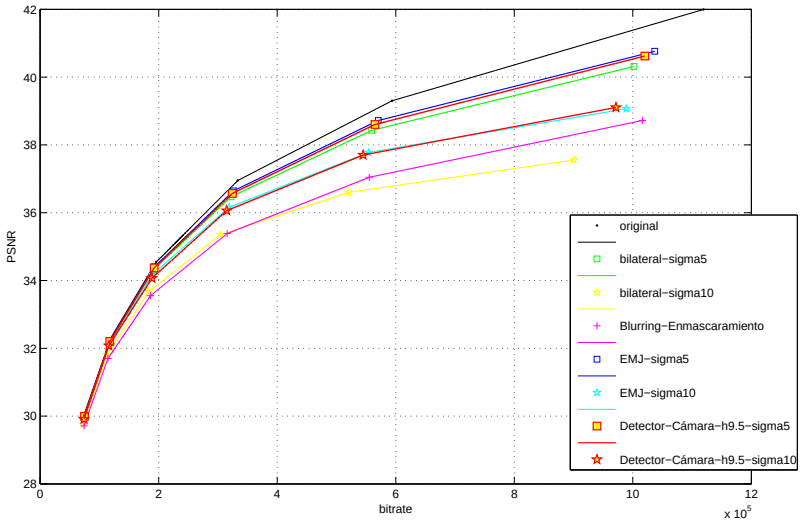


Figura 4.32: Gráfica PSNR/"Bitrate" de la secuencia "Foreman"

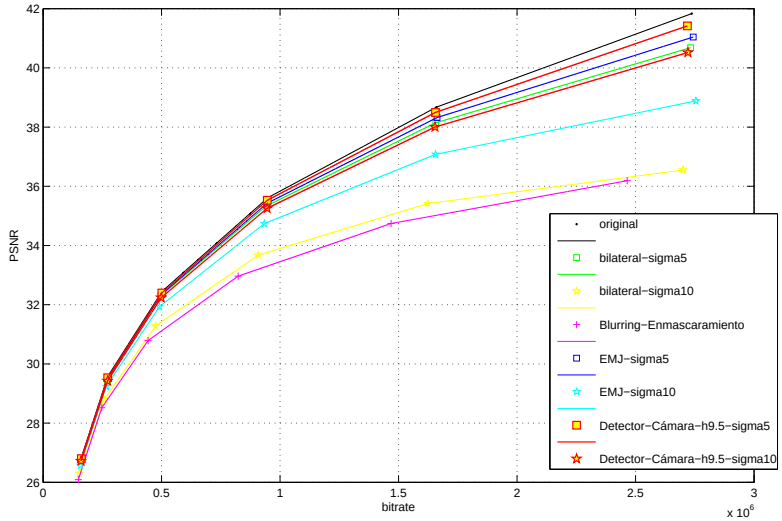


Figura 4.33: Gráfica PSNR/"Bitrate" de la secuencia "Stefan"

- Gráficas SSIM/"Bitrate"

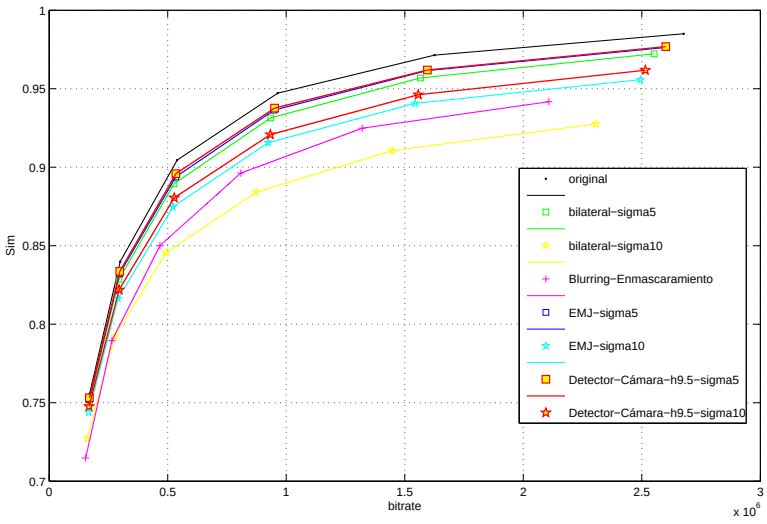


Figura 4.34: Gráfica SSIM/"Bitrate" de la secuencia "Bus"

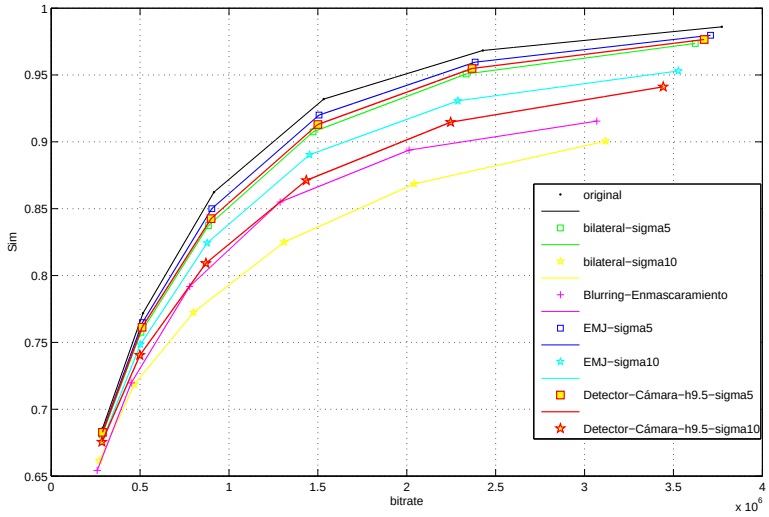


Figura 4.35: Gráfica SSIM/"Bitrate" de la secuencia "Football"

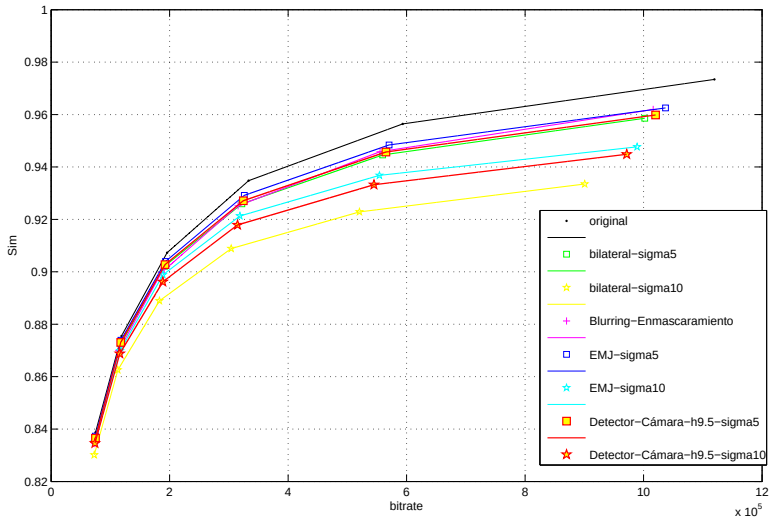


Figura 4.36: Gráfica SSIM/"Bitrate" de la secuencia "Foreman"

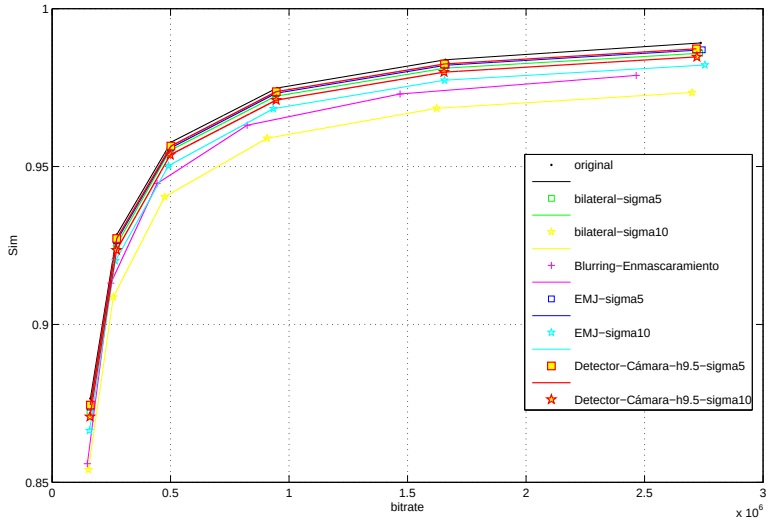


Figura 4.37: Gráfica SSIM/"Bitrate" de la secuencia "Stefan"

Prueba subjetiva Se muestran algunas capturas de pantalla de las secuencias decodificadas a partir de una codificación a tasa fija de la secuencia original y de tres variantes filtradas. En ellas se comprueba la eficiencia del filtro bilateral adaptado respecto a la del filtrado gaussiano basado en el algoritmo EMROI (visto en el apartado 4.4.6.3).

La configuración del codificador es:

- Tasa fija (Rate control activado): 128000, 256000, 384000, 512000 Kbps.
- Algoritmo VBR
- Frame rate: 26 fps
- Número de planos: 100
- Para el filtro con Estimación de Movimiento de Cámara se utilizan los siguientes parámetros:
  - Ventana: 2
  - α : 0.6
  - h: 6
  - σ_d: 3
- Para el filtro con estimación de movimiento jerárquico se utilizan los siguientes parámetros:
  - Ventana: 2
  - σ_d: 3

Bus

- Tasa: 128000 bits
- Plano: 34
- σ_r:5

(a) Secuencia original

(b) Filtro gaussiano basado en el algoritmo EMROI

(c) Filtro bilateral basado en el algoritmo EMROI

(d) Filtro bilateral basado en la EMC

Figura 4.38: Secuencia reconstruida a partir de distintas versiones de filtradas de "Bus" con tasa 128 kbps

(a) Zoom de la figura 4.38b (b) Zoom de la figura 4.38d

Figura 4.39: Zoom de las versiones filtradas de la secuencia "Bus"

Vemos que se aprecian algunas mejoras del filtro basado en la Estimación de Movimiento de Cámara, respecto al filtro selectivo gaussiano. Por ejemplo, el obelisco presenta menos efecto de bloque y se observa una menor degradación del coche, en primer plano, y de la flecha de indicación.

Football

- Tasa: 256000 bits
- Plano: 24
- σ_r:10

(a) Secuencia original

(b) Filtro gaussiano basado en el algoritmo EMROI

(c) Filtro bilateral basado en el algoritmo EMROI

(d) Filtro bilateral basado en la EMC

Figura 4.40: Secuencia reconstruida a partir de distintas versiones filtradas de "Football" con tasa 256 kbps

(a) Zoom de la figura 4.40b.

(b) Zoom de la figura 4.40d.

Figura 4.41: Zoom de las versiones filtradas de la secuencia "Football"

Vemos que en el filtro gaussiano adaptado, los números de la camiseta del jugador se ve más difuminados, además de unos molestos bloques borrosos en la parte inferior del césped. Estas diferencias pueden ser más evidentes en el zoom de ambas figuras.

Por otra parte, en el filtrado EMJ, se ve un efecto de arrastre más evidente que en el resto de las versiones.

Bus

- Tasa: 256000 bits
- Plano: 21
- σ_r: 5

(a) Secuencia original

(b) Filtro gaussiano basado en el algoritmo EMROI

(c) Filtro bilateral basado en el algoritmo EMROI

(d) Filtro bilateral basado en la EMC

Figura 4.42: Secuencia reconstruida a partir de distintas versiones filtradas de "Bus" con tasa 256 kbps

Al igual que en los casos anteriores, se aprecia una mejora de estos filtros respecto a la original y al filtro gaussiano adaptado.

Football

- Tasa: 384000 bits
- Plano: 74
- σ_r: 5

(a) Secuencia original

(b) Filtro gaussiano basado en el algoritmo EMROI

(c) Filtro bilateral basado en el algoritmo EMROI

(d) Filtro bilateral basado en la EMC

Figura 4.43: Secuencia reconstruida a partir de distintas versiones filtradas de "Football" con tasa 384 kbps

Stefan

- Tasa: 384000 bits
- Plano: 17
- σ_r: 5

(a) Secuencia original

(b) Filtro gaussiano basado en el algoritmo EMROI

(c) Filtro bilateral basado en el algoritmo EMROI

(d) Filtro bilateral basado en la EMC

Figura 4.44: Secuencia reconstruida a partir de distintas versiones filtradas de "Stefan" con tasa 384 kbps

Stefan

- Tasa: 384000 bits
- Plano: 19
- σ_r: 5

(a) Secuencia original

(b) Filtro gaussiano basado en el algoritmo EMROI

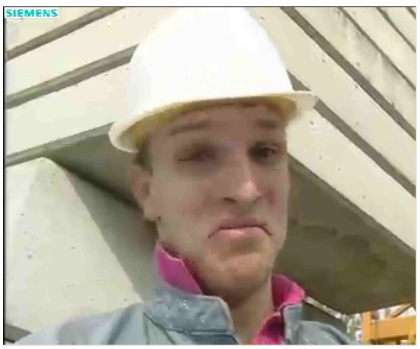
(c) Filtro bilateral basado en el algoritmo EMROI

(d) Filtro bilateral basado en la EMC

Figura 4.45: Secuencia reconstruida a partir de distintas versiones filtradas de "Stefan" con tasa 384 kbps

Foreman

- Tasa: 128000 bits
- Plano: 70
- σ_r: 5



(a) Secuencia original

(b) Filtro gaussiano basado en el algoritmo EMROI

(c) Filtro bilateral basado en el algoritmo EMROI

(d) Filtro bilateral basado en la EMC

Figura 4.46: Secuencia reconstruida a partir de distintas versiones filtradas de "Foreman" con tasa 128 kbps

Foreman

- Tasa: 256000 bits
- Plano: 17
- σ_r: 5

(a) Secuencia original

(b) Filtro gaussiano basado en el algoritmo EMROI

(c) Filtro bilateral basado en el algoritmo EMROI

(d) Filtro bilateral basado en la EMC

Figura 4.47: Secuencia reconstruida a partir de distintas versiones filtradas de "Foreman" con tasa 256 kbps

Pruebas de comparación entre el método ΔQP y el método de filtrado
Haciendo referencia a la técnica de pre-análisis que consistía en modificar la QP del codificador según un mapa de enmascarabilidad basado en la región de interés, se realizan pruebas de comparación subjetivas entre las distintas técnicas de pre-procesado y la técnica de pre-análisis (véase apartado 4.3.2).

La configuración del codificador es:

- Tasa fija (Rate control activado): 256000, 512000
- Algoritmo VBR
- Frame rate: 25 fps

- Número de planos: 100
- Para el filtro con estimación de cámara se utilizan los siguientes parámetros:
  - Ventana: 2
  - α : 0.6
  - h: 6
  - σ_d: 3
- Para el filtro con Estimación de Movimiento Jerárquico con estimación de movimiento de cámara se utilizan los siguientes parámetros:
  - Ventana: 2
  - σ_d: 3

Football

- Tasa: 512000 bits
- Planos: 83, 17
- σ_r:5

(a) Secuencia codificada para una ΔQP = 8

(b) Filtro gaussiano basado en el algoritmo EMROI

(c) Filtro bilateral basado en el algoritmo EMROI

(d) Filtro bilateral basado en la EMC

Figura 4.48: Secuencia reconstruida "Football" (plano 83) con tasa 512 kbps

(a) Secuencia codificada para una ΔQP = 8

(b) Filtro gaussiano basado en el algoritmo EMROI

(c) Filtro bilateral basado en el algoritmo EMROI

(d) Filtro bilateral basado en la EMC

Figura 4.49: Secuencia reconstruida "Football" (plano 17) con tasa 512 kbps

Bus

- Tasa: 256000 bits
- Planos: 21, 50
- σ_r:5

(a) Secuencia codificada para una ΔQP = 8

(b) Filtro gaussiano basado en el algoritmo EMROI

(c) Filtro bilateral basado en el algoritmo EMROI

(d) Filtro bilateral basado en la EMC

Figura 4.50: Secuencia reconstruida "Bus" (plano 21) con tasa 256 kbps

(a) Secuencia codificada para una ΔQP = 8

(b) Filtro gaussiano basado en el algoritmo EMROI

(c) Filtro bilateral basado en el algoritmo EMROI

(d) Filtro bilateral basado en la EMC

Figura 4.51: Secuencia reconstruida "Bus" (plano 50) con tasa 256 kbps

Foreman

- Tasa: 256000 bits
- Plano: 95
- σ_r: 5

(a) Secuencia codificada para una ΔQP = 8

(b) Filtro gaussiano basado en el algoritmo EMROI

(c) Filtro bilateral basado en el algoritmo EMROI

(d) Filtro bilateral basado en la EMC

Figura 4.52: Secuencia reconstruida "Foreman" con tasa 256 kbps

Stefan

- Tasa: 256000 bits
- Plano: 25
- σ_r: 5

(a) Secuencia codificada para una ΔQP = 8

(b) Filtro gaussiano basado en el algoritmo EMROI

(c) Filtro bilateral basado en el algoritmo EMROI

(d) Filtro bilateral basado en la EMC

Figura 4.53: Secuencia reconstruida "Stefan" con tasa 256 kbps

4.4.6.4. Valoración de los resultados subjetivos

- Se resuelve el problema del filtrado por bloques que realiza el filtro basado en el algoritmo EMROI. Como se puede ver, en este algoritmo se realiza el filtrado por píxel y no por bloques, lo que incrementa la eficacia del filtrado de imagen (véase figura 4.54).

(a) Secuencia filtrada a partir del algoritmo EMROI

(b) Secuencia filtrada a partir de la EMC

(c) Zoom del recuadro de la figura 4.54a

(d) Zoom del recuadro de la figura 4.54b

Figura 4.54: Comparación de las versiones filtradas de la secuencia “football”

Como se puede ver en ambas figuras, el filtrado es más efectivo, ya que se realiza un filtrado por píxel y no por bloques.

- En este algoritmo sólo se puede utilizar como referencia el "frame" previo al actual, porque sólo se tiene información del movimiento de cámara del "frame" anterior. Lo cual siendo una desventaja relativa tiene la ventaja de que computacionalmente es ligero.
- Al igual que las anteriores versiones de filtrado, el aumento del tamaño de la ventana implica un aumento del coste computacional y de la simplificación de la imagen.
- Señalar que el algoritmo proporciona buenos resultados para secuencias de vídeo donde la región de interés presenta un grado de movimiento muy importante.
- También es importante resaltar que las técnicas de pre-procesado basadas en los algoritmos EMROI y EMC aportan mejores resultados que la técnica de pre-análisis (variación de la QP) y la técnica de filtrado en el algoritmo EMROI.

Por otro lado, es importante comentar que existen algunas situaciones particulares donde el algoritmo presenta algunos inconvenientes.

- El movimiento de cámara presenta movimientos horizontales, verticales y zoom. Los problemas que surgen de los movimientos horizontales y verticales está en los extremos del "frame", porque al existir movimiento de cámara, los objetos podrían ser desplazados fuera del plano y de la misma manera, pueden aparecer objetos en un momento dado de los que no se tiene información en el instante anterior. Por ello, se opta por filtrar estas regiones consigo mismas aunque el filtrado no vaya a ser óptimo, ya que este tipo de situaciones ocurrirá en los extremos y, por tanto, es muy probable que el observador no sea capaz de detectar estas regiones, principalmente porque no centra su en mirada en ellos.

- Este método se basa en la similitud de dos "frames" contiguos, de modo que si una región del plano actual (compensando el movimiento de cámara) es diferente al "frame" anterior, indica que hay movimiento, y si por lo contrario son iguales es un indicativo de que se trata de una región estática. Surge un problema en regiones estáticas, porque píxeles que pertenecen a regiones estáticas de dos planos consecutivos son ligeramente diferentes (ver tablas: 4.1 y 4.2) cuando deberían ser exactamente iguales, por tanto al aplicar el algoritmo y calcular los pesos se comprobará que el peso no es nulo, es decir, que no se considera idealmente una región estática, por lo que es necesario fijar adecuadamente los valores de las variables del algoritmo.

Figura 4.55: El recuadro azul indica la región de la que se muestran los valores de intensidades de sus píxeles

128	114	147	68	41	141	159	48	71	55	71	86	74	77	74	76
133	115	144	68	43	147	160	62	69	55	70	84	75	77	69	76
125	110	143	67	45	145	122	37	73	55	71	84	79	78	70	77
123	115	146	67	50	79	111	78	61	59	71	85	77	76	69	76
121	114	147	72	44	127	190	85	62	60	73	86	76	79	70	75
129	122	149	70	47	132	197	63	66	60	73	84	79	82	70	78
132	129	148	71	46	113	167	49	75	57	74	85	77	79	76	77
128	117	139	74	42	134	153	43	77	57	76	86	77	76	73	76
125	116	138	72	42	158	170	38	76	56	78	90	77	76	82	76
125	116	131	73	50	114	140	65	66	58	75	84	79	73	74	72
129	117	146	77	43	113	188	70	68	59	78	83	78	76	75	77
123	108	139	78	42	151	156	52	72	56	76	85	79	80	71	75
111	95	136	76	42	148	162	41	74	56	79	88	87	80	74	72
118	116	147	73	44	124	185	47	73	60	81	91	86	80	73	69
141	142	158	71	48	125	178	50	74	61	78	87	79	80	78	73
135	140	154	67	47	141	175	43	77	59	80	90	79	78	77	74

Tabla 4.1: Región que pertenece al plano 2 de la secuencia "París"

128	115	148	67	42	141	160	49	72	56	71	85	73	76	73	76
133	114	144	66	43	150	158	64	70	55	70	84	74	79	68	75
124	110	144	65	45	145	120	37	74	52	73	85	79	78	70	76
121	115	145	67	53	79	109	76	63	59	71	84	77	77	69	75
121	114	146	71	45	128	191	86	61	58	77	86	77	81	72	74
128	123	149	70	46	133	198	63	66	60	74	85	78	81	71	77
130	129	147	71	45	115	170	47	74	57	74	86	77	78	77	77
128	115	140	75	41	131	156	41	76	58	76	86	77	77	73	77
124	116	139	76	41	149	174	40	76	57	77	89	76	77	81	76
125	116	132	74	50	115	141	66	67	58	76	83	78	73	78	72
129	119	143	76	44	116	186	65	67	59	79	84	78	76	77	76
122	109	139	78	41	152	157	52	72	56	78	84	79	80	71	77
112	94	138	75	42	151	161	42	75	57	79	87	87	80	74	73
118	116	149	73	44	123	188	47	73	60	81	92	88	81	74	72
141	142	158	69	47	127	179	49	78	60	79	88	80	79	79	74
135	137	158	69	47	138	180	44	78	61	79	87	79	80	77	72

Tabla 4.2: Región que pertenece al plano 3 de la secuencia "París"

- En lo referente a movimientos de cámara de tipo zoom, las regiones a comparar en este tipo de movimientos, son regiones iguales, pero de tamaños diferentes. No obstante, no se considera un problema porque cuando se realice un zoom de la imagen (región de interés para el observador), el algoritmo detectará movimiento y lo preservará.
- Por último, comentar que la detección de movimiento de cámara (EMC), obtenida a partir del algoritmo de Estimación de Movimiento Jerárquico, es computacionalmente costosa, sin embargo, existen otras alternativas que suponen menor complejidad.

Capítulo 5

Conclusiones y trabajos futuros

5.1. Conclusiones

El objetivo principal de la codificación perceptual de vídeo es diferenciar la información que puede ser, o no, detectada por el observador y así eliminar información redundante. Para conseguirlo, se basa en las propiedades del sistema visual humano (HVS), que no es capaz de detectar distorsiones en ciertas regiones de la imagen, ya sea debido a su textura o por pertenecer a regiones alejadas de la región de interés para el observador.

El objetivo de este proyecto es desarrollar técnicas aplicadas a la codificación perceptual de vídeo. Para ello, este proyecto parte de la técnica de pre-análisis y de la de pre-procesado desarrolladas por el Grupo Multimedia del Departamento de Teoría de la Señal y Comunicaciones de la Universidad Carlos III de Madrid.

La primera técnica consiste en un análisis de la estructura de la textura de las secuencias de vídeo que, en combinación con un mapa de importancia, obtenido a partir de una estimación de la región de interés, ajusta el parámetro de cuantificación del codificador.

La segunda realiza un filtrado selectivo de la secuencia de vídeo a partir del mapa de enmascaramiento basado en la región de interés, de modo que sea simplificada antes del proceso de codificación.

Sin embargo, estas dos técnicas tienen algunos inconvenientes que impiden obtener resultados óptimos:

1. La técnica de pre-análisis obtiene un mapa de enmascaramiento por texturas basado en la Transformada de Coseno Discreta (DCT) que realiza una correcta detección de bloques "detailed" (pertenecientes a bordes abruptos entre objetos que deben ser preservados de la distorsión) cuando los bordes son rectilíneos, pero no para bordes con estructuras más variables. También, tiene el inconveniente de detectar falsos "detailed".

2. La técnica de pre-procesado utiliza el filtro gaussiano, que produce un efecto de desenfocado para filtrados más intensos. Además es computacionalmente intensa.

Por estos motivos, el primer objetivo de este proyecto es el de encontrar un método para la caracterización de las texturas alternativo a la DCT.

Este método se basa en los Histogramas de Gradientes Orientados (HOG) para detectar texturas "detailed" y texturas "caotic" (que corresponden a regiones en que los límites de la sensibilidad del sistema visual humano permitirán enmascarar una mayor distorsión).

En primer lugar, se diseñó un clasificador de texturas basado en umbrales. Dicho mecanismo no es adecuado, ya que ofrece una clasificación que únicamente nos dice el tipo de macrobloque que es, pero no aporta el grado de "caotic" o "detailed" del macrobloque.

Por ello, se utiliza un sistema basado en redes neuronales que nos proporciona una salida continua que indica el tipo y grado del macrobloque. De los resultados observados, concluimos que este método soluciona, en la medida de lo posible, los inconvenientes de la DCT, ya que tiene una menor tasa de error en la detección de texturas "detailed" y consigue detectar bordes rectilíneos y no rectilíneos, lo que mejora la clasificación de texturas "detailed" en rostros, que son regiones especialmente sensibles. Sin embargo, este método acarrea otros tipos de problemas:

- Tiende a confundir bloques "caotic" con bloques normales.
- Problemas para detectar bloques "detailed" cuando contienen más de un borde de diferentes direcciones.

Finalmente, se desestimó el uso de el mapa de enmascaramiento por texturas y la codificación basada en el ajuste de la QP sólo tiene en cuenta el mapa de enmascaramiento que obtiene del algoritmo de Estimación de Movimiento Jerárquico con compensación de movimiento de cámara (EMROI). Los motivos para esta decisión fueron:

- Las regiones de interés pueden tener texturas caóticas.
- La clasificación por texturas basada en el algoritmo HOG no es totalmente correcta.

Ateniéndose al trabajo realizado, se concluye que la técnica de pre-procesado obtiene mejores resultados que el ajuste de la QP del codificador a partir de un análisis previo de la secuencia.

Para solucionar las carencias del pre-procesado gaussiano se empleó el filtro bilateral para realizar el filtrado de la secuencia (dicho filtro tiene en cuenta la textura de la imagen). Dado que los resultados subjetivos no mostraban mejoras, en comparación con el filtro gaussiano desarrollado por el Grupo de Multimedia, se optó por modificar el filtro para hacerlo adaptativo, de modo que a partir de un análisis temporal (similitud de las regiones) de la región actual con las regiones cosituadas detecta si se ha producido movimiento. Este filtro aportaba buenos resultados, excepto para secuencias con movimiento de cámara, por lo que se decidió modificar esta técnica de pre-procesado a partir de dos métodos que tuviesen en cuenta el mismo:

1. El primer método consiste en variar el parámetro del filtro bilateral, σ_r, en función del mapa de enmascarabilidad (EMROI), empleado ya en versiones anteriores.

2. Paralelamente, se desarrolla un segundo método debido a los inconvenientes que presenta el algoritmo de Estimación de Movimiento Jerárquico. Este método tiene el mismo objetivo que el anterior, variar el parámetro del filtro bilateral, pero con la diferencia de que realiza un análisis temporal de la región actual con la región del plano anterior donde la región actual estaría de no haberse producido un movimiento de cámara, el cual previamente se estima del algoritmo EMROI.

Se llevaron a cabo pruebas experimentales y subjetivas para comparar estos métodos entre ellos y, a su vez, compararlos con el filtrado gaussiano basado en el algoritmo EMROI.

Señalar que la configuración para este filtrado bilateral adaptativo es similar para las dos versiones, pero el contexto en el que se implementan es diferente: la primera controla la intensidad del filtrado a partir de un índice de enmascaramiento y la segunda la controla a partir de la similitud de las regiones del plano actual con movimiento de cámara y del plano anterior. Por tanto, la configuración del filtrado basada en el algoritmo EMROI es la siguiente:

- $Ventana = \{2, 3, 4, 5\}$
- $\sigma_r = [5 - 35]$

La configuración del filtrado bilateral basada en la EMC es:

- $Ventana = \{2, 3, 4, 5\}$
- $\sigma_r = [5 - 35]$
- $\alpha = [0{,}55 - 0{,}95]$
- $h = [6 - 12]$

Los rangos de valores de las variables se consideran adecuados para no degradar excesivamente la secuencia. Sin embargo, el valor a fijar dependerá de la tasa a la que se quiera trabajar y de la reducción de la misma que se quiera obtener.

Llegamos a la conclusión de que el filtrado bilateral basado en el algoritmo EMROI y el basado en la EMC ofrecen resultados similares y, a su vez, ofrecen resultados subjetivos mejorados con respecto al filtro gaussiano con consideraciones perceptuales, ya que a diferencia de este filtro, el filtro bilateral preserva la estructura de alto nivel.

Señalar que el filtrado bilateral basado en la EMC acarrea un coste computacional menor y ofrece unos resultados ligeramente mejor que el filtrado bilateral basado en el algoritmo EMROI. Es digno de mención que esta técnica de pre-procesado basada en la estimación de movimiento de cámara puede utilizar otros mecanismos de detección de movimiento menos costosos computacionalmente.

Por último, resaltar que esta técnica ofrece buenos resultados para secuencias donde las regiones de interés para el observador son aquellas zonas que presentan movimiento diferente al de cámara y además son significativas.

5.2. Líneas futuras

El campo de la codificación perceptual es un área muy importante en el que muchas líneas de investigación centran su atención. Estas investigaciones utilizan diferentes técnicas de procesos externos o internos al codificador para conseguir su objetivo. Hoy en día, la técnica de filtrado se ha convertido en una de las técnicas más populares.

Se propone finalizar todo el trabajo expuesto con unas líneas de trabajo futuras, con el objetivo de seguir potenciando las prestaciones del algoritmo y adaptarlo a otras partes del codificador.

5.2.1. Continuar con el control de la intensidad del filtrado

Recordemos que el principal objetivo es variar la intensidad del filtrado a partir del mapa de enmascaramiento con consideraciones perceptuales. En el capítulo 4 comprobamos que este mapa de enmascaramiento conlleva algunos inconvenientes. Se propone la búsqueda de mapas de calidad con menor coste computacional y una clasificación más acorde con la realizada por el sistema visual humano (HVS).

Asimismo, el filtro con el que se realizaron las pruebas es un filtro que tiene en cuenta las texturas de la imagen (bajo coste computacional). Sin embargo, se sugiere el estudio del control de parámetros de otros filtros más eficientes como el filtro de Gabor que, aunque acarrea un gran coste computacional, ofrece una calidad subjetiva bastante apreciable.

5.2.2. Trasladar la técnica de filtrado a otras partes del codificador

El filtrado no es exclusivo del pre-procesado y puede utilizarse en un proceso interno o en un proceso posterior al codificador. Se plantea trasladar el filtro bilateral dentro del codificador para eliminar información redundante del plano actual y del plano con compensación de movimiento. De esta manera, se minimiza la energía total de la señal resultante de la estimación y de la compensación de movimiento y se consigue un incremento de la eficiencia del codificador.

5.2.3. Adaptación a patrones de codificación

La secuencia de vídeo se compone de uno o más grupos de imágenes (GOP), que a su vez, están constituidos por imágenes de: I, P y B, según el tipo de predicción que realizan en la compensación de movimiento. Mientras que los planos Intra se codifican sin emplear imágenes previamente codificadas como referencia, los planos P utilizan una referencia anterior para realizar la compensación de movimiento, y los B realizan bipredicción, es decir, que emplean una referencia correspondiente a un plano anterior y otra correspondiente a un plano posterior en el orden de presentación. La diferencia fundamental entre los tipos está en el tamaño del plano codificado, siendo por lo general el mayor para I y el más pequeño para B.

En este proyecto, no se ha hecho ninguna distinción entre ellas y se llevado a cabo el filtrado de cada plano de la misma manera. Por tanto, es conveniente realizar un análisis del efecto de la codificación de los planos P y B dependiendo

del filtrado que se aplique a los planos "Intra", ya que estos últimos se utilizan como referencia para la codificación de los planos P y B y por tanto un menor filtrado que preserve los I puede ayudar a que funcione mejor la compensación de movimiento y baje la tasa general.

También es importante ver el efecto que se produce en la codificación al utilizar planos P y B filtrados de distinta manera.

Capítulo 6

Presupuesto

En este capítulo se realizan los cálculos correspondientes a los costes asociados a la realización de este proyecto. El presupuesto se desglosa en costes de materiales empleados, costes de honorarios de las personas encargadas de realizarlo y del coste total.

6.1. Coste del material

Los materiales empleados en este Proyecto, tanto software como hardware, son los que se detallan a continuación:

- *Espacio de trabajo.* Se añaden los costes de luz, calefacción, mantenimiento, mobiliario necesario, entre otros. Por ello, tiene en total un coste de unos 1000 €/mes. Al tratarse de un laboratorio compartido por aproximadamente 6 personas, tendrá un coste individual asociado de 167 €/mes. Debido a que la duración del proyecto ha sido de aproximadamente 12 meses, el coste relativo al espacio de trabajo asciende a 2004 €.

- *Ordenador personal.* Se emplea para la recopilación y estudio de documentación y programación y para realizar las pruebas necesarias durante el desarrollo del algoritmo. Debido a que estas pruebas requieren de gran capacidad de procesado, el ordenador deberá ser de una potencia considerable. El valor aproximado del equipo informático es de 900 €; como éste puede ser reutilizado tras la realización de este proyecto, su coste puede amortizarse hasta la cantidad de 310 €.

- *Un ordenador para pruebas.* Como ha sido necesario realizar baterias de pruebas muy intensivas se ha necesitado ocasionalmente de un ordenador adicional similar al anterior. Debido a su uso no habitual y que es compartido por el Departamento, su coste estimado es 100 €.

- *Licencias software.* En este proyecto se han empleado varios programas, cuyas licencias son las que se indican a continuación:

    - Sistema operativo Windows 7 Professional de Microsoft, licencia de cuyo coste es de aproximadamente de 309 € amortizados en 4 años, por lo tanto, el coste aplicable a este proyecto es de 77.25 €.

  - Matlab R2007b de MathWorks, utilizado en la implementación de los algoritmos, cuya licencia es de 2000 € (4 años de amortización), supone la cantidad de 500€.
  - Microsoft Office 2003, cuya licencia es de 198 €. Considerando una amortización de 4 años, el coste sería de 49,5 €.

- *Conexión ADSL.* La tarifa plana ADSL tiene un coste mensual de 60 €, lo que supone un total de 720 €, dividido entre las 6 personas del laboratorio 120 €.

- *Material de oficina.* En este apartado se incluye todo el gasto referente a material de oficina: papel, impresiones, bolígrafos, etc. El importe total se estimará en 60 €.

La siguiente tabla recoge todos los costes relacionados con el material utilizado en el desarrollo del proyecto:

Coste del material		
Descripción	Empresa	Coste imputable
Lugar de trabajo	-	2004 €
Ordenador personal	-	310 €
Ordenador de pruebas	-	100 €
Licencia Windows 7 Profesional	Microsoft	77.25 €
Licencia Matlab R2007b	MathWorks	500 €
Licencia Microsoft Office 2003	Microsoft	49.5 €
Conexión ADSL	-	120 €
Material de oficina	-	60 €
	Total: 3220,75 €	

Tabla 6.1: Coste de material

6.2. Coste personal

Para establecer el presupuesto asociado a los honorarios de las personas a cargo de este proyecto es necesario tener en cuenta, en primer lugar, la duración del proyecto y las horas de trabajo realizadas en el mismo. Por tanto, si la duración total es de 9 meses y se establece un horario de trabajo de 4 horas diarias y una dedicación de 5 días semanales, se obtiene un total de 720 horas laborables.

Para el cálculo completo del coste es necesario tener en cuenta los honorarios de un Ingeniero Técnico de Telecomunicación. Si nos atenemos al último dato proporcionado por el Colegio Oficial de Ingenieros de Telecomunicación [46], de 2008, los honorarios de un ingeniero de telecomunicación ascienden a 72 €/hora. Para tratar de actualizar este dato le aplicaremos una subida correspondiente al IPC entre los meses de enero de 2008 hasta agosto de septiembre 2011, que, según el Instituto Nacional de Estadística [?], ha sido un 6,8 %. Por lo tanto, teniendo en cuenta este dato, el coste por hora será de 76,896 €/hora.

Contando que el número de horas invertidas es de 720, los gastos por honorarios del ingeniero realizador del proyecto ascienden a 55.365,12 €.

Con respecto a los honorarios del director del proyecto, en general, se corresponden con un 7 % del coste total del proyecto. Considerando que el coste total del proyecto asciende a 58.585,87 €. Los honorarios correspondientes a dirección de proyecto corresponden a 4.101,01 €. El coste total por gastos de personal asciende a 59.466,13 €.

Personal		
Nombre y apellidos	Categoría	Coste (Euro)
Manuel de Frutos López	Ingeniero Senior	4.101,01€
Helen Mariela Medina Chanca	Ingeniero	55.365,12€
	Total	59.466,13€

Tabla 6.2: Coste de personal

6.3. Presupuesto total

Considerando todos los costes anteriormente detallados podemos calcular el coste total del Proyecto, que se muestra en la tabla.

Resumen de coste	
Concepto	Presupuesto Coste Total
Coste de personal	59.466,13€
Coste de material	3.220,75 €
Subtotal	62.686,88 €
IVA (16 %)	10.029,9 €
Total:	72.716,78€

Tabla 6.3: Coste total del proyecto

El presupuesto total del proyecto asciende a NOVENTA Y CINCO MIL SEISCIENTOS VEINTE Y TRES EUROS.

Fdo: Helen Mariela Medina Chanca

Ingeniera Técnica de Telecomunicación, especialidad Sistemas de Telecomunicación.

Anexo

Pruebas filtrado trilateral y bilateral con consideraciones temporales

En este apartado se muestran algunas imágenes a las que se les ha aplicado un filtrado trilateral y un filtrado bilateral con consideraciones temporales para distintos valores de σ_r. También se añaden planos de referencia: la imagen original y la imagen filtrada con el filtro bilateral. Con la intención de realizar una comparación subjetiva entre ellas.

Prueba 1 realizada con la siguiente configuración:

- Secuencia: "París"
- Formato: cif
- Ventana: 5
- σ_d: 3
- Número de frames para el filtro trilateral: 1

(a) Secuencia original

(b) Filtrado bilateral con $\sigma_r = 10$

(c) Filtrado trilateral con $\sigma_r = 10$

(d) Filtrado con consideraciones temporales con $\sigma_r = 10$

Figura 6.1: Versiones de filtrado para la secuencia "París"

Prueba 2 realizada con la siguiente configuración:

- Secuencia: "París"
- Formato: cif
- Ventana: 2
- σ_d: 3
- Número de frames para el filtro trilateral: 1

(a) Secuencia original

(b) Filtrado bilateral con $\sigma_r = 20$

(c) Filtrado trilateral con $\sigma_r = 20$

(d) Filtrado con consideraciones temporales con $\sigma_r = 20$

Figura 6.2: Versiones de filtrado para la secuencia "París"

Prueba 3 realizada con la siguiente configuración:

- Secuencia: "Football"
- Formato: cif
- Ventana: 2
- σ_d: 3
- Número de frames para el filtro trilateral: 1

(a) Secuencia original

(b) Filtrado bilateral con $\sigma_r = 30$

(c) Filtrado trilateral con $\sigma_r = 30$

(d) Filtrado con consideraciones temporales con $\sigma_r = 30$

Figura 6.3: Versiones de filtrado para la secuencia "Football"

Prueba 4 realizada con la siguiente configuración:

- Secuencia: "Tigre y dragón"
- Formato: cst
- Ventana: 2
- σ_d: 3
- Número de frames para el filtro trilateral: 1

(a) Secuencia original

(b) Filtrado bilateral con $\sigma_r = 45$

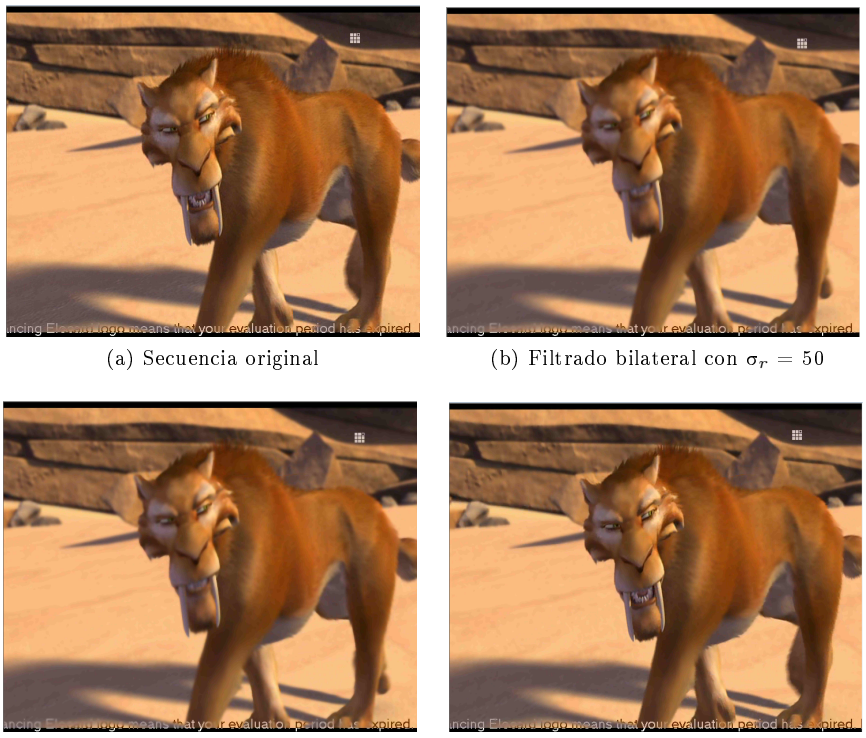
(c) Filtrado trilateral con $\sigma_r = 45$

(d) Filtrado con consideraciones temporales con $\sigma_r = 45$

Figura 6.4: Versiones de filtrado para la secuencia "Tigre y dragón"

Prueba 5 realizada con la siguiente configuración:

- Secuencia: "Iceage"
- Formato: cst
- Ventana: 2
- σ_d: 3
- Número de frames para el filtro trilateral: 1



(a) Secuencia original

(b) Filtrado bilateral con $\sigma_r = 50$

(c) Filtrado trilateral con $\sigma_r = 50$

(d) Filtrado con consideraciones temporales con $\sigma_r = 50$

Figura 6.5: Versiones de filtrado para la secuencia "Iceage"

De la pruebas realizadas, observamos diferencias entre las distintas versiones de filtrado.

El filtro trilateral ofrece mejores resultados que el filtrado bilateral para los vídeos de formato cif, pero para los de formato cst. En el filtrado bilateral y trilateral se aprecia un desenfocado muy evidente para valores mayores de σ_r.

Por otro lado, el filtro con consideraciones temporales consigue el objetivo de preservar las regiones de la imagen que presentan movimiento. Sin embargo éste filtro falla totalmente para secuencias con movimiento de cámara, ya que tiende a preservar toda la imagen.

Organización DVD adjunto

En este anexo se incluye una descripción del contenido del DVD adjunto que contiene las secuencias utilizadas en las pruebas realizadas.

Se incluyen las secuencias originales, las secuencias filtradas y las codificadas de todas las versiones empleadas en este proyecto. Adicionalmente, se incluye un archivo "Léeme.pdf" donde se indica el tamaño de los vídeos, tasas y las distintas configuraciones.

La siguiente tabla lista los distintos nombres de las secuencias.

Nombre archivo	Descripción
<secuencia>.avi	Secuencia original
<secuencia>_CP_blurring.avi	Filtrado gaussiano con consideraciones perceptuales
<secuencia>_w2sigmar_5_EMJ.avi	Filtrado bilateral basado en el algoritmo EMJ con σ_r= 5
<secuencia>_w2sigmar_10_EMJ.avi	Filtrado bilateral basado en el algoritmo EMJ con σ_r= 10
<secuencia>_w2sigmar_5h_6_detectorCamara.avi	Filtrado bilateral basado en la estimación del movimiento de cámara con σ_r= 5 y h = 6
<secuencia>_w2sigmar_10h_6_detectorCamara.avi	Filtrado bilateral basado en la estimación del movimiento de cámara con σ_r= 10 y h = 6

Tabla 6.4: Tabla de configuraciones

Se adjunta un diagrama que indica la organización de los archivos en el DVD.

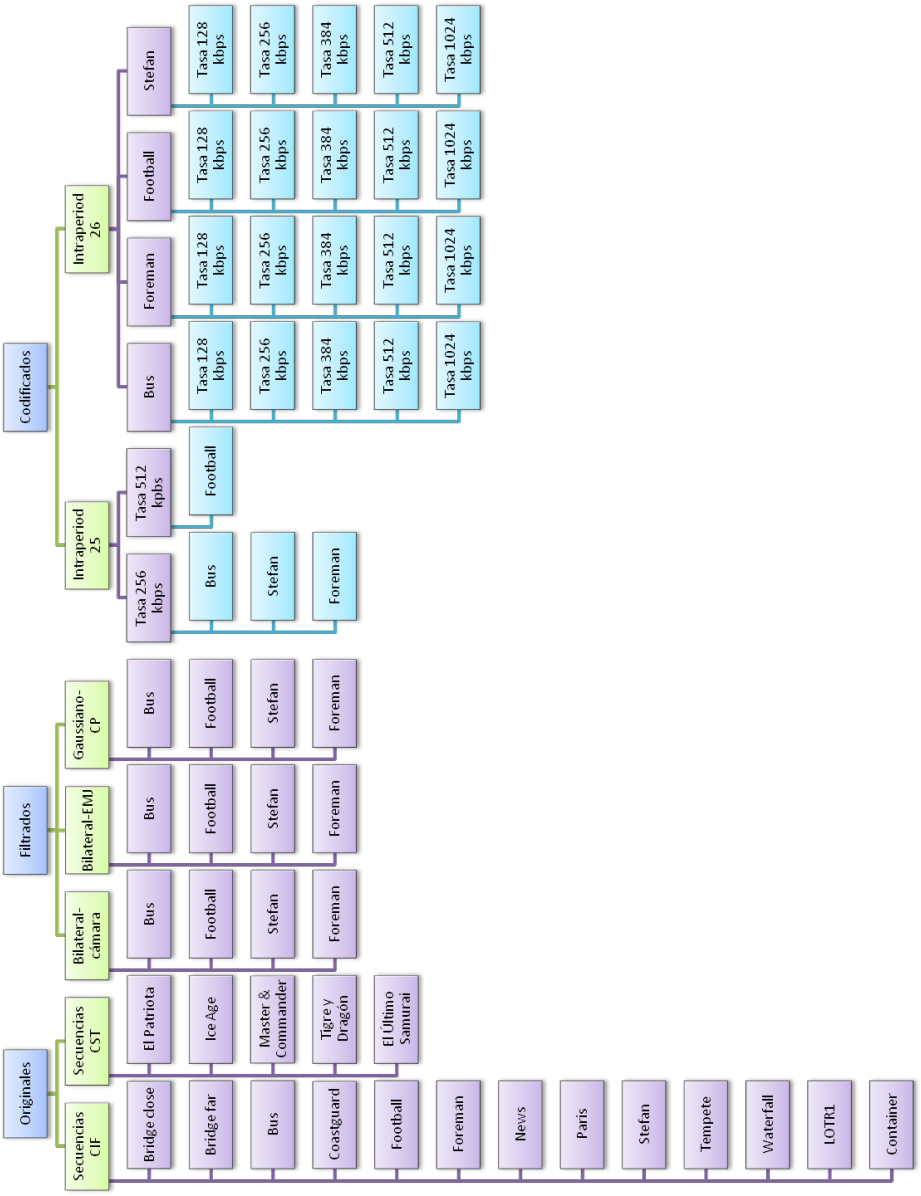


Figura 6.6: Diagrama de la organización del DVD adjunto

Asimismo, se incluye el presente Proyecto en formato pdf.

Glosario

AF:	*Anisotropic Filtering*
AC:	*Coeficiente de alterna*
CABAC:	*Context-based Adaptive Binary Arithmetic Coding*
CAVL:	*Context-Based Adaptive Variable Length Coding*
CODEC:	*Coder/Decoder*
CIF:	*Common Intermediate Format*
DWT:	*Discrete Wavelet Transform*
DCT:	*Discrete Cosine Tranform*
FFT2:	*Two- Dimension Fast Fourier Transform*
JND:	*Just Noticeable Distortion*
JME:	*Joint Motion Estimation*
HVS:	*Human Visual System*
HOG:	*Histogram of Oriented Gradients*
IDCT:	*Inverse Discrete Cosine Transform*
KLT:	*Karhunen-Loeve Transform*
MB:	*Macro Block*
ME:	*Motion Estimation*
MC:	*Motion Compensation*
MVD:	*Motion Vector Difference*
MSE:	*Mean Squared Error*
NL:	*Non-Local*
NNA:	*Artificial Neuronal Network*
ROI:	*Region Of Interest*
R-Q:	*Rate-Quantization*
RMSE:	*Root Mean Square Error*
PSNR:	*Peak Signal to Noise Ratio*
QP:	*Quantization Parameter*
SAD	*Sum of Absolute Dierences*
SIM:	*Space Image Model*
STD:	*Standard Deviation*
SSIM:	*Structural Similarity Index Method*

Bibliografía

[1] Iain E. G. Richardson, "H.264 and MPEG-4 Video Compression", Ed. Wiley, 2003.

[2] Ingo Höntsch, Lina J. Karam, "Adaptive Image Coding With Perceptual Distortion Control", IEEE Transactions on Image Processing, vol. 11, issue 3, pp. 213-222, marzo 2002.

[3] C. Tang, C. Chen, Y.Yu and C. Tsai, "Visual sensitivity guided bit allocation for video coding" IEEE Transaction on multimedia, Vol. 8, No 1, Feb. 2006, pp. 11-18.

[4] K. Minoo and T. Nguyen, "Perceptual video coding with H.264", Conference Record of the Thirty-Ninth Asilomar Conference on signal, Systems and Computers, pp. 741-745, noviembre 2005.

[5] G. Sorwar, A. Abraham, "Texture Classification Based on DCT and Soft Computing", IEEE International Conference on Fuzzy systems, vol. 3, pp. 545-548, diciembre 2001.

[6] Alfonso Fernández Sarría. "Estudio de técnicas basadas en la transformada Wavelet y optimización de sus parámetros para la clasificación por texturas de imágenes digitales", Universidad Politécnica de Valencia, Departamento de Ingeniería Cartográfica, Geodesia y Fotogrametría, febrero 2007.

[7] J. F. Canny, "A computational approach to edge detection" IEEE Trans, 1986.

[8] R. C. Gonzalez R. E. Woods, "Digital Image Processing". Reading, MA: Addison-Wesley, 1992.

[9] C. Christopoulos, J. Askelof, M. Larsson, "Efficient region of interest coding techniques in the upcoming JPEG2000 still image coding standard", IEEE Signal Processing Letters, vol. 7, issue 9, pp. 247-249, agosto 2002.

[10] Z. Wang, A. C. Bovik, "Bitplane-by-bitplane shift (BbBShift) - a Suggestion for JPEG2000 Region of Interest Image Coding", IEEE Signal Processing Letters, vol. 9, no. 5, mayo 2002.

[11] J. Han, M. Li, H. Zhang and Lei Guo, "Automatic Attention Object Extraction from Images", IEEE ICIP, vol. 3, pp. II-403-6, septiembre 2003.

[12] M. Bosch, F. Zhu and E. J. Delp, "Perceptual quality evaluation for texture and motion based coding", IEEE ICIP, pp. 2285-2288, febrero 2007.

[13] S. Sengupta, S. Gupta and J. Hannah, "Perceptually motivated bit-allocation for H.264 encoded video sequences", IEEE ICIP, vol. 2, pp. III-797-800, noviembre 2003.

[14] Laurent Itti, "Automatic Foveation for Video Compression Using a Neurobiological Model of Visual Attention", IEEE Transactions on Image Processing, vol. 13, no. 10, octubre 2004.

[15] Y. Sun, Dongdong Li; Ya-Qin, "Region-Based Rate Control and Bit Allocation for Wireless Video Transmission", IEEE Transactions on Multimedia, vol. 8, issue 1, pp. 1-10, enero 2006.

[16] H. Yu, F. Pan, Z. Lin and Y. Sun, "A Perceptual Bit Allocation Scheme for H.264", Proc. IEEE ICME 2005.

[17] H. H. Y. Tong, A. N. Venetsanopoulos, "A perceptual model for JPEG applications based on blocks classifications, texture masking and luminance masking", IEEE ICIP, vol 3, pp. 428-432, agosto 2002.

[18] Marcos Martín, "Ténicas clásicas de segmentación de imagen", 21 de mayo 2002.

[19] L. enrique Sucar, Giovani Gómez, "Visión computacional", Instituto de Ciencias Computacionales.

[20] Antoni Buades, Bartomeu Coll, Jean Michel M. "A non-local algorithm for image denoising", Departamento de Matemáticas.

[21] C. Tomasi, R. Manduchi. "Bilateral filtering for gray and colors images", IEEE Sixth International Conference on Computer Vision, pp. 839-846, agosto 2002.

[22] G. Bellino, F. Fernandez y G. Scarel. "Filtrado bilateral", junio 2007.

[23] Tuan Q. Pham, Lucas J. van Vliet. "Separable bilateral filtering for fast video preprocessing", IEEE International Conference on Multimedia Expo, 4 pp. octubre 2005.

[24] Bill Christmas, "Designing complex Gabor filters", noviembre 2007.

[25] V. Shiv Naga P., Justin Domke. "Gabor Filter Visualization", report, University of Maryland.

[26] Javier T. Movellan. "Tutorial on Gabor filters", Free Documentation License.

[27] Johannes Ballé. "Image simplification by frequency-selective means filtering", Picture Coding Symposium, pp 126-129, diciembre 2010.

[28] Alfonso Fernández Sarría. "Estudio de técnicas basadas en la transformada Wavelet y optimización de sus parámetros para la clasificación por texturas de imágenes digitales", febrero 2007.

[29] R. Dugad , N. Ahuja, "Video denoising by combining Kalman and Wiener estimates", ICIP, vol. 4, pp. 152-156, agosto 2002.

[30] Llach, Joan, Boyce, Jill M., "H.264 encoder with low complexity noise prefiltering" , Proceedings of SPIE, 2003.

[31] Cheong, H.-Y., Tourapis, A.M., Llach, J., Boyce, J.,"Adaptive spatio - temporal filtering for video de-noising", International Conference on Image Processing, vol. 2, pp. 965-968, 2004.

[32] G. Varghese and Zhou Wang, "Video denoising using a spatiotemporal statistical model of Wavelet coefficients", IEEE ICASSP, pp. 1257-1260, mayo 2008.

[33] J. Kim, J. W. Lee, R. Park, M Park, "Adaptive edge-preserving smoothing and detail enhancement for video preprocessing of H.263", International Conference on Consumer Electronics, pp. 337-338, febrero 2010.

[34] L. Mao-quan, X. Zheng-quan, "An adaptive preprocessing algortihm for low bitrate video coding", Journal of Zhejiang University SCIENCE A (JZUS A), julio 2006.

[35] Dissertation by Jian Xu, "Video preprocessing based on human perception for Telesurgery", University of Pittsburgh, agosto 2009.

[36] R. Kawada, A Koike, Y. Nakajima, "Prefilter control scheme for lowbitrate TV distribution", IEEE International Conference on Multimedia and Expo, pp. 769-772, Japan, diciembre 2006.

[37] N. Dalal and B. Triggs., "Histograms of oriented gradients fot human detection". In Proc of the IEEE Conf on Computer Vision and Pattern Recognition, San Diego, USA. Vol. II, pp. 886-893, 2005.

[38] Antoni Buades, Bartomeu Coll, Jean Michel M. "On image denoising methods".

[39] Luis F. Bertona, Tesis de Grado en Ingeniería Informática "Entrenamiento de Redes Neuronales basado en algoritmo evolutivos", Universidad de Buenos Aires, laboratorio de sistemas inteligentes, noviembre 2005.

[40] Grupo de Investigación Aplicada a la Ingeniería Química (GIAIQ) "Redes Neuronales: Conceptos Básicos y Aplicaciones".

[41] Ana B. Mejia Ocaña, Proyecto fin de carrera. "Segmentación del movimiento en secuencias de vídeo y su aplicación a la codificación perceptual de vídeo", Universidad Carlos III de Madrid, 2010.

[42] Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity", IEEE Transactions on Image Processing, vol. 13, No. 4, abril 2004.

[43] R. Rojas, "The Backpropagation Algorithm", Computer Sicence, Free University of Berlin, 1996.

[44] Banafshe A., Mark S. Nixon. "Robust log-Gabor filter for ear biometric", 19 th International Conference on Pattern Recognition, pp. 1-4, enero 2009.

[45] http://www.ine.es/varipc/index.do

[46] http://www.coit.es./Z