

ESTÁNDARES DE COMUNICACIÓN DE AUDIO Y VÍDEO (ECAV)

PRÁCTICA 2

IMPLEMENTACIÓN DE UN CODIFICADOR AVANZADO DE IMÁGENES

CURSO 2019-20

GRUPO	
FECHA	

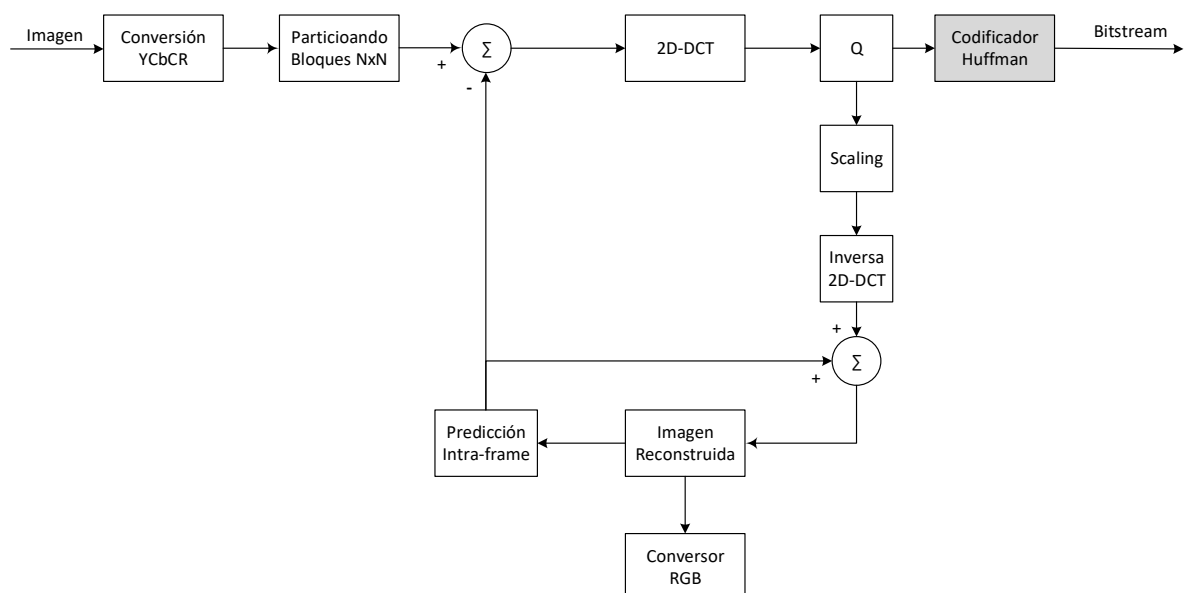
PARTICIPANTES DEL GRUPO

1. OBJETIVOS

Con el objetivo de afianzar las tecnologías de predicción, transformación y codificación entrópica para imágenes y secuencias de video, estudiadas en las sesiones teórica, en la presente práctica el alumn@ deberá programar su propio codificador de imágenes en algún lenguaje de alto nivel como Matlab o Python.

El diseño debe incluir funcionalidades y mejoras sobre el Perfil Baseline del estándar JPEG, que es el utilizado de modo masivo en todos los dispositivos de captura, comunicaciones, y redes sociales en las últimas décadas. Estas mejoras pueden incluir distintos tamaños de bloques alternativos al de 8x8, modos de predicción basadas en pixel y en bloque, distintos tipos y tamaños de transformada, cuantificación perceptual alternativa a la cuantificación uniforme.

El diseño debe incluir de **modo obligatorio** los bloques funcionales mostrados en el siguiente esquema, excepto el bloque sombreado que representa el codificador Huffman (opcional):



FUNCIONALIDADES OBLIGATORIAS

— Preprocesador:

- Conversión al espacio de color RGB al espacio YCbCr con submuestreo de croma 4:2:0
- En caso de que sea preciso, se debe ajustar la resolución de la imagen a un múltiplo entero del tamaño del bloque especificado.
- Particionado de la imagen en bloques $B(i, j)$ de tamaño 8x8 configurable.

- El particionado se aplicará a cada una de las componentes $Y(i, j)$, $Cb(i, j)$ y $Cr(i, j)$ de la imagen, y los siguientes procesos se aplicarán de modo independiente a cada una de ellas.
- Etapas de Predicción Intra-frame: se diseñará un predictor $P(i, j)$ basado en bloque con un mínimo de 2 modos. Como resultado se obtendrá un bloque de residuo $R(i, j)$.
- Etapas de transformación 2D-DCT: implementación de la 2D-DCT. Se puede utilizar la función de Matlab " $dct2()$ ". El resultado de la transformación será un bloque $C(u, v)$.
- Etapas de cuantificación: implementación de un cuantificador uniforme cuyo valor debe ser configurable. El resultado de la cuantificación será un bloque $Z(u, v)$.
- Etapas de escalador: implementación de la inversa de la cuantificación. El resultado de la cuantificación inversa será un bloque $Cq(u, v)$.
- Etapas de transformación 2D-DCT inversa: implementación de la inversa de la 2D-DCT. El resultado de la transformación inversa será un bloque $Rq(i, j)$.
- Reconstrucción del bloque decodificado: suma del bloque de predicción $P(i, j)$ con el bloque decodificado $Rq(i, j)$. El resultado de este proceso será un bloque $Bq(i, j)$.
- Post-procesador: la suma de los bloques decodificados $Bq(i, j)$ permitirá, reconstruir las componentes $Yq(i, j)$, $Cbq(i, j)$, $Crq(i, j)$. Se deberá implementar la conversión de dichas componentes al espacio de color RGB para su visualización como una variable RGB con la función de Matlab " $imshow$ ".
- Medida de calidad: se implementará la medida de calidad objetiva PSNR, entre la imagen original YCbCr y la decodificada YCbCrq, de modo independiente para cada una de las componentes, y se mostrará por pantalla el valor obtenido de PSNR(Y), PSNR(Cb), PSNR(Cr).

FUNCIONALIDADES OPCIONALES

- Preprocesador:
 - Particionado de la imagen en bloques $B(i, j)$ de tamaño $N \times N$ configurable, para valores de $N=4$ hasta 128, siempre potencias de 2.
- Etapas de Predicción Intra-frame: modos de predicción intra-frame basado en bloque adicionales a los 2 modos obligatorios, así como opción de predicción basada en pixel.
- Etapas de transformación 2D-DCT: implementación de la 2D-DCT de modo compacto, no basados en la función de Matlab " $dct2()$ " y ajustables al tamaño del bloque N .
- Etapas de cuantificación: implementación de un cuantificador perceptual.
- Etapas de codificación entrópica: implementación de un codificador Huffman de los coeficientes transformados $Z(u, v)$ incluyendo una etapa de Zig-Zag, y la escritura del bitstream en un fichero. Se contabilizará el número de bytes generados por el codificador con el objetivo de presentar por pantalla la relación de compresión obtenida.

IMPLEMENTACIÓN DEL CODIFICADOR

Como guía para el diseño del codificador, se ha proporcionado en la actividad un script en Matlab que sirve como modelo de la arquitectura del codificador “Codificador_JPEG-X.m”.

Para la prueba del codificador se pueden utilizar las imágenes de test proporcionadas en la práctica P1, o cualquier otra que no haya sido previamente comprimida (TIFF o PNG). La evaluación del codificador por parte del profesor se llevará a cabo con imágenes alternativas a estas en cuanto a resolución y características de textura y color.

EVALUACIÓN Y ENTREGA DEL DISEÑO

La evaluación de la práctica se realizará analizando y comprobando las funcionalidades del script del codificador implementado y subido al Aula Virtual, así como por una demostración y explicación del código diseñado, la cual se llevará a cabo de modo individual para cada miembro del grupo, y en la que se contestará a las cuestiones y dudas propuestas por el profesor.

Es imprescindible que el código entregado esté ampliamente comentado (%) con el objetivo de facilitar la comprensión de este. La entrega de una memoria de la práctica es opcional, pero se valorará en caso de ser entregada.

A partir del día **20 de abril**, los grupos que hayan entregado la actividad, solicitan al profesor la evaluación presencial de la práctica en el horario acordado por ambos.

- Implementación de bloques funcionales obligatorios: **5 PUNTOS**
- Implementación opcional del predictor Inter-frame: **1 PUNTOS**
- Implementación de una cuantificación perceptual: **1 PUNTO**
- Implementación de la transformada 2D-DCT de modo compacto: **1 PUNTO**
- Implementación de las funcionalidades para que todo el codificador funcione con cualquier tamaño de bloque N (ente 4 y 128): **2 PUNTOS**
- Implementación del bloque de codificación Huffman: **2 PUNTOS**
- La implementación de todas las opciones permitirá obtener dos puntos adicionales por encima de la puntuación máxima (**12/10 PUNTOS**).

