Universidad Rey Juan Carlos

Estándares de Comunicación de Audio y Vídeo (ECAV)

PRáctica 2

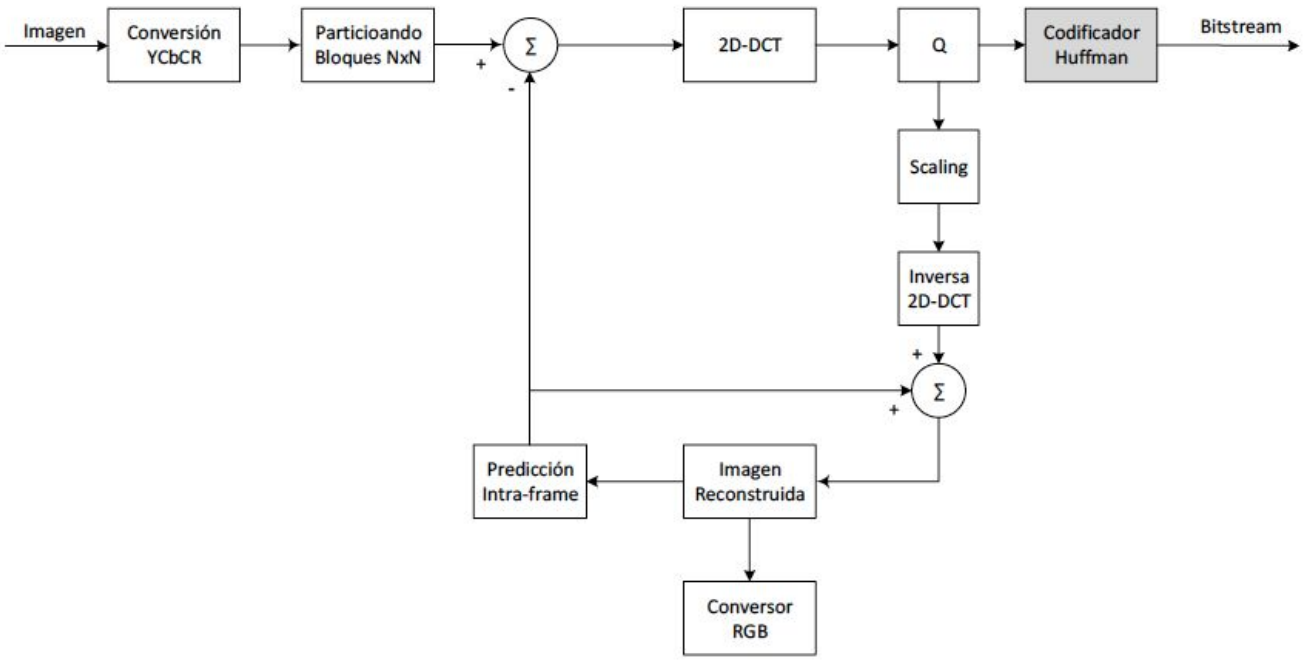
IMPLEMENTACIÓN DE UN CODIFICADOR AVANZADO DE IMÁGENES

Curso 2019-20

Participantes del grupo	
Yolanda Lillo Mata	
Elena Mª del Río Galera	
GRUPO	G10
FECHA	18/03/2020

1. OBJETIVOS

Describe brevemente el objetivo de la práctica.



FUNCIONALIDADES OBLIGATORIAS

Describe el código que has implementado para cada bloque funcional del codificador.

- Preprocesador: Para este bloque necesitamos Conviertir la imagen RGB al espacio de color YCbCr, ajustar la resolución de la imagen a un múltiplo entero del tamaño del bloque especificado, Particionar la imagen en bloques B(i,j) de tamaño 8x8 configurable. El código que hemos utilizado se puede consultar en el archivo de MATLAB adjunto.
- Etapa de Predicción Intra-frame: inicializamos el bloque predictor P(i,j) Como resultado se obtendrá un bloque de residuo R(i,j).

- Etapa de transformación 2D-DCT: implementación de la 2D-DCT. Calculo la Transformada bidimensional del Residuo C= dct2(R).

- Etapa de cuantificación: implementación de un cuantificador uniforme cuyo valor debe ser configurable.Cuantifico los coeficientes transformados del siguiente modo: Z = floor (C/Q); Adicionalmente utilizamos la codificación Huffman.
- Etapa de escalador: implementación de la inversa de la cuantificación.
- Etapa transformación 2D-DCT inversa: Implementación de la inversa de la 2D-DCT. Usamos la función de matlab idtc2().
- Reconstrucción del bloque decodificado: suma del bloque de predicción P(i,j) con el bloque decodificado Rq(i,j). Lo que hacemos es interpolar las componentes de Cb y Cr decodificadas Cbq y Crq a 4:4:4 con imresize().
- Post-procesador: Sumamos los bloques decodificados para reconstruir las componentes YCbCr y implementamos la conversión de dichas componentes al espacio de color RGB para su visualización en color (RGB) con la función imshow (función utilizada siempre que queremos mostrar la imagen)
- Medida de calidad: Para calcular la medida de calidad objetiva, más conocida como PSNR. Primero calculamos cada componente de la imagen original (YCbCr) y de la decodificada (YCbCrq). Después usando la fórmula del PSNR de cada componente para obtener su valor por separado.

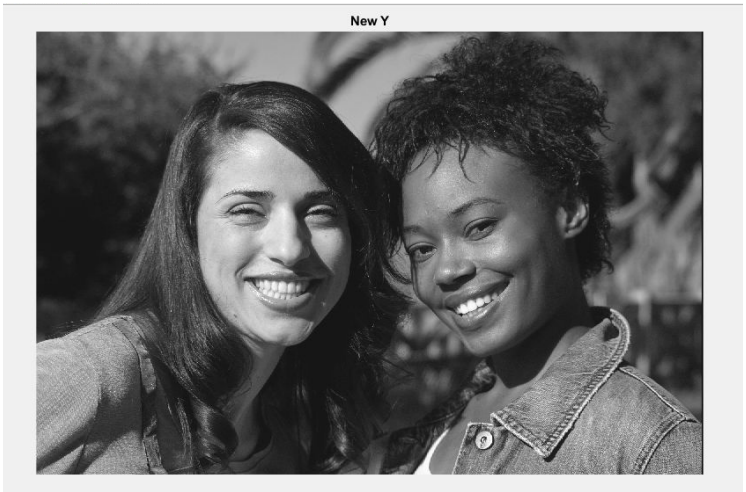
Funcionalidades opcionales

Describe el código que has implementado para cada bloque funcional opcional del codificador.

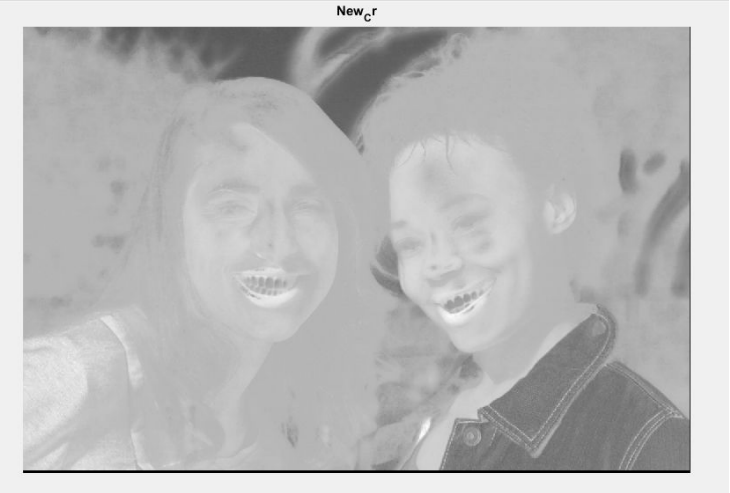
- Preprocesador:
- Etapa de Predicción Intra-frame:
- Etapa de transformación 2D-DCT:
- Etapa de cuantificación:

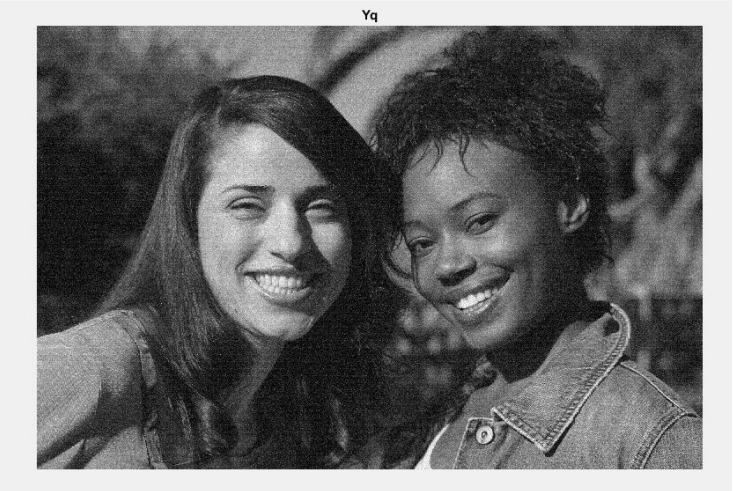
Implementación del codificador

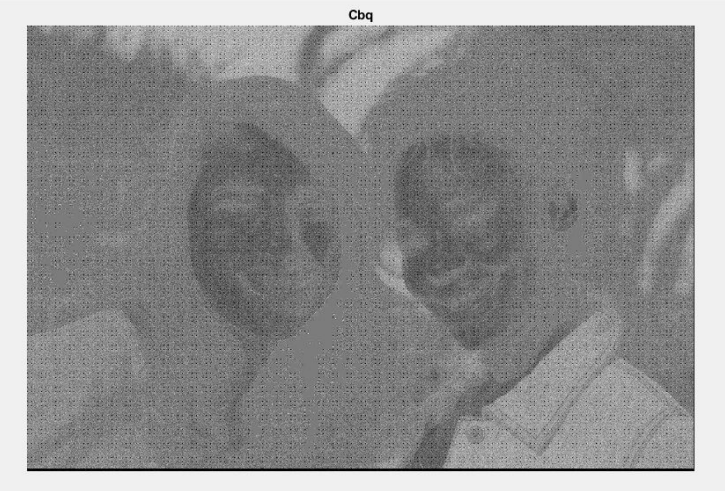
Describe los resultados obtenidos con el codificador que has implementado.

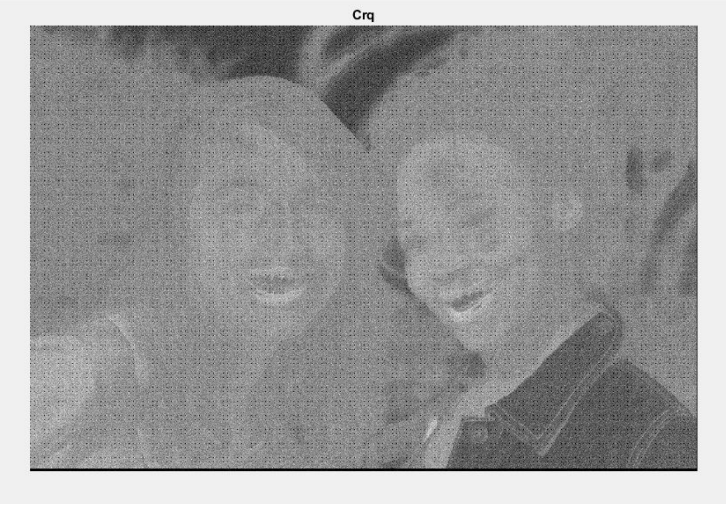

New Y

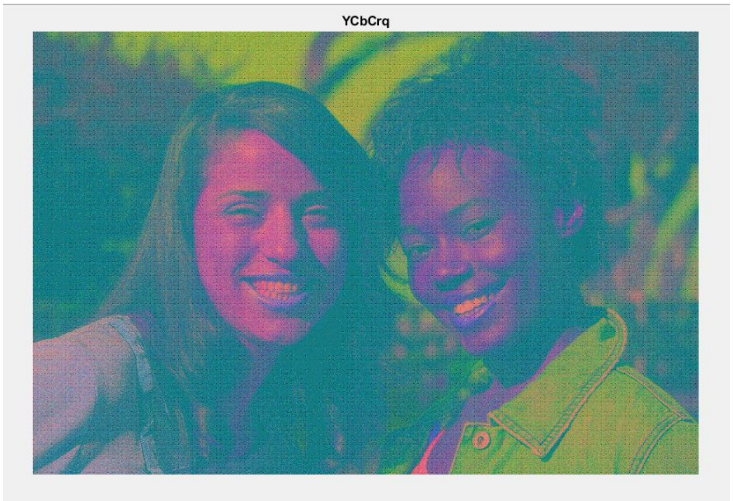

New$_C$b


New$_C$r


Yq


Cbq


Crq




```
>> Codificador_JPEG
PSNR(Y)= 22.991282
PSNR(Cb)= 23.806379
PSNR(Cr)= 23.754471
```

Imagen original

En estas imágenes podemos ver como afecta a nuestra imagen cada uno de los pasos que hemos ido realizando, podemos concluir que nuestra imagen decodificada difiere mucho de la imagen original, vemos nuestra imagen original pero con ruido.En cuanto a la imagen de arriba, respecto a los valores del PSNR (Peak Signal-To-Noise Ratio) , sabiendo que esto es la métrica más utilizada para la comparación de las prestaciones de los codecs de vídeo y un modo de indicar el error cuadrático medio en dB; Partiendo de que cuanto mayor es el valor del PSNR mayor es la calidad del codificador . Podemos decir que nuestro codificador tiene una calidad baja, tirando a un valor en torno a 23-24 dB, podemos decir que es bajo, lo que nos lleva a concluir que nuestro codificador tiene una calidad baja, entre 23 y 24.