

COMPRESION MULTIMEDIA - GUION DE PRACTICAS 3

AUDIO, IMÁGENES Y MAPAS DE COLORES EN MATLAB

1.- Escribe una función `testdct(fname)` que:

- Lea un archivo de imagen `fname` y genere una matriz truecolor RGB (el archivo puede estar en formato BMP o JPEG, y el tipo de color original puede ser indexado o truecolor)
- Convierta la matriz RGB leída del espacio de color RGB al espacio YCbCr
- Amplíe sus dimensiones a múltiplos de 8 píxeles
- Calcule la transformada DCT de la matriz ampliada en bloques de 8x8 píxeles
- Calcule la transformada inversa iDCT, también por bloques, a partir de la matriz de coeficientes obtenida en el paso anterior
- Convierte la matriz resultante al espacio RGB
- Reduzca sus dimensiones al tamaño original
- Almacene la matriz de imagen en un nuevo archivo BMP
- Muestre el error cuadrático medio MSE producido por la doble conversión y la mayor diferencia entre píxeles homólogos.

Para escribir la función, sigue las indicaciones dadas en la presentación de la práctica y usa las funciones auxiliares disponibles en la librería: `imlee`, `imdct`, `imidct` e `imescribe`. Aplícala al menos a cinco de los archivos de imagen (`Img00.bmp`, `Img01.bmp`, ... `Img20.jpg`) disponibles en la librería, e indica el máximo MSE obtenido y la máxima diferencia entre píxeles homólogos. ¿A qué se debe el error cometido?

2.- Analice el script `testQuantDCT8x8.m` y haga lo siguiente:

- Explique qué es lo que hace.
- Asigne a `escalon` el valor 1. Compare visualmente los valores originales y recuperados. Explique por qué son diferentes. Explique la situación para valores de escalón 0.5, 2, 5, ...
- Asigne a `escalon` varios valores, como 1, 2, 5, ... Compare visualmente el valor de `dct` y `quant`. ¿Cuándo se hacen nulos los valores cuantizados?
- Descomente la línea que permite visualizar el contenido de `seleccionados'` y asigne a `escalon` el valor `escMin` *para el resto de los apartados*.
- Para `umbral=0.5` ¿Cuál es el valor del escalón de cuantización? ¿Cuántos valores no nulos hay en la matriz `quant`? ¿Por qué? ¿Cuántos coeficientes descuantizados, `desquant`, quedan? Compare la imagen recuperada con los valores recuperados ¿Cuántas matrices base se están utilizando para recuperar la imagen? ¿por qué?
- Responda a las preguntas del apartado anterior para valores de `umbral` 0.3, 0.2, 0.1, 0.05 y 0.01.
- Como consecuencia de los dos apartados anteriores ¿Cuántos coeficientes transformados considera que son suficiente para codificar la imagen original sin perder demasiada calidad? Si sumara las 3 matrices base más relevantes ¿Qué figura obtendría? Tiene la 64 imágenes base de la DCT en las transparencias.
- Volviendo a `umbral=0.5` ¿cuál es el valor de los píxeles de la imagen recuperada? Teniendo en cuenta la cantidad de coeficientes transformados que han sido retenidos ¿podría predecir ese valor a partir de la imagen original?
- Renombre a la variable `q` y llámela `p` para los siguientes apartados.
- Asigne a `umbral` los valores 0.5, 0.3, 0.2, 0.1, 0.05 y 0.01. Para cada valor ¿Cuál es el valor del escalón de cuantización? ¿Cuántos valores no nulos hay en la matriz `quant`? ¿Por qué? ¿Cuántos coeficientes descuantizados, `desquant`, quedan? Compare la imagen recuperada con los valores recuperados ¿Cuántas matrices base se están utilizando para recuperar la imagen? ¿por qué?
- Como consecuencia del análisis anterior ¿Cuántos coeficientes transformados considera que son suficiente para codificar la imagen original sin perder demasiada calidad?
- Contraste su percepción con el valor del error cuadrático medio entre la imagen original y la recuperada.
- Justifique por qué los resultados para estas dos imágenes son tan diferentes.
- Construya otras imágenes `p` de 8x8 y prediga cómo serán sus coeficientes transformados y su recuperación para distintos umbrales y escalones

3.- Escribe una función `testquant(fname, caliQ)`, basada en la función `testdct(fname)`, que:

- Lea un archivo de imagen `fname` y genere una matriz truecolor RGB (el archivo puede estar en formato BMP o JPEG, y el tipo de color original puede ser indexado o truecolor)
- Convierta la matriz RGB leída del espacio de color RGB al espacio YCbCr
- Amplie sus dimensiones a múltiplos de 8 píxeles
- Calcule la transformada DCT de la matriz ampliada en bloques de 8x8 píxeles
- Cuantice la transformada DCT aplicando la tabla de cuantización estándar JPEG, multiplicada por un factor de calidad `caliQ` (entero positivo mayor o igual que 1)
- Descuantice la transformada DCT cuantizada de la imagen, aplicando la misma tabla de cuantización y el mismo factor de calidad
- Calcule la transformada inversa iDCT, también por bloques, a partir de la matriz de coeficientes obtenida en el paso anterior
- Convierte la matriz resultante al espacio RGB

- Reduzca sus dimensiones al tamaño original
- Almacene la matriz de imagen en un nuevo archivo BMP
- Muestre el error cuadrático medio MSE producido y la mayor diferencia entre píxeles homólogos.

Para escribir la función, sigue las indicaciones dadas en la presentación de la práctica y usa las funciones auxiliares disponibles en la librería. Aplícala al menos a cinco de los archivos de imagen (Img00.bmp, Img01.bmp, ... Img20.jpg), disponibles en la librería, usando al menos tres factores de calidad para cada una (50, 100 y 500). **¿En qué proporción** aumenta en promedio el MSE para un factor de calidad dado? ¿A qué se debe el aumento del MSE?

4.- (Opcional, pero recomendable para afianzar conceptos)

Escribe una función `wavtest(fname)` que:

- Lea un archivo WAV, cuyo nombre '`<file>.wav`' se da como argumento `fname`
- Lo reproduzca tres veces: a velocidad normal, más lenta y más rápida
- Visualice gráficamente todos los canales del archivo, cada uno en una ventana distinta
- Lo convierta a formato monoaural, promediando los distintos canales
- Almacene los 5 primeros segundos de señal del sonido mono en otro archivo llamado '`<file>short.wav`'
- Como ejemplo de archivo, se puede usar `hceste.wav`, proporcionado en la librería