

TRANSMISIÓN DE DATOS 2008/08		
Práctica 4: Codificación Transformacional		Grupo
		Puesto
Apellidos, nombre		Fecha
Apellidos, nombre		17/19/20 Nov. 2008

El objetivo de esta práctica es familiarizar al alumno con la codificación mediante transformaciones, en particular con la DCT (Discrete Cosine Transform – Transforma Discreta del Coseno).

Para llevar a cabo la práctica, desarrolle cada ejercicio en un fichero Matlab dentro del directorio P4.

Se proporcionan, accesibles desde la página de la asignatura en el fichero CodTrans.zip:

- o `cod_transf_ima.m`: se trata de una guía para realizar la práctica. Se recomienda usarla como plantilla, pero no olvidar que está escrita como pseudo-código, de forma que no compila directamente. Además, proporciona ya en Matlab las matrices A (matriz de datos) y Q (matriz de cuantificación) vistas en teoría.
- o `Lenna256.bmp` y `Lenna100.bmp`: contiene la imagen *Lenna* en formato bitmap en dos tamaños (256x256 y 100x100 píxeles).

Justo antes de finalizar la práctica, comprima el proyecto en un fichero `TxDatosP4GXzz.zip` (siendo X el grupo –A, B ó C–, y zz el número de pareja) conéctese al sistema de entrega de prácticas de la Intranet y entréguelo en el grupo que corresponda (A, B o C). Guárdese adicionalmente una **copia personal**, para posible reutilización del código en prácticas posteriores. Recuerde borrar su trabajo del ordenador del puesto de prácticas.

2.1 Codificación Transformacional mediante la DCT

La codificación mediante DCT permite cambiar a un espacio de representación de tipo frecuencial en el cuál la energía se concentra en unos pocos valores. Generalmente se aplica en bloques de 8x8 píxeles obteniendo conjuntos de 64 valores que, tras los pasos de transformación y cuantificación se recorren en zig-zag como paso previo a la codificación mediante un codificador entrópico (generalmente *Huffman* o *Aritmético*) y extensión de fuente (generalmente de carreras de ceros y marca fin de bloque –EOB–) de los 64 coeficientes de la DCT cuantificada. La decodificación sigue los pasos inversos.

En esta práctica se llevará a cabo el proceso de calcular la DCT, su cuantificación y posterior codificación con un codificador *Huffman* y cuantificación uniforme.

2.1.1 Ejercicio 1: Desarrollo de un codificador DCT de imágenes

A partir de `cod_transf_ima.m` escriba el código de un codificador DCT, que será guardado en `DCT_basica.m`

Calcule primero el codificador para el bloque A , de 8x8, proporcionado en el fichero.

En primer lugar genere la DCT. Para ello haga uso de la función `dct` de Matlab. Esta función realiza la DCT unidimensional, por lo que tendrá que hacer uso de la propiedad de separabilidad de la DCT, que permite calcular la DCT bidimensional sin más que aplicar la DCT primero por filas y luego por columnas (o viceversa). También es posible el cálculo mediante la utilización de la función `dct2` en caso de que la versión de Matlab instalada en el laboratorio lo permita.

Escriba los 64 coeficientes de la DCT calculada

DCT	1	2	3	4	5	6	7	8
1								
2								
3								
4								
5								
6								
7								
8								

A continuación cuantifique la DCT mediante la división por la matriz de cuantificación Q y el posterior redondeo.

Escriba los 64 coeficientes cuantificados

Fint	1	2	3	4	5	6	7	8
1								
2								
3								
4								
5								
6								
7								
8								

Escriba la extensión de fuente ([números de ceros, símbolo] y EOB cuando solamente queden ceros) que se pasaría a la fase de codificación entrópica.

A continuación, calcule el bloque reconstruido tras aplicar a la señal recibida (tras la decodificación entrópica que no realizaremos) una multiplicación por la matriz Q.

DeFint	1	2	3	4	5	6	7	8
1								
2								
3								
4								
5								
6								
7								
8								

Calcule después la señal reconstruida, aplicando la IDCT (mediante al función Matlab `idct` y la propiedad de separabilidad o bien `idct2`).

Escriba los 64 coeficientes de la IDCT calculada

IDCT	1	2	3	4	5	6	7	8
1								
2								
3								
4								
5								
6								
7								
8								

Este bloque es la señal que se reconstruye en recepción. Calcule la señal de error y su potencia de ruido.

Escriba los 64 coeficientes de error.

error	1	2	3	4	5	6	7	8
1								
2								
3								
4								
5								
6								
7								
8								

Potencia de ruido	
--------------------------	--

2.1.2 Ejercicio 2: Codificación DCT de imágenes

A partir de `DCT_basica.m` y `cod_transf_ima.m` haga un en `DCT_Imagen.m` con el siguiente interfaz. Recuerde salvar las imágenes resultantes a fichero como hasta ahora.

```
function [Perror] = DCT_Imagen (imagen)
```

```
#Perror: Potencia del error.
```

```
# imagen: imagen de entrada
```

Ejecútelos sobre la imagen 'Lenna256.bmp' y visualice los resultados. Coméntelos:

2.2 Efecto del cuantificador en las imágenes

2.2.1 Ejercicio 3: Desarrollo del efecto de la cuantificación en la codificación de imágenes

A partir de `DCT_basica.m` escriba una **función** en Matlab (`DCT_N.m`) que tras la cuantificación a 256 niveles que se realiza mediante la división por la matriz Q , incluya un cuantificador PCM uniforme de n bits (haga uso del código de la Práctica 2, aplicando como valor de sobrecarga el máximo, en valor absoluto, de cada bloque). Se habrá de respetar la siguiente sintaxis:

```
function [Perror] = DCT_N(imagen,n)

#Perror: Potencia del error.

# imagen: imagen de entrada

#n: número de bits usados para la cuantificación.
```

Compare (objetiva y subjetivamente) los resultados con los resultados de aplicar la cuantificación de N niveles sin la DCT (escriba este programa basándose en los de la Práctica 3 y guárdelo en `cuantif_N.m`).

```
function [Perror] = cuantif_N (imagen,n)

#Perror: Potencia del error.

# imagen: imagen de entrada

#n: número de bits usados para la cuantificación.
```

Para la “imagen” A, ejecute el programa para $N=64$ (6 bits), $N=32$ (5 bits), $N=16$ (4 bits), y $N=8$ (3 bits) y calcule la potencia de ruido.

	Cuantif_N.m	DCT_N.m
Potencia de ruido (N=64)		
Potencia de ruido (N=32)		
Potencia de ruido (N=16)		
Potencia de ruido (N=8)		

2.2.2 Ejercicio 4: Efecto de la cuantificación en la codificación de imágenes

Con la imagen de entrada `Lenna256.bmp`, usando las funciones desarrolladas en el apartado anterior (`DCT_N.m` y `cuantif_N`):

Ejecute el programa para $N=64$ (6 bits), $N=32$ (5 bits), $N=16$ (4 bits), y $N=8$ (3 bits) y calcule la potencia de ruido.

	Cuantif_N.m	DCT_ N.m
Potencia de ruido (N=64)		
Potencia de ruido (N=32)		
Potencia de ruido (N=16)		
Potencia de ruido (N=8)		

Comente los resultados subjetivos

2.2.3 Ejercicio 5:

En este apartado se simulará la codificación entrópica utilizando las funciones implementadas en la práctica 1 (codificación *Huffman*) y el código desarrollado en los anteriores ejercicios. Escriba un fichero `Ejercicio5.m`, en el que se lleve a cabo la codificación mediante DCT y posterior división entre la matriz Q de la imagen proporcionada *Lenna100.bmp* (se utilizará la versión reducida de la imagen para minimizar el tiempo de proceso). A continuación se deberá codificar mediante codificación *Huffman* los resultados obtenidos. Al ser los datos a codificar una matriz del mismo tamaño de la imagen original con posibles valores negativos lleve a cabo previamente las siguientes operaciones sobre la matriz:

- Convierta la matriz a un vector fila utilizando de la función 'reshape' (ver ayuda Malab).
- Convierta los valores a positivos restando a las muestras el valor mínimo de las mismas.

A continuación realice la secuencia usual para la codificación *Huffman* (cálculo de probabilidades, códigos *Huffman* y sustitución de símbolos). ¿Cuántos bits/bytes son necesarios para codificar la imagen DCT? Compárelos con los necesarios para codificar directamente la imagen original mediante codificación *Huffman* (tal y como se hacía en la práctica 1) y con el tamaño de la imagen sin codificar.

Explique los resultados:

Dado que la imagen está compuesta de 10000 píxeles (100x100) calcule el promedio de bits por píxel -bpp- necesarios para codificar la imagen.

	Imagen sin codificar	Huffman	DCT+Huffman
Bpp			

Utilizando la función `cuantif_N` para la cuantificación de la imagen (sin DCT) lleve a cabo la cuantificación de la imagen utilizando el número de bits por muestra (píxel) que más se aproximen a los obtenidos para codificación *DCT+Huffman*. Compare las potencias de error obtenidas para la codificación de la imagen con los dos métodos así como los resultados subjetivos. Comente los resultados explicando las diferencias encontradas y el motivo de las mismas.