Práctica 3:
Compresión de imágenes JPEG. Coeficientes AC

1. Introducción

El objetivo de la práctica es continuar la realizada en la sesión anterior incorporando ahora los coeficientes de alterna en el codificador JPEG. Comienza con el cuarto y quinto paso, que tienen como meta realizar una función totalmente operativa en la que, dada una imagen cualquiera, se genere el fichero jpeg correspondiente, para lo cual, además del trabajo realizado en los pasos anteriores, se deberá generar la cabecera adecuada, realizar el zigzag de los coeficientes de alterna, determinar los runlengths, codificarlos, etc.

Al igual que en la práctica anterior, los resultados de la práctica serán las funciones mijpeg_pasoN y se deberán enviar por email al profesor de laboratorio justo antes de finalizar cada sesión.

2. Paso 4. Codificación de los coeficientes AC

Ahora se utilizará el fichero `demo_paso4.m` para la generación de la imagen imC que tiene bloques con frecuencias horizontales y verticales crecientes y diversos colores.

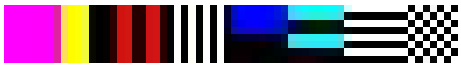

Figura 1: Imagen con contenido de alterna

Realiza una copia de la función `mijpeg_paso3` *con el nombre* `mijpeg_paso4` y amplíala para que codifique también los coeficientes AC de modo que la codificación JPEG sea completa. Se os facilita la funcion `jpeg_runlen.m` que obtiene la información de los pares (runlen, coef) de la matriz de coeficientes. Leed el help de la función para conocer más detalles. En el fichero `mijpeg_p4_ayuda.m` teneis indicado donde comienzan las tareas del paso 4, así como un breve recordatorio de lo que hay que hacer. Para ayudar a comprender el proceso y poder implementar la función, se presenta el diagrama de flujo de la figura 2.

Este paso es el más complicado. Si se tuviesen dificultades, se puede comprobar esta parte utilizando los ejemplos que se han preparado. Para ello se deben copiar las líneas correspondientes del fichero `mijpeg_p4_ayuda.m` al nuestro y descomentarlas. Los `bits_AC` resultantes se muestran en el cuadro 1.

Probad la función con varias imágenes reales (si no son múltiplo entero de 8, tomad una porción que sí lo sea). Sobre la imagen Lena de 256x256 en color, calcular el error cuadrático medio y el factor de compresión respecto al fichero .tif.

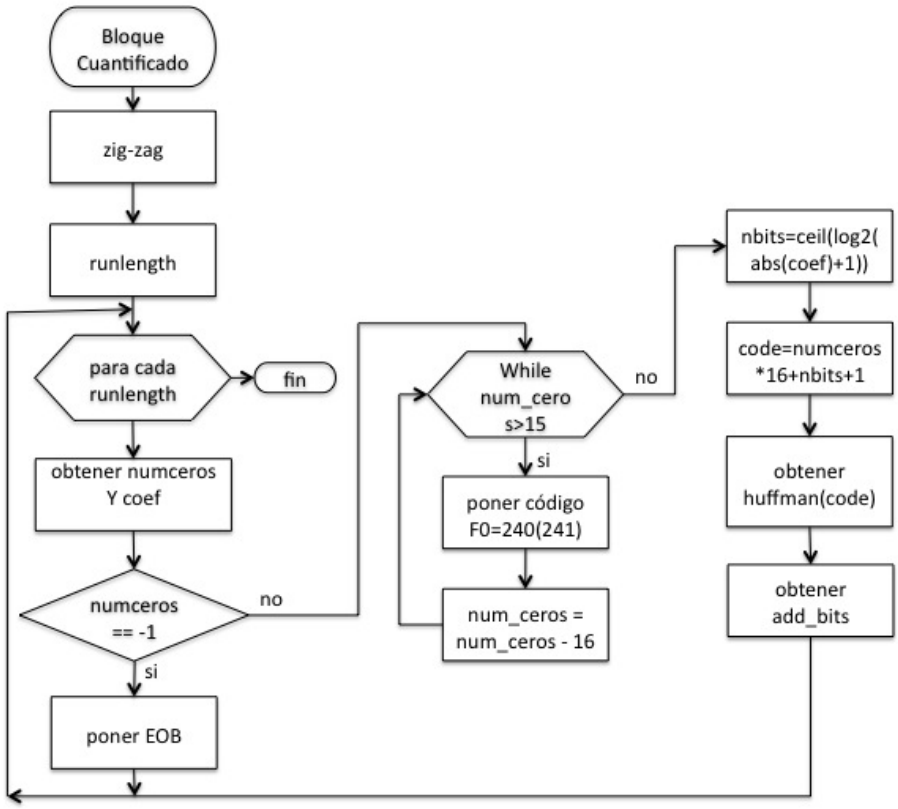

Figura 2: diagrama de flujo codificación bloques AC

3. Paso 5. Factor de Calidad

Ahora se utilizará el fichero `demo_paso5.m`. Realizar una copia de la función `mijpeg_paso4` con el nombre `mijpeg_paso5`. Introducir un parámetro de calidad en la función `mijpeg_paso5(im, nombrefich, calidad)`. Debe ser un entero entre uno y cinco, máxima calidad el cinco. Para implementar esta funcionalidad, en JPEG se modifican las tablas de cuantificación. A menor calidad, mayores valores de la tabla de cuantificación y por tanto al dividir los coeficientes de la DCT por estos valores menos bits se utilizarán y se comete mayor error. Para modificar las tablas `qty, qtc` se procederá de la siguiente manera :

- Si calidad es 5, qty = qty/2;
- Si calidad es 4, qty = qty;
- Si calidad es 3, qty = qty*1.5;
- Si calidad es 2, qty = qty*2;
- Si calidad es 1, qty = qty*3;

Para la tabla qtc se realiza la misma operación. Recordar que los valores resultantes deben ser enteros sin signo de 8 bits; para ello se puede hacer uso de la función `uint8( )` de matlab, `qty = uint8(qty )`. Una vez insertada la tabla en la cabecera hay que volver a pasar

runlength	Y	C
[-1 0]	1010 (EOB)	00 (EOB)
[3 4; -1 0]	1111 1111 0101, 100, 1010 (EOB)	1111 1110 00, 100, 00 (EOB)
[17 4; -1 0]	1111 1111 001(F0) 1111 001, 100 1010 (EOB)	1111 1110 10(F0) 1111 0110, 100 00 (EOB)
[32 4; -1 0]	1111 1111 001(F0) 1111 1111 001(F0) 100, 100 1010 (EOB)	1111 1110 10(F0) 1111 1110 10(F0) 1010, 100 00 (EOB)
[0 3; 32 4; -1 0]	01, 11 1111 1111 001(F0) 1111 1111 001(F0) 100, 100 1010 (EOB)	100, 11 1111 1110 10(F0) 1111 1110 10(F0) 1010, 100 00 (EOB)

Cuadro 1: Bits resultantes de la codificicación de los ejemplos del paso 4.

a `double` porque la operación `./` no está definida en matlab para el tipo `uint8` , `qty = double(qty)`

La tabla utilizada se debe insertar en la cabecera. El proceso para insertarla debeis averiguarlo a partir de la información disponible de JPEG. En la norma ITU T81, página 32, tabla B.1, podeis ver que el marcador que se debe utilizar es *ffdb* . La sintaxis está en el punto B.2.4.1 *Quantization table-specification syntax*, página 39 de la norma.

En nuestro caso particular, la posición de la etiqueta *ffdb* en la variable `cabecera` se encuentra en la posición 67 (byte ff) y posición 68 (byte db) (que os suministramos en el fichero mijpeg_esqueleto).

Poned en la función `mijpeg_paso5` la siguiente línea de código para que, si no se especifica parámetro de calidad (el número de argumentos es 2), tome valor 4 y se utilicen las tablas por defecto, `if(nargin == 2) calidad =4; end`

La función `demo_paso5.m` toma la imagen `lena_256_color.tif` y la codifica con las cinco calidades. Si dispone de tiempo realice la prueba también con la imagen `wikio.tif`

4. Paso 6. Tamaño arbitrario (Opcional)

Cuando la imagen no es múltiplo entero de 8, la codificación se puede realizar creando previamente una imagen múltiplo de 8 mayor que la original y copiando en ella la original a partir de la esquina superior izquierda. En las etiquetas de tamaño se ponen los valores de la imagen original.