

## Práctica 3

### Teoría de la Información. Codificación de audio

#### Resumen de la práctica

Esta práctica continúa el tema de la anterior, centrándonos ahora en el caso de archivos de audio.

## 1 Introducción

La codificación de audio es otro ejemplo típico de codificación con pérdidas. Al igual que como vimos para JPEG, para el caso de codificar ficheros de audio también es habitual trabajar en algún dominio transformado, y en concreto nosotros vamos a volver a utilizar la transformada DCT, ya que es la que se utiliza en los estándares de codificación MPEG.

## 2 Codificación de un fichero de audio

La codificación de ficheros de audio se parece mucho a lo que hemos hecho para el fichero de imagen. El procedimiento completo sería el siguiente:

1. Procesamos la señal en tramas de 2048 muestras con un solape de 1024 muestras entre trama y trama (para esto es útil la función `buffer`).
2. Enventanamos cada trama utilizando una ventana Hamming (hay una función `hamming` que lo hace).
3. Calculamos la DCT de la trama enventanada.
4. Se aplica la cuantificación, codificación entrópica, o lo que se desee.
5. Se aplica el inverso de todos los procesos del paso anterior.
6. Transformada DCT inversa de cada trama.
7. Se reconstruye el fichero uniendo las tramas mediante el procedimiento de solape-suma (la primera mitad de la trama se suma con la segunda mitad de la trama anterior).

## 3 Desarrollo de la práctica

Para leer el fichero de audio utilizad el comando: `[x,fs] = audioread('brahms_mono.wav','native');`. De esta forma garantizamos que los valores de las muestras de audio se leen como enteros de 16 bits entre  $-32767$  y  $32767$  en lugar de como números de punto flotante entre  $-1$  y  $1$ . Después se puede convertir la variable a `double` para que sea más cómodo trabajar a partir de ese momento. El archivo de audio sólo tiene un canal (es mono), aunque si tuviésemos un archivo en estéreo el procedimiento sería el mismo que el que vamos a hacer pero repetido para cada uno de los dos canales (L y R).

Estimad el tamaño del archivo de audio en Kbytes en bruto asumiendo una codificación de 16 bits por muestra.

Es recomendable tener alguna función que, dado un vector de datos, calcule el diccionario Huffman correspondiente así como la longitud de palabra media.

### 3.1 Codificación en el dominio del tiempo

En primer lugar, al igual que ya hicimos para la imagen, vamos a ver qué ocurriría si intentamos codificar el fichero de audio en el dominio del tiempo, sin realizar la transformada.

Cuantificad la señal de audio utilizando un escalón de cuantificación fijo (512). A partir de la señal cuantificada, calculad la longitud de palabra media según Huffman y con ella el tamaño del archivo en kBytes.

Deshaced la cuantificación, y escuchad el archivo resultante.

### 3.2 Codificación en el dominio de la frecuencia

Vamos a comparar el resultado anterior con el que obtendríamos trabajando en el dominio de la frecuencia. Para ello, entramos la señal de audio como se indicaba antes (tramas de 2048 muestras solapadas al 50%). Calculamos la DCT de cada trama, y sobre este vector en el dominio transformado, realizamos la cuantificación con el mismo escalón que en el caso anterior y estimamos el tamaño del archivo en KBytes utilizando codificación Huffman. Finalmente, escuchamos el resultado final.

### 3.3 Un paso más allá: perfilado del ruido (voluntario)

En la vida real la cuantificación es un proceso mucho más complejo que como lo hemos hecho nosotros aquí. Lo que se hace es utilizar un modelo psicoacústico para estimar el umbral de audición dada una determinada trama de audio y a partir de ahí utilizar un escalón de cuantificación distinto por cada banda de frecuencias de forma que el ruido de cuantificación quede perfilado siguiendo la forma de ese umbral (y por supuesto por debajo de él, para que no se oiga).

Podemos hacer algo parecido, de forma que utilizaremos un escalón de cuantificación distinto (en lugar del 512 de antes) para cada trama de la señal. Para calcular el valor de ese escalón, podemos asumir que la potencia del ruido de cuantificación viene dada por  $\Delta^2/12$ , siendo  $\Delta$  el valor del escalón de cuantificación. A partir de ahí podemos diseñar un codificador que intente mantener una relación señal a ruido de un determinado valor (pongamos  $50dB$ ) para todas las tramas.

Una opción para mejorar esto todavía más es trabajar por bandas de frecuencia. Así, lo que haríamos sería calcular un escalón de cuantificación distinto para cada banda de frecuencia de cada trama. De esta manera se consigue perfilar el ruido de cuantificación mejor, adaptándolo a la forma del espectro de la señal. Las bandas de frecuencia pueden ser cualesquiera, aunque es bastante razonable trabajar con bandas en octavas, que se asemejan en parte al comportamiento del oído humano. Así, una banda iría de la muestra 1 a la 2, la siguiente de la 3 a la 4, luego 5-8, 9-16, 17-32, 33-64, etc. En este caso es posible disminuir el valor de la SNR dejándolo quizás en unos 30dB.

Con todo lo anterior, calcula el nuevo tamaño del archivo en Kbytes y escucha el resultado. ¿Qué conclusiones puedes sacar? ¿Se te ocurre alguna pega a esta idea?

## 4 ¿Qué entregar?

- Código de las funciones generadas