

Práctica: Codificación de información multimedia

Universidad Carlos III de Madrid, Multimedia, Curso 2022-2023

Introducción y objetivos

El procesado de información multimedia es un campo que ofrece una gran diversidad de aplicaciones. Una de las más directas es la codificación, que a grandes rasgos consiste en transformar una señal multimedia para que pueda ser utilizada y almacenada en dispositivos digitales.

En esta práctica se profundizará en los conceptos principales sobre codificación que se revisaron en las clases de teoría. El desarrollo de estrategias de codificación depende directamente de la fuente elegida. Por lo tanto, la práctica se divide en 4 partes, que coinciden con las fuentes que se han estudiado: voz, audio, imagen y vídeo.

Para algunas partes será necesario implementar porciones de código, mientras que para otras se requerirá ejecutar código ya implementado y reflexionar sobre los resultados producidos.

Los objetivos se pueden resumir en los siguientes puntos:

- Revisar los principales conceptos de codificación vistos en clase.
- Profundizar en dichos conceptos mediante el estudio de un conjunto de casos prácticos.

Entrega de la memoria

Importante: para la evaluación de este ejercicio, es necesario subir a Aula Global un archivo comprimido (en formato zip) que contenga lo siguiente:

- Un documento **en formato .pdf** siguiendo el formato que se establece en el archivo **plantilla_practica**, disponible en Aula Global. El documento entregado en el archivo comprimido en formato zip tendrá un **máximo de 8 páginas** y debe contener las respuestas a las preguntas formuladas en las actividades presentes en las próximas secciones. **Se recomienda muy encarecidamente proporcionar figuras e imágenes que ilustren los argumentos propuestos en las respuestas.**
- El material suministrado para la práctica, siguiendo la misma jerarquía de directorios e incluyendo los scripts que el grupo haya necesitado añadir o modificar.

Nota: Es suficiente con que un único miembro del grupo suba el proyecto en Aula Global.

Codificación de voz

El objetivo de este apartado es profundizar en las ideas principales estudiadas en clase sobre las características de la señal de voz y los vocoders. El material para realizar esta parte se encuentra en el directorio `cvoz`. Cambie el directorio de trabajo de Matlab a esta ubicación. Se proporcionan dos scripts (**`cvoz1.m`** y **`cvoz2.m`**) y un directorio con señales de voz (**`muestras_voz`**).

Abra el script **`cvoz1.m`**. El script trabaja con un fichero de voz cuyo path se proporciona en la variable **`afile`** (de momento, utilice el path `\./muestras_voz/sentence.wav`). Primero se utiliza la función **`audioinfo`** para obtener una estructura que contiene la información proporcionada en la cabecera del fichero de audio. Acceda a sus campos y explore la información que contienen.

P1. Obtenga la frecuencia de muestreo y el número de bits por muestra del fichero `sentence.wav`.

La función **`audioread`** carga las muestras del fichero de audio en un vector, llamado **`x`**. Finalmente, la función **`stft`** realiza un análisis enventanado de la señal de voz. Ejecute el script **`cvoz1.m`** con el fichero `sentence.wav`. La figura muestra tres gráficas. En la primera se puede visualizar la señal completa y la posición de la ventana que se analiza en las gráficas inferiores. En la segunda se observa la ventana analizada en el dominio temporal. Y en la tercera la misma ventana pero en el dominio de la frecuencia. Puede pulsar cualquier tecla para que la ventana avance por el fichero de voz. Para detener la ejecución del script, haga click en la ventana de comandos y pulse **`Ctrl+C`**.

P2. ¿Por qué el eje frecuencial de la gráfica del espectro va de 0 a 4000 Hz? Justifique su respuesta.

P3. Busque una trama de un sonido claramente sonoro. Indique las características que ha observado para llegar a esta conclusión (en tiempo y frecuencia).

P4. Busque una trama de un sonido claramente sordo. Indique las características que ha observado para llegar a esta conclusión (en tiempo y frecuencia).

Ahora ejecute el script **`cvoz2.m`**. Es idéntico a **`cvoz1.m`**, salvo que en la gráfica en el dominio frecuencial, se muestra además la envolvente espectral, superpuesta a la gráfica del espectro. El objetivo es estudiar las características espectrales de diferentes hablantes cuando pronuncian el mismo fonema. Se ha elegido un hablante femenino y otro masculino como ejemplo de voces perceptualmente distintas (`w_oo.wav` y `m_oo.wav`).

P5. Comente las similitudes y diferencias espectrales entre los hablantes de `w_oo.wav` y `m_oo.wav`. Céntrese en dos aspectos: frecuencia fundamental y posición de los dos primeros formantes. Justifique su respuesta.

Codificación de audio

El objetivo de este apartado es revisar los conceptos de codificación diferencial vistos en clase y aplicarlos a la codificación de audio mono-canal. El material para realizar esta parte

se encuentra en el directorio `caudio`. Cambie el directorio de trabajo de Matlab a esta ubicación. Se proporcionan dos scripts (**`cod_amp.m`** y **`cod_dif.m`**) y un directorio (**`muestras_audio`**) con dos señales de audio. Dichas señales han sido previamente digitalizadas con la misma frecuencia de muestreo y el mismo número de bits por muestra.

Se requiere comprimir el fichero `bassoon.wav` lo máximo posible, tratando de minimizar la pérdida de calidad. Para ello se han implementado dos codificadores sencillos:

- **Opción 1:** consiste en cuantificar la amplitud de la señal directamente, pero usando un número menor de bits por muestra.
- **Opción 2:** consiste en cuantificar el error de predicción (codificación diferencial), con un número menor de bits por muestra.

El primer paso será realizar un análisis preliminar de los ficheros a codificar. El script **`hist_comp.m`** calcula la distribución de valores de la señal para ambas alternativas. Ejecute el script para el fichero `bassoon.wav`. La figura muestra a la izquierda la distribución de valores usando la **opción 1** y a la derecha la distribución usando la **opción 2**.

A continuación se valorará el ahorro de bits de cada propuesta, utilizando el fichero `bassoon.wav`. Ejecute el script **`cod_amp.m`**, el cual realiza una recuantificación de los valores de amplitud de la señal de audio directamente, sin transformar dicha señal. Al cuantificar con menos bits por muestra que la señal original, se consigue un ahorro de bits. Asegúrese de que entiende el código del script. Modifique la variable `nb` para experimentar con distintos números de bits por muestra.

P6. ¿A partir de qué valor de `nb` el ruido de cuantificación empieza a ser audible?

Ejecute el script **`cod_dif.m`**, el cual realiza una transformación de la señal antes de realizar la cuantificación. Concretamente, utiliza un predictor para estimar el valor de la señal en cada instante y dicha estimación se resta al valor original. La señal resultante es el residuo de predicción, y es lo que se cuantifica. De nuevo, se reducirá el número de bits por muestra para conseguir un ahorro de bits. Asegúrese de que entiende el código del script. Modifique la variable `nb` para experimentar con distintos números de bits por muestra.

P7. ¿A partir de qué valor de `nb` el ruido de cuantificación empieza a ser audible? Compárelo con el resultado que obtuvo en **P6** y trate de explicar los resultados.

P8. ¿Qué información utiliza el predictor para estimar el siguiente valor de la señal de entrada? Justifique su respuesta.

Modifique los scripts **`cod_amp.m`** y **`cod_dif.m`** para que calculen el error cuadrático medio (MSE) entre la señal original y la reconstruida. Añada también el cálculo de la tasa de compresión.

P9. Obtenga el MSE y la tasa de compresión de cada propuesta, para el valor de `nb` correspondiente que obtuvo en **P6** y **P7**. Relacione estos resultados con `nb`.

Ahora ejecute **`hist_comp.m`** con el fichero `wng.wav`.

P10. Compare los histogramas que obtiene en este caso con los que obtuvo a partir del fichero `bassoon.wav`. ¿A qué se deben las diferencias? Justifique su respuesta.

Codificación de imagen

El objetivo de este apartado es mejorar la comprensión del estudiante en las bases sobre codificación de imagen usando transformadas. Para ello, se propone el análisis de un conjunto de casos prácticos utilizando la Discrete Cosine Transform (DCT) vista en clase. El material para realizar esta parte se encuentra en el directorio `cimagen`. Cambie el directorio de trabajo de Matlab a esta ubicación. Se proporcionan dos scripts (**`cimagen1.m`** y **`cimagen2.m`**) y un directorio (`data`).

Cargue el fichero `filters.mat` y visualice los 4 filtros que se proporcionan (utilice la función **`imshow`**). El objetivo de estos filtros es seleccionar una región concreta del espectro de frecuencias espaciales de la imagen. Al multiplicar la transformada de una imagen por uno de los filtros, se preservan las frecuencias que fueron multiplicadas por 1 y se eliminan las frecuencias que fueron multiplicadas por 0. De esta manera se puede analizar qué información de la imagen está recogida en las frecuencias que preservamos.

Ejecute el script **`cimagen1.m`**, el cual aplica a la imagen original cada uno de los filtros mencionados anteriormente y muestra el resultado tanto en el dominio espacial como en el dominio frecuencial. En la *figura 1* puede observar la imagen original y su DCT. En las siguientes se muestran las imágenes filtradas en el dominio original y el transformado.

P11. Observe el resultado de aplicar los cuatro filtros. Compare la imagen original con las filtradas. ¿Qué contenido visual se ha preservado en cada caso? ¿Qué contenido se ha perdido? Relacione lo que observa en las imágenes filtradas con su espectro. Justifique su respuesta.

En JPEG no se calcula la transformada de toda la imagen. En su lugar, la imagen se divide en bloques de 8x8 y en cada uno de ellos se calcula la DCT. Este método proporciona un mayor ahorro de bits. Ejecute el script **`cimagen2.m`** y visualice el resultado de calcular la DCT, esta vez en bloques de 8x8.

P12. ¿En qué zonas de la imagen hay bloques en los que solo las bajas frecuencias tienen valores significativos? ¿A qué se debe? ¿En qué zonas las altas frecuencias son más prominentes?

Codificación de vídeo

El objetivo de este apartado es reforzar los conocimientos del estudiante sobre estimación de movimiento en codificación de vídeo. Para ello, se va a analizar la utilidad de este método de predicción en una serie de secuencias de vídeo. El material para realizar esta parte se encuentra en el directorio `cvideo`. Cambie el directorio de trabajo de Matlab a esta ubicación.

El material de este apartado presenta los siguientes directorios:

- `videos`: contiene las secuencias que se van a utilizar en las preguntas.
- `toolbox`: contiene el código necesario para realizar la estimación de movimiento.
- `outputs`: tras ejecutar el código, contendrá los frames de las secuencias analizadas con los vectores de movimiento superpuestos.

A su vez, en el directorio raíz se proporciona el script **`cvideo.m`**, que llama a una función para calcular la estimación de movimiento en una secuencia que se pasa como parámetro. Puede usar `help` para obtener una descripción de la función y de sus parámetros.

IMPORTANTE: antes de poder ejecutar **`cvideo.m`** es necesario que compile el código de la toolbox de estimación de movimiento. Para ello ejecute el script **`compilar.m`**.

P13. Ejecute el script con la secuencia `tennis` y visualice el resultado. Fíjese en el momento en que hay un *zoom out*. ¿Qué patrón presentan los vectores de movimiento cuando se produce este evento? ¿Existen zonas de la imagen que no lo presentan? ¿A qué se debe? Justifique su respuesta.

P14. Ejecute el script con la secuencia `coastguard` y visualice el resultado. Existen zonas del agua en las que los vectores son capaces de seguir el movimiento y otras en las que no. ¿A qué se debe? ¿Por qué el barco no presenta vectores de movimiento? Justifique sus respuestas.