

Grado en Ciencia e Ingeniería de Datos



PRÁCTICA 2: JPEG

Versión 1. Última actualización 2/05/2022 13:00.

Contenidos

Instrucciones	1
Material necesario	2
Objetivo	2
Metodología	3
Funciones proporcionadas	3
Funciones a implementar y <i>live script</i> principal.....	4
Entrega	5
Valoración	5
Rúbrica de evaluación	5

Instrucciones

- La práctica se realiza de forma individual.
- La entrega se realiza a través de la tarea correspondiente en el Campus Virtual (Moodle).
- **La fecha límite de entrega se indica en la tarea de entrega.**



- No entregar las imágenes comprimidas en ningún caso. Incluir únicamente la imagen original que se carga en el *live script* principal.
- Es muy importante que cada práctica sea original.

Material necesario

- i** • En esta práctica necesitas Matlab con el *toolbox* de procesamiento de imagen. La licencia campus de Matlab en la UDC está accesible en <https://matlab.udc.es>

Objetivo

En esta práctica se propone la realización de una implementación básica de un codificador y un decodificador de JPEG de imágenes en escala de grises que únicamente considera los bloques de la transformada discreta del coseno (DCT) y de la cuantificación.

El rendimiento del códec JPEG se evaluará ejecutando el par codificador-descodificador variando el factor de calidad de compresión y, para cada valor de éste, se calcularán las métricas SSIM y MSE¹. Cada métrica se dibujará en una gráfica independiente y se comparará con el resultado proporcionado por la implementación de Matlab con `imwrite()`. La Figura 1 muestra las curvas de la SSIM para la imagen `peppers.png`. Ten en cuenta que la gráfica depende de la imagen seleccionada.

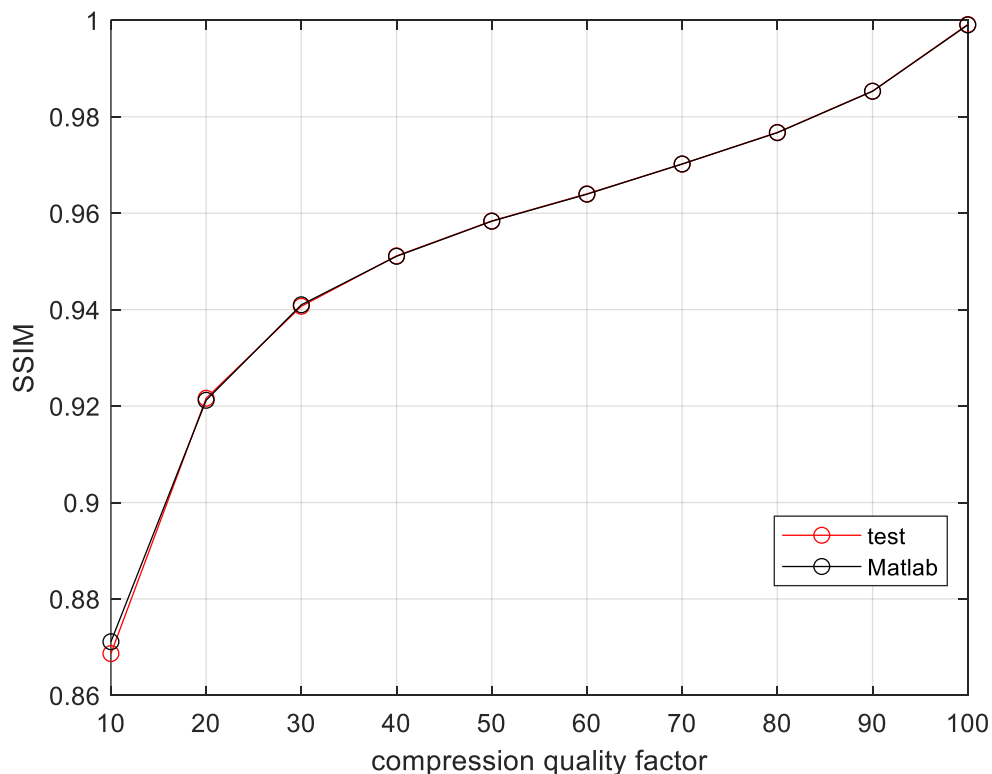


Figura 1: SSIM en función del factor de calidad de compresión para la imagen `peppers.png`.

¹ No se consideran métricas de compresión porque en la práctica no se implementa el codificador entrópico.

Metodología

Se implementará en un *live script* de Matlab el par codificador-descodificador JPEG con las siguientes características:

- El codificador y el descodificador asumirán imágenes con dimensiones múltiplos enteros de 8, de forma que la división en bloques de 8×8 píxeles sea sencilla. Se proporciona la función `resizeImageTo8()` que modifica las dimensiones de la imagen para que sean múltiplos enteros de 8.

i Ten en cuenta que la imagen que devuelve la función `resizeImageTo8()` deberá tomarse como la imagen de referencia para la evaluación de las métricas de calidad (SSIM y MSE).

- Se consideran imágenes en escala de grises únicamente. Si la imagen seleccionada es en color, se debe convertir primero a escala de grises.
- El codificador y el descodificador JPEG se implementan en live functions.
- La función del codificador acepta como entrada la matriz de píxeles de la imagen en escala de grises con tipo `uint8` y el valor de la calidad de compresión (debe ser un entero entre 1 y 100).
- La codificación consiste en:
 - Dividir la imagen en bloques de 8×8 píxeles y, para cada bloque:
 - Aplicar la transformada discreta del coseno.
 - Cuantificar con la matriz proporcionada de acuerdo con el parámetro de calidad.
 - Convertir el bloque en un vector de enteros de tipo `int16`.
- La salida del codificador consta de los siguientes elementos:
 - Vector de enteros de tipo `int16` en Matlab.
 - Resolución de la imagen como un vector de dos componentes.
 - Matriz de cuantificación utilizada.



Se recomienda implementar cada funcionalidad en el codificador y en el descodificador y probarla antes de pasar a la siguiente. El ejemplo para la imagen `tiny.png` puede ser de utilidad.

Funciones proporcionadas

Para la realización de la práctica se proporcionan las siguientes funciones:

i Estas funciones pueden modificarse y/o mejorarse. Se incluyen a modo de ayuda.

- `block2zigzag()` y `zigzag2block()`
 - A partir de una matriz 8×8 , `block2zigzag()` realiza el recorrido en zig-zag tal y como se indica en el estándar JPEG y devuelve el vector resultante. La función `zigzag2block()` realiza la operación recíproca.
- `computeDCTMatrix()`
 - Devuelve la matriz DCT tipo-II para bloques de 8×8 píxeles.



- `jpegLumaQuantizationMatrix()`
 - Devuelve la matriz de cuantificación para la luminancia.
- `resizeImageTo8()`
 - Utiliza la función `imresize()` de Matlab para garantizar que las dimensiones de la imagen son múltiplos enteros de 8. La imagen que devuelve esta función **deberá tomarse como la imagen de referencia para las métricas de calidad (SSIM y MSE). Esta operación debe realizarse fuera de la función que implementa el codificador.**
- `scaleQuantizationMatrix()`
 - Escala la matriz de cuantificación definida para calidad = 50 de acuerdo con el parámetro de calidad proporcionado. Si calidad = 100, entonces la matriz de cuantificación resultante contiene únicamente 1 (no hay cuantificación), lo que es útil para labores de depuración.

Funciones a implementar y *live script* principal

Se deberá implementar:

- Un único *live script*, cuyo prototipo se incluye en el fichero `TI_Practica_JPEG.mlx` y que contendrá el código, los resultados y **sus comentarios**.
- Las live functions `jpegEncoder()` y `jpegDecoder()` que implementan la codificación y decodificación JPEG, cuyos prototipos se incluyen con los ficheros de la práctica.

El *live script* principal evalúa la SSIM y el MSE del códec implementado para los valores del parámetro de calidad de compresión 10:10:100 y los comparará con los resultados producidos por la implementación de Matlab.

La evaluación se realizará con una imagen disponible en internet a seleccionar por parte del estudiante.

Se indican a continuación las tareas a realizar por el *live script*:

- Carga de la imagen original.
- Ajuste de sus dimensiones para que sean múltiplos enteros de 8 usando `resizeImageTo8()`.
- Inicialización de los vectores del parámetro de calidad y de los que se usarán para guardar los resultados de las métricas.
- Bucle que itera por todos los valores del parámetro de calidad de compresión:
 - Codificación mediante `jpegEncoder()`.
 - Decodificación mediante `jpegDecoder()`.
 - Codificación y decodificación con las funciones de Matlab `imwrite()` e `imread()`.
 - Cálculo de los valores de las métricas SSIM y MSE para el códec implementado y para el de Matlab.
- Dibujo de la imagen original y la imagen comprimida con calidad 90.
- Dibujo de las gráficas de SSIM, MSE con respecto al parámetro de calidad de compresión.
 - Las gráficas **deben comentarse debajo de cada una de ellas**.

Entrega

Se debe entregar el live script principal junto con todas las funciones utilizadas, con independencia de que hayan sido proporcionadas o implementadas.

Valoración

Se valorarán los siguientes aspectos:

- Corrección de los resultados obtenidos y de la implementación realizada.
- **Calidad del código** (nombrado de variables, estructuración, etc.) y de la documentación del código. En general, el código debe ser fácil de leer y entender.
- **Concisión** y **calidad** de los comentarios de los resultados obtenidos. Ten en cuenta que se pueden realizar ejecuciones y mostrar imágenes de resultado, no sólo las gráficas pedidas, para reforzar los comentarios realizados.
- **En los comentarios trata de evitar opiniones y/o conjeturas.**
- **El objetivo de la práctica no es aproximarse a los resultados que se obtienen con la implementación de Matlab disponible en `imwrite()`.**

Rúbrica de evaluación



	0 puntos	0,5 puntos	1 punto
Preprocesado y postprocesado de la imagen (incluye conversión al tipo de dato adecuado)	Hay errores importantes, como por ejemplo usar las funciones <code>im2double()</code> e <code>im2uint8()</code> , o faltan pasos fundamentales.	Hay algún error de importancia menor.	Está correcto.
Aplicación de la DCT y la DCT inversa a bloques 8x8 mediante multiplicación de matrices.	No se aplica correctamente o hay errores graves.	Hay algún problema menor con la implementación de la DCT y/o su inversa o se realiza de forma ineficiente (e.g., calculando la matriz para cada bloque).	La implementación es correcta y eficiente.
Escalado de la matriz de cuantificación según el parámetro de calidad	No se realiza el escalado o contiene errores muy graves.	Hay algún error en la implementación del escalado.	La implementación del escalado es correcta.
Aplicación de la cuantificación y la cuantificación inversa a bloques 8x8 mediante multiplicación de matrices.	No se aplica correctamente o hay errores graves, como por ejemplo que el resultado de la cuantificación no sea entero.	Hay algún problema menor con la implementación de la cuantificación y/o su inversa o se realiza de forma ineficiente, como por ejemplo calculando la matriz para cada bloque.	La implementación es correcta y eficiente.
Conversión del resultado en un vector de enteros.	No se realiza la conversión o contiene errores muy graves.	Se convierte a un vector o a entero, pero no ambos, o se utiliza un tipo entero incorrecto.	La implementación es correcta.
Implementación del códec JPEG con las funciones de Matlab	No se ha realizado la implementación o contiene errores muy graves.	Hay algún error en la implementación.	La implementación es correcta.
Calidad del código	El código es muy difícil de entender, las variables no siguen ningún tipo de criterio de nombrado.	Hay algunos problemas de estructuración del código, como por ejemplo no incluir todas las operaciones del codificador y decodificador en las funciones pedidas.	El código es de calidad y se entiende fácilmente.
Resultados SSIM	Los resultados no son correctos.	Alguno de los resultados no es correcto.	Los resultados son correctos.
Resultados MMSE	Los resultados no son correctos.	Alguno de los resultados no es correcto.	Los resultados son correctos.
Comentarios de los resultados	No hay ningún tipo de comentario.	Faltan comentarios en alguno de los resultados o son erróneos.	Todos los resultados están comentados.

