

Universidad de Murcia

Curso 2020/21

Grado de Ingeniería Informática, 4º curso

Facultad de Informática

**Compresión Multimedia**

Juan José Morell Fernández

juanjose.morellf@gmail.com

24418612H

Índice

[1. Abstract 3](#_Toc60520117)

[2. Introducción 4](#_Toc60520118)

[3. Metodología 5](#_Toc60520119)

[3.1 Procedimiento de compresión y descompresión 5](#_Toc60520120)

[3.2 Imágenes y Factores de calidad seleccionados 8](#_Toc60520121)

[3.3 Manual de usuario 8](#_Toc60520122)

[4. Resultados Experimentales 9](#_Toc60520123)

[5. Conclusiones 10](#_Toc60520124)

[6. Bibliografía 11](#_Toc60520125)

# 1. Abstract

# 2. Introducción

Una imagen normalmente está representada por una matriz, donde cada uno de sus valores corresponde a cada pixel y simboliza la intensidad de estos. Si el valor es alto significa que ese pixel tiene un nivel superior de brillo, mientras que si es un valor pequeño el nivel de brillo también lo será. Asimismo, las imágenes con color presentan diferentes canales por cada uno de los componentes de color existente. Los componentes más típicos son el rojo, verde y azul; formando la representación mas conocida de las imágenes, RGB.



Figura 1 – Espacio de color RGB

Aunque esta representación de las imágenes es la más común, no es como se almacenan en un disco duro. Por ejemplo, almacenar una imagen con color de 48 megapíxeles conlleva guardar un total de 48 millones de valores por cada uno de los canales, sumando un total de 144 millones de valores. Esto dependiendo del método de almacenamiento puede llegar a ocupar un total de 144 MB. Pero en la realidad, una imagen de esas características llega a ocupar menos de 14 MB. ¿Cómo se consigue lograr esto?

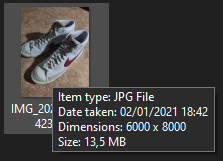


Figura 2 – Imagen de ejemplo, tomada con una cámara de 48 MP.

Justamente esta es la pregunta que hay que conseguir responder en este proyecto. Cuál es el proceso que lleva a una imagen, que supuestamente tendría que ocupar una gran cantidad de megabytes, pueda ocupar tan poco en disco.

La respuesta a esta pregunta es la compresión de imágenes, en este caso JPEG. Esta es una de las extensiones más conocidas mundialmente en las imágenes, y es un formato de compresión de imágenes.

# 3. Metodología

## 3.1 Procedimiento de compresión y descompresión

JPEG es un método de compresión con pérdida, lo que significa que la imagen final que se almacena en nuestro ordenador no tiene toda la información original. Cuanta más información se descarte en el proceso de compresión peor calidad tendrá la imagen comprimida. Este trato entre la calidad y la ocupación de la imagen en disco será uno de los puntos importantes dentro de este proyecto, y será comentado junto a las imágenes que se mostraran en el apartado 3.2.

En general, JPEG consigue una buena relación tamaño/calidad, ya que los datos que va a despreciar van a ser aquellos que no son realmente visibles por la vista humana.

El primer paso de la compresión es la conversión del espacio de color. Para no almacenar la imagen con un tamaño de 114 MB, se cambia el espacio de color RGB, a uno llamado YCbCr. Este se caracteriza por tener un canal que representa la intensidad de luz, y otros dos representan los colores, el azul y el rojo. Esto permite la separación entre la luminancia (Y) y la crominancia (Cb, Cr).

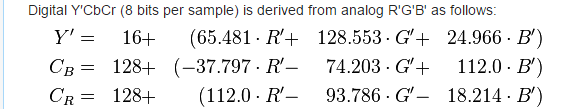


Figura 3 – Transformación de espacio RGB a YCbCr.

La transformación del espacio RGB a YCbCr se trata de una transformación lineal, que puede ser expresada mediante una multiplicación de matrices.

Las imágenes actuales son mucho más sensibles a los cambios de brillo que al de colores, por lo que se reducen los canales de color. Gracias al espacio YCbCr, se puede hacer esto sin problemas mediante el proceso de *downsample*, o también conocida como *chroma subsampling*.

Además de este cambio, otro factor importante a la hora de reducir el tamaño de las imágenes son las características de la vista humana. La idea reside en que los objetos o detalles pequeños que están representado en la imagen son más propensos a pasar desapercibidos que objetos grandes. Este efecto se denomina *sensibilidad de contraste dependiente de la frecuencia.*

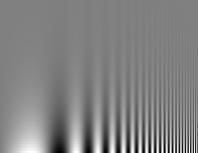


Figura 4 – Efecto de la frecuencia espacial

Un buen ejemplo de este fenómeno sería una imagen con líneas negras donde cada vez se van juntando más hasta que parece que no hay distancia, de izquierda a derecha. Mientras que el contraste va incrementando desde abajo hacia arriba de la imagen. Con esto se puede apreciar una curva en donde se desvanecen las líneas, y se debe a que nuestro ojo es más sensible a las variaciones donde el brillo sea mayor.

Se puede apreciar en la parte superior de la figura 4 que las líneas verticales casi no son visibles. Esto se debe a que se trata de la parte que presenta menor contraste y mayor frecuencia. Estos fenómenos nos dan espacio para poder comprimir más, ya que como se puede observar no detectamos tanta información.

JPEG lleva a cabo esto mediante la división de la imagen en bloques de 8 x 8 y cuantizandolos posteriormente, de forma individual, en una representación del dominio de frecuencias. Cada uno de estos bloques es comparado con un conjunto de 64 patrones de frecuencias donde la frecuencia espacial aumenta desde la esquina superior izquierda hasta la esquina inferior derecha.

Este proceso descompone la imagen en componentes de frecuencia, convirtiendo cada bloque de 8x8, que representaba el nivel de brillo de la imagen, en otro bloque de 8 x 8 que representa la presencia de un componente de frecuencia en particular. Este método se conoce como *discrete cosine transform* o *DCT*.

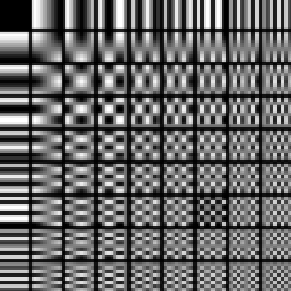


Figura 5 – Conjunto de 64 patrones de frecuencias DCT.

Mediante esta representación se pueden comprimir las frecuencias que son menos visibles al ojo humano, dividiendo la matriz del bloque 8 x 8 entre una matriz constante. Estas frecuencias que son menos visibles son divididas por constantes más grandes, al contrario que en las frecuencias más visibles que son divididas por constantes menores.

En nuestro caso, se utilizan las matrices de cuantización de JPEG por defecto, tanto para luminancia como para crominancia. También son conocidas por el nombre de ITU T.81.

La cuantización en este contexto se trata de redondear el resultado de la división de las matrices a los enteros más cercanos. Mediante el uso de números grandes el resultado de su división se acercará más a cero, lo que resulta en un ratio de compresión mayor, pero reduce también la calidad de la imagen.

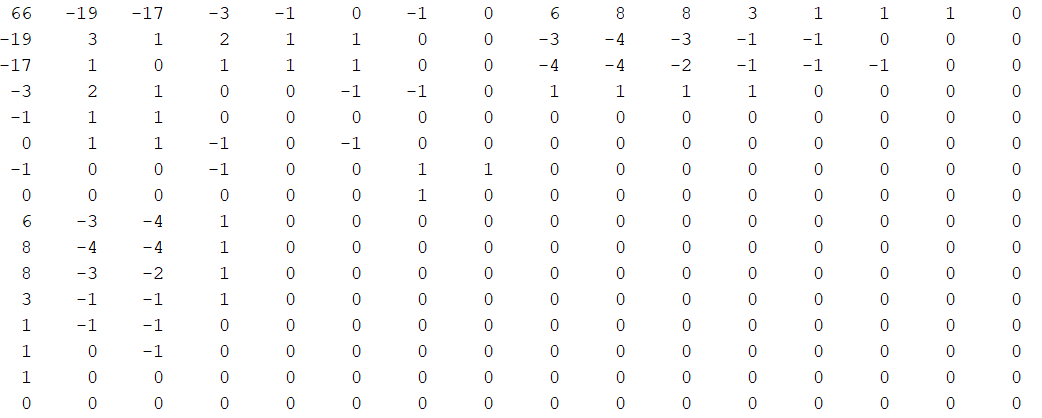


Figura 6 – Matriz tras la cuantización

Tras la cuantización, obtenemos unas matrices donde existen una gran cantidad de ceros en las frecuencias altas. Pero como se puede apreciar en la figura 6, los ceros están intercalados con diferentes números. Para ordenar el contenido de la matriz y poder almacenarlo de una forma mas eficiente se reordenan los coeficientes mediante un recorrido en zigzag desde la esquina superior izquierda a la esquina inferior derecha.

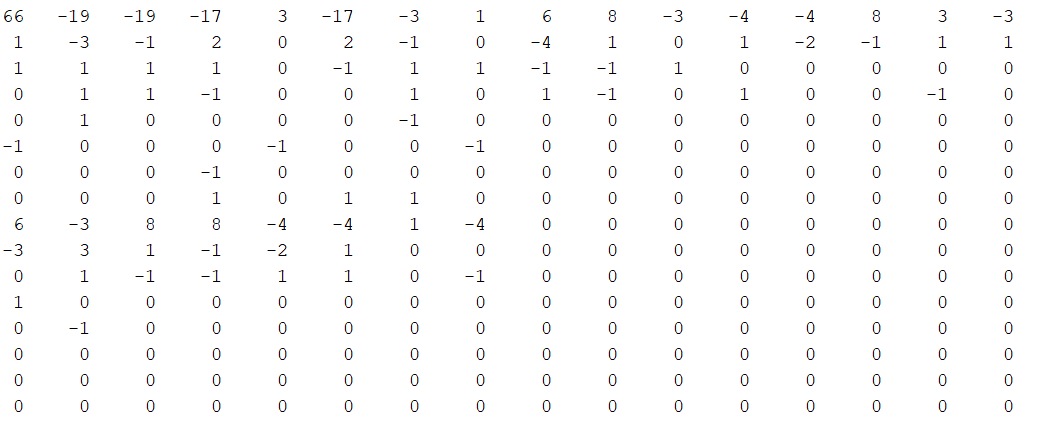


Figura 7 – Matriz tras recorrido en zigzag.

Una vez juntados los ceros, en vez de almacenar la matriz así se puede almacenar su valor junto al número de veces que aparece consecutivamente. Un ejemplo es que, si aparecen cinco ceros seguidos, en vez de almacenar los cinco, se almacena un 5 junto a un 0, representando que existen cinco ceros consecutivos. A esta técnica se le conoce como *Run-Length encoding*.

Otra forma de comprimir esto es teniendo en cuenta la frecuencia de aparición de cada elemento en la matriz, almacenando con más bits aquellos valores con más presencia y menor cantidad al resto. Esto consigue reducir el número de bits por símbolo. Esta técnica es conocida por *Entropy coding* o *Huffman coding*.

Ambas técnicas se recogen en el grupo de codificación sin perdidas, ya que no se pierde nada de la información original a la hora de comprimir. Se centran en almacenar los datos de una manera eficiente aprovechando la redundancia en los datos. Asimismo, estas técnicas también son utilizadas en otros ámbitos, como texto, audio o video.

[EXPLICAR ENTROPIA]

Todo este ejemplo se ha realizado mediante el programa creado tras el desarrollo de la práctica y una imagen en formato *bmp* de tamaño 10 x 10.

Por tanto, el proceso de compresión de una imagen toma el siguiente camino:

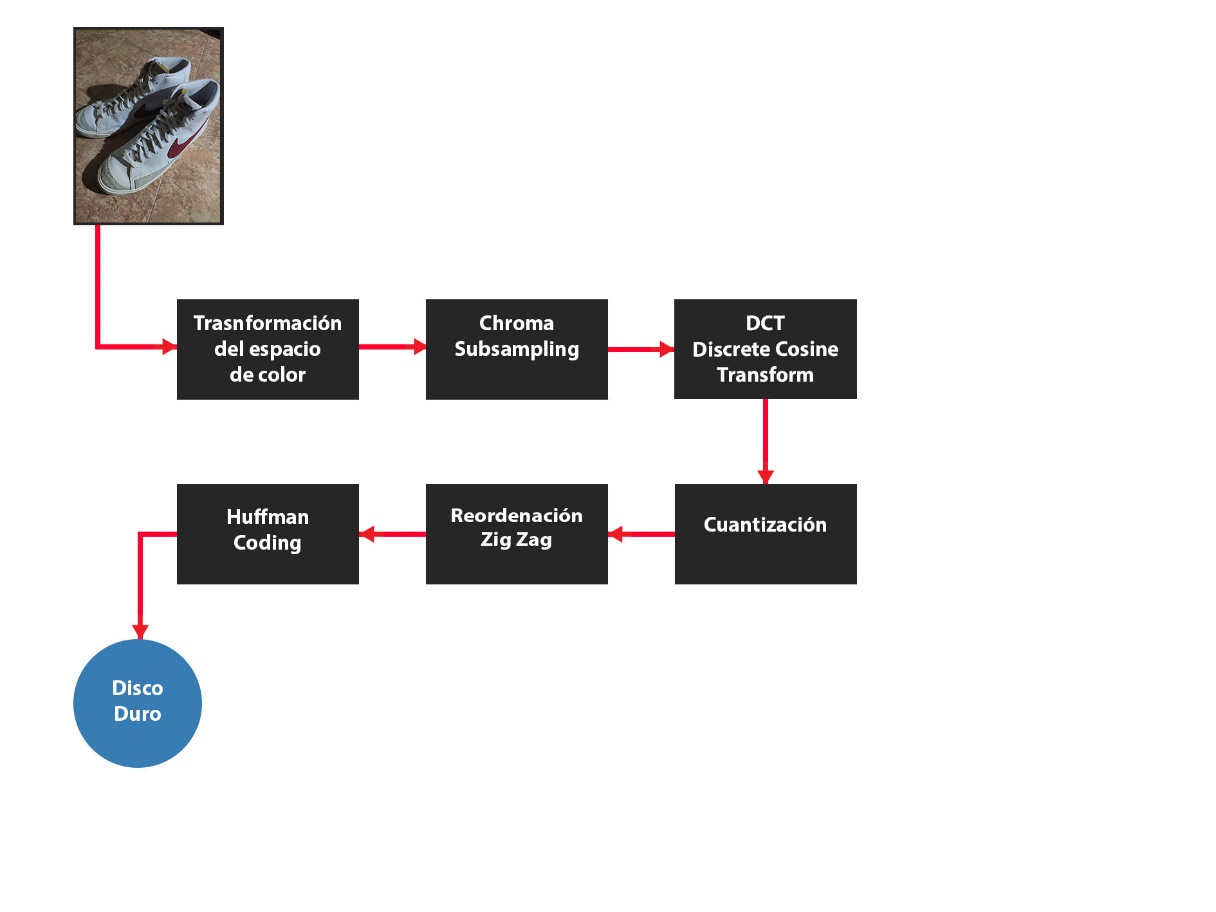


Figura – Esquema de compresión de una imagen

Para la decodificación de la misma imagen comprimida sería el proceso inverso. Debido a que parte de la información original se ha perdido tras los procesos de subsampling o cuantización, la imagen decodificada no va a ser idéntica a la original. Aun así, si la comprensión de la imagen se ha realizado con un correcto factor de calidad no se debería apreciar ninguna diferencia. Cuanto más alto sea dicho factor más notorios serán los efectos de la compresión.



Figura – Comparación entre diferentes compresiones

En esta figura no se podrá apreciar del todo bien el cambio entre imágenes, ya que dicha imagen esta también comprimida en formato jpeg. Si se desean ver las imágenes a su calidad total, se facilitarán las copias originales junto a esta documentación.

## 3.2 Imágenes y Factores de calidad seleccionados

Sadasds

## 3.3 Manual de usuario

dsadasd

# 4. Resultados Experimentales

# 5. Conclusiones

# 6. Bibliografía