

Compresión de imágenes mediante DCT Quantization

Martín Sturla, Darío Sneidermanis

Estudiantes Instituto Tecnológico de Buenos Aires (ITBA)

31 de Mayo de 2012

(Paper No Publicado)

Resumen—El siguiente paper busca analizar la compresión de imágenes utilizando *DCT Quantization* según la especificación JPEG original, comparando el nivel de compresión alcanzado y calidad perdida utilizando distintos niveles de cuantización.

Palabras clave—JPEG, compresión de imágenes, transformada discreta coseno, cuantización.

I. INTRODUCCIÓN

Históricamente la compresión de datos siempre ha sido de notable interés para la informática, ya sea para guardar algún archivo en algún dispositivo de capacidad limitada o para enviar un archivo por algún enlace utilizando la menor cantidad de banda ancha posible. Debido a estos requerimientos, surgen los primeros programas que haciendo uso de técnicas de compresión de datos, por ejemplo *Run Length Encoding* o codificación de *Huffman*, comprimen y descomprimen archivos. Algunos ejemplos incluyen *Winzip*, lanzado en 1991, o *Winrar*, lanzado en 1993. Sin embargo en el caso de las imágenes, debido a la naturaleza de sus datos, existen mejores alternativas para comprimirlas. Ya en el 1992, un comité conocido como *Joint Photographic Experts Group* comienza a publicar pautas para la compresión de imágenes, conocidas como la especificación *JPEG*.

La especificación original de JPEG, en la cual se centra este trabajo, indica que la compresión de las imágenes consiste en dos etapas fundamentales: una codificación con pérdida de información, es decir *lossy*, seguida de una compresión *lossless*, como por ejemplo las dos ya mencionadas. En esta primer etapa existen dos maneras de reducir la información: *chroma subsampling* y *DCT Quantization*. La primera, la cual no es analizada en este trabajo, consiste en aprovechar que el ojo humano tiene una menor agudeza para diferenciar colores que brillantez o *luma*, por lo cual se reduce la resolución de la croma. La segunda consiste en hacer uso de que el ojo humano es capaz de diferenciar cambios de luma en zonas grandes, pero no tan tenazmente en zonas pequeñas. Debido a esto, se eliminan estas componentes de frecuencia espaciales altas en zonas pequeñas

(casi imperceptibles para el ojo) y se suavizan los cambios de luma (también se efectúa un proceso similar con la croma). Debido a esta eliminación de información, existe mayor redundancia de los datos, por lo que un llamado posterior a un algoritmo de compresión será más efectivo.

El objetivo primordial del trabajo es analizar la técnica de *DCT Quantization* para distintas imágenes y matrices de cuantización, analizando en cada caso la tasa de compresión, el error introducido a la imagen recuperada y su distorsión gráfica. La sección 2 explica en profundidad la primer etapa de la compresión. La sección 3 trata sobre la segunda etapa de compresión. La sección 4 contiene los resultados.

II. ETAPA DE COMPRESIÓN CON PÉRDIDA DE INFORMACIÓN

A. Transformación del espacio de colores

La primer etapa de compresión se centra en el manejo de luma y croma, por lo cual es necesario transformar los datos de las imágenes, que normalmente contienen la cantidad de rojo, verde y azul en cada pixel (*RGB*). Estos datos son transformados a luma en un canal, Y' , y crominancia en dos: C_B y C_R . Estos últimos representan la proporción de verde con respecto al azul y rojo, respectivamente. Este cambio es también útil dado que la luma se concentra en sólo un canal (en *RGB* está disperso entre los tres). Dado que la luma es el componente más importante para el ojo humano, se puede hacer una mayor compresión de datos.

Las componentes $Y' C_B C_R$ son una combinación lineal de las componentes *RGB*. Las siguientes ecuaciones muestran más en profundidad la naturaleza de la conversión.

$$\begin{pmatrix} Y' \\ C_B \\ C_R \end{pmatrix} = \begin{pmatrix} +0,2990 & +0,5870 & +0,1140 \\ -0,1687 & -0,3313 & +0,5000 \\ +0,5000 & -0,4187 & -0,0813 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix}$$

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} +1,00000 & +0,00000 & +1,40200 \\ +1,00000 & -0,34414 & -0,71414 \\ +1,00000 & +1,77200 & +0,00000 \end{pmatrix} \begin{pmatrix} Y' \\ C_B - 128 \\ C_R - 128 \end{pmatrix}$$

31 de Mayo, 2012.

El apéndice A demuestra que las transformaciones son inversas. ¡TODO!!

B. Separación en bloques

Luego de la transformación a los canales $Y' C_B C_R$, la imagen se debe separar en bloques, que representan las "pequeñas zonas".^{en} las cuales se desea eliminar las frecuencias altas de luma y croma. El tamaño de los bloques depende de la tasa de submuestreo de croma usada, y varía entre cuadrados de 8 píxeles y 16 píxeles. Dado que no se utilizó submuestreo, se utilizaron bloques de 8.

En el caso de que las dimensiones de la imagen no sean múltiplos del tamaño del bloque, se procede a extenderla hasta que los bloques quepan perfectamente. Los nuevos píxeles son llenados utilizando información de sus vecinos más inmediatos. Otras alternativas analizadas incluyen llenar los canales de los nuevos píxeles con 0, sin embargo se observaron mayores distorsiones en los bordes.

C. Transformación del Coseno Discreta

El siguiente paso consiste en transformar los valores de los canales en cada bloque a sus frecuencias espaciales en dos dimensiones, utilizando la transformación del coseno discreta. Esta última consiste en una transformada de Fourier discreta utilizando únicamente cosenos. Para una cierta matriz cuadrada, la transformada está dada por la ecuación:

$$G_{u,v} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \frac{\alpha(u)\alpha(v)}{N} g_{x,y} \cos\left(\frac{\pi}{8}\left(x+\frac{1}{2}\right)u\right) \cos\left(\frac{\pi}{8}\left(y+\frac{1}{2}\right)v\right) \quad (1)$$

Donde $g_{x,y}$ representa el elemento de la fila x y la columna y del bloque de entrada, y $G_{u,v}$ el elemento de la fila u y columna v de la matriz de frecuencias de salida.

Análogamente la antitransformada está dada por:

$$g_{x,y} = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \frac{\alpha(u)\alpha(v)}{N} G_{u,v} \cos\left(\frac{\pi}{8}\left(x+\frac{1}{2}\right)u\right) \cos\left(\frac{\pi}{8}\left(y+\frac{1}{2}\right)v\right) \quad (2)$$

En ambas ecuaciones:

$$\alpha(k) = \begin{cases} \sqrt{1} & \text{si } k = 0 \\ \sqrt{2} & \text{en otro caso} \end{cases} \quad (3)$$

Nótese que los coeficientes de las frecuencias más bajas se encuentran en los valores bajos de x e y , es decir en la zona superior izquierda de la matriz de frecuencias. Similarmente los coeficientes de las frecuencias más altas se encuentran en la zona inferior derecha. Esta transformación concentra la señal de entrada en las frecuencias espaciales más bajas, es decir en la zona superior izquierda. En particular, comúnmente el mayor valor en módulo es $G_{0,0}$. Este valor representa la frecuencia con valor 0, y por lo tanto es un promedio de los 64 valores de entrada. A este valor se lo llama *DCcoefficient* dado que en circuitos el

promedio de una señal (es decir el valor asociado a la frecuencia cero) es la corriente efectiva directa (*DC*). Al resto de los valores se los llama *AC* debido a que representan la componente alterna de la corriente.

También es importante destacar que los valores de los canales son llevados a un rango alrededor del 0 antes de ser transformados, en el intervalo $[-127, 128]$. Esto se logra fácilmente restando 127 a todos los valores de entrada.

D. Cuantización

El paso final de la primer etapa de compresión, y el mayor responsable de la pérdida de información, es la cuantización de la matriz de coeficientes de frecuencias. Este paso consiste en dividir cada elemento de dicha matriz por un elemento de una matriz de cuantización, y redondear el valor obtenido. Las matrices de cuantización sugeridas por la especificación JPEG son las siguientes:

$$Q_y = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

$$Q_c = \begin{pmatrix} 17 & 18 & 24 & 47 & 47 & 99 & 99 & 99 \\ 18 & 21 & 26 & 26 & 66 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{pmatrix}$$

La fórmula para obtener los valores es la siguiente:

$$B(k,l) = \text{round} \left(\frac{A(k,l)}{Q_Y(k,l)} \right) \quad (4)$$

En otras palabras, se divide cada elemento de la matriz de coeficientes de frecuencias por su elemento correspondiente de la matriz de cuantización y se redondea. Para la luma se utiliza $Q_Y(k,l)$ y para croma $Q_c(k,l)$.

Nótese que existen varios valores de entrada que son transformados a la misma salida, es decir la relación no es inyectiva. Debido a esto, se pierde información. En particular, dado un coeficiente k de la matriz de cuantización, existen unos k valores que producirán la misma salida. Ergo, valores más grandes de la matriz de cuantización implican una mayor pérdida de información. Si se observan ambas matrices con detenimiento, se puede apreciar que los valores son mayores en la zona inferior derecha. Esto implica que la mayor pérdida de información ocurre en las frecuencias altas de los canales, que es lo que se buscaba en un principio. Asimismo, debido a que la transformada coseno discreta concentra la señal en la zona superior izquierda, y los coeficientes de cuantización son mayores en la zona inferior derecha, el resultado de la cuantización tendrá una densidad de ceros considerable en esta última zona.

III. CODIFICACIÓN ENTRÓPICA

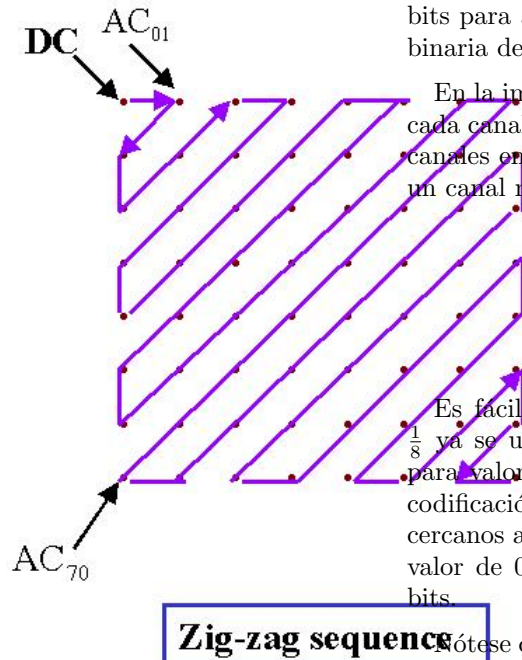
A. Reordenamiento y codificación

La segunda etapa de la compresión consiste en aplicar una codificación entrópica a los datos cuantizados. Debido a la gran densidad de ceros producida por la transformación y cuantización, este tipo de algoritmos logra una compresión más efectiva. La especificación JPEG admite dos algoritmos: la codificación de *Huffman* y codificación aritmética. Ambos consisten en sustituir instancias de patrones frecuentes por cadenas de bits más cortas, con la diferencia que *Huffman* reemplaza cada patrón de entrada por su código respectivo, mientras que la aritmética lo transforma en un número real. La codificación aritmética por lo general obtiene resultados entre 5 y 7% más compactos, pero por lo general se utiliza *Huffman* debido a problema de patentes y su mayor velocidad.

Sin importar qué algoritmo se utilice, la gran densidad de ceros se traduce a que existe un patrón extremadamente frecuente, el cual puede ser reemplazado por una cadena de bits particularmente corta para reducir el tamaño del archivo. Sin embargo, se pueden modificar algunos datos para reducir aún más la tasa de compresión.

A.1 Reordenamiento

Antes de ejecutar el algoritmo de codificación, se pueden reordenar los valores de la matriz cuantificada, ordenando los valores según qué frecuencia representan.



Zig-zag sequence

Figura 1: Reordenamiento de los valores según su frecuencia

Debido a que los valores asociados a mayores frecuencias eran mayormente cero, este reordenamiento casuará cadenas largas de ceros en los datos antes de codificar. Esto resulta en una codificación más compacta.

A.2 Resta del coeficiente DC

El valor *DC* suele ser el mayor valor de cada bloque. Si se pueden acotar los valores de los bloques a un intervalo pequeño en un entorno del 0, la codificación sería más eficaz. Para eliminar los valores altos de *DC*, se puede hacer uso de que el *DC* representa el promedio de los canales de un bloque. Asumiendo una imagen normal, es muy posible que el promedio de los canales en pixeles adyacentes sea similar. Debido a esto, en vez de guardar el *DC* se puede guardar la diferencia entre el *DC* actual y el del pixel anterior. Esto reduciría la cantidad de valores posibles (pudiendo incrementar incluso la frecuencia de ceros marginalmente).

B. Cálculo de cota superior razonable

Para hallar una cota superior razonable y simple al tamaño del archivo resultante luego de la compresión, se puede utilizar la frecuencia relativa de valores de canales de pixeles con valor en 0. Utilizando una codificación de *Huffman*, se le podría asignar a estos pixeles un único bit, en 0. En el peor de los casos, el resto de los valores varía uniformemente entre -127 y 128 , por lo cual se necesitan 9 bits para almacenarlos (un 1 seguido de la representación binaria del número).

En la imagen original, se utilizan 24 bits por pixel, 8 por cada canal. Asumiendo que p es la frecuencia de valores de canales en 0, es fácil verificar que en promedio el valor de un canal necesitará:

$$b_{pixel} = 3(p + 9(1 - p)) \quad (5)$$

Es fácil ver también que para valores de p mayores a $\frac{1}{8}$ ya se usan menos de 24 bits (nótese sin embargo que para valores tan bajos de p seguramente haya una mejor codificación que la asumida; esta última asume valores más cercanos a 1 que 0, como se observa en la práctica). Con un valor de 0,9, por ejemplo, esta cantidad se reduce a 5,4 bits.

Nótese que esta es simplemente una cota superior razonable. Utilizando restas de coeficientes *DC* y reordenamientos la cantidad de bits seguramente será menor. La estimación no contempla el espacio requerido por la tabla de codificaciones que se guarda en el encabezado del archivo. Este se puede despreciar asumiendo que la imagen posee dimensiones considerables.

IV. PROCEDIMIENTO

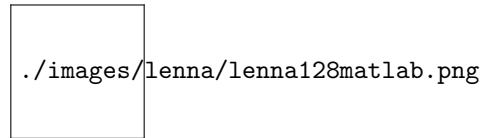


Figura 2: Imagen de Lenna de 128x128 pixeles
figure

Imagen	Tamaño en pixeles	Densidad de 0
Lenna512	512x512	0.01%
Nukem	610x685	0.01%
Eye	389x353	0.01%
Landscape	1280x1024	0.07%

Tabla 1: Tamaño de las imágenes de muestra

REFERENCIAS

- [1] Fierens, P. (2011), *Análisis Armónico: Guía 02*, Buenos Aires: Instituto Tecnológico de Buenos Aires.
- [2] Khayam, S. (2003), *The Discrete Cosine Transform: Theory and Application*, Michigan State University (MI).
- [3] Hsu, H. (1987), *Análisis de Fourier*, Addison - Wesley Iberoamericana (DE).
- [4] *JPG* - <http://en.wikipedia.org/wiki/Jpg>, Wikipedia.
- [5] <http://www.cmlab.csie.ntu.edu.tw/cml/dsp/training/coding/jpeg/jpeg/encoder.htm>, Wikipedia.