

Codificador JPG

Fabio Andres Ortiz Gasca
20171020051
andresrenzas@gmail.com

Resumen—El formato JPG para la comprensión de imagen es de los más comunes en la web y con los que más están familiarizados las personas sin información técnica acerca del procesamiento de imagen. Aún así, pese a ser ampliamente utilizado, no es de conocimiento común cuáles son los métodos detrás de su funcionamiento o aplicación. En vistas de realizar una correcta explicación del entendimiento, se procede a la construcción de un codificador (y por ende decodificador) sin utilizar ninguna librería que no fuera explícitamente necesaria (el proyecto está realizado en Python). Evidentemente basada en la idea de que para entenderlo, se necesita construir desde la base. En el paso a paso se explicará la función del mismo, y como fue abordado lógicamente.

I. INTRODUCCIÓN

JPG es junto a PNG, el formato de imagen más abundante en la Web, las ventajas inherentes a su uso son la razón tras la basta presencia a través de toda la red. Factible en cualquier tipo de navegador, afinado desde el planteamiento sobre la búsqueda de una mejora en la optimización de datos, en la actualidad se cuentan con varios cambios en el protocolo, optimizados para la conservación de datos o el ocupar menos cantidad de espacio. Para comprender que implica este éxito se necesita entender cuáles son los procedimientos, metodologías y formas que construyen paso a paso al JPG, cómo funciona a sus niveles más básicos y que auguraría para futuros protocolos de codificación.

II. OBJETIVOS

- Comprensión éxito a nivel histórico del formato JPG.
- Entendimiento metodología sobre la construcción del JPG.
- Identificar cuáles son las fortalezas y debilidades del formato JPG.

III. HISTORIA

La primera mención a JPEG (acrónimo de Joint Photographic Experts Group) se remonta a 1992, esta especificación incluía numerosos procesos de papers de investigación y patentes citados por el CCIT (ITU TeleCommunication Standardization Sector). Entre estos se incluye el uso de la Transformada Discreta de Coseno, sustentada por diversos Papers como técnica para la comprensión de imagen, también se citan papers que justifican el uso de la matriz de cuantificación, y en últimas, una aclaración acerca del algoritmo de Huffman más utilizado. También cuenta con mejoras sugeridas referentes a la codificación, como una variación guiada a la aritmética.

Joint Photographic Experts Group se funda en 1986, durante el final de la década de los 80 formalizan el protocolo JPG. Después de examinar varios métodos de comprensión de la

información, escogieron la Transformada Discreta de Coseno como el método que mejor se desempeñaba al conservar la información. Aprobado en Septiembre de 1992 bajo el nombre de ITU-T Recommendation T.81 y en 1994 bajo la denominación ISO/IEC 10918-1.



Figura 1. JPG Logo

La especificación incluye al CODEC, que define cómo se comprime la imagen en bytes, así mismo, la descompresión del mismo. No obstante, no particulariza cuál es el formato que debe contener esa información codificada, durante la presente construcción a favor de evitar problemáticas se opta por utilizar un .txt. No obstante, en las aplicaciones online usan formatos Exif y JFIF. Ambos están especializados en almacenar óptimamente la información, a tal punto que desde 2009, la renovación sobre el funcionamiento del JPEG clasifica a JFIF como formato de intercambio.

A lo largo de los años consecutivos han habido varias demandas por el aparente conflicto de patentes. No obstante, las justificaciones presentes en el informe sobre el JPEG le asignaban suficiente validez para defender la validez del mismo. Dado que se cumplieron 20 años desde presentado el informe, ya está en dominio público así que estas dejaron de ocurrir con la misma frecuencia. Eso sí, algunas alegaciones resultaron exitosas más no en lo referente a la patente del mismo, sino sobre las patentes en que se basaron y lo que implica sobre ciertas aplicaciones.

Desde su instauración, el JPG ha logrado hacerse con una importante baza de popularidad y uso, gracias al éxito detrás de su conceptualización, valores de conservación de imagen suficientemente mejores respecto a los demás protocolos, lo

que resultaba óptimo para los primeros intercambios globales en Internet, a finales de la década de los 90. Las continuas mejoras respecto a la conservación de los datos, sin que fuera explícitamente necesario el requerimiento de más recursos, lo ha mantenido vigente durante el paso de los años. Y seguramente este esfuerzo sobre la mejora continua le asegurará todavía mayor tiempo como uno de los formatos más utilizados de la red.

IV. CODEC

La codificación conformada a través de pasos, donde cada cuál profundiza o se ve alimentado a través de diversos procesos alternos, con seguridad se puede afirmar que se puede ir tanto como el conocedor requiera. Tal como se menciona previamente, el objetivo es dar tratado superficial acerca del mismo, donde la comprensión mejore más no requiera entender completamente cada uno de los elementos convergentes, en orden de lo estipulado, se dividirá cada procedimiento, junto al enfoque lógico adaptado (más no se adjuntará el código, este se encuentra anexo en un repositorio de Github).

- Procesado de la matriz.
- RGB a YCbCr.
- Aplicación Transformada Discreta de Coseno.
- Aplicación matriz de cuantificación.
- Codificación de Huffman.
- Descompresión

V. PROCESADO DE LA MATRIZ

Para la ejemplificación a lo largo de aplicación de codificación JPG, utilizaremos a la **figura 2** que llamaremos "Prueba", es de formato .tiff. Mide 24x24, siendo un total de 576 píxeles, cada uno de estos se compone de 3 valores correspondientes a las gamas del RGB, ósea, la imagen tiene 1728 datos. El tamaño es de 2066 Bytes.



Figura 2. Prueba.tiff

El codificador sólo contempla imágenes cuya altura y anchura sean múltiplos de 8, debido a que el primer paso de la preparación es dividir la imagen en bloques 8x8. Este codificador no permite entradas cuya altura o anchura difieran de esta condición, sin embargo, cabe aclarar, que obviamente tales métodos existen y cuentan con su propia serie de especificaciones. Referente a prueba, la división daría como resultado 9 matrices, debido a que se organiza a través de un 3x3. Tal como se observa en **figura 3**.

Se necesita brindar algún tipo de recorrido sobre la matriz para iterar sobre cada dato, para facilitar esta navegación se procede a dividir en 8 filas cada matriz con sus respectivos 8 datos. De esta manera cada una se compone de la misma cantidad de datos, permitiendo métodos de recorridos utilizables para cada una. La **figura 4** representa la división en filas

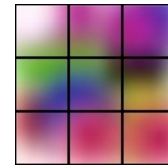


Figura 3. Prueba Dividida

y celdas, a través de una tabla similar. Hay que considerar que cada dato está dividido en otros 3 datos, ósea que nuestra matriz podría considerarse que cuenta con una profundidad de 3.

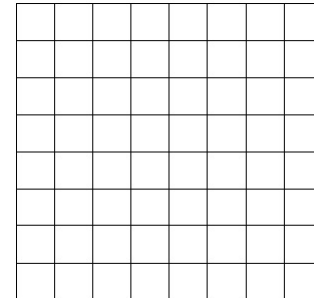


Figura 4. Prueba Dividida en Filas

VI. RGB A YCbCr

La transformación de valores RGB a escala YCbCr está guiada a la idea de tener menor cantidad de información a codificar. Debido a que el codificador construido sólo se prueba a través de imágenes bastante pequeñas (menor a los 100p). Esta reescala no ejercería una codificación lo suficientemente considerable como para elaborar su aplicación. Igual necesita ser abordado.

El YCbCr no es un espacio de color absoluto, ósea una escala independiente que tiene sus propias formas de clasificar el espectro, sino que es una forma de codificación de RGB. La Y es representación del componente Luma, mientras que la Cb y Cr son los componentes de crominancia diferencia de azul y diferencia de rojo.

Las señales de los colores básicos de la luz (rojo, verde y azul) no son óptimos para el almacenamiento y transmisión directa, debido a que cuentan con una gran cantidad de redundancia. La codificación YCbCr transforma los colores primarios en información lo suficientemente significativa. Lo cuál puede inducir a una gran cantidad de errores durante el procesado, comprimido o tratado, debido a que los errores serían lo "suficientemente significativos".

En la transformación digital se utilizan las siguientes fórmulas:

$$Y = 0 + (0.299 * R) + (0.587 * G) + (0.114 * B)$$

$$Cb = 128 - (0.168736 * R) - (0.331264 * G) + (0.5 * B)$$

$$Cr = 128 + (0.5 * R) - (0.418688 * G) - (0.081312 * B)$$

Estos datos van acorde a la especificación del formato. De las cuáles son inferibles las fórmulas inversas:

$$\begin{aligned}
R &= Y + (1.402 * (Cr - 128)) \\
G &= Y - (0.34414 * (Cb - 128)) - (0.71414 * (Cr - 128)) \\
B &= Y + (1.772 * (Cb - 128))
\end{aligned}$$

Para finalizar la conversión en datos más simples para ser codificados, se suma 127 a cada dato (número particularizado en la aplicación).

VII. TRANSFORMADA DISCRETA DE COSENO

Transformada basada en la Transformada de Fourier discreta, utilizando únicamente números reales. Cuenta con una sobresaliente capacidad de compactación de la energía al dominio transformado, lo que infiere en que la transformada de coseno discreta concentra mayor parte de la información en pocos coeficientes transformados.

La transformación es independiente de los datos. Y no varía según los datos que recibe, al contrario de otros métodos utilizados en demás algoritmos de comprensión. Produce pocos errores en los límites de los bloques imagen. Con una minimización o litigación de errores se permite reducir estas redundancias, logrando amortiguar el efecto de bloque en las imágenes reconstruidas.

Cuenta con una interpretación frecuencial sobre los componentes transformados. Con el enfoque apropiado acerca de la interpretación se puede aprovechar aún más notoriamente sus capacidades.

$$C = \frac{1}{2} \begin{pmatrix} \cos(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) \\ \cos(\frac{\pi}{16}) & \cos(\frac{3\pi}{16}) & \cos(\frac{5\pi}{16}) & \cos(\frac{7\pi}{16}) & \cos(\frac{9\pi}{16}) & \cos(\frac{11\pi}{16}) & \cos(\frac{13\pi}{16}) & \cos(\frac{15\pi}{16}) \\ \cos(\frac{2\pi}{16}) & \cos(\frac{6\pi}{16}) & \cos(\frac{10\pi}{16}) & \cos(\frac{14\pi}{16}) & \cos(\frac{18\pi}{16}) & \cos(\frac{22\pi}{16}) & \cos(\frac{26\pi}{16}) & \cos(\frac{30\pi}{16}) \\ \cos(\frac{3\pi}{16}) & \cos(\frac{9\pi}{16}) & \cos(\frac{15\pi}{16}) & \cos(\frac{21\pi}{16}) & \cos(\frac{27\pi}{16}) & \cos(\frac{33\pi}{16}) & \cos(\frac{39\pi}{16}) & \cos(\frac{45\pi}{16}) \\ \cos(\frac{4\pi}{16}) & \cos(\frac{12\pi}{16}) & \cos(\frac{20\pi}{16}) & \cos(\frac{28\pi}{16}) & \cos(\frac{36\pi}{16}) & \cos(\frac{44\pi}{16}) & \cos(\frac{52\pi}{16}) & \cos(\frac{60\pi}{16}) \\ \cos(\frac{5\pi}{16}) & \cos(\frac{13\pi}{16}) & \cos(\frac{21\pi}{16}) & \cos(\frac{29\pi}{16}) & \cos(\frac{37\pi}{16}) & \cos(\frac{45\pi}{16}) & \cos(\frac{53\pi}{16}) & \cos(\frac{61\pi}{16}) \\ \cos(\frac{6\pi}{16}) & \cos(\frac{14\pi}{16}) & \cos(\frac{22\pi}{16}) & \cos(\frac{30\pi}{16}) & \cos(\frac{38\pi}{16}) & \cos(\frac{46\pi}{16}) & \cos(\frac{54\pi}{16}) & \cos(\frac{62\pi}{16}) \\ \cos(\frac{7\pi}{16}) & \cos(\frac{15\pi}{16}) & \cos(\frac{23\pi}{16}) & \cos(\frac{31\pi}{16}) & \cos(\frac{39\pi}{16}) & \cos(\frac{47\pi}{16}) & \cos(\frac{55\pi}{16}) & \cos(\frac{63\pi}{16}) \end{pmatrix}$$

Figura 5. Transformada Discreta de Coseno

En el proceso de transformación se realizará una multiplicación de matrices, por izquierda se multiplicará con la DCT, el resultado se multiplicará con la inversa de DCT. Este proceso es irreversible, ósea, cuando se realice al contrario, no regresará los valores iniciales. Dado tanto por las particularidades intrínsecas de la transformada y porque la multiplicación de matrices no es conmutativa.

VIII. MATRIZ DE CUANTIFICACIÓN

Corresponde al alivio respecto a los cambios referente al brillo en áreas considerablemente grandes, más no si el brillo cambia repetidamente en espacios más significativamente pequeños, por lo tanto, se opta por eliminar las altas frecuencias, sin que se pierda significativamente la calidad visual.

Aplicado con una división de cada dato del dominio de frecuencia entre una constante asignada a ese componente y redondeando al entero más cercano. Se agudizan la pérdida (y calidad) de la información por el método de redondeo. En consecuencia a su aplicación, los componentes de las altas frecuencias tienden a a cero, mientras que los demás están entre números positivos y negativos pequeños.

En JPEG se utiliza como matriz de cuantificación a la matriz de Losheller.

$$C = \frac{1}{2} \begin{pmatrix} \cos(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) & \cos(\frac{\pi}{4}) \\ \cos(\frac{\pi}{16}) & \cos(\frac{3\pi}{16}) & \cos(\frac{5\pi}{16}) & \cos(\frac{7\pi}{16}) & \cos(\frac{9\pi}{16}) & \cos(\frac{11\pi}{16}) & \cos(\frac{13\pi}{16}) & \cos(\frac{15\pi}{16}) \\ \cos(\frac{2\pi}{16}) & \cos(\frac{6\pi}{16}) & \cos(\frac{10\pi}{16}) & \cos(\frac{14\pi}{16}) & \cos(\frac{18\pi}{16}) & \cos(\frac{22\pi}{16}) & \cos(\frac{26\pi}{16}) & \cos(\frac{30\pi}{16}) \\ \cos(\frac{3\pi}{16}) & \cos(\frac{9\pi}{16}) & \cos(\frac{15\pi}{16}) & \cos(\frac{21\pi}{16}) & \cos(\frac{27\pi}{16}) & \cos(\frac{33\pi}{16}) & \cos(\frac{39\pi}{16}) & \cos(\frac{45\pi}{16}) \\ \cos(\frac{4\pi}{16}) & \cos(\frac{12\pi}{16}) & \cos(\frac{20\pi}{16}) & \cos(\frac{28\pi}{16}) & \cos(\frac{36\pi}{16}) & \cos(\frac{44\pi}{16}) & \cos(\frac{52\pi}{16}) & \cos(\frac{60\pi}{16}) \\ \cos(\frac{5\pi}{16}) & \cos(\frac{13\pi}{16}) & \cos(\frac{21\pi}{16}) & \cos(\frac{29\pi}{16}) & \cos(\frac{37\pi}{16}) & \cos(\frac{45\pi}{16}) & \cos(\frac{53\pi}{16}) & \cos(\frac{61\pi}{16}) \\ \cos(\frac{6\pi}{16}) & \cos(\frac{14\pi}{16}) & \cos(\frac{22\pi}{16}) & \cos(\frac{30\pi}{16}) & \cos(\frac{38\pi}{16}) & \cos(\frac{46\pi}{16}) & \cos(\frac{54\pi}{16}) & \cos(\frac{62\pi}{16}) \\ \cos(\frac{7\pi}{16}) & \cos(\frac{15\pi}{16}) & \cos(\frac{23\pi}{16}) & \cos(\frac{31\pi}{16}) & \cos(\frac{39\pi}{16}) & \cos(\frac{47\pi}{16}) & \cos(\frac{55\pi}{16}) & \cos(\frac{63\pi}{16}) \end{pmatrix}$$

Figura 6. Matriz de Cuantificación (Losheller)

Notese como mientras los valores más están hacia la derecha y hacia abajo, cada vez van aumentando, correspondiendo a las frecuencias superiores, contrario a donde los brillos menos varían, en la esquina superior izquierda de la transformada.

IX. CODIFICACIÓN HUFFMAN

Aplicada a través de una variante denominada codificación entrópica, consistiendo en recorrer los datos en Zig Zag. Corresponde a la formación de la transformada donde los últimos grupos suelen ser el mismo dato repetido consecutivamente, al aplicar la codificación de Huffman se agrupan los últimos bajo un mismo grupo. También es factible utilizar la codificación aritmética, sin embargo, su uso necesita de una gran cantidad de patentes.

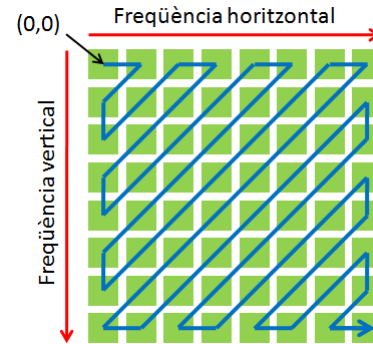


Figura 7. Recorrido en Zig Zag

La aplicación del método mejorado de Huffman no está en el codificador debido a que el pequeño tamaño de las imágenes dedicadas a comprobar no lo requerían siendo necesario. En cambio, se aborda en la siguiente forma: Cada matriz fue dividida en 3 listas, estas corresponden a las cantidades R, G y B. Dentro de estas se aplicó la aplicación estándar de Huffman.

Cada uno de los símbolos es clasificado acorde a la cantidad de veces que se repita dentro de la lista, posteriormente se van agrupando conforme a esa frecuencia de aparición. Se formalizan diccionarios insertados dentro de una enciclopedia sobre la información de codificación.

Primero se enlistan los valores conforme a cuál sea su mayor cantidad de apariciones, porque infiere cuál sería la probabilidad que tiene algún valor escogido al azar de resultar siendo ese. Se van agrupando acorde a la menor cantidad de

Original source			Source reduction							
Symbol	Probability	Code	1	2	3	4				
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

Figura 8. Ejemplo codificación de Huffman

apariciones, hasta que sólo quedan dos valores. De ahí se escoge intercalando entre 0 y 1 las posibilidades. En el último paso se seleccionan los dos factibles valores.

X. DECODIFICACIÓN

Los mismos pasos que construyen la codificación a la inversa, del archivo donde fue insertado la codificación se procede a decodificar. De la anterior operación se conserva la variable ".enciclopedia", con los diccionarios respectivos para la traducción de los datos binarios.

Una vez traducido, se debe volver a agrupar en tríadas (de la forma RGB), luego se reinsertan dentro de las matrices. Procede a multiplicarse con los valores particulares asignados a través de la matriz de cuantificación, posteriormente se aplica la inversa del DCT, lo que implica contar con la matriz, multiplicarla por izquierda con la inversa de la Transformada Discreta de Coseno y por la derecha con la original de la Transformada Discreta de Coseno.

Se suma 127 a cada dato, se realiza la conversión a RGB si los valores estaban en YCbCr. Finalmente se cuenta con la imagen original. En el correspondiente caso, respecto a Prueba.tiff, el resultado final es Prueba.jpg, cuyo peso es de .

Los valores de la primera fila de la primera matriz antes de comenzar eran de:

- (255, 244, 253), (255, 245, 253), (254, 243, 252), (252, 237, 249), (247, 221, 241), (238, 195, 227), (226, 155, 207), (212, 111, 186)

Y posteriormente, después de la codificación:

- (252, 244, 255), (255, 240, 251), (255, 238, 246), (255, 236, 249), (247, 224, 251), (232, 193, 238), (220, 151, 209), (214, 119, 184)

Habiendo una pérdida de:

- (3, 0, 2), (0, 5, 2), (1, 5, 8), (3, 1, 0), (0, 3, 10), (6, 2, 11), (6, 4, 2), (2, 8, 2)

Existió pero es pasablemente menor, satisfactorio sobre lo esperado. Cabe aclarar que hay varios valores que pueden contar con una pérdida mayores, alrededor de hasta 30. Sin embargo, considerando todo el desempeño general, puede considerarse como exitoso.

XI. MEJORAS Y CONCLUSIONES

Durante varios años se han propuesto consecutivas mejoras a la especificación, como son reforzar mediante la correlación entre los valores dentro de los bloques, entre los adyacentes y hasta lo referente a la relación entre canales.

Así mismo se han propuesto diversos usos con los coeficientes de la DCT, siendo la reescala a bajo nivel a través

de un factor homónimo. También se proponen otros métodos respecto a la codificación de Huffman, como son la ya mencionada codificación aritmética, o la codificación progresiva (cuyo bitrate es menor acorde a la codificación independiente de cada coeficiente). Hay mejoras como reordenar los coeficientes y bloques para predecir nuevos valores, incluso dividirlos entre pequeños modelos acorde a sus características y valores adyacentes.

Estas mejoras pueden conllevar hasta una optimización del 65 por cierto, basta con notar como este codificador ya se desempeña competentemente aún sin haber pasado los valores a otra escala ni aplicar la codificación entrópica.

Entre las variaciones más reconocidas del JPG se encuentran:

- Estereoscópico: Formato utilizado para efectos 3D desde imágenes 2D, consistiendo en la contraposición de dos imágenes, una dedicada al ojo izquierdo, la otra al derecho. No se necesita un visualizador particular, basta con el mismo del formato común.
- Formato Multi-Imagen: Almacena varias imágenes en un mismo archivo.
- XT: Soporta mayores bits de profundidad (hasta 16). mayor rango dinámico y codificación con punto flotante y hasta codificación del canal Alfa.
- XL: Alta resolución, con un máximo de 10 bits por componente, codificación RGB/YCbCr/ICtCp, imágenes animadas, codificación Alfa y Gamma, mejora en la codificación utilizando gradientes y hasta mapas de bits, incluyendo tratado sobre regiones de la misma.

REFERENCIAS

- b1 Lezama, J. (2017). COMPRENSIÓN DE IMÁGENES CODIFICACIÓN DE HUFFMAN. Revista De Educación Matemática, (32), 25-36.
- [0] [1] Lezama, J. (2017). COMPRENSIÓN DE IMÁGENES: FORMATO JPEG. Revista De Educación Matemática, (32), 23-34.
- [2] G. Sapiro, Digital image processing: p008 - Huffman coding. Duke university, 2013.
- [3] "How JPG Works", freeCodeCamp.org, 2016. [Online]. Available: <https://www.freecodecamp.org/news/how-jpg-works-a4dbd2316f35/>. [Visitado el 4 de agosto de 2020].
- [4] M. Khalid, Understanding and Decoding a JPEG Image using Python - Yasoob Khalid", Yasoob.me, 2020. [Online]. Available: <https://yasoob.me/posts/understanding-and-writing-jpeg-decoder-in-python/>. [Visitado el 4 de agosto de 2020].
- [5] "Working with Images in Python - GeeksforGeeks", GeeksforGeeks, 2017. [Online]. Available: <https://www.geeksforgeeks.org/working-images-python/>. [Visitado el 4 de agosto de 2020].
- [6] "Jpeg Compression - Huffman Coding", Robertstocker.co.uk, 2020. [Online]. Available: <http://www.robertstocker.co.uk/jpeg/jpeg-new-11.htm>. [Visitado el 4 de agosto de 2020].
- [7] R. Moilina and V. Gil-Costa, Conversión de RGB a YCbCr en System Generation y HLS. 2020.