

Comparación de algoritmos de compresión de imágenes

Arturo Moreno Báez, José Ismael De la Rosa, y Jesús Villa

Maestría en Ingeniería,

Cuerpo Académico de Procesamiento Digital de Señales,

Universidad Autónoma de Zacatecas.

TEL: +(492)9239407, ext. 1518

correo-e: morenob20@hotmail.com, ismaelrv@ieee.org

Resumen — En este trabajo se presenta la comparación de dos métodos ya utilizados para la codificación y compresión de imágenes fijas basados en técnicas de transformación. La aportación principal es la implementación en programación de alto nivel de una interface gráfica que contiene diversas rutinas para llevar a cabo la codificación de imágenes fijas utilizando las siguientes metodologías de transformación: 1) codificador basado en la transformada del coseno discreta o DCT, de sus siglas en inglés, que se implementa en la norma JPEG, 2) codificador basado en la transformada wavelet discreta (DWT, de sus siglas en inglés), en donde por el momento se utilizan bases de Haar. Los resultados son consistentes con los encontrados en la literatura existente y han sido validados con MATLAB[®].

Abstract— In this paper we present the comparison of two transform methods for codification and compression. The main contribution of this work, is in fact, the implementation in high level languages of a graphic interface which contains several algorithms for compression of still images according with the following two methodologies: 1) codification based on the discrete cosine transform (DCT), which is imple-

mented in the JPEG norm, and 2) codification based in the discrete wavelet transform (DWT), where we use for instance the Haar bases. The obtained results with the designed interface are congruent with those reported in literature and also were validated using MATLAB[®].

Descriptores — Codificación, compresión, métodos de transformación, wavelets.

I. INTRODUCCIÓN

LA norma más reciente establecida por la ISO/ITU-T en lo referente a la compresión de imágenes fijas es la JPEG 2000 (compresión con pérdidas y sin pérdidas), dicha norma supera por mucho en calidad a la norma estándar durante los 90's JPEG [7]. Otros estándares recientemente actualizados son JPEG-LS, MPEG-4 VTC (Visual Texture Coding) y MPEG-4/7/21. Estas últimas utilizan el núcleo utilizado por la norma JPEG (pero para el caso de la transmisión/recepción de video), es decir, la DCT. En el trabajo de D. Santa-Cruz [7] se introducen de manera un poco más detallada los esquemas presentados en las Figuras 1 y 2, en

donde además se lleva a cabo una comparación con respecto a otros esquemas importantes, sobre todo en el caso de compresión sin pérdidas. Lo importante del esquema de JPEG es que esta norma inicialmente propuesta en los 80's divide la imagen en bloques de 8×8 píxeles y a cada bloque se le aplica la DCT. Los bloques transformados son cuantizados con un cuantizador escalar uniforme, son barridos en zig-zag y son codificados con un codificador entrópico de Huffman (codificador de símbolos). La cantidad de pasos de cuantización se lleva a cabo para cada uno de los 64 coeficientes DCT y se especifica en tablas las cuales permanecen fijas para todos los bloques. Los coeficientes DCT de todos los bloques se codifican por separado, utilizando un esquema predictivo.

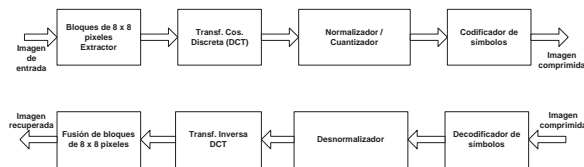


Figura 1. Diagrama para JPEG.

El núcleo de la norma JPEG 2000 [8], [9], es la DWT [6], en donde también se utiliza la cuantización escalar, un modelado contextual, codificación aritmética y post-compresión. La DWT utiliza wavelets diádicas y cuando se utilizan filtros reversibles se tendrá compresión sin pérdidas, o en el caso de filtros no reversibles se tendrá compresión con pérdidas, pues se pueden alcanzar altas tasas de compresión. La DWT hace una descomposición de la imagen en sub-bandas, y cada sub-banda es dividida en bloques de 64×64 , luego se utiliza codificación entrópica utilizando simultáneamente un codificador aritmético de planos de bits. Los datos codificados se organizan por niveles (i.e. layers), los cuales son niveles de calidad. Finalmente se utiliza una post-compresión de acuerdo a una

asignación de longitud de palabra prevista, la cual será una palabra código de salida (paquetes de salida).

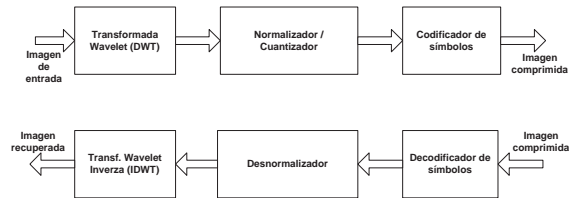


Figura 2. Diagrama para JPEG 2000.

El resto del artículo está organizado de la siguiente forma: en la sección II se presenta brevemente la teoría relacionada con los dos métodos de transformación comparados en este trabajo, en seguida se presentan algunos resultados obtenidos a partir de la interfase gráfica en la sección III. Finalmente se vierten algunas conclusiones y se comenta sobre trabajo a futuro en la sección IV.

II. BREVE DESCRIPCIÓN DE LOS DOS MÉTODOS DE TRANSFORMACIÓN

La compresión de datos nace en la etapa temprana de la computación cuando era necesario el uso de métodos para la reducción del espacio necesario para almacenar la información, ya fuera ésta textos, programas, imágenes o cualquier tipo de información (en la Figura 3 se muestra un diagrama general de compresión), y la evolución ha sido tal que en este momento que los medios de almacenamiento se han vuelto económicos a comparación con los costos de almacenamiento en las primeras etapas de la computación. En las dos últimas décadas se han propuesto diferentes metodologías, en este trabajo se rescatan dos que forman parte importante de las dos normas más recientes a nivel mundial para la compresión de imágenes fijas.

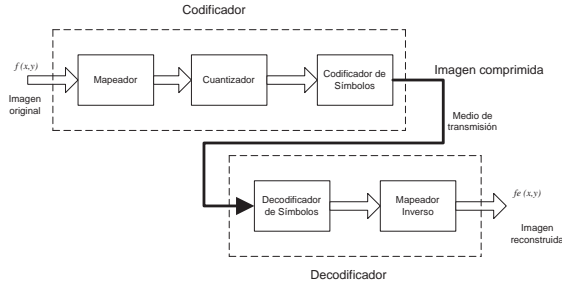


Figura 3. Diagrama general de compresión / descompresión.

A. Transformada del coseno discreta

La DCT para una imagen $f(x)$ de tamaño $N \times N$ se define como (en [5] se describe un poco más a detalle)

$$C(u, v) = \alpha(u)\alpha(v) \sum_{x_1=0}^{N-1} \sum_{x_2=0}^{N-1} f(x) \times \cos\left(\frac{(2x_1+1)u\pi}{2N}\right) \cos\left(\frac{(2x_2+1)v\pi}{2N}\right), \quad (1)$$

en donde x es bi-variable $x = [x_1, x_2]$, $C(u, v)$ son los coeficientes en el dominio frecuencial, y la transformada inversa (IDCT) es

$$f(x) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u)\alpha(v)C(u, v) \times \cos\left(\frac{(2x_1+1)u\pi}{2N}\right) \cos\left(\frac{(2x_2+1)v\pi}{2N}\right), \quad (2)$$

para $x_1, x_2, u, v = 0, 1, \dots, N-1$, en donde

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}}, & \text{para } u = 0, \\ \sqrt{\frac{2}{N}}, & \text{para } u = 1, 2, \dots, N-1. \end{cases}$$

B. Transformada wavelet discreta

La idea detrás de la teoría de wavelets u ondículas es poder representar una señal como una serie de funciones desplazadas y escaladas por medio de factores llamados pesos. Comenzaremos definiendo la función base para la representación de señales

$$\phi(x) = \begin{cases} 1, & 0 \leq x < 1 \\ 0, & \text{Cualquier otro caso.} \end{cases} \quad (3)$$

La función $\phi(x)$ la tomaremos como base para la representación de señales más complejas, para ello deberemos de poder desplazar y escalar la función, definiendo una nueva función

$$\phi_{j,k}(x) = \sqrt{2^j} \phi(2^j \cdot x - k), \quad (4)$$

donde $j \geq 0$ y $0 \leq k \leq 2^j - 1$, tal que j es el factor de escalado y k el de desplazamiento. Con $\phi_{j,k}(x)$ podremos representar cualquier señal dada para lo cual multiplicaremos cada una de nuestras funciones desplazadas por un factor C_i , como se muestra a continuación

$$f(x) = C_0 \cdot \phi_{j_0, k_0}(x) + C_1 \cdot \phi_{j_0, k_1}(x) + \dots + C_{n-1} \cdot \phi_{j_0, k_{n-1}}(x), \quad (5)$$

para calcular cada uno de estos pesos usaremos la siguiente ecuación:

$$C_k = \langle f, \phi_{j,k} \rangle. \quad (6)$$

Entre mayor sea el número de segmentos, se tendrá una aproximación mejor, pero los coeficientes no mejoran la representación de la información ni reducen el número de datos necesarios para su representación.

B.1 Mejorando la DWT

Una mejora a el método anterior es calcular con cada nuevo nivel de análisis una aproximación más detallada a la señal, con ello cada nuevo grupo de coeficientes mejorará la representación de la señal, con el acomodo adecuado de los coeficientes tendremos al principio los coeficientes con frecuencias bajas y al final coeficientes de frecuencias altas, los últimos representan los detalles de la señal. Se procede de la siguiente forma, considere que

$$f(x) \approx \begin{cases} V_0 \\ V_1 = V_0 \oplus W_0 \\ V_2 = V_1 \oplus W_1 = V_0 \oplus W_0 \oplus W_1 \\ V_3 = V_2 \oplus W_2 = V_1 \oplus W_1 \oplus W_2 \\ V_3 = V_0 \oplus W_0 \oplus W_1 \oplus W_2 \\ \vdots \end{cases} \quad (7)$$

donde V_i es la aproximación a la función $f(x)$ y W_i son los detalles de la función, para la reconstrucción a la función original. Entre mayor sea el nivel de reconstrucción nuestra aproximación a $f(x)$ sera mejor, la forma de calcular cada uno, tanto la aproximación como los detalles se realizan con las siguientes funciones:

$$\varphi(x) = \begin{cases} 1, & 0 \leq x < 1, \\ 0, & \text{En otro caso,} \end{cases} \quad (8)$$

$$\psi(x) = \begin{cases} 1, & 0 \leq x < \frac{1}{2}, \\ -1, & \frac{1}{2} \leq x < 1, \\ 0, & \text{En otro caso.} \end{cases} \quad (9)$$

Donde φ es la función de escalamiento de la ondícula y ψ es la función que describe a la ondícula madre, en este caso hemos descrito la función de la ondícula madre de base **Haar**, teniendo la definición básica de la ondícula definiremos las funciones que usaremos para el análisis, las cuales deben de tener la característica de ser escalables y desplazables

$$\varphi_{j,k}(x) = \sqrt{2^j} \varphi(2^j \cdot x - k), \quad (10)$$

$$\psi_{j,k}(x) = \sqrt{2^j} \psi(2^j \cdot x - k). \quad (11)$$

Las cuales se utilizan para calcular tanto la aproximación, como los detalles para ello se procederá de la siguiente forma

$$\begin{aligned} V_0 &= c_{00} \varphi_{0,0}(x) \\ W_0 &= d_{00} \psi_{0,0}(x) \\ W_1 &= d_{10} \psi_{1,0}(x) + d_{11} \psi_{1,1}(x) \\ W_2 &= d_{20} \psi_{2,0}(x) + d_{21} \psi_{2,1}(x) + \\ &\quad d_{22} \psi_{2,2}(x) + d_{23} \psi_{2,3}(x) \\ &\vdots \end{aligned}$$

Para el cálculo de cada uno de los coeficientes de ponderación se procede de la siguiente forma

$$\begin{aligned} c_{00} &= \langle f, \varphi_{0,0} \rangle \\ d_{00} &= \langle f, \psi_{0,0} \rangle \\ d_{10} &= \langle f, \psi_{1,0} \rangle \\ d_{11} &= \langle f, \psi_{1,1} \rangle \\ d_{20} &= \langle f, \psi_{2,0} \rangle \\ d_{21} &= \langle f, \psi_{2,1} \rangle \\ d_{22} &= \langle f, \psi_{2,2} \rangle \\ &\vdots \end{aligned}$$

Calculando los factores y teniendo la función de escalamiento y de ondícula para la reconstrucción (IDWT), sólo tendremos que usar la siguiente ecuación (define el algoritmo à trous [6])

$$f(x) = \underbrace{c_{00} \varphi_{0,0}(x)}_{V_0} + \underbrace{d_{00} \psi_{0,0}(x)}_{W_0} + \underbrace{d_{10} \psi_{1,0}(x) + d_{11} \psi_{1,1}(x)}_{W_1} + \dots$$

$$\underbrace{\hspace{15em}}_{V_1=V_0 \oplus W_0}$$

$$\underbrace{\hspace{25em}}_{V_2=V_1 \oplus W_1=V_0 \oplus W_0 \oplus W_1} \quad (12)$$

B.2 Transformada wavelet rápida (FWT)

La formula para el cálculo de los detalles de la secuencia es la siguiente:

Aproximación

$$W_\varphi(j, k) = h_\varphi(-n) * W_\varphi(j+1, n) \mid_{n=2k, k \geq 0}, \quad (13)$$

Detalles

$$W_\psi(j, k) = h_\psi(-n) * W_\varphi(j+1, n) \mid_{n=2k, k \geq 0}, \quad (14)$$

donde h_ψ es la función de la ondícula madre y h_φ es la función de escala de la ondícula.

$$h_\psi(n) = (-1)^n h_\varphi(1-n). \quad (15)$$

III. ALGUNOS RESULTADOS DE LA INTERFACE

Como ya se comentó, la aportación principal de este trabajo se basa en la construcción de una interface gráfica en la cual se utilizan dos lenguajes de programación de alto nivel como lo son el C++ y el Visual Basic, se generó código en C++ para las diferentes funciones de procesamiento de imágenes enfocadas a la codificación y compresión (codificador Shannon-Fano, de Huffman, algoritmo zig-zag, DCT, DWT, etc.) y por otro lado, se elaboró código en Basic para el diseño gráfico propiamente dicho (ventanas, colores de ventanas, menús, etc.). En la Figura 4 se

presenta una fotografía de la interface gráfica nombrada ANAI (Análisis de Imágenes), más adelante se presentarán algunos resultados propiamente de los dos codificadores comparados en este trabajo. Por el momento, en la Figura 4 se visualizan algunos resultados obtenidos con la DCT sobre la imagen clásica de “Barbara”.

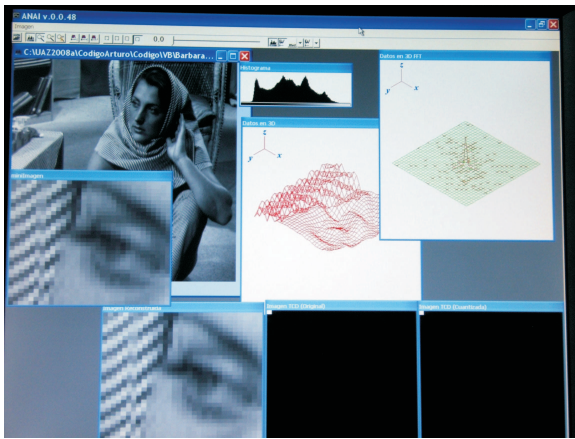


Figura 4. Interface gráfica diseñada (ANAI).

La interface consta de dos menús principales en modo texto y una barra de menú en donde se especifican operaciones tales como la apertura de imágenes a ser tratadas, en un inicio se lleva a cabo el análisis por bloques y para ello hay cuatro botones que permiten seleccionar la ventana de análisis, que va de bloques de 4×4 a bloques de 32×32 . También hay otros tres botones que permiten ver las tablas de códigos generadas por Shannon-Fano, Huffman y aritmética. Es posible también visualizar el histograma de la imagen tratada al momento. También se tienen varios botones que permiten visualizar el bloque de la imagen que se analiza, los valores de los pixeles en términos numéricos, así como un perfil tridimensional (3D) en escala de grises y modo gráfico. Se puede obtener también la transformada rápida de Fourier en dos dimensiones, y se tienen dos sub-menús que por su parte están dedicados a varias op-

eraciones que se llevan a cabo para la DCT y para la DWT. En la Figura 5 se muestra una comparación visual de los resultados obtenidos para la transformación de la imagen de “Lena,” en este caso la tasa de compresión es casi 1:1 por lo que las imágenes recuperadas son de muy buena calidad visual se parecen bastante a la imagen original, se puede ver en la Figura 5 (d) y (e) que corresponden a la IDCT e IDWT con bases Haar que la diferencia con (c) que es la original no hay cambios perceptibles (se utiliza un bloque de análisis de 32×32). Por su parte (a) y (b) corresponden a los coeficientes de la DCT y la DWT.

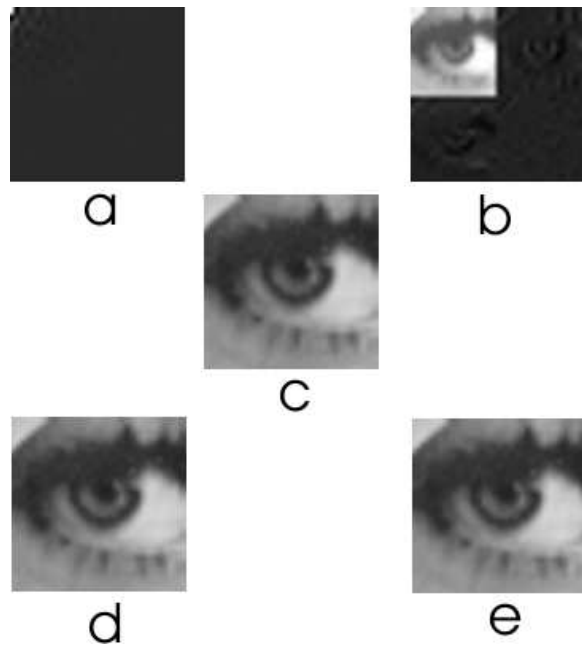


Figura 5. Comparación visual entre la DCT y la DWT de un ojo de Lena, con una tasa de compresión de 1:1.

En la Figura 6 se observa por otro lado, una segunda comparación ahora utilizando otra imagen de prueba conocida como “Goldhill,” en este caso se utilizó la codificación Shannon-Fano, y se tiene una tasa de compresión de 1:20 aproximadamente, se puede ver como la imagen reconstruida cuando se usa la IDCT comienza a sufrir distorsión (a), mientras que

la imagen recuperada a partir de la IDWT no ha sufrido cambios substanciales al compararla con el segmento de la imagen original (c). Las subimágenes (d) y (e) muestran los coeficientes de la DCT y la DWT cuantizados (unque no se notan mucho los valores numéricos asignados a los píxeles).

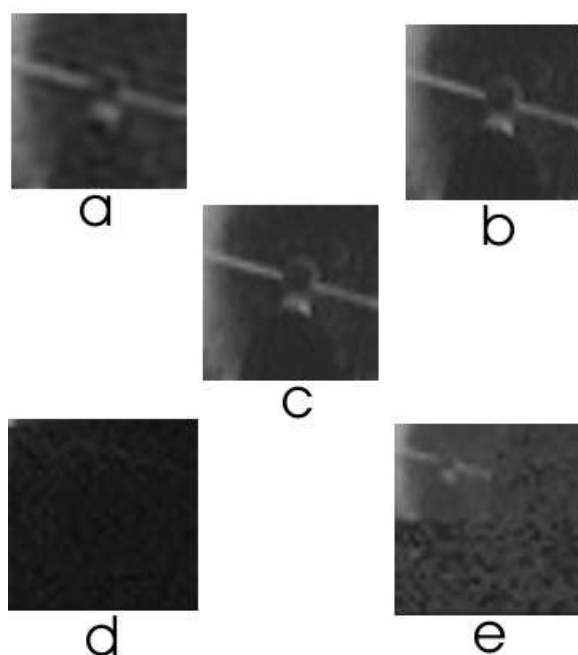


Figura 6. Comparación visual entre la DCT y la DWT de hombre caminando en la imagen Goldhill, con una tasa de compresión de 1:20 aprox.

Finalmente, se comenta que el bloque de análisis se puede desplazar a lo largo y ancho de toda la imagen, llevando a cabo un análisis prácticamente en tiempo real, además, también se pueden realizar otras operaciones adicionales que no describimos en este documento, como la obtención de tablas de códigos, histogramas donde se muestra la variación de la entropía dependiendo de la tasa de compresión deseada, entre otras.

IV. CONCLUSIONES Y TRABAJO FUTURO

Se ha construido una interface gráfica que tiene como propósito quedar en código abierto para seguir incluyendo otras técnicas de codificación, así como técnicas de restauración de imágenes, los resultados obtenidos por el momento muestran consistencia con los reportados en la literatura. Dentro de las técnicas adicionales que se pretenden incluir a corto término, se tiene como meta:

- utilizar bases diferentes a las Haar, como las de Daubechies [10],
- bases de Meyer,
- obien, la transformada ridgelet [1], [2],
- transformada curvelet [3], [4].

RECONOCIMIENTOS

Nuestros más sinceros agradecimientos al Programa de Mejoramiento al Profesorado (PROMEP) de la Secretaria de Educación Pública (SEP) de México, este trabajo fué parcialmente apoyado por PROMEP-SEP bajo registro UAZAC PTC 24-103.5/03/1127.

REFERENCIAS

- [1] Candès, E. J., *Ridgelets: Theory and Applications*, Ph.D. Thesis, Department of Statistics, Stanford University, 1998.
- [2] Candès, E. J., y Guo, F., “New multiscale transforms, minimum total variation synthesis: Applications to edge-preserving image reconstruction”, *Journal of Signal Processing*, Vol. 82, pp. 1519–1543, Jan. 2002.
- [3] Candès, E. J., y Donoho, D. L., “Curvelets and Curvilinear Integrals”, *Journal of Approx. Theory*, Vol. 113, pp. 59–90, 2000.
- [4] Donoho, D. L., y Duncan, M. R., “Digital Curvelet Transform: Strategy, Implementation and Experiments”, in *Proc. Aerosense 2000, Wavelet Applications VII*, SPIE, Vol. 4056, pp. 12–29, 2000.
- [5] R. C. González, R. E. Woods, “Digital image processing,” Ed. Prentice Hall, Second edition, 2002.

- [6] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, Second Edition, 1999.
- [7] D. Santa-Cruz, y T. Ebrahimi, “A study of JPEG 2000 still image coding versus other standards,” *Proc. of the X European Signal Processing Conference*, Vol. 2, pp. 673–676, Tampere, Finland, Sept. 5 - 8, 2000.
- [8] A. Skodras, C. Christopoulos, T. Ebrahimi, “The JPEG 2000 still image compression standard,” *IEEE Signal Processing Magazine*, Vol. 18, No. 5, pp. 3 –58, Sept. 2001.
- [9] B.E. Usevitch, “A tutorial on modern lossy wavelet image compression: foundations of JPEG 2000,” *IEEE Signal Processing Magazine*, Vol. 18, No. 5, pp. 22–35, Sept. 2001.
- [10] I. Daubechies, *Ten lectures on wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, Philadelphia, Pennsylvania, 1992.