Compresión de Imágenes Fijas utilizando la Trasformada Wavelet

Natalia Fournier[1], Gabriela Castro[2], Claudia Russo[3] y Oscar Bria[4]

o.bria@ieee.org

Depto de Informática, Cs. Ex., UNLP, Argentina

RESUMEN

En este trabajo presentamos la aplicación de un algoritmo de compresión de imágenes fijas utilizando la Transformada Wavelet.

La transformada Wavelet es una herramienta coveniente para el análisis multirresolución de señales y en particular se ajusta naturalmente a la compresión de imágenes al adaptar el ancho de banda requerido en forma automática.

Este algoritmo estudia las características de las imágenes en tonos de gris para permitir explotar aspectos importantes del sistema visual humano. El ojo humano es menos sensitivo a las frecuencias espaciales altas (bordes de una imagen) que a las frecuencias espaciales bajas (texturas de una imagen). El método utilizado consiste en codificar con pocos bits los coeficientes que representan frecuencias altas y con más bits los coeficientes de frecuencias bajas.

Las etapas de la compresión son:

- La descomposición Wavelet utilizando diferentes filtros FIR, entre ellos los de Haar y Daubechies.
- La cuantificación durante la cual se lleva a cabo la compresión efectiva, y que comprende dos pasos: la asignación de bits y el umbralamiento y cuantificación.
- la codificación que incluye el método de Run-Length seguido de una codificación de Huffmann dinámica o estática.

La decompresión comprende procesos inversos de los anteriores.

El algoritmo resulta ser efectivo en cuanto a la calidad de las imágenes comprimidas y en pruebas preliminares se han alcanzado índices de compresión del orden de diez veces.

[1]Alumna de Licenciatura en Informática, Facultad de Ciencias Exactas, UNLP.
[2]Alumna de Licenciatura en Informática, Facultad de Ciencias Exactas, UNLP.
[3]Profesora Adjunta, Depto. de Informática, Facultad de Ciencias Exactas, UNLP.
[4]Profesor Adjunto, Depto. de Informática, Facultad de Ciencias Exactas, UNLP.

INTRODUCCIÓN

En este trabajo presentamos la aplicación de un algoritmo de compresión de imágenes fijas utilizando la Transformada Wavelet.

La transformada Wavelet es una herramienta conveniente para el análisis multirresolución de señales y en particular se ajusta naturalmente a la compresión de imágenes al adaptar el ancho de banda requerido en forma automática.

Este algoritmo estudia las características de las imágenes en tonos de gris para permitir explotar aspectos importantes del sistema visual humano. El ojo humano es menos sensitivo a las frecuencias espaciales altas (bordes de una imagen) que a las frecuencias espaciales bajas (texturas de una imagen). El método utilizado consiste en codificar con pocos bits los coeficientes que representan frecuencias altas y con más bits los coeficientes de frecuencias bajas.

La cantidad de información almacenada, transmitida y manipulada por las computadoras ha ido creciendo exponencialmente durante las últimas décadas. Dos desarrollos recientes han contribuido particularmente a este efecto. Uno es la aparición de sistemas multimedia junto con numerosas aplicaciones que las mismas motivaron. La época en que las computadoras manipulaban sólo números y texto hace tiempo ha terminado y ha sido reemplazada por una era de sonido, imágenes, películas y realidad virtual. Otro desarrollo es la creciente disponibilidad de la Internet, que ha hecho que la información esté disponible para una gran cantidad de usuarios.

Este desarrollo solo fue posible por la rápida evolución del hardware. La performance de la CPUs, discos y canales de transmisión ha crecido exponencialmente. Sin embargo uno puede fácilmente encontrar ejemplos donde el hardware corriente es inadecuado (técnica o económicamente).

Las técnicas de compresión proveen una solución. Si podemos representar la información en un formato comprimido podemos obviamente:

- ahorrar almacenamiento,
- ahorrar tiempo de CPU,
- ahorrar tiempo de transmisión.

En vista de lo anterior nos hemos propuesto: Implementar un algoritmo en lenguaje C para la compresión de imágenes fijas utilizando la Transformada Wavelet. Y además implementar una interfaz versátil para experimentar con el algoritmo anterior haciendo uso de diversos tipos de filtros, factores de compresión y otros parámetros.

Los algoritmos de compresión de señales intentan minimizar el espacio requerido minimizando de alguna manera la redundancia de información. Las técnicas de compresión de imágenes se dividen en dos grandes grupos:

- con pérdida,
- sin pérdida.

La compresión sin pérdida se aplica cuando la señal reconstruida debe ser exacta a la señal original, por ejemplo archivos de texto. Con este tipo el radio de compresión es limitado.

En el caso de la compresión con pérdida, se permite un error mientras la calidad después de la compresión sea aceptable. Este esquema tiene la ventaja de que se puede lograr radios de compresión mucho más altos que con compresión sin pérdida.

En cualquier esquema de compresión se pretende eliminar la correlación presente en los datos. Existen varios tipos de correlación:

1- Correlación espacial: Se puede predecir el valor de un pixel en una imagen mirando a los pixels vecinos.

2- Correlación espectral: La transformada de Fourier de una señal es a menudo suave. Esto significa que uno puede predecir un componente frecuencial mirando las frecuencias vecinas.

3- Correlación temporal: En video digital, la mayoría de los pixels de dos frames consecutivos cambian muy poco en la dirección del tiempo (por ejemplo: background)

Uno de los estandares para compresión con pérdida es a través de codificación por transformada. La idea es representar los datos usando una base matemática diferente que revele o no la correlación. En esta nueva base la mayoría de los coeficientes serían tan pequeños que podrán ser seteados a cero.

Un compresor de imágenes comprende de una fase de Codificación y otra de Decodificación.

Codificación : Dada una imagen original I, se aplica una transformación T que consiste en mapear los pixels desde un dominio, por ejemplo espacial, hacia otro, por ejemplo frecuencial, en busca de redundancia. Luego en base a un criterio se determina cúales y cuántos coeficientes deben seleccionarse. Posteriormente, este conjunto pasa por un módulo de cuantificación. En éste proceso se llevan los coeficientes, en base a una tabla específica, a un espacio con menor precisión, requiriendo así menos bits para su representación. Este conjunto de coeficientes cuantificados pasa a un proceso de Codificación ; generalmente se usa un código entrópico que se encarga de asignar de manera eficiente pocos bits a los coeficientes más frecuentes y muchos bits a los menos frecuentes. Finalmente obtenemos un conjunto de datos que representan la imagen comprimida.

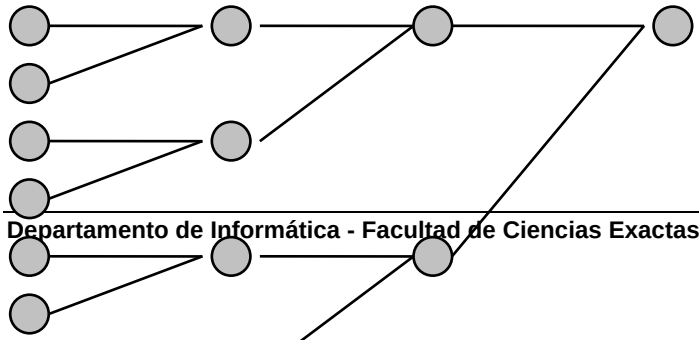
Decodificación : Dada una imagen comprimida I', se aplica en forma inversa los procesos inversos de la etapa de Codificación, es decir : Decodificación, Decuantificación y Antitransformación para reconstriur la imagen similar a la original.

TRANSFORMADA WAVELET

La transformada Wavelet puede verse como un sistema de análisis y síntesis estruucturado piramidalmente. Las pirámides pueden ser usadas para el filtrado de imágenes, construyendo valores a partir de la señal original.

Esta pirámide puede verse como el resultado de aplicar filtros a escala sobre la señal. Para n valores iniciales, tenemos $\log_2(n)$ pasos y $2n$-1 nodos. Sólo se tuvieron que computar n-1 sumas. Sin embargo este no es un buen esquema para reconstrucción, pues hay redundancia en los datos. Llamando $s_{i,j}$ al j-ésimo elemento del nivel y (0 es el tope de la pirámide, k=$\log_2$(n) el nivel más bajo) tenemos:

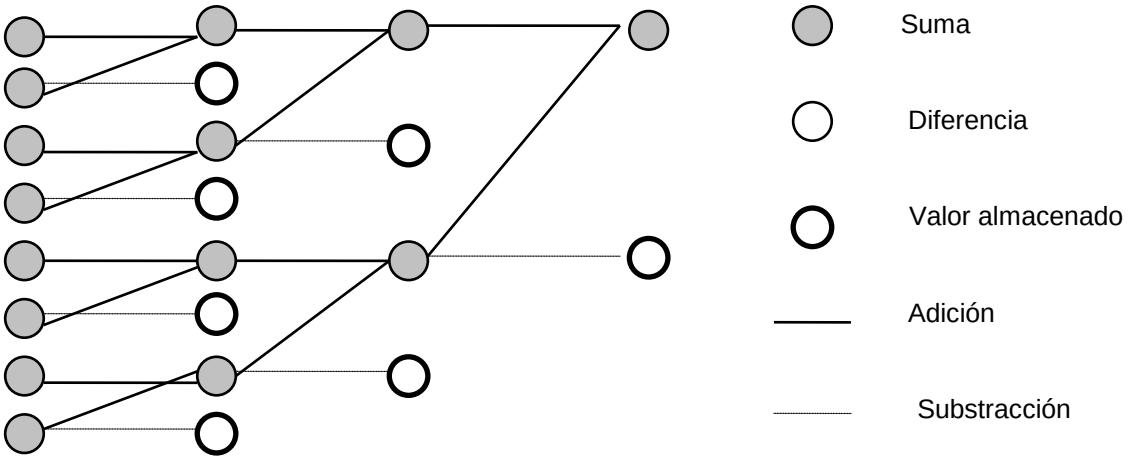
$$s_{i,j} = \frac{\left(S_{i+1,2j} + S_{i+1,2j+1}\right)}{2}$$



Podemos ahora almacenar $s_{0,0}$ como antes pero en el nivel de abajo almacenamos:

$$s'_{1,0} = \frac{(s_{1,0} - s_{1,1})}{2}$$

Es claro que sumando $s_{0,0}$ y $s'_{1,0}$ recobramos $s_{1,0}$ y restando $s_{0,0}$ y $s'_{1,0}$ recobramos s1,1. Por lo tanto tenemos la misma información con un elemento menos. La misma modificación aplicada recursivamente a través de la pirámide resulta en n-1 valores almacenados en k-1 niveles. Como necesitamos el valor tope así como $s_{0,0}$, y las sumas como resultados intermedios, el esquema computacional ahora es el siguiente:



Ahora podemos generalizar el concepto de pirámides de imagen a lo que se conoce como esquema de codificación subbanda. Se aplican recursivamente dos operadores a la señal para submuestrearla por 2. El primero es la función caja, y es un suavizado, o un filtro pasa-bajo, y el otro es la wavelet básica de Haar, o detalle o un filtro pasa-alto. En general, si tenemos un filtro pasa-bajo h(n), un filtro pasa-alto g(n), y una señal f(n), podemos computar la versión suavizada y submuestreada:

$$a(k) = \sum_i f(i)h(-i + 2k)$$

y la versión detallada submuestreada:

$$d(k) = \sum_i f(i)g(-i + 2k)$$

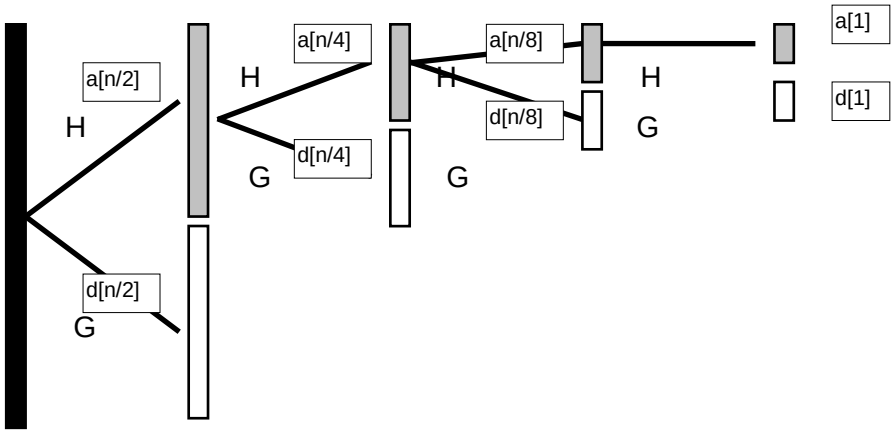
Si el filtro de suavizado es ortogonal a su traspuesto, entonces los dos filtros están relacionados por:

$$g(L-1-i) = (-1)^i h(i)$$

(donde L es la longitud del filtro). La reconstrucción es entonces exacta , y es computada como:

$$f(i) = \sum_k [a(k)h(-i+2k) + d(k)g(-i+2k)]$$

Podemos aplicar este esquema recursivamente a la nueva señal suavizada, que es de la mitad del tamaño de f(), hasta que tenga dos vectores a() y d() de longitud 1 después de $\log_2(n)$ aplicaciones. El esquema computacional es el siguiente:



COMPRESIÓN DE IMÁGENES

Uno de los algoritmos más usados para la compresión de imágenes es el JPEG. El algoritmo divide la imagen en bloques de 8X8 pixels usando en cada uno la Transformada Discreta del Coseno (DCT). La desventaja de esto es que la imagen comprimida revela los bloques y no puede aprovecharse la correlación entre bloques.

Un algoritmo de compresión esencialmente consta de tres pasos: transformada, cuantificación, y codificación.

Transformada wavelet

Para la elección de una determinada wavelet, deben considerarse las siguientes propiedades:

- Soporte compacto: los filtros deben ser FIR finitos.
- Coeficientes racionales: permiten evitar las operaciones de punto flotante.
- Suavidad: si la wavelet no es suave el error será fácil de detectar visualmente.

Longitud de los filtros: son preferibles los filtros cortos, pero hay un trade-off entre estos y la suavidad ya que la misma es proporcional a la longitud de los filtros.

Cuantificación

Un problema que impide la codificación eficiente es el hecho de que los coeficientes de la transformada pueden tener valores arbitrarios. El propósito de la cuantificación es restringir los valores de los coeficientes a un número limitado de posibilidades.

Codificación

El paso de codificación involucra reemplazar, de un modo reversible, el string de símbolos de entrada del cuantificador por un flujo de bits.

Las dos categorías principales son codificación de longitud fija y de longitud variable. En un codificador de longitud fija cada símbolo es reemplazado con el mismo número de bits. Es por lo tanto esencial usar un buen cuantificador. Un ejemplo es el algoritmo de Lloyd-Max.

Una variante más poderosa usa codificación de longitud variable. La idea es asignar las palabras más cortas a los símbolos más frecuentes. Supongamos que una palabra de código k_i tiene probabilidad p_i con

$$\sum_i p_i = 1$$

El contenido de información o entropía ahora está dado por

$$H = -\sum_i p_i \log_2 p_i$$

y esta es la cantidad mínima teórica de bits necesarios por palabra de código. El problema es que *H* no es necesariamente un número natural.

Los codificadores de longitud variable(o codificadores entrópicos) tratan de acercarse lo más posible a este mínimo. Los dos métodos más populares son Huffman y la codificación aritmética.

Debe tenerse en mente que estos codificadores son sólo óptimos es el caso en el que las probabilidades p_i son conocidas. En la práctica uno usualmente tiene que estimar p_i basado en los datos o en alguna información a priori.

Evidentemente, la posición de los coeficientes que fueron seteados a cero tiene que ser codificada también. Esto puede ser hecho con codificación Run Length, la cual es usualmente seguida de codificación entrópica de las longitudes de las corridas.

DESCRIPCIÓN DEL ALGORITMO

Estructuras de Datos

Los datos del árbol de descomposición wavelet no son guardados en una estructura que refleje su forma pues haría más lento el proceso y aumentaría el uso de memoria. Por ello no se usan más que grandes buffers y los archivos definitivos en los cuales se guarda la información.

En el archivo de descomposición los nodos del árbol van guardándose en orden breath-first-search, es decir, por niveles. Por otro lado, los nodos intermedios que son usados en descomposiciones recursivas son mantenidos en archivos temporarios.

Los buffers son fijos y son manipulados como matrices de dos dimensiones. Del mismo modo se manejan los coeficientes de los filtros, en arreglos dinámicos que simulan matrices cuadradas.

Fases de la aplicación

Compresión

- *Ingreso de datos*

  Para la ejecución se requiere la entrada de un archivo de imagen (RAW) cuadrada cuya longitud de lado sea potencia de 2, y los siguientes datos a elegir:

  $\Rightarrow$ coeficientes para generar los filtros (c0...cn-1), y la longitud de los mismos.

⇒ un grado de compresión(cuantificación) que varía entre bajo, mediano y alto.

⇒ opción de usar o no codificación Run-Length.

⇒ opción de usar o no codificación Huffman.

⇒ opción esntre codificación Huffman estática o dinámica.

⇒ si se usa Huffman estático, la tabla huffman a usar o aceptar la tabla por default.

I. *Transformación wavelet*

Comprende el pasaje de la imagen del dominio espacial al frecuencial a través de la aplicación sucesiva de filtros pasabajos y pasaaltos obteniendo los coeficientes wavelet que la reperesentan. Involucra los siguientes pasos:

A. *Construcción de filtros:* A partir de la longitud y coeficientes ingresados se generan los cuatro filtros QMF 2-D, LL(pasabajos en ambas direcciones),LH(pasabajos horizontal y pasaaltos vertical), HL(pasaaltos horizontal y pasabajos vertical), HH(pasaaltos en ambas direcciones).

Estos filtros se arman a partir de los filtros pasabajos (L) y pasaalto(H) de 1-D, los cuales a su vez se arman a partir de c0..cn-1 de la siguiente forma:

El filtro L es un vector de longitud n compuesto por cn-1..c0 en ese orden; los coeficientes del filtro H están dados por ci*(-1)i+1, i:0..n-1.

Cada elemento de la matriz del filtro 2-D, se calcula de la siguiente forma:

LL(i,j)=L(j)*L(i)

LH(i,j)=L(j)*H(i)

HL(i,j)=H(j)*L(i)

HH(i,j)=H(j)*H(i), donde i=0..n-1, j=0..n-1.

B. *Lectura:* La lectura de la imagen se efectúa por ventanas rectangulares cuyo ancho es el de la señal y el largo es MaxRead/ancho.

Como el filtrado es cíclico, la primera vez se leen las últimas filas de la matriz seguidas de las primeras. Las siguientes veces se leen las últimas filas de la ventana leída anteriormente y se completa con las subsiguientes filas de la matriz.

C. *Descomposición:* El proceso de naturaleza recursiva, se realiza por niveles a partir de la imagen original. Inicialmente se aplican los cuatro filtros sobre la matriz de la imagen completa, obteniéndose cuatro matrices que componen el nivel siguiente del proceso, cuyo tamaño se reduce en cuatro por la decimación. En los sucesivos niveles la aplicación del filtro se efectúa sobre la matriz generada por el filtro LL.

1. *Filtrado de una matriz*

Consiste en aplicar el filtro en apoyándolo sobre elementos intercalados de la matriz cada vez. De esta forma se produce la decimación tanto en las filas como en las columnas de la matriz de la imagen filtrada. Como la decimación es por dos en cada dimensión, la decimación total es por cuatro, es decir la matriz filtrada tendrá la cuarta parte del tamaño de la matriz de la cual proviene.

Para no agregar ceros (equivalentes a negro pues es cero luiminosidad) alrededor de la imagen para el caso de filtrar los bordes, se efectúa un filtrado circular. Tal forma de filtrado consiste en completar los datos faltantes "debajo" del filtro, con los dartos del otro extremo de la fila o columna de la matriz de la imagen. De este modo cada fila o columna de la matriz funciona como un arreglo circular sobre el cual se aplican los filtros.

2. *Aplicación del filtro*

Dada F la matriz del filtro a aplicar y M la submatriz de la imagen sobre la cual se aplica el filtro, cada elemento de la matriz filtrada se calcula:

- sum=0
- for i=0..n
- for j=0..n
- sum=sum + M(n-1-i,n-1-j) * F(i,,j)

(depende de n, longitud del filtro).

Como puede leerese en el seudocódigo, el producto de convolución entre la submatriz y el filtro, se realiza hacia adelante en el filtro, pero hacia atrás en la matriz, de manera que se comienza a filtrar sobre el extremo inferior derecho de la submatriz de imagen, con el extremo superior izquierdo del filtro.

II. *Cuantificación:* Con el árbol de descomposición armado se procede a reducir el tamaño de la información con una pérdida irreversible. Éste es el paso que realiza la compresión *real* de la señal.

B. *Normalización*

Sobre los datos del árbol de descomposición wavelet se realiza una normalización que consiste en restarle a los coeficientes de cada nodo la media del mismo, obteniendo así una distribución de probabilidades con media 0.

C. *Cálculo de coeficientes para la asignación de bits*

Primero se busca el coeficiente de mayor valor absoluto para cada nodo, pues es el factor principal en la asignación de los bits. Simultáneamente se calcula la proporción de energía que aporta cada nodo al total. Se considera como energía a la suma del cuadrado de los valores de cada nodo y la proporción es tomada como el cociente entre la energía de un nodo y la energía total de la imagen transformada. Se toma en cuenta además un peso de cada nodo, que es el logaritmo de un valor inversamente proporcional al tamaño del lado de cada uno.

D. Asignación de bits

Todos los coeficientes calculados en el paso anterior, junto con un nivel de compresión ingresado por el usuario(bajo, mediano o alto), es utilizado para el cálculo de los bits que le serán alocados a cada nodo.

E. Umbralamiento y cuantificación

Una vez asignados los bits, se recorre el árbol seteando a 0 los coeficientes cuyo módulo es menor que un umbral dado y luego efectivizando la cuantificación. Esto último se logra dividiendo los coeficientes de cada nodo por un valor que reduce su magnitud tal que el máximo coeficiente pueda representarse con la cantidad de bits asignados.

F. Almacenamiento

Sólo los bits asignados a cada nodo son guardados para cada valor cuantificado. Juntamente con esta información, se almacena la cantidad de bits asignados a cada nodo, para la posterior decuantificación.

- *Codificación*

Una vez cuantificados los coeficientes se codifican entrópicamente para suprimir la redundancia existente en el lenguaje binario y acercar la longitud promedio lo más posible a la entropía de los datos.

Esto se logra con codificación Run Length y luego codificación Huffman.

G. Codificación Run Length:

Para suprimir los ceros consecutivos que aparecen naturalmente después de cuantificar, por el umbralamiento y la reducción de precisión, se utiliza el código Run Length, que es tanto mejor cuanto más largas sean las corridas de ceros. Se da la posibilidad al usuario de utilizar o no estos códigos para comprobar su efectividad y aporte a la compresión total.

H. *Codificación Huffman:*

Con el objeto de acercar la longitud promedio de los datos al límite inferior que es la entropía, se aplica el código Huffman. Se da la posibilidad al usuario de elegir entre Huffman estático o dinámico, pudiendo además en el primer caso proveer una tabla o aceptar la que está por default. En el caso de codificación dinámica hay un paso anterior que no se efectúa en la codificación estática:

1. *Armado de la tabla Huffman*

Para armar la tabla de codificación dinámica deben hacerse estadísticas sobre los datos a codificar con lo cual los mismos son recorridos dos veces, una al armar la tabla y otra al reemplazar los datos por sus respectivos códigos.

a) Estructuras de datos

La tabla se manipula a través de una lista implementada con un arreglo, cuyos componentes contienen un símbolo, su probabilidad, la dirección del elemento anterior de la lista y del siguiente (ordenados de mayor a menor por probabilidad), la dirección de los dos sumandos que componen la probabilidad (si es resultado de una suma).

De este modo se facilita la ordenación de las probabilidades iniciales(una por cada símbolo), y la inserción de las nuevas probabilidades, que son resultado de la suma de las dos menores.

b) Extracción de estadísticas

En esta parte del proceso se recorre el archivo de datos contando la cantidad de veces que aparece cada símbolo y luego dividiendo esa cantidad por el total de símbolos contenidos en el archivo, lo cual constituye la probabilidad de aparición de cada símbolo en ese conjunto de datos.

c) Formación de los códigos

Una vez calculadas las probabilidades de los símbolos, las mismas son ordenadas de mayor a menor. Entonces se van agrupando las dos últimas probabilidades para formar una nueva, igual a la suma de ambas, sucesivamente hasta llegar a tener sólo dos. Durante este procedimiento, se van haciendo los enganches necesarios en la lista que manipula la tabla Huffman para el siguiente paso.

Seguidamente se vuelve hacia atrás restaurando los sumandos de cada nueva probabilidad, y al mismo tiempo asignando un nuevo bit de código a cada par de probabilidades mayores, hasta tener nuevamente la lista de probabilidades de los símbolos originales, con sus respectivos códigos.

Una vez completo el proceso de asiganción de códigos, se vuelca el contenido de la lista que manipula la tabla en un arreglo unidimensional de longitud igual a la cantidad de símbolos codificados, donde cada posición guarda el código igual a su índice.

2. *Asignación de códigos*

Con los códigos contenidos en la tabla de Huffman, se recorre el archivo reemplazando cada símbolo por su correspondiente código Huffman.

Descompresión

- *Decodificación*

Se aplica el decodificador Huffman, si así corresponde, y luego el descompresor Run Length.

I. *Decodificador Huffman*

Haciendo uso de la tabla Huffman, almacenada en archivo aparte si es estático, o en el archivo comprimido si es dinámico, se restauran los valores originales a partir de los códigos Huffman.

Se obtiene un bit por vez y se busca en la tabla si la concatenación de bits actual corresponde a un código Huffman. Si es así, se reemplaza el código por el símbolo respectivo, si no, se obtiene otro bit y se repite el proceso.

J. *Descompresor Run Length*

Se descomprime el archivo volviendo a poner en su lugar los ceros suprimidos. Esto se logra escribiendo la cantidad indicada de ceros cada vez que se encuentra una marca de Run Length seguida del número correpondiente de ceros codificados.

II. *Decuantificación*

Se revierte en parte el proceso de cuantificación multiplicando los coeficientes de cada nodo por el valor correspondiente, volviendo así a la representación floating point. La pérdida se produce por la umbralación, que no tiene proceso inverso, y por la reducción de precisión en la representación de los coeficientes.

Desnormalización: Para completar la restauración de los datos wavelet se le suma a cada coeficiente la media de su nodo, para que éste recupere la distribución de probabilidades original.

III. *Antitransformada wavelet*

La imagen, que para ser comprimida había sido transformada pasando del dominio espacial al frecuencial, ahora debe ser antitransformada para retornar al dominio original. Para ello se aplica la transformada inversa wavelet, que consiste en aplicar los filtros inversos , tanto pasabajos como pasaaltos, a la imagen transformada desde las hojas del árbol de descomposición(último nivel), hasta la raíz. El proceso involucra los siguientes pasos:

A. *Construcción de filtros inversos:* A partir de la longitud y coeficientes ingresados se generan los cuatro filtros QMF 2-D, LL'-LH'-HL'-HH' de manera análoga a los filtros usados para la descomposición.

Los filtros inversos de una dimensión son los traspuestos de L y H. Esto significa que $L_i^{-1}=L_{n-1-i}$ y $H_i^{-1}=H_{n-1-i}$ donde i:0,..,n-1 y n=longitud de los filtros

B. *Filtrado de los nodos:* Se inicia desde el último nivel de descomposición integrado por las cuatro matrices generadas por los filtros LL-LH-HL-HH. A partir de éstas, se aplica a cada una el correspondiente filtro HH-1,HL-1,LH-1, y LL-1 y se suma elemento a elemento para obtener la matriz generada por el filtro LL del nivel anterior. Así sucesivamente hasta obtener la raíz, que es la imagen original.

1. Aplicación de un filtro inverso: Se apoya la matriz que representa al filtro sobre una parte del nodo cada vez, realizando nuevamente un producto por convolución entre la imagen y el filtro. Como en la descomposición se decimaba la aplicar el filtro ahora hay que interpolar para recuperar el tamaño original. La interpolación consiste en considerar nulos los valores de los elementos intermedios de las matrices, lo cual se implementa salteando un coeficiente del filtro en ambos sentidos al efectuar el producto por convolución, representando la multiplicación por cero.

RESULTADOS

A continuación se muestran imágenes en tonos de gris y las respectivas imágenes resultantes de la compresión, descompresión wavelet y la imagen diferencia, de cuya comparación puede observarse la relación factor de compresión - calidad visual.


lena.bmp

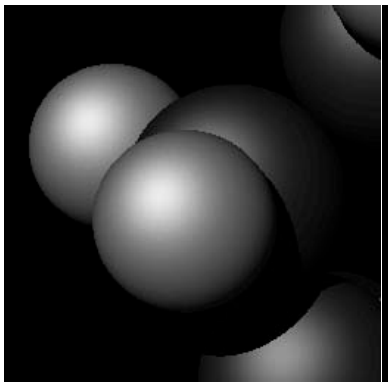

lena7.bmp

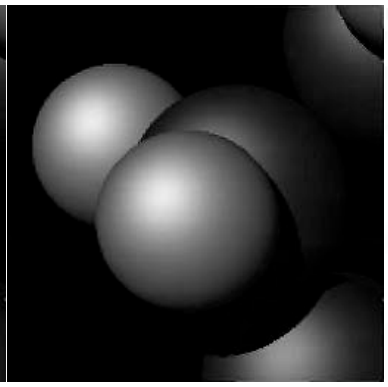

pérdida


land.bmp


land7.bmp


pérdida


mole.bmp


mole7.bmp

pérdida

Factor de Compresión : bueno

(Con filtros de Daubechies)

IMAGEN	TAMAÑO ORIGINAL (BITS)	COMPRESIÓN		COMPRESIÓN S/HUFFMAN		COMPRESIÓN S/RUNLENGTH		COMPRESIÓN S/HUFFMAN Y RUNLENGTH	
		BITS	RADIO	BITS	RADIO	BITS	RADIO	BITS	RADIO
LENA	65.536	6359	10,3	7964	8,22	7358	8,90	21651	3,02
MOLE	65.536	4157	15,7	4711	13,91	5886	11,13	23277	2,81
LAND	65.536	10281	6,37	12103	5,41	10747	6,09	19203	3,41

(Con JPEG Base Line)

IMAGEN	TAMAÑO ORIGINAL (BITS)	COMPRESIÓN RADIO
LENA	65.536	8.80
MOLE	65.536	25.24
LAND	65.536	4.03

CONCLUSIONES

Analizando los resultados del cuadro anterior, concluimos que la aplicación de la transformada wavelet sobre los datos conduce a una pérdida menos perceptible de calidad visual debida a la cuantificación. Puede agregarse que si bien por este último paso se reduce notablemente el tamaño de la imagen, es la codificación entrópica, llevada a cabo con códigos RunLength y Huffman, la que permite un acercamiento de la longitud promedio de los mismos al límite marcado por la entrpía.

El uso de la transformada wavelet para compresión de imágenes resulta en una buena relación entre factor de compresión y calidad visual, puesto que a través del filtrado son separadas las distintas componentes frecuenciales y luego son comprimidas de acuerdo a la importancia que tengan para el "ojo humano". La aplicación de la transformada sobre toda la imagen en lugar de por bloques, impide que los mismos se noten en la imagen resultante, como ocurre con la transformada del coseno.

Como tabajo futuro se sugiere la compresión de imágenes color y video, así como también la aplicación de cuantificación vectorial en lugar de escalar. También podría extenderse la aplicación a otros formatos de imágenes.

BIBLIOGRAFÍA

- *Wavelets and their Application in Computer Graphics,* SIGGRAPH '94 Course Notes. Alain Fournier, university of British Columbia.
- *Fundamentals of digital image processing*. Anil K. Jain.
- *JPEG-Like Image Compression*. Craig A. Lindley, Dr. Dobb's. 1995.
- Discrete Wavelet Transforms: *Theory and Implementation*. Tim Edwards, Stanford University, 1991.
- *Binary Tree Predictive Coding*. John A. Robinson, University of Waterloo.
- *The Fast Wavelet Transform*. Mac. A. Cody, Dr. Dobb's. 1992.
- *The Wavelet Packet Transform*. Mac A. Cody, Dr. Dobb's. 1994.
- *Wavelet Filter Evaluation for Image Compression*. John D. Villaseñor, Benjamin Belzer, Judy Liao. 1995.

- *Representación adaptiva de imágenes en bases de datos ortogonales óptimas, y aplicaciones a compresión*. Félix Safar, Fabián Blasetti, Juan Pablo Villa, Daniel Cocimano, Sergio Katz. 1991.
- Trabajo de Grado : *"Un algoritmo Paralelo Adaptativo de Compresión de Imágenes en tonos de gris".* Juan Pablo Villa.
- *Wavelet Sampling Tecniques*. Wim Sweldens, Robert Piessens.
- *Recent Advances in Wavelet Technology*. R.O. Weels. 1994.
- A Theory for Multiresolution Signal Descomposition. The Wavelet Representation. Stephan G. Mallat.
- *Design of Linear Phase Cosine Modulated Filter Bank for Subband Image Compression*. J.E. Odegard, R.a. Gophinat, C.S. Burrus.
- *Linear-Phase M-Band Wavelets with Application to Image Coding*. Peter M. Heller, Truong Q. Nguyen, Hemant Singh, W. Knox Carey.
- *Image Restoration with Noise Suppression Using the Wqavelet Transform*. Jean-Luc Starck. 1994.
- *Vector Quantization and Signal Compression.* Allen Gersho, Robert M. Gray.
- Digital Coding of Waveforms: *Principles and Applications to Speech and Video*. N.S. Jayant.