# UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

ESCUELA DE INGENIERÍA UNIDAD ENSENADA


# ANÁLISIS DE WAVELETS (ONDÍCULAS) PARA EL PROCESAMIENTO DIGITAL DE SEÑALES

TESIS PROFESIONAL QUE PARA OBTENER EL TÍTULO DE INGENIERO EN ELECTRÓNICA

PRESENTA:


David Milton Vizcarra Llamas


DIRECTOR DE TESIS

M.C. Humberto Cervantes de Ávila



Ensenada, B.C. Junio de 1998

# Índice


| | Página |
|---|---|
| Índice de tablas y figuras. | 3 |
| I. Introducción. | 4 |
| 1.1 Objetivo. | 4 |
| 1.2 Procesamiento Digital de Imágenes. | 4 |
| 1.3 Pasos fundamentales para el procesamiento de imágenes. | 4 |
| 1.4 Análisis de Fourier. | 6 |
| II. Análisis de Wavelets. | 8 |
| 2.1 Antecedentes. | 8 |
| 2.2 Bancos de filtros. | 10 |
| 2.3 Base Wavelet Haar. | 15 |
| 2.4 Procesos de promediación y diferenciación. | 18 |
| III. Aplicaciones de la Transformada Wavelet. | 20 |
| 3.1 Gráficas en computadora. | 20 |
| 3.2 Procesamiento Digital de Señales. | 20 |
| 3.3 Transmisión progresiva de imágenes. | 22 |
| 3.4 Compresión de señales. | 22 |
| 3.4.1 Compresión de imágenes. | 23 |
| 3.4.2 Implementación de la Transformada Wavelet en Matlab | 26 |

## Índice


| | Página |
|---|---|
| Índice de tablas y figuras. | 2 |
| I. Introducción. | 4 |
| 1.1 Objetivo. | 4 |
| 1.2 Procesamiento Digital de Imágenes. | 4 |
| 1.3 Pasos fundamentales para el procesamiento de imágenes. | 4 |
| 1.4 Análisis de Fourier. | 6 |
| II. Análisis de Wavelets. | 8 |
| 2.1 Antecedentes. | 8 |
| 2.2 Bancos de filtros. | 10 |
| 2.3 Base Wavelet Haar. | 15 |
| 2.4 Procesos de promediación y diferenciación. | 18 |
| III. Aplicaciones de la Transformada Wavelet. | 20 |
| 3.1 Gráficas en computadora. | 20 |
| 3.2 Procesamiento Digital de Señales. | 20 |
| 3.3 Transmisión progresiva de imágenes. | 22 |
| 3.4 Compresión de señales. | 22 |
| 3.4.1 Compresión de imágenes. | 23 |
| 3.4.2 Implementación de la Transformada Wavelet en Matlab | 26 |

IV. Conclusiones. 37

V. Bibliografía. 38

5.1 Libros y artículos científicos. 38

5.2 Páginas de Internet. 39

VI. Apéndice. 40

6.1 Guía para utilizar los códigos en Matlab. 40

6.2 Contenido del disco anexo. 42

## Índice de tablas y figuras


| Figuras | Página |
|---|---|
| 2.1 Descomposición Wavelet de una imagen en estructuras. | 6 |
| 2.2 Banco de filtros. | 8 |
| 2.3 Banco de filtros pasabajas recursivos. | 9 |
| 2.4 Función de dilación. | 12 |
| 2.5 Función Wavelet. | 12 |
| 3.1 Imagen del macaco. | 21 |
| 3.2 Transformada Wavelet de la imagen del macaco. | 22 |
| 3.3 Imagen comprimida del macaco. | 23 |
| 3.4 Fotografía del planeta Saturno. | 23 |
| 3.5 Transformada Wavelet de la imagen de Saturno. | 24 |
| 3.6 Imagen comprimida con $\varepsilon = 20$. | 24 |
| 3.7 Imagen comprimida con $\varepsilon = 100$. | 25 |

| Tablas | Página |
|---|---|
| 3.1 Factores de compresión para la imagen del macaco. | 22 |
| 6.1 Imágenes contenidas en el disco anexo. | 31 |
| 6.2 Código Matlab contenido en el disco anexo. | 31 |

# I. Introducción.

## 1.1 Objetivo.

Investigar e implementar en el área de Procesamiento Digital de Señales una alternativa a la Transformada de Fourier para el procesamiento de imágenes que nos permita identificar características en la frecuencia sin perder información en el espacio.

## 1.2 Procesamiento Digital de Imágenes.

El Procesamiento Digital de Imágenes es el área del Procesamiento de Señales para funciones en 2 ó más dimensiones. Su interés radica en dos principales áreas: el mejoramiento de información pictográfica para la interpretación humana y el procesamiento de datos escenarios para la percepción de máquinas automáticas.

## 1.3 Pasos fundamentales para el procesamiento de imágenes.

*Adquisición de imágenes.*

Este proceso requiere un sensor de imágenes y la capacidad de digitalizar la señal producida por el sensor. Este sensor puede ser una cámara de TV a color o monocromática que produce una imagen completa del dominio a analizar cada 1/30 de segundo. Puede ser una cámara de rastreo que produce una sola imagen cada vez. Si la salida de la cámara o sensor de imagen no se encuentra en forma digital se requiere de un convertidor analógico - digital. La naturaleza del sensor y la imagen que produce son determinadas por la aplicación.

*Preprocesamiento.*

Ya que la imagen fue obtenida, el siguiente paso es el preprocesamiento de la imagen. La función clave del preprocesamiento es mejorar la imagen de tal manera que se aumente la probabilidad de éxito de los siguientes procesos. En este ejemplo, el preprocesamiento típicamente trata con técnicas para mejorar el contraste, reducir el ruido, aislar regiones donde la textura indica información correlacionada o alfanumérica.

*Segmentación.*

En este proceso se divide la imagen en sus partes u objetos constituyentes. En general la segmentación autónoma es una de las tareas más difíciles en el procesamiento de imágenes. En términos de reconocimiento de caracteres. El papel clave de la segmentación es extraer caracteres individuales y palabras del fondo. La salida de la etapa de segmentación es usualmente datos crudos de pixeles, que constituyen ya sea los limites de la región o todos los puntos en esa región.

*Representación.*

De una u otra manera, convertir los datos a una manera apropiada para su representación en una computadora es necesario. Y la decisión sobre la forma de representación recae en si el foco de la información es sobre las propiedades internas como la textura de una pieza ósea.

*Descripción.*

Un método debe ser especificado para describir los datos para que los rasgos de interés sean acentuados. La descripción, también llamada selección de rasgos, trata con la extracción de rasgos para diferenciar una clase de objetos de otra. En términos de reconocimiento de caracteres, los descriptores como los lagos (hoyos) y bahías son rasgos poderosos que ayudan a diferenciar una parte del alfabeto de otra.

*Reconocimiento e interpretación.*

La última etapa envuelve reconocimiento e interpretación. El reconocimiento es el proceso que asigna una etiqueta a un objeto basado en la información provista por sus descriptores. La interpretación envuelve el asignar un significado a un ensamble de objetos reconocidos.

## 1.4 Análisis de Fourier.

En comunicaciones la teoría de Fourier es esencia para entender como se comportará una señal al pasar por un filtro, amplificador o canal de comunicación. Aún en el caso de la comunicación digital donde unos y ceros son transmitidos, existe información en la frecuencia. La mayoría de los filtros son designados para atenuar ciertos componentes de frecuencia de una señal, ya sea pasabajas, pasaaltas o pasa/rechaza banda, pero la Transformada de Fourier no se limita al análisis de filtros sino que es una herramienta valiosa para analizar la señal en el dominio de la frecuencia antes y después del filtrado.

En el contexto de procesamiento de imágenes se ha investigado una alternativa al análisis de Fourier basado en una función prototipo simple, y sus escalas y traslaciones. La transformada de Fourier, provee información sobre el dominio de la frecuencia, sin embargo la información localizada en el tiempo se pierde en el proceso. El problema con esto es la incapacidad de asociar características en el dominio de la frecuencia con su localización en el tiempo, como un cambio en el espectro de frecuencia resultará en cambios a través del dominio del tiempo.

# II. Análisis de Wavelets.

## 2.1 Antecedentes.

Wavelet u ondícula es un diminutivo de *wave* (onda) y representa una familia de funciones relacionadas por un parámetro de tamaño.

Al igual que el análisis de Fourier, el análisis de wavelets descompone una señal en las partes que la componen para sus análisis pero donde la Transformada de Fourier descompone la señal en series de senos de diferentes frecuencias, la transformada wavelet la descompone en sus ondículas (o wavelets), versiones escaladas y trasladadas de cierta wavelet madre.

Cuando se analizan señales de naturaleza no estacionaria, es un beneficio el ser capaz de adquirir una correlación entre los dominios de tiempo y frecuencia de una señal. En contraste con la transformada de Fourier, la transformada wavelet permite localización excepcional tanto en el dominio del tiempo por medio de traslaciones de la wavelet madre, como en la dominio de escala (frecuencia) por medio de dilaciones. Las operaciones de traslación y dilación aplicadas a la wavelet madre son realizadas para calcular los coeficientes wavelet, que representan la correlación entre la wavelet y una sección localizada de la señal. Los coeficientes wavelet son calculados para cada segmento wavelet, dando una función en la escala de tiempo relacionando la correlación de la wavelet con la señal. La señal original es descompuesta sucesivamente en componentes de baja resolución, mientras los componentes de alta frecuencia son analizados inmediatamente. El máximo número de dilaciones que pueden ser realizadas depende del tamaño de la señal a ser analizada, con 2N muestras de datos permitiendo la descomposición de la señal en N niveles discretos utilizando la transformada discreta wavelet.

Para generar una descomposición en tamaños, debemos reducir la onda original para aplicar al área más pequeña. Este procedimiento (mostrado en la Figura 2.1) es similar a dividir una página en cuatro partes iguales y dividir la esquina superior izquierda de la misma manera, si se continúa este proceso se formaran 10, 13 y después 16 rectángulos. Usando la Transformada Wavelet, cada uno de los rectángulos contendrá información que representa un aspecto de los datos en la imagen.

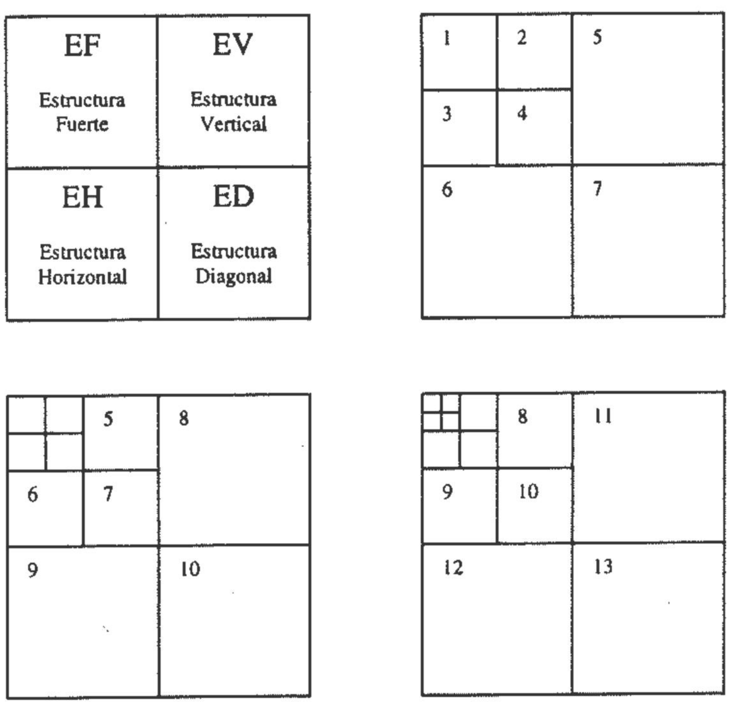



Figura 2-1. Descomposición Wavelet de una imagen en estructuras.

El proceso de descomposición del 25% del área de la última descomposición mostrado como "Fuerte" da lugar a la idea de las wavelets. El propósito de la descomposición en wavelets es tomar ventaja del sistema visual humano. Por ejemplo, la combinación ojo/cerebro es particularmente sensible a las líneas horizontales y verticales, así que en general, los valores en las porciones "Vertical" (EV) y "Horizontal" (EH) de la descomposición son más importantes que las de la porción Diagonal. Los valores en la porción "Fuerte" (EF) son más importantes que los valores en las otras tres áreas, ya que el ojo es más sensible a los valores representados aquí. Dado este fenómeno fisiológico, decidir como descartar datos para compresión no es difícil. Generalmente, los datos son descartados en la porción ED luego en las porciones EH y EV de la imagen. El aspecto artístico del proceso de compresión wavelet es ahora como distribuir la cantidad de datos descartados en estas porciones cada vez más pequeñas de una manera tal que se engañe al ojo/cerebro en no notar el impacto causado por la remoción de estos datos.

**2.2 Bancos de fitros.**

La primera wavelet (y la única por bastante tiempo) fue encontrada por Haar en el inicio de este siglo. Pero la construcción de wavelets más generales para formar bases para funciones integrables cuadradas fue investigada en los ochenta, así como algoritmos más eficientes en el proceso de señales que han surgido. Una formalización de tales construcciones creó un marco de trabajo para las expansiones llamado análisis de multirresolución, y estableció enlaces con métodos usados en otros campos.

El proceso de descomposición del 25% del área de la última descomposición mostrado como "Fuerte" da lugar a la idea de las wavelets. El propósito de la descomposición en wavelets es tomar ventaja del sistema visual humano. Por ejemplo, la combinación ojo/cerebro es particularmente sensible a las líneas horizontales y verticales, así que en general, los valores en las porciones "Vertical" (EV) y "Horizontal" (EH) de la descomposición son más importantes que las de la porción Diagonal. Los valores en la porción "Fuerte" (EF) son más importantes que los valores en las otras tres áreas, ya que el ojo es más sensible a los valores representados aquí. Dado este fenómeno fisiológico, decidir como descartar datos para compresión no es difícil. Generalmente, los datos son descartados en la porción ED luego en las porciones EH y EV de la imagen. El aspecto artístico del proceso de compresión wavelet es ahora como distribuir la cantidad de datos descartados en estas porciones cada vez más pequeñas de una manera tal que se engañe al ojo/cerebro en no notar el impacto causado por la remoción de estos datos.

## 2.2 Bancos de filtros.

La primera wavelet (y la única por bastante tiempo) fue encontrada por Haar en el inicio de este siglo. Pero la construcción de wavelets más generales para formar bases para funciones integrables cuadradas fue investigada en los ochenta, así como algoritmos más eficientes en el proceso de señales que han surgido. Una formalización de tales construcciones creó un marco de trabajo para las expansiones llamado análisis de multirresolución, y estableció enlaces con métodos usados en otros campos.

Paralelamente a los avances en matemáticas puras y avanzadas estuvieron los de procesamiento de señales, pero en el contexto de señales discretas en el tiempo. Influenciada por tales aplicaciones como la compresión de voz e imágenes, se propuso un método de codificación de subbandas en los setenta, que ocasionó el estudio de bancos perfectos de filtros reconstrucción, un problema resuelto en los ochenta.

De una configuración particular, más específicamente cuando el banco de filtros tiene bandas octavas, se obtiene una serie de wavelets discreta en el tiempo. Tal configuración ha sido muy popular en el procesamiento de señales no tanto por sus propiedades matemáticas sino porque una banda octava o espectro logarítmico es más natural para ciertas aplicaciones como la compresión de audio ya que emula el proceso de audición. Tal banco de filtros de bandas octavas puede ser usado, bajo ciertas condiciones, para generar bases de wavelets, como la wavelet Daubechies. La función de dilación de la transformada discreta wavelet puede ser representada como un árbol de filtros pasabajas y pasa altas, con cada paso transformando el filtro pasabajas.

Un banco de filtros H transforma una entrada $x$ en una salida y = H($x$). La Figura 2.2 muestra el ejemplo familiar de un banco de filtros de síntesis que separa las frecuencias bajas y altas de una señal de entrada discreta.

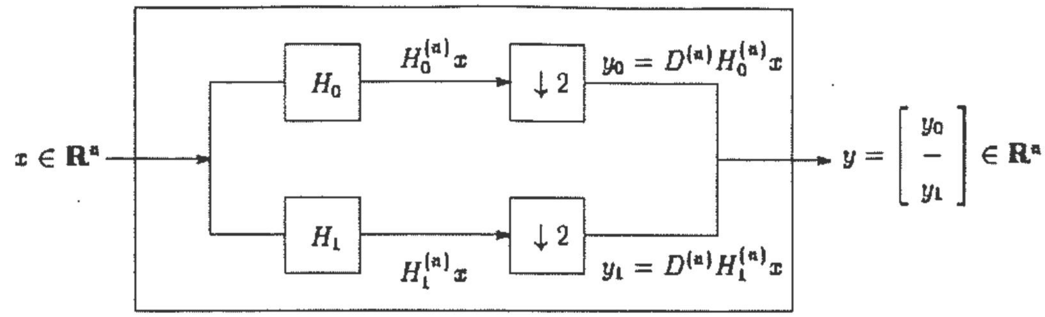



Figura 2.2 Banco de filtros

Si el filtro de bancos es lineal, entonces las correspondientes transformaciones pueden ser representadas como una matriz H, que se obtiene calculando el producto de la matriz transformación con la matriz entrada ($y = Hx$).

A modo de ejemplo, podemos calcular la matriz H para el banco de filtros de síntesis H, asumiendo una entrada $x$ que consiste de n muestras ($x \in \mathbf{R}^n$). Podemos recordar que el filtro pasa bajas Haar $H_0$ simplemente promedia elementos adyacentes de su entrada. Esta operación puede ser representada por la matriz:

$$H_0^{(n)} = \frac{1}{2}\begin{bmatrix} 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & 1 & 1 & & \\ & & & \ddots & & \\ & & & & 1 & 1 \\ & & & & & 1 \end{bmatrix} \in \mathbf{R}^{n \times n}.$$

El filtro pasaaltas $H_1$ calcula la mitad de la diferencia entre muestras sucesivas de la entrada, y puede ser representada como:

$$H_1^{(n)} = \frac{1}{2}\begin{bmatrix} 1 & -1 & & & & & \\ & & 1 & -1 & & & \\ & & & & 1 & -1 & \\ & & & & & \ddots & \\ & & & & & 1 & -1 \\ & & & & & & 1 \end{bmatrix} \in \mathbf{R}^{n\times n}.$$

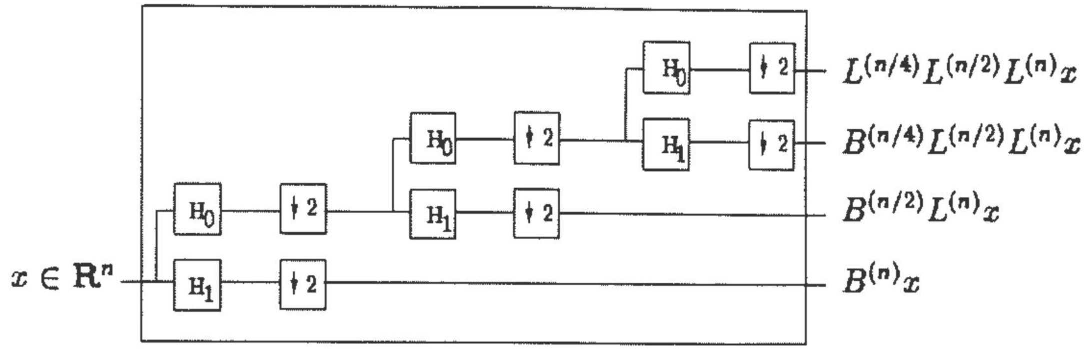
La operación de submuestreo se representa por la matriz:

$$D^{(n)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \in \mathbf{R}^{\frac{n}{2}\times n}$$

De manera similar el efecto combinado del filtro pasaaltas y submuestreo es la matriz de la ecuación 2.1:

$$B^{(n)} = D^{(n)} H_1^{(n)} = \frac{1}{2}\begin{bmatrix} 1 & -1 & & & & & & \\ & & 1 & -1 & & & & \\ & & & & \cdots & & & \\ & & & & & 1 & -1 & \\ & & & & & & 1 & -1 \end{bmatrix} \in \mathbf{R}^{\frac{n}{2}\times n}. \qquad (2.1)$$

La salida de la rama superior del banco de filtros es una versión burda de la señal de entrada. Podemos construir una representación jerárquica de la señal filtrando recursivamente la salida pasabajas del banco de filtro. Este proceso se ilustra en la figura 2.3.




$$y = \begin{bmatrix} L^{(n/4)}L^{(n/2)}L^{(n)} \\ B^{(n/4)}L^{(n/2)}L^{(n)} \\ B^{(n/2)}L^{(n)} \\ B^{(n)} \end{bmatrix} x = H_{sys}x \in \mathbf{R}^n$$

Figura 2.3.Banco de filtros pasabajas recursivos.

En cada etapa de la recursión, la tasa de la señal disminuyó en un factor de 2. Si la señal es discreta y finita (y con una longitud potencia de 2), entonces eventualmente se alcanzará una señal con una muestra. En el caso del banco de filtros de síntesis Haar, esta muestra será el promedio de los elementos de la imagen original. La suma de las magnitudes de todas las salidas pasaaltas del último procedimiento y la salida pasabajas es igual al tamaño de la entrada original. La representación jerárquica final de una señal de entrada es una colección de detalles de la señal a varios niveles de resolución (escalas) y una versión burda de la señal original (la cual será la salida del filtro pasabajas final). El proceso completo puede ser representado con una sola matriz de transformación $H_{sys}$ que uno puede pensar como en reescribir la entrada $x$ en términos de otra base para producir la salida $y = H_{sys}\, x$.

Cuando la entrada al sistema es una imagen (una señal en 2 dimensiones), la primera aplicación del banco de filtros produce una versión borrosa de la imagen (la salida pasabajas) y los detalles de la imagen original (los bordes pronunciados) que no son contenidos en la versión borrosa (la salida del filtro pasaaltas). La segunda aplicación del banco de filtros toma la versión borrosa y la empaña aun más. Las diferencias entre la segunda imagen borrosa son capturadas en la salida del segundo filtro pasabajas. Este proceso recursivo transforma una imagen en una colección de imágenes que capturan los detalles de la imagen a diferentes escalas y una imagen final burda.

La versatilidad y poder de la Transformada Discreta Wavelet (TDW) puede ser incrementada significativamente usando la forma generalizada, en Análisis de Paquetes Wavelets (APW). A diferencia de la TDW que solo descompone los componentes de baja frecuencia (aproximaciones), el APW utiliza tanto los componentes bajos de frecuencia (aproximaciones), como los de alta frecuencia (detalles) y permite descomponer la señal en cualquiera de 2N diferentes esquemas de codificación de señal. Escoger el sistema óptimo para la señal particular se logra usualmente determinando la mejor base para el árbol, normalmente mediante el criterio basado en la entropía.

## 2.3 Base Wavelet Haar.

La descomposición jerárquica vía banco de filtros escribe una señal en términos de una nueva base. Para describir esta nueva base debemos analizar las ecuaciones de dilación:

$$\phi(t) = 2\sum_k h_0(k)\phi(2t-k)$$

y de wavelet :

$$w(t) = 2\sum_k h_1(k)\phi(2t-k)$$

La función $\phi(t)$ es llamada función escaladora, y la función w(t) es llamada función wavelet. La ecuación de dilación y la ecuación wavelet deben ser válidas para todo valor

$$\phi(2^{j-1}t) = 2\sum_k h_0(k)\phi(2^j t-k) \qquad (2.2)$$

de t. Reemplazando t por $2^{j-1}$t obtenemos:

$$w(2^{j-1}t) = 2\sum_k h_1(k)\phi(2^j t-k) \qquad (2.3)$$

Este último sistema de ecuaciones es más conveniente para describir la construcción de la base wavelet correspondiente al banco de filtros.

Ahora ilustramos la construcción de la base wavelet por medio de las ecuaciones de dilación usando el banco de filtros Haar. Recordemos que el filtro pasabajas Haar $H_0$ esta definido por $h_0(0)=h_0(1)=1/2$ (todos los demás coeficientes son cero). Sustituyendo el filtro pasabajas Haar en la ecuación de dilación obtenemos:

$$\phi(t) = \phi(2t) + \phi(2t-1).$$

La solución a esta recurrencia es la función escaladora Haar

$$\phi(t) = \begin{cases} 1 & \text{si } t \in [0,1) \\ 0 & \text{de lo contrario} \end{cases}$$

Las funciones $\phi(t)$, $\phi(2t)$, y $\phi(2t-1)$ se muestran en la figura 2.4.

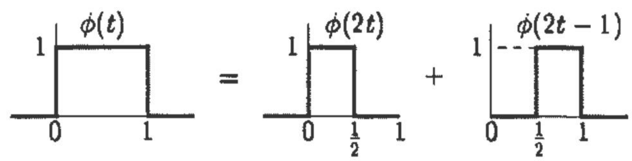



Figura 2.4 Función de dilación.

El filtro pasabajas $H_1$ esta definido por $h_1(0)=1/2$ y $h_1(1)=1/2$. Sustituyendo en la ecuación wavelet, obtenemos:

$$w(t) = \phi(2t) - \phi(2t-1).$$

Podemos concluir que la función wavelet Haar es

$$w(t) = \begin{cases} 1 \text{ si } t \in [0,1/2) \\ -1 \text{ si } t \in [1/2,1) \\ 0 \text{ de lo contrario} \end{cases} .$$

La función wavelet w(t) se muestra en la figura 2.5.

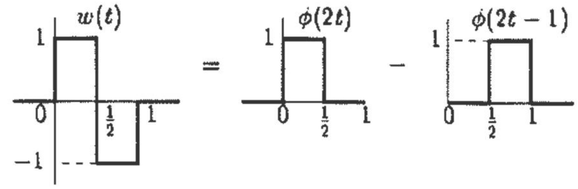



Figura 2.5 Función Wavelet.

La función escaladora $\phi(t)$ es la analogía continua del filtro pasabajas $H_0$. Aplicando $\phi(t)$ a f(t) obtenemos:

$$<\phi, f> = \int_{-\infty}^{\infty} \phi(t) f(t)\, dt = \int_{0}^{1} f(t)\, dt,$$

## 2.4 Procesos de promediación y diferenciación.

El proceso de filtrado pasabajas es implementado por promediación y el de filtrado pasaaltas por diferenciación. Estos procesos son reversibles: podemos manejar los renglones en la tabla para tener los renglones anteriores, por medio de las adiciones y sustracciones apropiadas. En otras palabras, no se ha perdido nada al transformar la cadena. Claramente, este proceso puede ser generalizado para cadenas de cualquier longitud: cadenas de tamaño $2^k$ requieren k rondas de promediación y diferenciación, y cualquier cadena puede ser agregada con ceros al final, hasta que la longitud sea igual a una potencia de 2.

El propósito de la transformada wavelet es que regiones de pequeñas variaciones en los datos originales se manifiestan como elementos pequeños o ceros en la versión transformada wavelet. Una matriz con una alta proporción de entradas cero se denomina dispersa. Para muchas matrices, su versión correspondiente a la transformada wavelet es más dispersa que la original. Las matrices más dispersas son más fáciles de almacenar y transmitir que matrices ordinarias del mismo tamaño. Debido a que matrices ralas pueden ser especificadas en un archivo de datos solo en términos de las locaciones y valores de sus entradas no-cero.

El proceso de promediación y diferenciación que es el corazón de la transformada wavelet, envuelve el proceso repetido de ciertos pares (a,b) para lograr pares ((a+b)/2, (a-b)/2). En la transformada wavelet normalizada, se procesa cada par (a,b) para lograr ((a+b)/√2,(a-b)/√2).

# III. Aplicaciones de la Transformada Wavelet.

## 3.1 Gráficas en computadora.

En visión de computadoras, las técnicas de multirresolución han sido usadas en varios problemas, desde la estimación de movimiento hasta reconocimiento de objetos. Las imágenes son aproximadas sucesivamente empezando por una versión burda yendo hacia una versión de resolución fina. En gráficas por computadora, un método llamado refinamiento sucesivo interpola iterativamente curvas o superficies, y el estudio de tales interpoladores está relacionado con la construcción de wavelets por medio de bancos de filtros.

Finalmente los procedimientos computacionales usan el concepto de aproximaciones sucesivas, algunas veces alternando entre resoluciones finas y burdas. Los métodos multicuadrícula usados para la solución de ecuaciones diferenciales parciales son un ejemplo.

## 3.2 Procesamiento Digital de Señales.

Existen algunas diferencias evidentes entre una señal senoidal y las wavelets. En comparación con la señal senoidal que es suave e infinita en duración, la wavelet es irregular en forma y soportada compactamente, es decir tiene un valor igual a cero fuera de cierto intervalo finito. Son estas propiedades las que convierten a las wavelets en herramientas ideales para el análisis de señales de naturaleza no estacionaria. Su forma irregular las lleva a analizar señales con discontinuidades o cambios bruscos, mientras que su soporte compacto permite la localización temporal de los rasgos de una señal.

### 3.3 Transmisión progresiva de imágenes

En el caso de obtención de imágenes, por ejemplo de Internet, la técnica de compresión que se ha discutido permite un tipo de transmisión progresiva de imágenes. Cuando una imagen P es solicitada electrónicamente, una versión T codificada en wavelets se obtiene del almacenamiento, y bits de información sobre ella se envían sobre el canal, empezando con el promedio general y los coeficientes de detalle más grandes, y siguiendo así hasta los coeficientes de detalle más pequeños. Esta información al ser recibida por el usuario, es usada para desplegar una reconstrucción de P, empezando con una aproximación muy cruda de la imagen que, rápidamente actualizada y refinada, se aprecia notablemente mejor mientras se usen más coeficientes wavelet.

Eventualmente (asumiendo que el usuario ha juzgado que vale la pena esperar por la imagen) todos los coeficientes habrán sido transmitidos y una copia perfecta de P desplegada. Si el usuario pierde interés o paciencia en el transcurso, puede detener el proceso.

El navegador Netscape, que muchas personas utilizan para accesar la Web, actualmente usa una forma de compresión de imágenes JPEG para transmisión progresiva. Programas basados en wavelets, usando una generalización de la transformada wavelet Haar se encuentra disponibles, así como un aditamento (plugin) para Netscape que tiene mejores resultados que el basado en JPEG.

## 3.4 Compresión de señales.

Compresión es la ciencia y arte de remover información de una imagen original. La compresión puede ser con pérdidas o sin pérdidas. La compresión es sin pérdidas cuando la cantidad de información a ser removida puede ser recuperada exactamente donde fue removida, y es con perdidas cuando los datos removidos no pueden ser recuperados durante el proceso de descompresión.

### 3.4.1 Compresión de imágenes.

La compresión con pérdidas puede impactar la calidad visual de la imagen. La tasa de compresión, el grado de impacto de la compresión con pérdidas en una imagen, puede variar desde mínima hasta extrema dependiendo de la cantidad de información removida. Para la compresión sin pérdidas, la tasa de compresión puede ser muy baja, entre 2:1 y 3:1. Para una imagen desplegada en una pantalla de computadora de tamaño 640X480 pixeles solamente, la cantidad de datos es 1 millón de bytes. Con compresión sin pérdidas, la imagen resultante requiere de 1/3 a ½ de tal cantidad de datos. La compresión con pérdidas puede variar desde un valor tan bajo como la de compresión sin pérdidas hasta varios de cientos a uno.

El verdadero beneficio en la transmisión wavelet no es tanto la expectación de esparcimiento en las matrices transformadas, es el hecho de que podemos jugar con las versiones "mas detalladas" para crear muchas entradas cero, podemos alterar las matrices transformadas, tomando ventaja de las "regiones de baja actividad", entonces aplicar la transformada inversa wavelet a esta versión, para obtener una aproximación de los datos originales.

Así llegamos a la puerta de la compresión wavelet: Fijar un valor de umbral $\varepsilon$, y decretar que cualquier coeficiente de detalle en la versión transformada wavelet cuya magnitud es menor o igual a $\varepsilon$ será convertido en cero (con fortuna, esto nos llevará a una matriz relativamente esparcida), luego reconstruir una aproximación de los datos originales usando esta versión transformada wavelet de los datos. Este proceso se llama compresión sin pérdidas cuando $\varepsilon$ es igual a cero, de otra manera se llama compresión con pérdidas.

En general, para una matriz particular P y un umbral $\varepsilon$, la tasa de compresión se entiende como la razón del número de valores no-cero en la matriz transformada T al número de entradas en la matriz reducida D.

Los codificadores/decodificadores de compresión están basados en una de cuatro técnicas para lograr compresión con pérdidas: fractales, cuantización de vectores, la transformada discreta de coseno (TDC) y wavelets. Estas cuatro técnicas tienen características comunes, y aun nivel burdo, realizan similares funciones elementales. Cada una transforma la imagen en una nueva representación (el dominio de la trasformada), manipula los valores en este dominio eliminando valores que representan contribuciones pequeñas o finas dentro de la imagen original y suavizando el resto de los valores para hacerlos más homogéneos. Los datos perdidos impactan la calidad resultante cuando la imagen es descomprimida. Ya que cada técnica de compresión describe su propio dominio de la transformada, la eliminación de valores y el proceso de suavizamiento impactará la imagen resultante en diferentes maneras, produciendo una deformidad característica de la transformada usada para comprimir la imagen.

Al incrementarse la tasa de compresión, esta deformidad característica se hará más pronunciada hasta que eventualmente todos los rasgos reconocibles sean perdidos. Ya que cada técnica realiza la supresión de datos en la imagen en diferentes maneras, cada una de las técnicas de compresión tiene sus ventajas y desventajas, y cada una produce un resultado visual diferente, si el nivel de compresión es suficientemente alto para los que los observadores aprecian la deformidad.

Los fractales y la cuantización de vectores requieren significantes recursos computacionales para la compresión, pero por varias sofisticadas razones matemáticas, no son soluciones óptimas. La técnica de compresión TDC divide una imagen en pequeños bloques y después los trata como imágenes individuales. A niveles moderados de compresión, los resultados son imágenes con bloques perceptible que no son aceptables. Una de las principales diferencias entra la transformada wavelet y la transformada discreta del coseno (que es la base para JPEG y MPEG) es que la wavelet es aplicada a través de la imagen completa en vez de los pequeños (8 x 8) segmentos que la componen que es el tamaño efectivo del marco del proceso TDC. La transformada wavelet trata la imagen en forma completa, lo cual permite a los usuarios obtener tasas de compresión más altas sin temor a los desagradables artefactos generados por los algoritmos JPEG o MPEG.

Bajo el liderazgo del Dr. Paul Fisher, la compañía INFINOP pasó su primer año de trabajo (1991) bajo el patrocinio de la Fuerza Naval de los Estados Unidos, realizando un análisis objetivo de las cuatro técnicas y determinó que las wavelets serían la tecnología ganadora, y sobre estos resultados el estándar nacional que se ha escogido para ser adoptado en el año 2000 será basado en un proceso wavelet.

### 3.4.2 Implementación de la Transformada Wavelet Haar en Matlab

Para implementar la Transformada Wavelet Haar es necesario crear una matriz que realice las operaciones de promediación y diferenciación. Esta matriz deberá ser de dimensiones idénticas a la de la matriz de la imagen.

Matlab provee herramientas para procesamiento de imágenes, en especial una para utilización de wavelets. Estas opciones; sin embargo, se encuentran solo en la versión profesional. La versión utilizada en universidades es la versión para estudiantes que sólo contiene librerías básicas.

Para leer imágenes y convertirlas en una matriz en dos dimensiones, Matlab posee una función que acepta imágenes en diferentes formatos, todas las imágenes utilizadas para este proyecto son JPEG, 256X256 pixeles a excepción de la foto del macaco que es de 512X512.

A modo de ilustración y para realizar más rápidamente los procesos, las imágenes son convertidas de color a tonos de grises, así solo se trabaja con una matriz, en lugar de las 3 matrices de los colores primarios (Rojo, Verde y Azul). Este proceso se muestra a continuación en código de Matlab, la función se llama "getimage.m" y lee imágenes en varios formatos:

```
imagefile=input('Nombre del archivo de la imagen: ');
a=imread(imagefile);
a1=a(:,:,1);
imagesize=size(a);
n=imagesize(1);
```

A continuación se muestra "transform.m" función que fabrica una matriz Wavelet del mismo tamaño que la imagen (n). Esta matriz es para el tipo Wavelet normalizada (con factor raíz cuadrada de dos) que ofrece mejores resultados que la Wavelet normal mostrada en la ecuación 2.1.

```
cum=0;
k=log2(n);
a=ones(n,n,k);
for d=1:k
   m=zeros(n,n);
   for c=1:n/(2^d)
      m(2*c+n*(c-1)-1)=1;
      m(2*c+n*(c-1))=1;
      m(2*c+n*(c-1+n/(2^d))-1)=1;
      m(2*c+n*(c-1+n/(2^d)))=-1;
   end
   m=m/sqrt(2);
   for c=1:cum
      m(n/(2^(d-1))*n+n/(2^(d-1))+c+(c-1)*n)=1;
   end
   cum=cum+n/2^d;
   a(:,:,d)=m;
end
```

El siguiente paso es transformar la imagen utilizando esta matriz wavelet. Para esto se transpone la matriz wavelet se multiplica por la matriz de la imagen y el resultado se multiplica ahora por la matriz wavelet directa.

```
w=ones(n,n);
w=a(:,:,1);
for d=2:k
   w=w*a(:,:,d);
end
t=w'*double(a1)*w;
```

La matriz resultante es la versión transformada Wavelet de la imagen, que es una versión que ocuparía menos espacio de almacenamiento y tomaría menos tiempo al ser transmitida. Pero será comprimida más aún por la función "compress.m" que elimina todos los valores de la versión transformada menores a cierto valor delta definido previamente.

```
for i=1:n*n
   if abs(t(i))<=delta
      t(i)=0;
   end
end
```

Podemos calcular el factor de compresión utilizando la función "compfact.m", que divide el número de valores diferentes de cero de la matriz original entre los valores diferentes de cero de la nueva matriz comprimida.

```
nnz1=0;
nnz2=0;
for ind=1:n*n
    if a1(ind) > 0
       nnz1=nnz1+1;
    end
    if t(ind) > 0
       nnz2=nnz2+1;
    end
end
comp_factor=nnz1/nnz2
```

Por último se aplicará la Transformada Inversa Wavelet con la función "invtrans.m" que transformará la matriz comprimida en forma apropiada para despliegue y rearchivamiento.

```
a2=(w^(-1))'*t*w^(-1);
j=zeros(n,n,3);
j(:,:,1)=a2;
j(:,:,2)=a2;
j(:,:,3)=a2;
image(uint8(j));
title('Image Comprimida');
outputfile=input('Nombre del archivo de salida: ');
imwrite(outputfile);
```

La primera imagen utilizada será una de las más clásicas en procesamiento digital de imágenes, es la foto de un macaco.

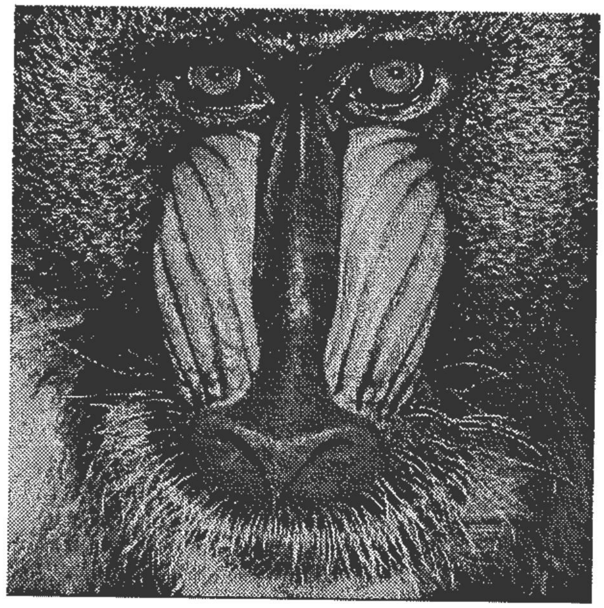


Figura 3.1 Imagen del macaco.

Esta imagen, como se muestra a continuación en su espectro, tiene alto contenido en las frecuencias bajas al tener grandes áreas de color casi constante lo que es muy afortunado para este algoritmo. Podemos notar que la mayoría de los valores se encuentra en la esquina superior izquierda, área de los valores pasabajas.

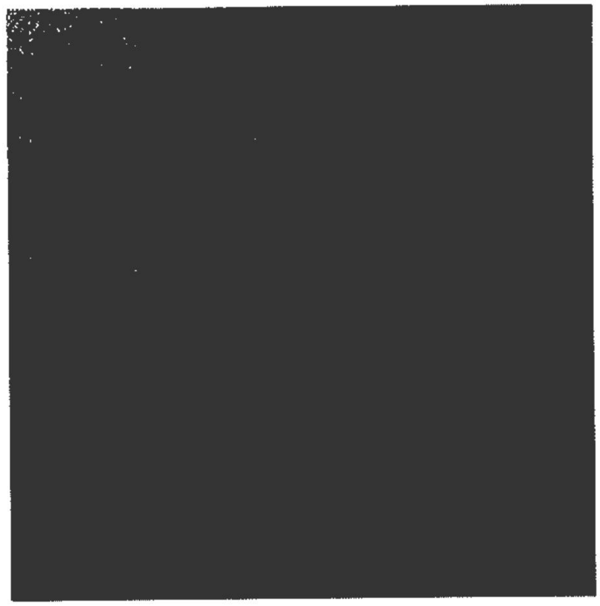


Figura 3.2 Transformada Wavelet de la imagen del macaco.

Se utilizaron valores ε (delta en el código Matlab) de 5, 20 y 100. Los factores de compresión resultante se muestran en la Tabla 3.1.

| ε = 5 | ε = 20 | ε = 100 |
|---|---|---|
| 2.91 | 7.52 | 148.52 |

Tabla 3.1 Factores de compresión para la imagen del macaco.

La imagen resultante con ε = 100 es de bastante buena calidad, considerando el enorme ahorro en coeficientes.

Figura 3.3 Imagen comprimida del macaco.

La siguiente imagen, una foto tomada del planeta Saturno, tiene una alta cantidad de información contenida en las frecuencias más altas (parte inferior derecha en la Transformada Wavelet) y por tanto no ofreció resultados muy positivos, pero podemos apreciar en la imagen de su Transformada Wavelet como, la imagen es descompuesta en valores de frecuencia distintos, conservando además información en el espacio.

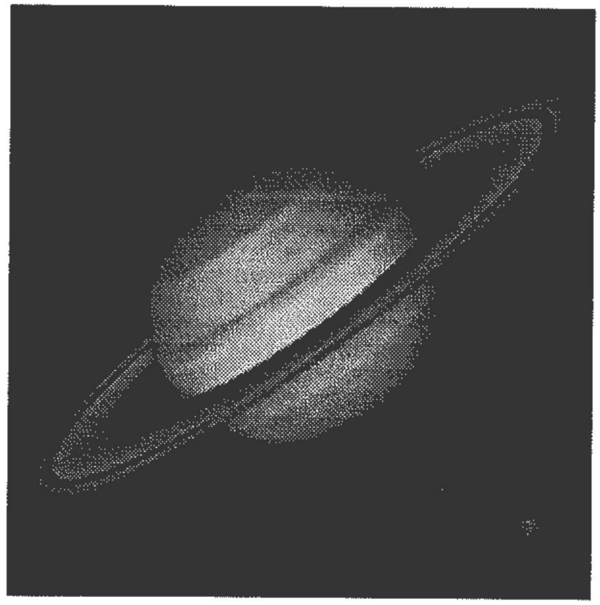


Figura 3.4 Fotografía del planeta Saturno.

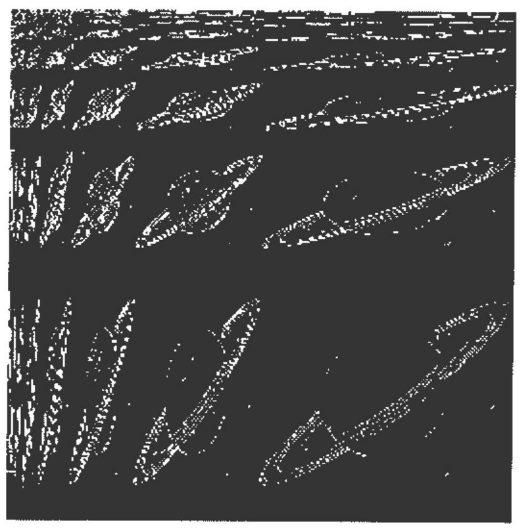


Figura 3.5 Transformada Wavelet de la imagen de Saturno

Comparando los resultados con la imagen del macaco y la del planeta Saturno, los factor de compresión en esta última fueron mayores para los mismos valores de delta utilizados (11.29, con $\varepsilon = 20$), pero los resultados visuales no fueron agradables, pues se apreciaron artefactos y grandes bloque que eliminan detalles importantes de la imagen original, ya que esta contenía mucha información en las altas frecuencias.

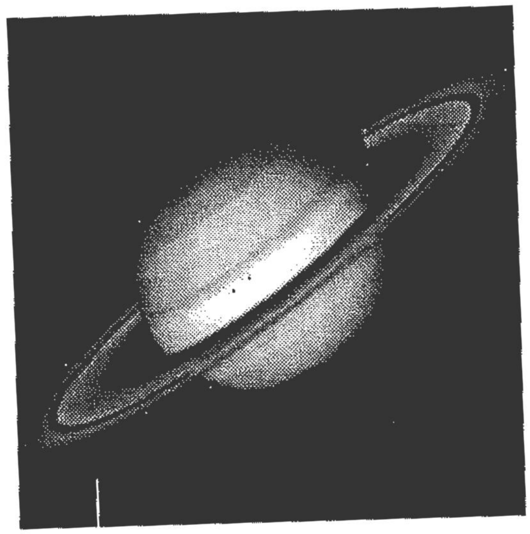


Figura 3.6 Imagen comprimida con $\varepsilon = 20$.

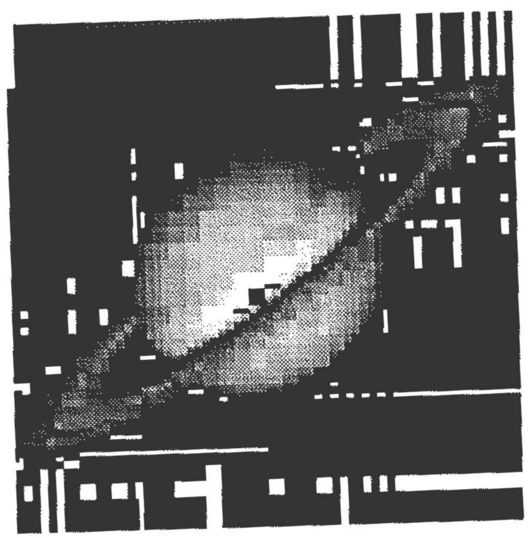


Figura 3.7 Imagen comprimida con $\varepsilon = 100$.

Una de las mejoras futuras a este código podría ser el tener valores diferentes de $\varepsilon$ para las áreas pasabajas/pasaaltas y pasaaltas/pasabandas y otra para la pasaaltas, ya que las imágenes con gran detalle no pudieron ser comprimidas en gran medida sin deteriorar significativamente la imagen.

Otra desventaja es el que solo se pueden utilizar imágenes cuadradas (mismo número de renglones que de columnas) en vez de utilizar la complementación por ceros (ya sea de columnas o de renglones) que sería más apropiada. Además se podría incluir, como se mencionó anteriormente, la capacidad de comprimir imágenes a colores sin tener que convertirlas a tonos de gris.

## IV. Conclusiones.

El análisis de Wavelets resulta una muy útil alternativa al análisis de Fourier, pues además de proporcionar información sobre el espectro de las funciones, muestra la correlación entre el espacio de la señal y las frecuencias que la componen.

Estos algoritmos en vez de ser implementados en un lenguaje de programación se opto por utilizar el programa Matlab que maneja matrices muy eficientemente y escribe y lee imágenes con una sola línea de código.

Una de las wavelets más sencillas (Haar) resultó relativamente fácil de implementar en este programa, otras wavelets o modificaciones de la wavelet Haar pueden implementarse de manera similar tanto en Matlab como en lenguajes de programación o ensamblador para su utilización en circuitería.

## V. Bibliografía.

### 5.1 Libros y artículos científicos.


DIGITAL IMAGE PROCESSING

González y Woods

Addison-Wesley Publishing Company.

IMAGE COMPRESSION USING THE HAAR WAVELET TRAANSFORM.

Colm Mulcahy

Spelman Science and Math Journal

WAVELETS, FILTER BANKS AND MULTIRESOLUTION SIGNAL PROCESSING.

A WAVELET APPROACH FOR LOSSLESS INTEGER COMPRESSION

Hongyang Chao et al

Zhongshan University.

IMAGE COMPRESSION USING THE HAAR WAVELET TRANSFORM

Peggy Morton y Arne Petersen

College of the Redwoods.

WAVELETS: WHAT'S NEXT?

Win Sweldens

## 5.2 Páginas de Internet


http://www.wavelet.org

http://paos.colorado.edu/research/wavelets/wavelet1.html

http://www.monash.edu.au/cmcm/wavelet/wbasic.html

http://www.infinop.com

http://robotics.stanford.edu/~scohen/lect02/lect02.html

http://www.amara.com/current/wavelet.html

# VI. Apéndice

## 6.1. Guía para utilizar los códigos en Matlab

1. Iniciar Matlab.
2. Cambiar el directorio para tener acceso el disco flexible.

```
cd a:\
```

3. Verificar que los archivos correspondan con los de las Tablas 6.1 y 6.2.

```
ls
```

4. Leer una imagen del subdirectorio y convertirla a una matriz (a).

```
Getimage
```

Al desplegar "Nombre del archivo de la imagen:" escribir el nombre de uno de los archivos de la Tabla 6.1 entre apóstrofes ('), el programa es sensible a mayúsculas.

Si se desea desplegar esta imagen tecléese:

```
image(a)
```

5. Crear la matriz de transformación wavelet:

```
Matrix
```

6. Realizar la transformación wavelet de la imagen:

```
Transform
```

7. Para comprimir la imagen se debe definir un valor adecuado de ε (delta en el programa) se pueden tratar valores a prueba y error (20 por ejemplo). Al final se debe dar un nombre a la imagen comprimida:

```
delta = 20;
compress
```

Para calcular y desplegar el factor de compression, tecléese:

```
compfact
```

Para desplegar la imagen de la transformada wavelet o grabarla en un archivo, tecléese respectivamente:

```
displaytr
imwrite ('Nombre nuevo de la imagen.jpg');
```

8. Para tratar con distintos índices de compresión, debe asignarse otros valores a delta antes de comprimir la imagen (paso 7). Si se desea leer otra imagen o el valor de delta en es menor al anterior, se debe recalcular la matriz de la imagen (paso 4), ya que algunos valores son eliminados.

## 6.2. Contenido del disco anexo.

| Nombre | Descripción | Tamaño | Colores |
|---|---|---|---|
| arizona.jpg | Desierto | 512X512 | 256 colores |
| baboon.jpg | Macaco | 512X512 | 256 colores |
| bab-100.jpg | Imagen anterior comprimida | 512X512 | |
| Birdgirl.jpg | Niña con pájaros en un parque | 128X128 | 256 tonos de gris |
| Bullfigt.jpg | Ruedo de toros | 128X128 | 256 colores |
| obsunam.jpg | Observatorio de San Pedro Mártir (UNAM) | 256X256 | 256 tonos de gris |
| Painter.jpg | Pintor | 128X128 | 256 tonos de gris |
| saturn.jpg | Planeta Saturno | 256X256 | 256 colores |
| telescop.jpg | Imagen tomada del telescopio de la UNAM | 256X256 | 256 tonos de gris |
| Trainsta.jpg | Estación de tren | 128X128 | 256 tonos de gris |

Tabla 6.1 Imágenes contenidas en el disco.

| Nombre | Descripción |
|---|---|
| getimage.m | Lee imagen tipo JPEG, BMP, PCX, TIFF y la convierte en una matriz. |
| matrix.m | Crea una matriz de transformación del tamaño de la imagen. |
| transform.m | Realiza la Transformada Wavelet multiplicando la imagen con la matriz de transformación. |
| displtr.m | Despliega la imagen Transformada Wavelet. |
| compress.m | Comprime la imagen eliminando elementos menores que cierto valor delta. |
| compfact.m | Calcula el factor de compresión. |
| invtrans.m | Convierte la imagen en una forma apropiada para su despliegue. |
| display.m | Despliega imagen comprimida. |
| wavelet.m | Todas las funciones anteriores. |

Tabla 6.2 Código Matlab contenido en el disco.