

Universidad EAFIT
Departamento de Ciencias Básicas

Compresión de imágenes usando wavelets

Gloria Puetamán Guerrero

Hernán Salazar Escobar

Trabajo de investigación presentado como requisito parcial para optar el
título de Magíster en Matemáticas Aplicadas

Director

Jairo Villegas Gutiérrez

Departamento de Ciencias Básicas

Universidad EAFIT

Medellín

Agradecimientos

El desarrollo de este trabajo fue posible gracias a la constante dedicación e interés que el profesor Jairo Villegas Gutiérrez mostró durante el proceso de construcción y elaboración del mismo; las discusiones sostenidas durante las reuniones y sus consejos fueron de vital importancia para comprender muchos tópicos inherentes a la teoría wavelet y que en su momento fueron oscuros y azarosos para los autores.

De igual forma, queremos agradecer a las Universidades EAFIT y San Buenaventura, Seccional Medellín, por el apoyo en cuanto a los recursos bibliográficos, el espacio físico, y lo más importante, por la ayuda financiera que la Universidad EAFIT nos brindó para llevar a feliz término nuestra maestría. Al Doctor Carlos Mario García Ramírez, director de investigaciones de la USB, por la colaboración y el interés para llevar a cabo el proyecto Compresión de Imágenes Vía Wavelets.

Índice general

Introducción	1
1. Preliminares	3
1.1. Terminología	3
1.2. Transformada de Fourier	5
1.2.1. Serie de Fourier	6
1.2.2. Teorema de muestreo de Shannon	7
1.2.3. Filtrado	8
1.3. Espacio de probabilidad	10
1.3.1. Variable aleatoria	11
1.3.2. Procesos estocásticos	12
2. Introducción a las wavelets	15
2.1. Introducción	15
2.2. Transformadas wavelets	16
2.2.1. Transformada wavelet continua	16
2.2.2. Transformada wavelet discreta	20
2.3. Análisis Multirresolución	22
2.4. Ecuación de escala	27
2.5. Construcción de la función de escala	29
2.6. Descomposición y reconstrucción	32
2.6.1. Algoritmo de descomposición	32
2.6.2. Algoritmo de reconstrucción	35

3. Compresión de imágenes usando wavelets	37
3.1. El problema de la compresión de imágenes	38
3.1.1. Formulación matemática	39
3.1.2. Compresión lineal y no lineal	41
3.2. Introducción a la teoría de códigos	47
3.2.1. Elementos de la teoría de codificación	48
3.2.2. Matriz generadora y de verificación de paridad	52
3.2.3. Códigos Huffman	55
3.3. Algoritmo de compresión	58
3.3.1. Discretización de una imagen	58
3.3.2. Algoritmo para la compresión	63
4. Manual del usuario y anexos	69
4.1. Instalación e Invocación	70
4.2. Estructura de la interface gráfica	71
4.3. Manejo básico de la herramienta	76
4.3.1. Abrir Imagen	76
4.3.2. Cambiar el mapa de colores	76
4.3.3. Realizar una compresión básica	77
4.3.4. Realizar compresión	77
4.3.5. Zoom sobre las imágenes	78
4.4. Ejemplos de uso	79
4.4.1. Código de barras	80
4.4.2. Huella digital	81
4.4.3. Imagen del río	81
4.5. Código Fuente	82
Bibliografía	89

Introducción

Las wavelets¹ y el análisis de multirresolución constituyen una potente herramienta para afrontar problemas fundamentales en el tratamiento de señales. Entre ellos se encuentran la reducción del ruido, la compresión de señales (de mucha importancia tanto en la transmisión de grandes cantidades de datos como en su almacenamiento) o la detección de determinados patrones o irregularidades locales en ciertos tipos de señales (electrocardiogramas, huellas digitales, vibraciones de motores, defectos de soldadura entre placas de acero, entre otras) (ver, p.e., [1], [7], [9], [11], [12], [18], [20], [23], [24], [30], [42], [47]). Esta moderna teoría ha experimentado un gran desarrollo en las dos últimas décadas mostrándose muy eficiente donde otras técnicas, como por ejemplo, la transformada rápida de Fourier no resultaban satisfactorias.

Algunos de los principales problemas que afectan el tratamiento de señales digitales es la compresión de datos para su posterior almacenamiento o transmisión, la eliminación del ruido y detección de ciertos fenómenos locales, ha permitido el desarrollo tanto teórico como computacional de este campo del análisis armónico. Otra de las principales virtudes de las wavelets es que permiten modelar mejor procesos que dependen fuertemente del tiempo y para los cuales su comportamiento no tiene porqué ser suave. Una de las principales ventajas de las wavelets frente a los métodos clásicos, como la transformada de Fourier, es que en el segundo caso se maneja una base de funciones bien localizada en frecuencia pero no en tiempo, mientras que la mayoría de las wavelets interesantes presentan una buena localización en

¹Se utilizará este término en lugar de ondícula, palabra también usada por algunos autores.

2 Introducción

tiempo y en frecuencia, disponiendo incluso de bases de wavelets con soporte compacto.

Las wavelets proporcionan un conjunto de herramientas flexibles para detectar problemas prácticos en ciencia e ingeniería. Entre estas herramientas se tienen la transformada wavelet que está asociada con el Análisis Multirresolución de una señal, es decir, a distintos niveles de resolución se tendrá una base de wavelets. Concretamente, cuando mayor detalle se pretenda obtener en una señal (mayor resolución), mayor número de funciones por unidad de longitud se tendrán en la base de wavelets. Además, no existe una transformada wavelet única, ni que resuelva todos los problemas, a partir de la modelación del proceso y de un análisis a priori del tipo de señal tratada y del objetivo que se pretenda (compresión, eliminación del ruido u otro) se busca la familia de wavelets (Haar, Daubechies, Coiflets,...) que mejor coincida con las características de la señal a estudiar (ver, p.e., [7], [8], [15], [27], [30], [33], [34], [42], [47]).

El propósito de este trabajo es mostrar de manera coherente y práctica la teoría matemática de la compresión de imágenes usando wavelets. En ningún momento se pretende que el material expuesto en esta monografía sea completamente original, nuestro aporte está en presentar algunos desarrollos matemáticos, que por lo regular, en los artículos estudiados no están.

El trabajo comienza con la descripción de la terminología necesaria para abordar las wavelets, partiendo de los resultados básicos del análisis de Fourier; se discuten algunos conceptos de la teoría de probabilidad. El capítulo 2 es una introducción a la teoría wavelet, prestando especial interés a las transformadas wavelets tanto discretas como continuas y terminando con un corto estudio del análisis multirresolución; puntos claves en la construcción de bases wavelets y en análisis de señales. El capítulo 3 es el estudio de algunos conceptos básicos en el procesamiento de señales. Acá se presenta la teoría matemática de la compresión de imágenes, también se muestra de manera muy breve la teoría algebraica de codificación y se termina con el problema de la compresión de imágenes, mirada como un proceso de compresión de datos, en donde los datos comprimidos se representan mediante señales digitales. Finalmente, a manera de ejemplo de aplicación utilizando Matlab 7.0, el capítulo 4 es el manual de usuario para codificar y comprimir una imagen usando transformada wavelet.

CAPÍTULO 1

Preliminares

1.1. Terminología

En este corto capítulo se presentará alguna terminología necesaria para la lectura de esta monografía. En particular, se hará un resumen de resultados básicos de análisis de Fourier omitiendo sus pruebas, las cuales se pueden encontrar en algunos de los siguientes textos [4], [21], [39], [57], [61]. También se tratará algo sobre teoría de probabilidad y series de tiempo, ver por ejemplo, [3], [10], [13], [25], [38], [45], [62].

Recuerde que $L_1(\mathbb{R})$ es el espacio de todas las funciones $f : \mathbb{R} \rightarrow \mathbb{C}$, tal que $\int_{\mathbb{R}} |f(t)| dt = \|f\|_{L_1} < \infty$. De igual forma se tiene $L_2(\mathbb{R})$, el espacio de las funciones cuadrado-integrables, cuya norma es

$$\|f\|_{L_2} = \left(\int_{\mathbb{R}} |f(t)|^2 dt \right)^{1/2} < \infty.$$

Este espacio se dota con el producto escalar

$$\langle f, g \rangle_{L_2} = \int_{\mathbb{R}} f(t) \overline{g(t)} dt,$$

donde $\overline{g(t)}$ denota el conjugado complejo de $g(t)$. Con este producto interno el espacio $L_2(\mathbb{R})$ es de Hilbert. Las funciones $f, g \in L_2(\mathbb{R})$ son ortogonales si $\langle f, g \rangle_{L_2} = 0$. En general, $L_p(\mathbb{R})$ ($p \geq 1$), es el espacio de todas las funciones (clases de equivalencia) $f : \mathbb{R} \rightarrow \mathbb{C}$, tal que $\int_{\mathbb{R}} |f(t)|^p dt = \|f\|_{L_p}^p < \infty$, o de

4 Preliminares

manera equivalente

$$\|f\|_{L_p} = \left(\int_{\mathbb{R}} |f(t)|^p dt \right)^{1/p}$$

es la norma de f en $L_p(\mathbb{R})$. Otro espacio que se utilizará es $\ell_2(\mathbb{Z})$, el de las sucesiones (x_j) , $j \in \mathbb{Z}$, tal que $\sum_j |x_j|^2 < \infty$.

Sea $F = \mathbb{C}$ o \mathbb{R} , X y Y espacios normados (espacios vectoriales equipados con una norma). Un operador lineal es una función $T : X \rightarrow Y$ tal que $T(au + bv) = aT(u) + bT(v)$, para cada $a, b \in F$ y cada $u, v \in X$. El operador T es continuo en u_0 si para cada $\epsilon > 0$ existe $\delta > 0$ tal que si

$$\|u - u_0\|_X < \delta \quad \text{entonces} \quad \|Tu - Tu_0\|_Y < \epsilon. \quad (1.1.1)$$

Si (1.1.1) se cumple para cada $u_0 \in X$ se dice que T es continuo en X . Si δ no depende del punto u_0 se dice que T es uniformemente continuo en X .

El operador T es acotado si y sólo si existe una constante $c > 0$ tal que $\|Tu\|_Y \leq c\|u\|_X$ para cada $u \in X$.

Si $f, g \in L_1(\mathbb{R})$, entonces la convolución de f y g , denotada $f * g$, se define por

$$(f * g)(t) = \int_{\mathbb{R}} f(t - z)g(z)dz.$$

Un sistema de funciones $\{\varphi_j, j \in \mathbb{Z}\}$, $\varphi_j \in L_2(\mathbb{R})$, se llama ortonormal si

$$\int_{\mathbb{R}} \varphi_j(t) \overline{\varphi_k(t)} dt = \delta_{jk},$$

donde δ_{jk} es la delta de Kronecker. Es decir,

$$\delta_{jk} = \begin{cases} 1, & \text{si } j = k; \\ 0, & \text{si } j \neq k. \end{cases}$$

Un sistema ortonormal se llama una base en un subespacio V de $L_2(\mathbb{R})$ si cualquier función $f \in V$ tiene una representación de la forma

$$f(t) = \sum_j c_j \varphi_j(t),$$

donde los coeficientes c_j satisfacen $\sum_j |c_j|^2 < \infty$. En lo que sigue se utilizará la notación $\sum_j = \sum_{j=-\infty}^{\infty}$, $\int_{\mathbb{R}} = \int_{-\infty}^{\infty}$, $\|f\|_{L_2} = \|f\|_2$ y \langle, \rangle_2 .

La función característica del conjunto A , χ_A , se define por

$$\chi_A(t) = \begin{cases} 1, & t \in A; \\ 0, & t \notin A. \end{cases}$$

También se utilizará la notación $I\{A\}$ para denotar esta función y la llaman función indicadora.

El soporte de una función $f : A \subseteq \mathbb{R} \rightarrow \mathbb{C}$, denotado $\text{Sop}f$, se define por $\text{Sop}f = \{x \in A : f(x) \neq 0\}$.

1.2. Transformada de Fourier

En esta sección se recordará la definición y algunas propiedades importantes de la transformada de Fourier.

Definición 1.2.1. Sea $f \in L_1(\mathbb{R})$ y $\omega \in \mathbb{R}$. La transformada de Fourier de f en ω se define por

$$\hat{f}(\omega) := \int_{\mathbb{R}} f(t)e^{-i\omega t} dt.$$

Como

$$\int_{\mathbb{R}} |f(t)| |e^{-it\omega}| dt = \int_{\mathbb{R}} |f(t)| dt = \|f\|_{L_1} < \infty$$

se tiene que la transformada de Fourier está bien definida. La aplicación $f \mapsto \hat{f}$ se llama transformación de Fourier y se denota por \mathcal{F} ($\mathcal{F}(f) = \hat{f}$). La función \hat{f} es continua y tiende a cero cuando $|\omega| \rightarrow \infty$ (Lema de Riemann-Lebesgue). Es claro que $\mathcal{F}(af + bg) = a\mathcal{F}(f) + b\mathcal{F}(g)$, para cada $a, b \in \mathbb{R}$.

En general \hat{f} no es una función integrable, por ejemplo, sea

$$f(t) = \begin{cases} 1, & |t| < 1; \\ 0, & |t| > 1. \end{cases}$$

Entonces

$$\begin{aligned} \hat{f}(\omega) &= \int_{-1}^1 e^{-it\omega} dt = \left[\frac{e^{-i\omega} - e^{i\omega}}{-i\omega} \right] \\ &= \frac{\text{sen } \omega}{\omega} \notin L_1(\mathbb{R}). \end{aligned}$$

Si $\hat{f}(\omega)$ es integrable, entonces existe una versión continua de f y se puede obtener la fórmula de inversión de Fourier

$$f(t) = \mathcal{F}^{-1}(\hat{f}(\omega)) = \frac{1}{2\pi} \int_{\mathbb{R}} \hat{f}(\omega) e^{i\omega t} d\omega. \quad (1.2.1)$$

La siguiente proposición recoge algunas propiedades fundamentales de la transformada de Fourier.

Proposición 1.2.2. Sean $f, g \in L_1(\mathbb{R})$, entonces

1. $\widehat{(T_x f)}(\omega) = e^{-i\omega x} \hat{f}(\omega)$, donde $(T_a f)(t) = f(t - a)$.
2. $(T_x \hat{f})(\omega) = \widehat{(e^{ix(\cdot)} f)}(\omega)$
3. $\widehat{f * g} = \hat{f} \hat{g}$

6 Preliminares

4. Si $\epsilon > 0$ y $g_\epsilon(t) = g(\epsilon t)$ entonces $\hat{g}_\epsilon(\omega) = \epsilon^{-1}\hat{g}(\omega/\epsilon)$.

Otro resultado útil es el siguiente: Si $f, g \in L_1(\mathbb{R}) \cap L_2(\mathbb{R})$, entonces

$$\|f\|_2^2 = \frac{1}{2\pi} \int_{\mathbb{R}} |\hat{f}(\omega)|^2 d\omega \quad (\text{fórmula de Plancherel}) \quad (1.2.2)$$

$$\langle f, g \rangle_2 = \frac{1}{2\pi} \int_{\mathbb{R}} \hat{f}(\omega) \overline{\hat{g}(\omega)} d\omega \quad (\text{fórmula de Parseval}). \quad (1.2.3)$$

Por extensión, la transformada de Fourier se puede definir para cualquier $f \in L_2(\mathbb{R})$. En virtud a que el espacio $L_1(\mathbb{R}) \cap L_2(\mathbb{R})$ es denso en $L_2(\mathbb{R})$. Luego, por isometría (excepto por el factor $1/2\pi$) se define \hat{f} para cualquier $f \in L_2(\mathbb{R})$, y las fórmulas (1.2.2) y (1.2.3) permanecen válidas para todo $f, g \in L_2(\mathbb{R})$.

En teoría de señales, la cantidad $\|f\|_2$ mide la energía de la señal, mientras que $\|\hat{f}\|_2$ representa el espectro de potencia de f .

Si f es tal que $\int_{\mathbb{R}} |t|^k |f(t)| dt < \infty$, para algún entero $k \geq 1$, entonces

$$\frac{d^k}{d\omega^k} \hat{f}(\omega) = \int_{\mathbb{R}} (-it)^k e^{-i\omega t} f(t) dt. \quad (1.2.4)$$

Recíprocamente, si $\int_{\mathbb{R}} |\omega|^k |\hat{f}(\omega)| d\omega < \infty$, entonces

$$(i\omega)^k \hat{f}(\omega) = \mathcal{F}(f^{(k)})(\omega). \quad (1.2.5)$$

1.2.1. Serie de Fourier

Sea f una función 2π -periódica en \mathbb{R} . Se escribirá $f \in L_p(0, 2\pi)$ si

$$f(t)\chi_{[0, 2\pi]}(t) \in L_p(0, 2\pi), \quad p \geq 1.$$

Cualquier función f , 2π -periódica en \mathbb{R} , tal que $f \in L_2(0, 2\pi)$, se puede representar por una serie de Fourier convergente en $L_2(0, 2\pi)$

$$f(t) = \sum_n c_n e^{int},$$

donde los coeficientes de Fourier son dados por

$$c_n = \frac{1}{2\pi} \int_0^{2\pi} f(t) e^{-int} dt.$$

Se puede verificar que si $f \in L_1(\mathbb{R})$, entonces la serie

$$S(t) = \sum_k f(t + 2k\pi) \quad (1.2.6)$$

converge casi para todo t y pertenece a $L_1(0, 2\pi)$. Además, los coeficientes de Fourier de $S(t)$ están dados por

$$c_k = \frac{1}{2\pi} \hat{f}(k) = \mathcal{F}^{-1}(f)(-k).$$

En efecto, para ver la expresión (1.2.6), basta probar que

$$\int_0^{2\pi} \sum_k |f(t + 2k\pi)| dt < \infty.$$

Para la segunda parte se calcula los coeficientes de Fourier

$$\frac{1}{2\pi} \int_0^{2\pi} \left[\sum_k f(t + 2k\pi) \right] e^{-ikt} dt.$$

Intercambiando la suma con la integral se obtiene

$$\begin{aligned} \sum_k \frac{1}{2\pi} \int_0^{2\pi} f(t + 2k\pi) e^{-ikt} dt &= \sum_k \frac{1}{2\pi} \int_{2\pi k}^{2\pi(k+1)} f(z) e^{-ikz} dz \\ &= \frac{1}{2\pi} \hat{f}(k). \end{aligned}$$

1.2.2. Teorema de muestreo de Shannon

Una de las principales aplicaciones de la transformada de Fourier es el teorema de muestreo de Shannon, el cual afirma que una señal de banda limitada se puede reconstruir a partir de ciertos valores muestrales (ver [9], [37] o [41]).

Una señal es una función f definida en todo \mathbb{R} que tiene energía finita $\int_{-\infty}^{\infty} |f(t)|^2 dt$, es decir, $f \in L_2(\mathbb{R})$. La señal f es de banda limitada si su transformada de Fourier \hat{f} tiene soporte compacto, es decir, existe $\lambda > 0$ tal que $\hat{f}(\omega) = 0$ si $|\omega| > \lambda$, o de manera equivalente, $\hat{f}(\omega) \neq 0$ para $|\omega| \leq \lambda$. En la práctica, λ se elige de manera que sea el menor número que cumple con esta condición. En este caso, λ es el ancho de banda de la señal.

La reconstrucción de una señal f de banda limitada a partir de valores muestrales tomados en tiempos apropiados, se puede realizar utilizando la fórmula de inversión de Fourier (1.2.1). En consecuencia, si f es banda limitada entonces

$$f(t) = \frac{1}{2\pi} \int_{-\lambda}^{\lambda} \hat{f}(\omega) e^{i\omega t} d\omega. \tag{1.2.7}$$

8 Preliminares

Por otro lado, la serie de Fourier para $\hat{f}(\omega)$ en $[-\lambda, \lambda]$ es

$$\hat{f}(\omega) = \sum_n c_n e^{n\pi i \omega / \lambda}, \quad (1.2.8)$$

donde

$$c_n = \frac{1}{2\lambda} \int_{-\lambda}^{\lambda} \hat{f}(\omega) e^{-n\pi i \omega / \lambda} d\omega.$$

Al comparar c_n con $f(t)$ en la ecuación (1.2.7), se obtiene

$$c_n = \frac{\pi}{\lambda} f\left(-\frac{n\pi}{\lambda}\right).$$

Reemplazando en (1.2.8) se tiene

$$\hat{f}(\omega) = \sum_n \frac{\pi}{\lambda} f\left(\frac{n\pi}{\lambda}\right) e^{n\pi i \omega / \lambda}. \quad (1.2.9)$$

Observe que se reemplazó n por $-n$ debido a que n toma todos los valores enteros en esta sumatoria. Al sustituir (1.2.9) en (1.2.7) se tiene

$$f(t) = \frac{1}{2\lambda} \int_{-\lambda}^{\lambda} \sum_n f\left(\frac{n\pi}{\lambda}\right) e^{i\omega(t-n\pi/\lambda)} d\omega.$$

Al intercambiar la sumatoria con la integral se obtiene

$$f(t) = \sum_n f\left(\frac{n\pi}{\lambda}\right) \frac{\text{sen}(\lambda t - n\pi)}{\lambda t - n\pi}. \quad (1.2.10)$$

En consecuencia, una señal $f(t)$ se puede reconstruir mediante el muestreo en los tiempos $0, \pm\pi/\lambda, \pm2\pi/\lambda, \dots$, una vez conocidos los valores de $f(t)$ para estos tiempos, entonces la ecuación (1.2.10) reconstruye toda la señal. La ecuación (1.2.10) se conoce como el teorema de muestreo de Shannon.

1.2.3. Filtrado

En teoría de comunicación un operador lineal se llama sistema lineal (ver [37], [41] o [50]). Un sistema lineal H se llama invariante en el tiempo si para cada señal de entrada $f(t)$ con salida $g(t)$ se cumple

$$H(f(t - t_0)) = g(t - t_0), \quad \forall t_0 \in \mathbb{R}.$$

El término filtro se utilizará para describir un sistema lineal e invariante en el tiempo. Los filtros se clasifican según sus características en el dominio de

la frecuencia como paso bajo, paso alto, paso banda y de banda eliminada. El filtrado se emplea en procesamiento digital de señales de diferentes maneras, por ejemplo, eliminación de ruido, detección de señales en radares, análisis espectral de señales y en otras tantas aplicaciones [37], [41] o [50].

Si f es una señal, no necesariamente de banda limitada, el espectro de f está dado por su transformada de Fourier \hat{f} . Cuando f no es de banda limitada, f se puede reemplazar con una señal de banda limitada f_λ con ancho de banda que no exceda el número positivo λ , aplicando un filtro de paso bajo que elimina las frecuencias fuera del rango $[-\lambda, \lambda]$. Esto es, sea

$$\hat{f}_\lambda(\omega) = \begin{cases} \hat{f}(\omega), & \text{para } |\omega| \leq \lambda; \\ 0, & \text{para } |\omega| > \lambda. \end{cases}$$

Esto define la transformada de la función f_λ a partir de la cual se puede recuperar f_λ por medio de la fórmula de inversión de Fourier

$$f_\lambda(t) = \frac{1}{2\pi} \int_{\mathbb{R}} \hat{f}_\lambda(\omega) e^{i\omega t} d\omega = \frac{1}{2\pi} \int_{-\lambda}^{\lambda} \hat{f}(\omega) e^{i\omega t} d\omega.$$

El proceso de aplicar un filtro de paso bajo es equivalente a multiplicar por la función característica $\chi_{[-\lambda, \lambda]}$, es decir,

$$\hat{f}_\lambda(\omega) = \chi_{[-\lambda, \lambda]}(\omega) \hat{f}(\omega). \quad (1.2.11)$$

La transformada inversa de Fourier de la función característica es

$$\mathcal{F}^{-1}(\chi_{[-\lambda, \lambda]})(t) = \frac{1}{2\pi} \int_{-\lambda}^{\lambda} e^{i\omega t} d\omega = \frac{\text{sen } \lambda t}{\pi t}.$$

Un hecho conocido en procesamiento de señales (ver p.e., [41] o [50]), el filtrado tanto digital como analógico se hace por convolución. En consecuencia, si $\phi(t)$ es una función filtro, entonces el efecto de filtrar una señal f por ϕ es una nueva función g definida por $g(t) = (\phi * f)(t)$ y al aplicar la propiedad 3 de la Proposición (1.2.2) se tiene $\hat{g}(\omega) = \hat{\phi}(\omega) \hat{f}(\omega)$. Por tanto, (1.2.11) queda después de aplicar la fórmula de inversión

$$f_\lambda(t) = \left(\frac{\text{sen } \lambda t}{\pi t} * f(t) \right).$$

Esto da el filtrado de paso bajo de f como la convolución de la función $\frac{\text{sen } \lambda t}{\pi t}$ con f .

1.3. Espacio de probabilidad

A partir de la teoría de la medida, la teoría de probabilidad ha alcanzado un alto grado de formalización. En las siguientes líneas se presentan algunos elementos básicos sobre el tema, para un estudio profundo se puede consultar [3].

Definición 1.3.1. Sea Ω un conjunto no vacío y \mathcal{A} una colección de subconjuntos de Ω . \mathcal{A} es una σ -álgebra sobre Ω si y sólo si se satisfacen las siguientes condiciones

i) $\Omega \in \mathcal{A}$

ii) Si A_1, A_2, \dots es una sucesión contable de elementos de \mathcal{A} , entonces $\bigcup A_n \in \mathcal{A}$

iii) Si $A \in \mathcal{A}$, entonces $A^c \in \mathcal{A}$, donde A^c es el complemento de A en Ω .

La pareja (Ω, \mathcal{A}) se llama espacio medible y a los elementos de \mathcal{A} , conjuntos medibles.

Definición 1.3.2. Sea \mathcal{C} una colección de subconjuntos de Ω . Por σ -álgebra minimal que contiene a \mathcal{C} o la σ -álgebra que genera a \mathcal{C} , denotada $\sigma(\mathcal{C})$, se entiende una σ -álgebra de subconjuntos de Ω tal que si \mathcal{K} es otra σ -álgebra que contiene a \mathcal{C} , entonces $\mathcal{C} \subset \sigma(\mathcal{C}) \subset \mathcal{K}$.

La σ -álgebra \mathcal{B} generada por todos los conjuntos abiertos de \mathbb{R}^n , se llama álgebra de Borel y los elementos en \mathcal{B} se llaman conjuntos de Borel. Esta σ -álgebra es de gran interés en diversos campos de la matemática, en particular en la teoría de probabilidades.

Definición 1.3.3. Una probabilidad P es una medida normalizada sobre un espacio medible (Ω, \mathcal{A}) ; esto es, P es una función de valor real la cual asigna a todo $A \in \mathcal{A}$ el número $P(A)$ tal que

i) $P(\Omega) = 1$

ii) Si A_1, A_2, \dots es una sucesión contable de elementos de \mathcal{A} disjuntos dos a dos, entonces

$$P\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} P(A_n)$$

iii) $P(A) \geq 0$ para todo $A \in \mathcal{A}$.

La tripla (Ω, \mathcal{A}, P) se llama espacio de probabilidad. $P(A)$ se lee como la probabilidad del evento A .

Algunas consecuencias de la definición (1.3.3) son:

1. $P(\emptyset) = 0$.
2. Sean A y B eventos. Si $A \subset B$, entonces $P(A) \leq P(B)$.
3. Si A_1, A_2, \dots, A_n son eventos disjuntos dos a dos, entonces

$$P\left(\bigcup_{k=1}^n A_k\right) = \sum_{k=1}^n P(A_k).$$

4. $P(A^c) = 1 - P(A)$, para todo $A \in \mathcal{A}$.
5. Si $\{A_n\}$ es una sucesión contable de eventos, entonces

$$P\left(\bigcup_{n=1}^{\infty} A_n\right) \leq \sum_{n=1}^{\infty} P(A_n), \quad \text{desigualdad de Boole.}$$

1.3.1. Variable aleatoria

Definición 1.3.4. Una variable aleatoria X es una función de valor real cuyo dominio es Ω y la cual es \mathcal{A} -medible, esto es, para cualquier número real x , $\{\omega \in \Omega : X(\omega) \leq x\} \in \mathcal{A}$.

El conjunto $\{\omega \in \Omega : X(\omega) \leq x\}$ se llama conjunto de eventos elementales, se denotará por $[X \leq x]$.

Si X es una variable aleatoria, la función de distribución F_X se define por

$$F_X(x) = P[X \leq x], \quad \text{para todo } x \in \mathbb{R}.$$

Note que diferentes variables aleatorias pueden tener la misma función de distribución. Por ejemplo, sea $\Omega = \{C, S\}$, si $P(C) = P(S) = 1/2$ y si X y Y son variables aleatorias definidas por $X(C) = 1, X(S) = 0, Y(C) = 0$ y $Y(S) = 1$, entonces

$$F_X(x) = F_Y(x) = \begin{cases} 0, & \text{si } x < 0 \\ 1/2, & \text{si } 0 \leq x < 1 \\ 1, & \text{si } x \geq 1. \end{cases}$$

Si X es una variable aleatoria, entonces la función de distribución F_X tiene las siguientes propiedades:

12 Preliminares

1. F_X es no decreciente, es decir, si $-\infty < a < b < \infty$, entonces

$$F_X(a) \leq F_X(b)$$

2. $\lim_{x \rightarrow \infty} F_X(x) = 1$ y $\lim_{x \rightarrow -\infty} F_X(x) = 0$

3. F_X es continua por la derecha, esto es,

$$\lim_{h \downarrow 0} F_X(x+h) = F_X(x), \quad \forall x.$$

Una función de distribución F se llama absolutamente continua, si existe una función medible Borel f sobre \mathbb{R} tal que

$$F(x) = \int_{-\infty}^x f(t)dt \quad \forall x.$$

La función f se llama la densidad de F .

Si X es una variable aleatoria con función de distribución absolutamente continua y densidad f , entonces el valor esperado de X es dado por

$$E(X) = \int_{\mathbb{R}} xf(x)dx,$$

siempre que la integral sea finita.

1.3.2. Procesos estocásticos

Una variable aleatoria siempre tiene asociada una distribución de probabilidad que mide la probabilidad de ocurrencia de sus distintos resultados. Cuando la variable aleatoria cambia con el tiempo, se le puede asociar una distribución de probabilidad que también varía con el tiempo. En tales ambientes resulta útil definir un proceso estocástico [45].

Definición 1.3.5. Sea $I \subseteq \mathbb{R}$ un conjunto de índices y (Ω, \mathcal{A}, P) un espacio de probabilidad. Una función $X : I \times \Omega \rightarrow \mathbb{R}^n$ es un proceso estocástico si para cada $t \in I$ fijo, la función $X_t : \Omega \rightarrow \mathbb{R}^n$ es una variable aleatoria, que representa el valor del proceso $X(t, \omega)$, $\omega \in \Omega$. Si $\omega \in \Omega$ es fijo, la aplicación $I \rightarrow \mathbb{R}^n$ tal que $t \mapsto X_t(\omega)$ se llama la trayectoria o realización del proceso X .

Los valores que toma el proceso en \mathbb{R}^n se llaman estados del proceso. Si el conjunto I es contable, el proceso estocástico X se dice que es de tiempo discreto. Por otro lado, si I es un intervalo de los reales no negativos, el proceso estocástico es de tiempo continuo.

Si X es un proceso estocástico continuo, entonces

- i) X es independiente si para todo $t, s \in I$ $s \neq t$, las variables aleatorias asociadas X_s y X_t son independientes.
- ii) X es independientemente distribuida, si la distribución de probabilidad F_{X_t} es la misma para cada $t \in I$.
- iii) X tiene incrementos independientes si para cada $n \geq 1$ y para cualquier partición del intervalo I , $t_0 < t_1 < \dots < t_n$, las diferencias

$$X_{t_1} - X_{t_0}, X_{t_2} - X_{t_1}, \dots, X_{t_n} - X_{t_{n-1}}$$

son variables aleatorias independientes.

- iv) X tiene incrementos estacionarios si $X_t - X_s \stackrel{d}{=} X_{t+h} - X_{t+s}$ para cada $t, s, t+h, s+h$ en I , $s < t$ y $h > 0$. El símbolo $\stackrel{d}{=}$ significa que los términos en comparación tienen la misma distribución de probabilidad. La estacionariedad de un proceso estocástico alude a que la distribución de probabilidad de la diferencia entre dos variables aleatorias permanece invariante bajo cualquier traslación temporal.

Una serie de tiempo es la realización de un proceso estocástico. En otras palabras, una serie de tiempo se puede considerar como una colección de variables aleatorias $\{X_t : t \in I\}$ (ver p.e., [25], [38] o [62]).

CAPÍTULO 2

Introducción a las wavelets

2.1. Introducción

El origen de la descomposición de una señal en wavelets está en la necesidad de conocer las características y particularidades de la señal en diferentes instantes de tiempo. La principal virtud de las wavelets es que permite modelar procesos que dependen fuertemente del tiempo y para los cuales su comportamiento no tiene porqué ser suave [1], [9], [11], [12], [20], [23], [24]. Una de las ventajas de las wavelets frente a los métodos clásicos, como la transformada de Fourier, es que en el segundo caso se maneja una base de funciones bien localizada en frecuencia pero no en tiempo, esto es, el análisis en frecuencia obtenido del análisis de Fourier es insensible a perturbaciones que supongan variaciones instantáneas y puntuales de la señal como picos debidos a conmutaciones o variaciones muy lentas como tendencias. En otras palabras, si f es una señal (f es una función definida en todo \mathbb{R} y tiene energía finita $\int_{-\infty}^{\infty} |f(t)|^2 dt$). La transformada de Fourier $\hat{f}(\omega)$ proporciona la información global de la señal en el tiempo localizada en frecuencia. Sin embargo, $\hat{f}(\omega)$ no particulariza la información para intervalos de tiempo específicos, ya que

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt$$

y la integración es sobre todo tiempo (ver [21]). Así, la imagen obtenida no contiene información sobre tiempos específicos, sino que sólo permite calcular

el espectro de amplitud total $|\hat{f}(\omega)|$, mientras que la mayoría de las wavelets interesantes presentan una buena localización en tiempo y en frecuencia, disponiendo incluso de bases de wavelets con soporte compacto.

En este capítulo se presenta una introducción a la teoría wavelets, en particular se estudiará la transformada wavelet y el análisis multirresolución en $L_2(\mathbb{R})$. Con este concepto se ilustra como construir otras bases wavelets, y además, permite analizar funciones (señales) en $L_2(\mathbb{R})$ en varias escalas (niveles de resolución) [6], [9], [15], [20]. Para ello, se utiliza versiones escaladas de un conjunto ortonormal en $L_2(\mathbb{R})$. Para tal descomposición de una función $f \in L_2(\mathbb{R})$, sólo se necesitan los coeficientes de la expansión de f en dicho conjunto ortonormal.

2.2. Transformadas wavelets

El análisis wavelets es un método de descomposición de una función o señal usando funciones especiales, las wavelets. La descomposición es similar a la de la transformada de Fourier, donde una señal $f(t)$ se descompone en una suma infinita de armónicos $e^{i\omega t}$ de frecuencias $\omega \in \mathbb{R}$, cuyas amplitudes son los valores de la transformada de Fourier de f , $\hat{f}(\omega)$:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{i\omega t} d\omega, \quad \text{donde} \quad \hat{f}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt.$$

El análisis de Fourier tiene el defecto de la no localidad: el comportamiento de una función en un conjunto abierto, no importa cuán pequeño, influye en el comportamiento global de la transformada de Fourier. No se captan los aspectos locales de la señal tales como cambios bruscos, saltos o picos, que se han de determinar a partir de su reconstrucción.

2.2.1. Transformada wavelet continua

La teoría wavelets se basa en la representación de una función en términos de una familia biparamétrica de dilataciones y traslaciones de una función fija ψ , la wavelet madre que, en general, no es senoidal. Por ejemplo,

$$f(t) = \int_{\mathbb{R}^2} \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right) \mathcal{W}_\psi f(a, b) da db$$

en donde $\mathcal{W}_\psi f$ es una transformada de f definida adecuadamente. También se tiene de modo alterno un desarrollo en serie

$$f(t) = \sum_{j,k} c_{j,k} 2^{j/2} \psi(2^j t - k)$$

en donde se suma sobre las dilataciones en progresión geométrica. Para conservar la norma en $L_2(\mathbb{R})$ de la wavelet madre ψ , se insertan los factores $\frac{1}{\sqrt{|a|}}$ y $2^{j/2}$, respectivamente.

Definición 2.2.1. Una wavelet ψ es una función cuadrado integrable tal que la siguiente condición de admisibilidad se tiene

$$C_\psi := \int_{\mathbb{R}} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty, \quad (2.2.1)$$

donde $\hat{\psi}(\omega)$ es la transformada de Fourier de ψ .

Observación 2.2.1. Si además $\psi \in L_1(\mathbb{R})$, entonces la condición (2.2.1) implica que $\int_{\mathbb{R}} \psi(t) dt = 0$. En efecto, por el Lema de Riemann-Lebesgue (ver p.e., [39]), $\lim_{\omega \rightarrow \infty} \hat{\psi}(\omega) = 0$ y la transformada de Fourier es continua, lo cual implica que $0 = \hat{\psi}(0) = \int_{\mathbb{R}} \psi(t) dt$.

Sea $\psi \in L_2(\mathbb{R})$. La función dilatada y trasladada se define por

$$\psi_{a,b}(t) := \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right), \quad a, b \in \mathbb{R}, \quad a \neq 0.$$

Esta función se obtiene a partir de ψ , primero por dilatación en el factor a y, luego, por traslación en b . Es claro que $\|\psi_{a,b}\|_2 = \|\psi\|_2$.

Definición 2.2.2. Para $f, \psi \in L_2(\mathbb{R})$, la expresión

$$\mathcal{W}_\psi f(a, b) := \int_{\mathbb{R}} f(t) \overline{\psi_{a,b}(t)} dt \quad (2.2.2)$$

se llama la transformada wavelet de f .

Por la desigualdad de Cauchy, se ve que $\mathcal{W}_\psi f$ es una función acotada con $|\mathcal{W}_\psi f(a, b)| \leq \|f\|_2 \|\psi\|_2$. Note también que

$$\mathcal{W}_\psi f(a, b) = \langle f, \psi_{a,b} \rangle_{L_2(\mathbb{R})} = \langle f, \psi_{a,b} \rangle.$$

La transformada wavelet $\mathcal{W}_\psi f$ de f puede ser descrita en términos del producto de convolución. La convolución de dos funciones $f, g \in L_2(\mathbb{R})$ es dada por

$$(f * g)(t) = \int_{\mathbb{R}} f(t-z)g(z)dz.$$

Observe que esta fórmula está definida para al menos todo $t \in \mathbb{R}$, pero $f * g$ no necesariamente está en $L_2(\mathbb{R})$. Usando la notación $\tilde{\psi}(t) = \overline{\psi(-t)}$, se tiene $\mathcal{W}_\psi f(a, b) = (f * \tilde{\psi}_{a,0})(b)$. Note también que $\hat{\psi}_{a,b}(\omega) = \sqrt{|a|} \tilde{\hat{\psi}}(a\omega) e^{-i\omega b}$. Estos hechos se aplicarán en la prueba de la siguiente proposición, la cual establece la fórmula de Plancherel para la transformada wavelet.

Proposición 2.2.3. Sea $\psi \in L_2(\mathbb{R})$ y satisfice la condición (2.2.1). Entonces para cualquier $f \in L_2(\mathbb{R})$, las siguientes relaciones se tienen

1. Isometría

$$\int_{\mathbb{R}} |f(t)|^2 dt = \frac{1}{C_\psi} \int_{\mathbb{R}^2} |\mathcal{W}_\psi f(a, b)|^2 db \frac{da}{a^2}$$

2. Fórmula de inversión

$$f(t) = \frac{1}{C_\psi} \int_{\mathbb{R}^2} \mathcal{W}_\psi f(a, b) \psi_{a,b}(t) db \frac{da}{a^2}$$

Demostración. 1. Es fácil verificar que $(f * \tilde{\psi}_{a,0})(b) = \sqrt{|a|} \mathcal{F}^{-1}\{\hat{f}(\omega) \tilde{\psi}(a\omega)\}$. En consecuencia,

$$\begin{aligned} \int_{\mathbb{R}^2} |\mathcal{W}_\psi f(a, b)|^2 db \frac{da}{a^2} &= \int_{\mathbb{R}} \int_{\mathbb{R}} |(f * \tilde{\psi}_{a,0})(b)|^2 db \frac{da}{a^2} \\ &= \int_{\mathbb{R}} \int_{\mathbb{R}} |a| |\mathcal{F}^{-1}(\hat{f}(\cdot) \tilde{\psi}(a\cdot))(\omega)|^2 d\omega \frac{da}{a^2} \\ &= \int_{\mathbb{R}} \int_{\mathbb{R}} |\hat{f}(\omega)|^2 |\hat{\psi}(a\omega)|^2 d\omega \frac{da}{|a|} \\ &= \int_{\mathbb{R}} |\hat{f}(\omega)|^2 \left[\int_{\mathbb{R}} |\hat{\psi}(a\omega)|^2 \frac{da}{|a|} \right] d\omega \\ &= C_\psi \int_{\mathbb{R}} |\hat{f}(\omega)|^2 d\omega = C_\psi \|f\|_2^2. \end{aligned}$$

Observe que se utilizó el teorema de Fubini y la fórmula de Plancherel para la transformada de Fourier.

2. Para simplificar los cálculos en la fórmula de inversión, suponga que $f, \hat{f} \in L_1(\mathbb{R})$.

$$\begin{aligned} \int_{\mathbb{R}} \mathcal{W}_\psi f(a, b) \psi_{a,b}(t) db &= \sqrt{|a|} \int_{\mathbb{R}} \mathcal{F}^{-1}(\hat{f}(\cdot) \tilde{\psi}(a\cdot))(\omega) \psi_{a,b}(t) d\omega \\ &= \sqrt{|a|} \int_{\mathbb{R}} \hat{f}(\omega) \tilde{\psi}(a\omega) \mathcal{F}^{-1}(g)(\omega) d\omega, \end{aligned}$$

donde $g(b) := \psi_{a,b}(t)$. Ahora, la transformada inversa de Fourier de g es

$$\begin{aligned} \mathcal{F}^{-1}(g)(\omega) &= \frac{1}{2\pi} \int_{\mathbb{R}} g(b) e^{i\omega b} db \\ &= \frac{1}{2\pi} \sqrt{|a|} \int_{\mathbb{R}} \psi(z) e^{-ia\omega z} e^{i\omega t} dz \\ &= \frac{1}{2\pi} \sqrt{|a|} \hat{\psi}(a\omega) e^{i\omega t}. \end{aligned}$$

Sustituyendo e integrando respecto a $a^{-2}da$ se obtiene

$$\begin{aligned}
 \int_{\mathbb{R}^2} \mathcal{W}_\psi f(a, b) \psi_{a, b}(t) db \frac{da}{a^2} &= \frac{1}{2\pi} \int_{\mathbb{R}} |a| \left[\int_{\mathbb{R}} \hat{f}(\omega) |\hat{\psi}(a\omega)|^2 e^{i\omega t} d\omega \right] \frac{da}{a^2} \\
 &= \frac{1}{2\pi} \int_{\mathbb{R}} \hat{f}(\omega) \left[\int_{\mathbb{R}} |\hat{\psi}(a\omega)|^2 \frac{da}{|a|} \right] e^{i\omega t} d\omega \\
 &= C_\psi \frac{1}{2\pi} \int_{\mathbb{R}} \hat{f}(\omega) e^{i\omega t} d\omega \\
 &= C_\psi f(t).
 \end{aligned}$$

□

Otro resultado de interés que se presentará en la siguiente proposición, es la fórmula de Parseval para la transformada wavelet.

Proposición 2.2.4. *Sea $\psi \in L_2(\mathbb{R})$ y satisface la condición (2.2.1). Entonces para cualquier $f, g \in L_2(\mathbb{R})$, se tienen*

$$\langle f, g \rangle_{L_2(\mathbb{R})} = \frac{1}{C_\psi} \int_{\mathbb{R}^2} \mathcal{W}_\psi f(a, b) \overline{\mathcal{W}_\psi g(a, b)} \frac{dad b}{a^2}$$

Demostración. Como $(f * \tilde{\psi}_{a,0})(b) = \sqrt{|a|} \mathcal{F}^{-1}\{\hat{f}(\omega) \tilde{\psi}(a\omega)\}$ o de manera equivalente, $\mathcal{F}(f * \tilde{\psi}_{a,0})(\omega) = \sqrt{|a|} \hat{f}(\omega) \tilde{\psi}(a\omega)$, entonces

$$\int_{\mathbb{R}} \mathcal{W}_\psi f(a, b) \overline{\mathcal{W}_\psi g(a, b)} db = |a| \int_{\mathbb{R}} \hat{f}(\omega) \tilde{g}(\omega) |\hat{\psi}(a\omega)|^2 d\omega,$$

ahora, integrando respecto a $a^{-2}da$ se sigue

$$\begin{aligned}
 \int_{\mathbb{R}^2} \mathcal{W}_\psi f(a, b) \overline{\mathcal{W}_\psi g(a, b)} db \frac{da}{a^2} &= \int_{\mathbb{R}} |a| \left[\int_{\mathbb{R}} \hat{f}(\omega) \tilde{g}(\omega) |\hat{\psi}(a\omega)|^2 d\omega \right] \frac{da}{a^2} \\
 &= \int_{\mathbb{R}} \hat{f}(\omega) \tilde{g}(\omega) \left[\int_{\mathbb{R}} |\hat{\psi}(a\omega)|^2 \frac{da}{|a|} \right] d\omega \\
 &= C_\psi \int_{\mathbb{R}} \hat{f}(\omega) \tilde{g}(\omega) d\omega \\
 &= C_\psi \langle \hat{f}, \tilde{g} \rangle_{L_2(\mathbb{R})} = C_\psi \langle f, g \rangle_{L_2(\mathbb{R})}.
 \end{aligned}$$

Note que se aplicó el teorema de Fubini, y en el último renglón de la expresión anterior, la fórmula de Parseval para la transformada de Fourier. □

En la siguiente proposición se listan algunas propiedades.

Proposición 2.2.5. *Sean ψ y φ wavelets y $f, g \in L_2(\mathbb{R})$. Entonces*

1. $\mathcal{W}_\psi(\alpha f + \beta g)(a, b) = \alpha \mathcal{W}_\psi f(a, b) + \beta \mathcal{W}_\psi g(a, b)$, $\alpha, \beta \in \mathbb{R}$.
2. $\mathcal{W}_{\alpha\psi + \beta\varphi} f(a, b) = \alpha \mathcal{W}_\psi f(a, b) + \beta \mathcal{W}_\varphi f(a, b)$, $\alpha, \beta \in \mathbb{R}$.
3. $\mathcal{W}_\psi(T_c f)(a, b) = \mathcal{W}_\psi f(a, b - c)$, donde T_c es el operador traslación definido por $T_c f(t) = f(t - c)$.
4. $\mathcal{W}_\psi(D_c f)(a, b) = \sqrt{c} \mathcal{W}_\psi f(ca, cb)$, donde D_c es el operador dilatación definido por $D_c f(t) = \sqrt{c} f(ct)$.

2.2.2. Transformada wavelet discreta

La transformada wavelet continua introduce cierta redundancia, pues la señal original se puede reconstruir completamente calculando $\mathcal{W}_\psi f(a, \cdot)$ para una cantidad numerable de escalas, por ejemplo, potencias enteras de 2. Esto es, si se elige la escala $a = 2^{-j}$ para cada $j \in \mathbb{Z}$, y también se discretiza en el dominio del tiempo en los puntos $b = 2^{-j}k$, $k \in \mathbb{Z}$, la familia de wavelets será ahora dada por

$$\psi_{2^{-j}, 2^{-j}k}(t) = \frac{1}{\sqrt{2^{-j}}} \psi\left(\frac{t - 2^{-j}k}{2^{-j}}\right) = 2^{j/2} \psi(2^j t - k), \quad \forall j, k \in \mathbb{Z}.$$

Se utilizará la notación ψ_{jk} para denotar la wavelet ψ comprimida 2^j y trasladada el entero k , es decir, $\psi_{jk}(t) = 2^{j/2} \psi(2^j t - k)$.

Con la elección de $a = 2^{-j}$ y $b = 2^{-j}k$, observe que el muestreo en el tiempo se ajusta proporcionalmente a la escala, es decir, a mayor escala se toma puntos más distantes, ya que se busca información global, mientras que a menor escala se buscan detalles de la señal, por tal motivo se muestrea en puntos menos distantes entre si. Para otras elecciones de a y b se puede consultar [8].

Definición 2.2.6. Una función $\psi \in L_2(\mathbb{R})$ es una wavelet si la familia de funciones ψ_{jk} definidas por

$$\psi_{jk}(t) = 2^{j/2} \psi(2^j t - k), \quad \forall j, k \in \mathbb{Z}, \quad (2.2.3)$$

es una base ortonormal en el espacio $L_2(\mathbb{R})$.

Una condición suficiente para la reconstrucción de una señal f es que la familia de dilatadas y trasladadas ψ_{jk} forme una base ortonormal en el espacio $L_2(\mathbb{R})$, ver [15] y [27] para más detalles. Si esto se tiene, cualquier función $f \in L_2(\mathbb{R})$ se puede escribir como

$$f(t) = \sum_{j,k} d_{j,k} \psi_{jk}(t) \quad (2.2.4)$$

o teniendo en cuenta (2.2.3) como

$$f(t) = \sum_{j,k} d_{j,k} 2^{j/2} \psi(2^j t - k),$$

donde $d_{j,k} = \langle f, \psi_{2^{-j}, 2^{-j}k} \rangle = \mathcal{W}_\psi f(2^{-j}, 2^{-j}k)$.

Definición 2.2.7. Para cada $f \in L_2(\mathbb{R})$ el conjunto bidimensional de coeficientes

$$d_{j,k} = \langle f, \psi_{jk} \rangle = \int_{\mathbb{R}} 2^{j/2} f(t) \overline{\psi(2^j t - k)} dt$$

se llama la transformada wavelet discreta de f .

En consecuencia, la expresión (2.2.4) se puede escribir en forma alterna como

$$f(t) = \sum_{j,k} \langle f(t), \psi_{jk}(t) \rangle \psi_{jk}(t). \tag{2.2.5}$$

La serie (2.2.5) se llama representación wavelet de f .

Ejemplo 2.2.1. El ejemplo más clásico es la wavelet de Haar, la cual es dada por

$$\psi(t) = \chi_{[0, \frac{1}{2})} - \chi_{[\frac{1}{2}, 1)} = \begin{cases} 1, & 0 \leq t < \frac{1}{2}; \\ -1, & \frac{1}{2} \leq t < 1. \end{cases}$$

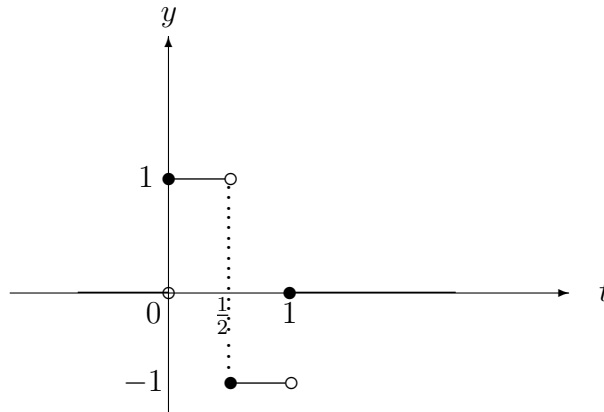


Figura 2.2.1

Observación 2.2.2. 1) $\psi_{jk}(t)$ es más apropiada para representar detalles finos de la señal como oscilaciones rápidas. Los coeficientes wavelet $d_{j,k}$ miden la cantidad de fluctuación sobre el punto $t = 2^{-j}k$ con una frecuencia determinada por el índice de dilatación j .

2) Las wavelets gozan de la “propiedad zoom,” esto hace que las bases wavelet sean excelentes detectores de singularidades, en otras palabras, las singularidades producen coeficientes wavelet grandes.

3) La propiedad zoom es común en todos los sistemas wavelet, constituye la mayor diferencia con los sistemas de Fourier para la detección de singularidades. En problemas de teoría de señales, las singularidades llevan información esencial como la presencia de esquinas en las imágenes. Esto hace de las bases wavelet una herramienta muy útil para el procesamiento de imágenes, en detrimento del análisis de Fourier.

4) Es interesante notar que

$$d_{j,k} = \mathcal{W}_\psi f(2^{-j}, 2^{-j}k)$$

es la transformada wavelet de f en el punto $(2^{-j}, 2^{-j}k)$. Estos coeficientes analizan la señal mediante la wavelet madre ψ .

2.3. Análisis Multirresolución

El sistema de Haar no es muy apropiado para aproximar funciones suaves. De hecho, cualquier aproximación de Haar es una función discontinua [15], [27]. Se puede probar que si f es una función muy suave, los coeficientes de Haar decrecerán muy lentamente. Por tanto se pretende construir wavelets que tengan mejor propiedades de aproximación, y una forma de hacerlo es a través del análisis multirresolución (AMR) [34], [33],[32], [31], [15].

Sea $\varphi \in L_2(\mathbb{R})$, la familia de trasladadas de φ ,

$$\{\varphi_{0k}, k \in \mathbb{Z}\} = \{\varphi_{0k}(\cdot - k), k \in \mathbb{Z}\}$$

es un sistema ortonormal (con el producto interno de $L_2(\mathbb{R})$). Acá y en lo que sigue

$$\varphi_{jk}(t) = 2^{j/2}\varphi(2^j t - k) = D_{2^j}T_k\varphi(t), \quad j \in \mathbb{Z}, \quad k \in \mathbb{Z},$$

recuerde que $D_a f(t) = a^{1/2}f(at)$ y $T_a f(t) = f(t - a)$ son los operadores dilatación y traslación, respectivamente.

Se definen los espacios vectoriales

$$\begin{aligned} V_0 &= \left\{ f(t) = \sum_k c_k \varphi(t-k) : \sum_k |c_k|^2 < \infty \right\}, \\ V_1 &= \left\{ h(t) = f(2t) : f \in V_0 \right\}, \\ &\vdots \\ V_j &= \left\{ h(t) = f(2^j t) : f \in V_0 \right\}, j \in \mathbb{Z} \\ &= \text{gen} \{ \varphi_{jk}(t) = 2^{j/2} \varphi(2^j t - k) : k \in \mathbb{Z} \}. \end{aligned}$$

Note que φ genera la sucesión de espacios $\{V_j, j \in \mathbb{Z}\}$. Suponga que la función φ se escoge de tal forma que los espacios estén encajados $V_j \subset V_{j+1}$, $j \in \mathbb{Z}$, y $\bigcup_{j \geq 0} V_j$ es denso en $L_2(\mathbb{R})$, estos dos hechos fundamentales hacen parte de la definición de análisis multirresolución.

Definición 2.3.1. *Un análisis multirresolución en $L_2(\mathbb{R})$ es una sucesión creciente de subespacios cerrados V_j , $j \in \mathbb{Z}$, en $L_2(\mathbb{R})$, $\dots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \dots$ tales que*

1. $\bigcup_{j \in \mathbb{Z}} V_j$ es denso en $L_2(\mathbb{R})$,
2. $\bigcap_{j \in \mathbb{Z}} V_j = \{0\}$,
3. $f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1}$, $j \in \mathbb{Z}$,
4. $f(t) \in V_0 \Leftrightarrow f(t-k) \in V_0$, $j \in \mathbb{Z}$,
5. Existe una función $\varphi \in L_2(\mathbb{R})$ tal que el conjunto de funciones $\{\varphi(t-k)\}_{k \in \mathbb{Z}}$ es una base ortonormal para V_0 .

La función φ se llama función de escala. En el espacio V_{j+1} las funciones (señales) se describen con más detalle que en el espacio V_j , la resolución es mejor en el espacio “más grande”. Esto es, las funciones en V_{j+1} que no están en V_j realzan la resolución [9], [20]. Es usual reunir estos “sintonizadores finos” en un nuevo subespacio $W_j = V_{j+1} \setminus V_j$. Sin embargo, la elección de estos subespacios no es única. Pero se puede escoger a W_j como el complemento ortogonal de V_j en V_{j+1} . Es decir,

$$W_j = V_{j+1} \cap V_j^\perp, \quad j \in \mathbb{Z},$$

o de manera equivalente

$$V_{j+1} = V_j \oplus W_j, \quad j \in \mathbb{Z}. \tag{2.3.1}$$

Informalmente, esto quiere decir que si se tiene una función (señal) f a resolución 2^{j+1} y se proyecta a resolución inferior 2^j entonces

$$f = P_j f + \sum_{k \in \mathbb{Z}} \langle f, \psi_{jk} \rangle \psi_{jk},$$

acá P_j representa la proyección ortogonal en el espacio V_j donde se recoge la versión “suavizada” de f y la diferencia $f - P_j f$ representa el “detalle” de f , que está en W_j y se expresa como $\sum_{k \in \mathbb{Z}} \langle f, \psi_{jk} \rangle \psi_{jk}$. Recuerde que

$$P_j f = \sum_{k \in \mathbb{Z}} \langle f, \varphi_{jk} \rangle \varphi_{jk}, \quad j \in \mathbb{Z}.$$

En otras palabras, W_j contiene los detalles en V_{j+1} que no se representan en V_j , y cada función (señal) en W_j es ortogonal a toda función en V_j (ver p.e., [4]).

El conjunto de funciones linealmente independientes φ_{jk} que generan a V_j son las funciones de escala, mientras que el conjunto de funciones linealmente independientes ψ_{jk} que generan a W_j son las wavelets.

Por definición, el subespacio W_j es cerrado. Note también que si $f \in V_0$, entonces por 5 de la definición anterior se tiene

$$f(t) = \sum_k \langle f, T_k \varphi \rangle T_k \varphi(t).$$

Además, por la ortogonalidad de $\{T_k \varphi(t)\}_{k \in \mathbb{Z}}$,

$$\sum_k |\langle f, T_k \varphi \rangle|^2 = \|f\|_2^2.$$

Observe que al aplicar la descomposición (2.3.1) en cada V_j se obtiene

$$\begin{aligned} V_N &= V_{N-1} \oplus W_{N-1} = V_{N-2} \oplus W_{N-2} \oplus W_{N-1} \\ &= \dots = V_{-N} \oplus \left(\bigoplus_{j=-N}^{N-1} W_j \right), \end{aligned}$$

y cuando $N \rightarrow \infty$ se tiene

$$\overline{\bigcup_{j \in \mathbb{Z}} V_j} = \bigoplus_{j \in \mathbb{Z}} W_j \oplus \bigcap_{j \in \mathbb{Z}} V_j.$$

Usando las condiciones 1 y 2 de la definición de AMR se obtiene

$$\bigoplus_{j \in \mathbb{Z}} W_j = L_2(\mathbb{R}).$$

Por definición, también los subespacios W_j satisfacen las condiciones 3 y 4 de la definición de AMR o de manera directa como se prueba en el siguiente lema.

Lema 2.3.2. *Sea $(V_j)_{j \in \mathbb{Z}}$ un AMR y $W_j = V_{j+1} \cap V_j^\perp$. Entonces*

- i) $f \in W_j \Leftrightarrow Tf \in W_j$, para cada $j \in \mathbb{Z}$.
- ii) $f \in W_j \Leftrightarrow Df \in W_{j+1}$, para cada $j \in \mathbb{Z}$.

Demostración. Sea $f \in W_j$, esto significa que $f \in V_{j+1}$ y $\langle f, D_{2^j} T_k \varphi \rangle = 0$ para cada $k \in \mathbb{Z}$. Por la condición 3 y 4 de AMR, la primera relación es equivalente a $Tf \in V_{j+1}$ y $Df \in V_{j+2}$. Además de la relación $TD_2 = D_2T_2$, se sigue inmediatamente, que la segunda relación es equivalente a

$$\langle Tf, TD_{2^j} T_k \varphi \rangle = \langle Tf, D_{2^j} T_{k+2^j} \varphi \rangle = 0, \quad \forall k \in \mathbb{Z}.$$

Por tanto $Tf \in V_{j+1} \cap V_j^\perp$, y así, $Tf \in W_j$.
La segunda relación también es equivalente con

$$\langle Df, D_{2^{j+1}} T_k \varphi \rangle, \quad \forall k \in \mathbb{Z},$$

de lo cual se sigue que $Df \in V_{j+1}^\perp$ que junto con $Df \in V_{j+2}$ se obtiene $Df \in W_{j+1}$. \square

La siguiente proposición justifica los comentarios hechos arriba y es útil en futuros resultados.

Proposición 2.3.3. *Sea $(V_j)_{j \in \mathbb{Z}}$ un análisis multirresolución con función de escala φ . Entonces para cada $j \in \mathbb{Z}$, el conjunto de funciones*

$$\{\varphi_{jk}(t) = 2^{j/2} \varphi(2^j t - k), k \in \mathbb{Z}\}$$

es una base ortonormal para V_j .

Demostración. Para probar que $\{\varphi_{jk}(t), k \in \mathbb{Z}\}$ genera a V_j , se debe ver que toda $f(t) \in V_j$ se puede escribir como combinación lineal de funciones de $\{\varphi(2^j t - k), k \in \mathbb{Z}\}$. La propiedad 3 de la definición de AMR, implica que la función $f(2^{-j} t)$ pertenece a V_0 y por tanto $f(2^{-j} t)$ es combinación lineal de $\{\varphi(t - k), k \in \mathbb{Z}\}$. Haciendo la transformación $t \mapsto 2^j t$, se tiene que $f(t)$ es combinación lineal de $\{\varphi(2^j t - k), k \in \mathbb{Z}\}$.

Resta probar que $\{\varphi_{jk}(t), k \in \mathbb{Z}\}$ es ortonormal. Para ello se debe ver que

$$\langle \varphi_{jk}, \varphi_{jm} \rangle = \delta_{jk} = \begin{cases} 0, & \text{si } j \neq k; \\ 1, & \text{si } j = k \end{cases}$$

o

$$2^j \int_{-\infty}^{\infty} \varphi(2^j t - k) \overline{\varphi(2^j t - m)} dt = \delta_{km}.$$

Para establecer esta igualdad, basta hacer el cambio de variable $z = 2^j t$, para obtener

$$2^j \int_{-\infty}^{\infty} \varphi(2^j t - k) \overline{\varphi(2^j t - m)} dt = \int_{-\infty}^{\infty} \varphi(z - k) \overline{\varphi(z - m)} dz = \delta_{km},$$

en virtud de la propiedad 5 de la definición de AMR. \square

El siguiente lema contiene dos resultados utilizados en la existencia de los sistemas AMR, bajo hipótesis apropiadas. Por motivos de completitud se hará la prueba del primer apartado, la del segundo se puede encontrar en [57]. Recuerde que P_j es la proyección ortogonal sobre el espacio V_j .

Lema 2.3.4. *Para cualquier $f \in L_2(\mathbb{R})$,*

i) $\lim_{j \rightarrow -\infty} P_j f = 0.$

ii) $\lim_{j \rightarrow \infty} P_j f = f.$

Demostración. i) Puesto que $\|P_j\| = 1$, basta probar el resultado para funciones en $L_2(\mathbb{R})$ con soporte compacto. Si f tiene soporte en $[-a, a]$, entonces al aplicar las desigualdades de Cauchy-Schwarz y de Minkowski se tiene

$$\begin{aligned} \|P_j f\|_2^2 &= \left\| \sum_{k \in \mathbb{Z}} \langle f, \varphi_{jk} \rangle \varphi_{jk} \right\|_2^2 \\ &= \sum_{k \in \mathbb{Z}} |\langle f, \varphi_{jk} \rangle|^2 \\ &= \sum_{k \in \mathbb{Z}} \left| \int_{-a}^a f(t) 2^{j/2} \varphi(2^j t - k) dt \right|^2 \\ &\leq \sum_{k \in \mathbb{Z}} \left(\int_{-a}^a |f(t)|^2 dt \right) 2^{j/2} \left(\int_{-a}^a |\varphi(2^j t - k)| dt \right)^2 \\ &= \|f\|_2^2 \sum_{k \in \mathbb{Z}} \left(\int_{-2^j a - k}^{2^j a - k} |\varphi(z)| dz \right)^2. \end{aligned}$$

Si $2^j a < 1/2$, entonces estas integrales están definidas sobre intervalos ajenos cuya unión se escribe $\Omega_j = \cup_{k \in \mathbb{Z}} (-2^j a - k, 2^j a - k)$, con $\cap_j \Omega_j = \mathbb{Z}$, el cual tiene medida cero. Por tanto,

$$\|P_j f\|_2^2 \leq \|f\|_2^2 \int_{\Omega_j} |\varphi(z)|^2 dz \rightarrow 0, \quad j \rightarrow -\infty$$

por el teorema de la convergencia dominada de Lebesgue. \square

2.4. Ecuación de escala

Puesto que el conjunto $\{\varphi(\cdot - k)\}_{k \in \mathbb{Z}}$ constituye una base ortonormal de V_0 entonces cada $f \in V_0$ se puede expresar como

$$f = \sum_{n \in \mathbb{Z}} a_n \varphi_{0n}, \quad \varphi_{0n}(x) = \varphi(x - n).$$

Ahora, como $\varphi \in V_0$, y $V_0 \subset V_1$, se tiene entonces $\varphi \in V_1$. Pero la propiedad de dilatación implica que $\varphi(2^{-1}\cdot) \in V_0$. En consecuencia, se puede expandir

$$\varphi(2^{-1}t) = \sum_{n \in \mathbb{Z}} g_n \varphi(t - n), \quad t \in \mathbb{R},$$

para algunos coeficientes $(g_n)_{n \in \mathbb{Z}}$. O de manera equivalente,

$$\varphi(t) = \sum_{n \in \mathbb{Z}} g_n \varphi(2t - n), \quad t \in \mathbb{R}, \quad (2.4.1)$$

en donde las constantes de estructura (los (g_n)) satisfacen $\sum_{n \in \mathbb{Z}} |g_n|^2 < \infty$. La relación (2.4.1) se llama ecuación de escala. Los coeficientes g_n constituyen un filtro $\mathbf{g} = (g_n)_{n \in \mathbb{Z}}$ asociado a la función de escala.

Ejemplo 2.4.1. Si $\varphi(t) = \chi_{[0,1)}(t)$, entonces claramente

$$\varphi(t) = \varphi(2t) + \varphi(2t - 1)$$

es la ecuación de escala, con las constantes de estructura $g_0 = 1$, $g_1 = 1$ y $g_n = 0$, en otro caso.

A continuación se dan algunas propiedades de las constantes de estructura.

Proposición 2.4.1. *Los coeficientes de la ecuación de escala satisfacen las siguientes propiedades:*

$$g_n = 2 \int_{\mathbb{R}} \varphi(t) \overline{\varphi(2t - n)} dt, \quad n \in \mathbb{Z} \quad (2.4.2)$$

$$\sum_{k \in \mathbb{Z}} g_k \bar{g}_{2n+k} = 2\delta_{0n} \quad (\text{delta de Kronecker}). \quad (2.4.3)$$

$$\sum_{n \in \mathbb{Z}} |g_n|^2 = 2 \quad (2.4.4)$$

Si también $\varphi \in L_1(\mathbb{R})$, $\int_{\mathbb{R}} \varphi \neq 0$ y la ecuación (2.4.1) converge en $L_1(\mathbb{R})$, entonces

$$\sum_{n \in \mathbb{Z}} g_n = 2. \quad (2.4.5)$$

Demostración. Puesto que $g_n/\sqrt{2}$ son los coeficientes de Fourier de $\varphi \in V_1$ con respecto a la base ortonormal $\sqrt{2}\varphi(2t - n)$, se tiene

$$\frac{g_n}{\sqrt{2}} = \int_{\mathbb{R}} \varphi(t) \sqrt{2} \overline{\varphi(2t - n)} dt,$$

o lo que es lo mismo,

$$g_n = 2 \int_{\mathbb{R}} \varphi(t) \overline{\varphi(2t - n)} dt.$$

De la propiedad 5 de la definición de AMR se tiene

$$\int_{\mathbb{R}} \varphi(t - n) \overline{\varphi(t)} dt = \delta_{0n}.$$

Al sustituir (2.4.1) y aplicar la identidad de Parseval y la ortogonalidad se tiene

$$\begin{aligned} \delta_{0n} &= \sum_{k, m \in \mathbb{Z}} g_k \bar{g}_m \int_{\mathbb{R}} \varphi(2t - 2n - k) \overline{\varphi(2t - m)} dt \\ &= \frac{1}{2} \sum_{2n+k=m} g_k \bar{g}_m, \end{aligned}$$

lo cual es lo mismo que (2.4.3). En particular, si se toma $n = 0$ en la última expresión, se obtiene

$$\sum_{k=m} g_k \bar{g}_k = \sum_{k \in \mathbb{Z}} |g_k|^2 = 2.$$

Si, además, se tiene $\varphi \in L_1(\mathbb{R})$ con $\int_{\mathbb{R}} \varphi(t) dt \neq 0$, entonces al integrar (2.4.1) término a término se llega

$$\begin{aligned} \int_{\mathbb{R}} \varphi(t) dt &= \sum_{n \in \mathbb{Z}} g_n \int_{\mathbb{R}} \varphi(2t - n) dt \\ &= \frac{1}{2} \sum_{n \in \mathbb{Z}} g_n \int_{\mathbb{R}} \varphi(t) dt, \end{aligned}$$

al dividir por $\int_{\mathbb{R}} \varphi$ se obtiene $\sum_{n \in \mathbb{Z}} g_n = 2$. □

2.5. Construcción de la función de escala

Para construir la función de escala, es necesario encontrar los coeficientes g_n . Una forma de hacerlo es vía la transformada de Fourier, puesto que de manera directa es difícil (ver p. e., [15], [27]). En consecuencia, al aplicar la transformada de Fourier a la ecuación (2.4.1) se obtiene

$$\begin{aligned}\hat{\varphi}(\omega) &= \frac{1}{2} \sum_{n \in \mathbb{Z}} g_n e^{-in\omega/2} \hat{\varphi}\left(\frac{\omega}{2}\right) \\ &= \hat{\varphi}\left(\frac{\omega}{2}\right) P(e^{-i\omega/2})\end{aligned}\tag{2.5.1}$$

donde el polinomio P es dado por

$$P(z) = \frac{1}{2} \sum_{n \in \mathbb{Z}} g_n z^n.$$

Al iterar (2.5.1) se tiene

$$\begin{aligned}\hat{\varphi}(\omega) &= P(e^{-i\omega/2}) \hat{\varphi}\left(\frac{\omega}{2}\right) \\ \hat{\varphi}\left(\frac{\omega}{2}\right) &= P(e^{-i\omega/2^2}) \hat{\varphi}\left(\frac{\omega}{2^2}\right)\end{aligned}$$

$$\hat{\varphi}(\omega) = P(e^{-i\omega/2}) P(e^{-i\omega/2^2}) \hat{\varphi}\left(\frac{\omega}{2^2}\right).$$

Continuando de esta manera se obtiene

$$\begin{aligned}\hat{\varphi}(\omega) &= P(e^{-i\omega/2}) \dots P(e^{-i\omega/2^n}) \hat{\varphi}\left(\frac{\omega}{2^n}\right) \\ &= \left(\prod_{j=1}^n P(e^{-i\omega/2^j}) \right) \hat{\varphi}\left(\frac{\omega}{2^n}\right).\end{aligned}$$

Para una función de escala dada φ , la ecuación precedente se tiene para cada n . En el límite, cuando $n \rightarrow \infty$, la última ecuación se transforma

$$\hat{\varphi}(\omega) = \left(\prod_{j=1}^{\infty} P(e^{-i\omega/2^j}) \right) \hat{\varphi}(0).$$

Si φ satisface la condición de normalización $\int_{\mathbb{R}} \varphi = 1$, entonces $\hat{\varphi}(0) = 1$ y así,

$$\hat{\varphi}(\omega) = \prod_{j=1}^{\infty} P(e^{-i\omega/2^j}).\tag{2.5.2}$$

Por tanto, si el producto $\prod_{j=1}^{\infty} P(e^{-i\omega/2^j})$ converge, entonces la función de escala queda determinada salvo un factor no nulo $\hat{\varphi}(0)$, que es su media. En consecuencia, la única función de escala asociada al filtro \mathbf{g} está dada por (2.5.2). Es decir, si la función P asociada al filtro \mathbf{g} cumple cierta propiedad de convergencia, entonces se tiene $\hat{\varphi}$ y, antitransformando, se obtiene φ . En resumen, se tiene

Proposición 2.5.1. *Sea \mathbf{g} un filtro y P el polinomio dado por*

$$P(z) = \frac{1}{2} \sum_{n \in \mathbb{Z}} g_n z^n.$$

Si la función Φ definida por

$$\Phi(\omega) = \lim_{n \rightarrow \infty} \prod_{j=1}^n P(e^{-i\omega/2^j})$$

está en $L_2(\mathbb{R})$ y verifica $\lim_{|\omega| \rightarrow \infty} \Phi(\omega) = 0$. Entonces existe una función de escala φ asociada al filtro \mathbf{g} y determinada por $\hat{\varphi} = \Phi$ con $\int \varphi = 1$.

La siguiente proposición permite dar condiciones sobre la ortonormalidad de la base $\{\varphi_{0k}\}_{k \in \mathbb{Z}}$ en términos de los coeficientes g_k .

Proposición 2.5.2. *El sistema $\{\varphi_{0k}\}_{k \in \mathbb{Z}}$ es ortonormal si y sólo si la transformada de Fourier de φ satisface*

$$2\pi \sum_{k \in \mathbb{Z}} |\hat{\varphi}(\omega + 2k\pi)|^2 = 1.$$

Demostración. Como $\varphi(t - k)$ forma una base ortonormal en V_0 , entonces al aplicar el teorema de Plancherel (ver p.e., [39]) se tiene

$$\begin{aligned} \delta_{0m} &= \int_{\mathbb{R}} \varphi(t) \overline{\varphi(t - m)} dt \\ &= \int_{\mathbb{R}} \hat{\varphi}(\omega) \overline{\hat{\varphi}(\omega) e^{-im\omega}} d\omega \\ &= \int_{\mathbb{R}} |\hat{\varphi}(\omega)|^2 e^{im\omega} d\omega \\ &= \sum_{k \in \mathbb{Z}} \int_{2\pi k}^{2\pi(k+1)} |\hat{\varphi}(\omega)|^2 e^{im\omega} d\omega \\ &= \int_0^{2\pi} e^{im\omega} \left(\sum_{k \in \mathbb{Z}} |\hat{\varphi}(\omega + 2k\pi)|^2 \right) d\omega. \end{aligned}$$

Sea $F(\omega) = 2\pi \sum_{k \in \mathbb{Z}} |\hat{\varphi}(\omega + 2k\pi)|^2$, entonces

$$\frac{1}{2\pi} \int_0^{2\pi} e^{im\omega} F(\omega) d\omega = \delta_{0m}. \quad (*)$$

La función F es 2π -periódica ya que

$$\begin{aligned} F(\omega + 2\pi) &= 2\pi \sum_{k \in \mathbb{Z}} |\hat{\varphi}(\omega + 2\pi(k+1))|^2 \\ &= 2\pi \sum_{j \in \mathbb{Z}} |\hat{\varphi}(\omega + 2\pi j)|^2 = F(\omega). \end{aligned}$$

Como F es periódica, su serie de Fourier, $\sum_{m \in \mathbb{Z}} c_m e^{imt}$, donde los coeficientes de Fourier son dados por $c_m = \frac{1}{2\pi} \int_0^{2\pi} F(\omega) e^{-im\omega} d\omega$. Por tanto, la condición de ortonormalidad (*), es equivalente a $c_{-m} = \delta_{m0}$, lo cual a su vez es equivalente a $F(\omega) = 1$. \square

Como consecuencia de este resultado se tiene la siguiente condición necesaria sobre el polinomio $P(z)$ para la existencia de un AMR.

Corolario 2.5.3. *El polinomio $P(z) = \sum_{n \in \mathbb{Z}} g_n z^n$ satisface*

$$|P(e^{-it})|^2 + |P(e^{-i(t+\pi)})|^2 = 1, \quad 0 \leq t \leq 2\pi. \quad (2.5.3)$$

Demostración. De los resultados anteriores se tiene

$$\begin{aligned} \sum_{n \in \mathbb{Z}} |\hat{\varphi}(\omega + 2n\pi)|^2 &= \frac{1}{2\pi} \quad \text{y} \quad \hat{\varphi}(\omega) = P(e^{-i\omega/2}) \hat{\varphi}\left(\frac{\omega}{2}\right). \\ \frac{1}{2\pi} &= \sum_{n \in \mathbb{Z}} |\hat{\varphi}(\omega + 2n\pi)|^2 = \sum_{n \text{ par}} + \sum_{n \text{ impar}} \\ &= \sum_{k \in \mathbb{Z}} |\hat{\varphi}(\omega + (2k)2\pi)|^2 + \sum_{k \in \mathbb{Z}} |\hat{\varphi}(\omega + (2k+1)2\pi)|^2 \\ &= \sum_{k \in \mathbb{Z}} \left(|P(e^{-i(\frac{\omega}{2} + 2k\pi)})|^2 \left| \hat{\varphi}\left(\frac{\omega}{2} + 2k\pi\right) \right|^2 \right. \\ &\quad \left. + |P(e^{-i(\frac{\omega}{2} + (2k+1)\pi)})|^2 \left| \hat{\varphi}\left(\frac{\omega}{2} + (2k+1)\pi\right) \right|^2 \right) \\ &= |P(e^{-i\frac{\omega}{2}})|^2 \sum_{k \in \mathbb{Z}} \left| \hat{\varphi}\left(\frac{\omega}{2} + 2k\pi\right) \right|^2 \\ &\quad + |P(-e^{-i\frac{\omega}{2}})|^2 \sum_{k \in \mathbb{Z}} \left| \hat{\varphi}\left(\left(\frac{\omega}{2} + \pi\right) + 2k\pi\right) \right|^2 \\ &= |P(e^{-i\frac{\omega}{2}})|^2 \frac{1}{2\pi} + |P(-e^{-i\frac{\omega}{2}})|^2 \frac{1}{2\pi}. \end{aligned}$$

En consecuencia,

$$1 = |P(e^{-i\frac{\omega}{2}})|^2 + |P(-e^{-i\frac{\omega}{2}})|^2.$$

□

Corolario 2.5.4. *En términos de g_k , (2.5.3) se transforma en*

$$\sum_{n \in \mathbb{Z}} g_{n-2k} \bar{g}_{n-2m} = 2\delta_{k-m,0}.$$

Demostración. Observe que (2.5.3) en términos de los coeficientes g_n está dado por

$$\frac{1}{4} \sum_{n,k \in \mathbb{Z}} g_n \bar{g}_k e^{-i(n-k)t} + \frac{1}{4} \sum_{n,k \in \mathbb{Z}} g_n \bar{g}_k (-1)^{n-k} e^{-i(n-k)t} = 1.$$

Los términos impares se cancelan y por lo tanto se tiene

$$\sum_{n-k \text{ par}} g_n \bar{g}_k e^{-i(n-k)t} = \sum_{j \in \mathbb{Z}} c_j e^{-2ijt} = 2$$

donde

$$c_j = \sum_{n-k=2j} g_n \bar{g}_k.$$

Esto es válido para todo t , luego $c_j = 2\delta_j$. Por tanto

$$\sum_{n \in \mathbb{Z}} g_n \bar{g}_{n-2j} = 2\delta_j.$$

De hecho, esto es equivalente con la afirmación a probar. □

2.6. Descomposición y reconstrucción

En esta sección se describirán algoritmos de descomposición y reconstrucción asociados a un AMR. Estos algoritmos se utilizarán junto con el análisis multirresolución en la descomposición y reconstrucción de señales en donde tanto la función de escala como la wavelet son funciones continuas.

2.6.1. Algoritmo de descomposición

Sean $c_{j,k}$ y $d_{j,k}$ los coeficientes de la función de escala φ y de la wavelet ψ , respectivamente, para $j, k \in \mathbb{Z}$, definidos por

$$c_{j,k} := \int_{\mathbb{R}} f(t) \varphi_{jk}(t) dt \tag{2.6.1}$$

$$d_{j,k} := \int_{\mathbb{R}} f(t)\psi_{jk}(t)dt, \quad (2.6.2)$$

donde

$$\varphi_{jk}(t) = 2^{j/2}\varphi(2^j t - k), \quad \psi_{jk}(t) = 2^{j/2}\psi(2^j t - k).$$

φ y ψ son, respectivamente, la función de escala y la wavelet madre.

Ahora bien, como $\varphi_{jk}(t) = 2^{j/2}\varphi(2^j t - k)$, entonces existe h_m tal que

$$\begin{aligned} \varphi_{jk}(t) &= \sum_{m \in \mathbb{Z}} h_m 2^{j/2} \varphi_{1m}(2^j t - k) \\ &= \sum_{m \in \mathbb{Z}} h_m 2^{(j+1)/2} \varphi(2^{j+1} t - 2k - m) \\ &= \sum_{m \in \mathbb{Z}} h_m \varphi_{j+1, m+2k}(t) \\ &= \sum_{m \in \mathbb{Z}} h_{m-2k} \varphi_{j+1, m}(t). \end{aligned} \quad (2.6.3)$$

Reemplazando este valor en (2.6.1), se obtiene

$$\begin{aligned} c_{j,k} &= \int_{\mathbb{R}} f(t) \sum_{m \in \mathbb{Z}} h_{m-2k} \varphi_{j+1, m}(t) dt \\ &= \sum_{m \in \mathbb{Z}} h_{m-2k} \int_{\mathbb{R}} f(t) \varphi_{j+1, m}(t) dt \\ &= \sum_{m \in \mathbb{Z}} h_{m-2k} c_{j+1, m}, \end{aligned}$$

por tanto,

$$c_{j,k} = \sum_{m \in \mathbb{Z}} h_{m-2k} c_{j+1, m}. \quad (2.6.4)$$

Como $V_0 \subset V_1$, para cada $\varphi \in V_0$ también se satisface $\varphi \in V_1$. Además, $\{\varphi_{1k}, k \in \mathbb{Z}\}$ es una base ortonormal para V_1 , entonces existe una sucesión $(h_k) \in \ell_2(\mathbb{Z})$ tal que

$$\varphi(t) = \sum_{k \in \mathbb{Z}} h_k \varphi_{1k}(t), \quad (2.6.5)$$

por tanto, los elementos de la sucesión se puede escribir como

$$h_k = \langle \varphi, \varphi_{1k} \rangle \quad \text{y} \quad (h_k) \in \ell_2.$$

La ecuación (2.6.5) relaciona funciones con diferentes factores de escala, también se conoce como ecuación de dilatación. Para la base de Haar se tiene

$$h_k = \begin{cases} 1/\sqrt{2}, & k = 0, 1 \\ 0, & \text{en otro caso.} \end{cases}$$

Si φ es la función de escala de un AMR, entonces la wavelet madre ψ se relaciona con φ por medio de la ecuación

$$\psi(t) = \sum_{k \in \mathbb{Z}} (-1)^k h_{1-k} \varphi_{1k}(t). \tag{2.6.6}$$

Al sustituir (2.6.6) en (2.6.2) se obtiene

$$d_{j,k} = \sum_{p \in \mathbb{Z}} (-1)^p h_{1-p+2k} c_{j+1,p}. \tag{2.6.7}$$

Si los coeficientes de escala en cualquier nivel j son dados, entonces todos los coeficientes de la función escala de nivel inferior para $J < j$, se pueden calcular recursivamente usando la ecuación (2.6.4), mientras que todos los coeficientes wavelet de nivel inferior ($J < j$) se calculan aplicando (2.6.7).

Si $c_{j,\cdot}$ y $d_{j,\cdot}$ representan los coeficientes de la función de escala y wavelet en el nivel j , respectivamente, la Figura 2.6.1 representa el algoritmo de descomposición en forma esquemática. Por ejemplo, la flecha que relaciona los coeficientes c_{j-1} y c_{j-2} , indica que c_{j-2} se calcula sólo usando c_{j-1} .

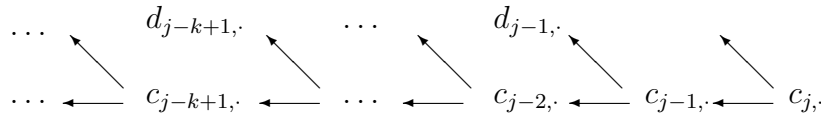


Figura 2.6.1

Observe que las fórmulas (2.6.4) y (2.6.7) comparten un hecho interesante, esto es, en cada ecuación, si el índice de dilatación k se incrementa en uno, todos los índices de (h_m) se desplazan en dos unidades; lo cual significa que si existe solamente un número finito de términos no nulos en la sucesión (h_m) , entonces aplicando el algoritmo de descomposición a un conjunto de coeficientes de escala no nulos en el nivel $j + 1$, se obtendrá sólo la mitad de coeficientes no nulos en el nivel j . Este proceso en teoría de señales se conoce como *downsampling*. Un resultado análogo se tiene para los coeficientes wavelet.

Para expresar lo anterior en la terminología de filtros, recuerde que la convolución de dos sucesiones en $\ell_2(\mathbb{Z})$

$$x = (\dots, x_{-1}, x_0, x_1, \dots) \quad \text{y} \quad y = (\dots, y_{-1}, y_0, y_1, \dots)$$

se define por

$$(x * y)_m := \sum_{k \in \mathbb{Z}} x_k y_{m-k}.$$

En consecuencia, (2.6.4) se puede expresar como

$$c_{j-1,k} = \sum_{m \in \mathbb{Z}} \tilde{h}_{2k-m} c_{j,m} = (\tilde{h} * c_j)_{2k}, \quad (2.6.8)$$

note que se reemplazó j por $j - 1$ y para simplificar se utilizó la notación $\tilde{y}_m = y_{-m}$. Si se define el operador downsampling para la sucesión x como

$$((\downarrow 2)x)_k := x_{2k}, \quad k \in \mathbb{Z},$$

entonces (2.6.8) se puede escribir

$$c_{j-1,\cdot} = (\downarrow 2)(\tilde{h} * c_j). \quad (2.6.9)$$

De un procedimiento similar se obtiene, con $g_m = (-1)^m h_{1-m}$,

$$d_{j-1,\cdot} = (\downarrow 2)(\tilde{g} * c_j). \quad (2.6.10)$$

Al algoritmo de descomposición, Mallat lo llamó *algoritmo piramidal* [31], mientras Daubechies lo llamó *algoritmo de cascada* [15].

2.6.2. Algoritmo de reconstrucción

Recuerde que dado un AMR, el conjunto de funciones linealmente independientes φ_{jk} que generan a V_j son las funciones de escala, mientras que el conjunto de funciones linealmente independientes ψ_{jk} que generan a W_j son las wavelets. En otras palabras, $\{\varphi_{jk}\}_{k \in \mathbb{Z}}$ y $\{\psi_{jk}\}_{k \in \mathbb{Z}}$ son generadas, respectivamente, por φ y ψ , esto es,

$$\varphi_{jk}(t) = 2^{j/2} \varphi(2^j t - k) \quad \text{y} \quad \psi_{jk}(t) = 2^{j/2} \psi(2^j t - k), \quad \forall k \in \mathbb{Z}$$

forman las bases ortonormales para V_j y W_j , respectivamente. Definiendo

$$a_{2k} = \langle \varphi_{10}, \varphi_{0k} \rangle, \quad a_{2k-1} = \langle \varphi_{11}, \varphi_{0k} \rangle$$

$$b_{2k} = \langle \varphi_{10}, \psi_{0k} \rangle, \quad b_{2k-1} = \langle \varphi_{11}, \psi_{0k} \rangle,$$

donde $a_k = h_{-k}$ y $b_k = (-1)^k h_{k+1}$. Entonces

$$\begin{aligned} c_{j,k} &= \sum_{m \in \mathbb{Z}} a_{2m-k} c_{j-1,m} + b_{2m-k} d_{j-1,m} \\ &= \sum_{m \in \mathbb{Z}} h_{k-2m} c_{j-1,m} + (-1)^k h_{2m-k+1} d_{j-1,m}. \end{aligned} \quad (2.6.11)$$

Observe que esta última expresión es casi la suma de dos convoluciones. La diferencia está en que el índice para la convolución es $k - m$ mientras acá aparece $k - 2m$. En otras palabras, (2.6.11) es una convolución pero sin los términos impares (falta $h_{k-(2m-1)}$). Para que (2.6.11) sea una convolución, se altera la sucesión original intercalando ceros entre sus componentes y obteniendo una nueva sucesión que contiene ceros en todas sus entradas impares. Este procedimiento se llama *upsampling*, denotado por $(\uparrow 2)$. Más explícitamente, si $x = (\dots, x_{-2}, x_{-1}, x_0, x_1, x_2, \dots)$, entonces

$$((\uparrow 2)x)_k = (\dots, x_{-2}, 0, x_{-1}, 0, x_0, 0, x_1, 0, x_2, \dots)$$

o de manera equivalente,

$$((\uparrow 2)x)_k = \begin{cases} x_{k/2}, & \text{si } k \text{ es par,} \\ 0, & \text{si } k \text{ es impar.} \end{cases}$$

En consecuencia,

$$c_{j,k} = (((\uparrow 2)c_{j-1}) * h)_k + (((\uparrow 2)d_{j-1}) * g)_k. \quad (2.6.12)$$

La Figura 2.6.2 representa el algoritmo de reconstrucción en forma esquemática.

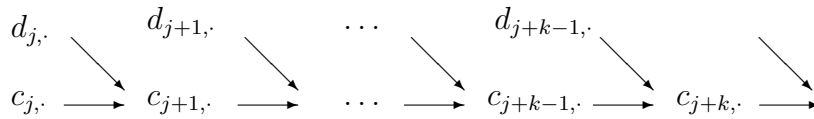


Figura 2.6.2

CAPÍTULO 3

Compresión de imágenes usando wavelets

El propósito de este capítulo es presentar algunos conceptos básicos en el procesamiento de señales. En particular en la compresión de imágenes, la cual se puede mirar como un proceso de compresión de datos, en donde los datos comprimidos siempre son representaciones digitales de señales bidimensionales. La compresión de imágenes se puede situar en el contexto más general de la teoría de la información y de la codificación en donde muchos problemas de comunicación digital, la información se transmite en forma de cadenas de ceros y unos a través de un canal en el cual la transmisión puede producir perturbaciones debidas a interferencias climatológicas, problemas eléctricos, error humano, deterioro de una cinta o disco, que podrían causar que un cero se reciba como uno o viceversa, estas perturbaciones, en general, se llaman ruido. En estas circunstancias, las transformaciones erróneas de los dígitos en una palabra enviada, pueden conllevar a que una palabra recibida sea diferente a la que se envió.

El interés por la teoría de comunicación, desde la compresión de datos se centra principalmente en la codificación eficiente de la información en ausencia del ruido. Sin embargo, existen muchas aplicaciones, por ejemplo en medicina o astronomía, en donde no es aceptable ningún deterioro de la imagen porque toda la información contenida, incluso la que se estima como ruido, se considera imprescindible.

En varias áreas de trabajo, las necesidades de compresión provienen principalmente de problemas de transmisión y manipulación de información, y en menos ocasiones por problemas de almacenamiento. Es evidente que las imágenes, incluso comprimidas, requieren grandes cantidades de memoria.

Debido a este hecho, existen básicamente dos tipos de compresores de imágenes: compresores sin pérdida (lossless compressors) y, compresores con pérdida (lossy compressors). Estos últimos, a costa de eliminar la información menos relevante para el observador, alcanzan cotas de compresión muy eficientes. A este tipo de compresores pertenece el estándar JPEG (Joint Photographic Experts Group) [12], [24], [36], el cual usa la transformada discreta coseno, mientras la nueva versión JPEG2000 está completamente basada en transformada wavelet [56].

3.1. El problema de la compresión de imágenes

Así como las señales son tratadas como funciones de una variable, las imágenes se representarán por funciones de dos variables. Por lo tanto, dada una imagen f , que pertenece a un espacio de Hilbert \mathcal{H} , el problema de la compresión de imágenes es el de encontrar una representación aproximada \tilde{f} de f con las siguientes tres condiciones:

- a) La señal \tilde{f} es dada por un número fijo de coeficientes N
- b) Existe un control conocido del error $E(N) := \|f - \tilde{f}\|$
- c) Existe un algoritmo rápido que reproduce a \tilde{f} .

El objetivo de un algoritmo de compresión (procedimiento o método) es el de representar ciertas imágenes con menor información que la usada en la representación de la imagen original. Para un algoritmo sin pérdida, la imagen original y la comprimida será la misma, y el error entre ellas será cero; mientras un algoritmo con pérdida, la imagen original difiere de la comprimida. Este tipo de algoritmo es de suma importancia en la minimización de transmisión de datos o almacenamiento de información.

Otro hecho importante es el estudio del error, es decir, encontrar una base $\{e_j : j \in \mathbb{Z}\}$ en donde la señal procesada \tilde{f} se pueda expresar como una combinación lineal finita de la forma $\sum_{j=1}^N \langle f, e_j \rangle e_j$ de tal manera que el error de aproximación sea lo más pequeño posible, esto es, $E(N) = O(N^{-\alpha})$. Existen dos formas estándar de construir tal aproximación: la lineal y la no lineal y las representaciones más usuales para \tilde{f} son la serie de Fourier o la serie wavelet. Los espacios más usados, en donde se definirá la métrica o norma que mide el error, son los espacios de Sobolev y Besov.

Finalmente, otro punto de interés en esta teoría, es entender la interrelación entre la calidad de la imagen comprimida y el número de

coeficientes empleados. El propósito de esta sección es estudiar las dos primeras condiciones (a y b), siguiendo los artículos [7], [18] y [35]. El estudio del algoritmo se deja para la última sección.

3.1.1. Formulación matemática

Una imagen se puede considerar como una función f definida sobre el cuadrado unitario $Q = [0, 1] \times [0, 1]$. Se verá que el procesamiento de señales está estrechamente ligado con el análisis wavelet. El concepto de wavelets en dos dimensiones es relevante en este estudio.

Sea φ la función de escala y $\psi(x)$ la correspondiente wavelet madre, entonces las tres funciones

$$\begin{aligned}\psi_1(x, y) &= \psi(x)\psi(y) \\ \psi_2(x, y) &= \psi(x)\varphi(y) \\ \psi_3(x, y) &= \varphi(x)\psi(y)\end{aligned}\tag{3.1.1}$$

forma, por dilatación y traslación, una base ortonormal para $L_2(\mathbb{R}^2)$; esto es,

$$\{2^{j/2}\psi_m(2^j x - k_1, 2^j y - k_2)\}, \quad j \in \mathbb{Z}, \quad k = (k_1, k_2) \in \mathbb{Z}^2,$$

$m = 1, 2, 3$ es una base ortonormal para $L_2(\mathbb{R}^2)$.

Por tanto, cada $f \in L_2(\mathbb{R}^2)$ se puede representar como

$$f = \sum_{j,k \in \mathbb{Z}} d_{j,k} \psi_{jk},\tag{3.1.2}$$

donde ψ es cualquiera de las tres funciones $\psi_m(x, y)$ y $d_{j,k} = \langle f, \psi_{jk} \rangle$.

Un modelo matemático natural en el procesamiento de señales, es considerar una señal f como un elemento de un espacio de Hilbert \mathcal{H} . Usualmente, \mathcal{H} se toma como $L_2(\mathbb{R}^d)$, $L_2([0, N]^d)$ o $\ell_2(N^d)$, dependiendo de la aplicación en estudio. Por ejemplo, las imágenes de la vida real se pueden ver como funciones $f(x, y)$ que corresponden a la intensidad del campo de luz en $Q = [0, 1] \times [0, 1]$. Se explicará brevemente como este tipo de funciones se pueden discretizar con el propósito de manipulación y almacenamiento en el computador [7].

Es de notar que la función intensidad $f(x, y)$ asociada con la imagen no puede, en general, medirse en todos los puntos $0 \leq x, y \leq 1$. En la práctica, un dispositivo (un fotómetro) mide promedios de intensidad de luz sobre pequeños cuadrados que se llaman pixels (o elementos de imagen) distribuidos diádicamente a lo largo de toda la imagen en Q . Para m grande (usualmente,

$m \geq 8$), se puede codificar la imagen como una sucesión de 2^{2m} coeficientes

$$p_j = p_j^{(m)} = \frac{1}{|Q_{m,j}|} \iint_{Q_{m,j}} f(x,y) dx dy, \quad 0 \leq j_1, j_2 < 2^m, \quad (3.1.3)$$

donde $j = (j_1, j_2)$ y $Q_{m,j}$ (el j -ésimo pixel) denota el cuadrado diádico $[\frac{j_1}{2^m}, \frac{j_1+1}{2^m}] \times [\frac{j_2}{2^m}, \frac{j_2+1}{2^m}]$. A cada uno de estos cuadrados se le asocia el número p_j (usualmente entre 0 y 2^8), el cual representa la escala de gris de la imagen en ese punto. De esta manera la imagen queda digitalizada y se puede almacenar y procesar por computador.

Observe que hasta el momento no se ha definido una función f que represente a la imagen como una colección de pixels. Esto se puede hacer considerando una sucesión $\{\varphi_{mj}\}$ de funciones de valor real con soporte en $Q_{m,j}$ y construyendo la *imagen observada*

$$f_o(x) = \sum_j p_j \varphi_{mj}(x), \quad x \in \mathbb{R}^2. \quad (3.1.4)$$

Usualmente, $\varphi_{mj}(x) = \varphi(2^m x - j)$, para una función fija φ , la cual se puede escoger simplemente como χ_Q (la característica de Q), o versiones suavizadas tales como splines o funciones de escala. En el caso de escoger una wavelet, por ejemplo, las wavelets de Haar en el cuadrado Q , y denotando la función característica de Q por $\chi = \chi_Q$ y la función característica normalizada de $Q_{m,j}$ por

$$\chi_{mj} = 2^m \chi_{Q_{m,j}} = 2^m \chi(2^m x - j), \quad x \in \mathbb{R}^2,$$

entonces (3.1.3) se puede expresar como

$$\begin{aligned} p_j &= 2^m \int_{Q_{m,j}} \chi(2^m x - j) f(x) dx \\ &= 2^m \langle \chi_{mj}, f \rangle. \end{aligned}$$

De esta manera, la imagen observada (3.1.4) queda expresada como

$$\begin{aligned} f_o(x) &= \sum_j p_j \chi(2^m x - j) \\ &= \sum_j \langle \chi_{mj}, f \rangle \chi_{mj}. \end{aligned}$$

Por tanto, si la expansión en wavelet del campo de intensidad f es

$$f = \sum_{k \geq 0} \sum_j d_{j,k} \psi_{jk}, \quad (3.1.5)$$

entonces la expansión en wavelet de la imagen f_o es

$$f_o = \sum_{0 \leq k < m} \sum_j d_{j,k} \psi_{jk}.$$

Observe que f_o es la proyección de f sobre $\text{gen}\{\chi_{mj}\}_j = \text{gen}\{\psi_{jk}\}_{0 \leq k < m, j}$ en $L_2(Q)$. Note también que para m suficientemente grande, f_o es prácticamente una copia indistinguible de la imagen original f . El problema de la compresión de imágenes es entonces representar f_o con una cantidad suficiente de coeficientes para no perder la semejanza o parecido visual de la imagen original.

En el apartado que sigue se presentarán las dos técnicas de aproximación utilizando bases wavelet, y se estudiará el problema del error en este caso.

3.1.2. Compresión lineal y no lineal

Considere el espacio de Hilbert \mathcal{H} . Sea $\{e_k\}_{k=1}^{\infty}$ una base ortonormal fija. Entonces, para cada $f \in \mathcal{H}$ se tiene

$$f = \sum_{k=1}^{\infty} c_k e_k,$$

donde $c_k = \langle f, e_k \rangle$ son los coeficientes base. Hay dos maneras de construir y aproximar una señal f_N con sólo N coeficientes base.

1. El método de aproximación lineal: corresponde a minimizar el funcional

$$\left\| f - \sum_{k=1}^N a_k e_k \right\|, \quad \text{para todo } a_1, \dots, a_N \in \mathbb{C}.$$

Para la base $\{e_k\}$ el mínimo es dado por

$$f_N = \sum_{k=1}^N \langle f, e_k \rangle e_k,$$

es decir, f_N es la proyección ortogonal de f sobre $V_N := \text{gen}\{e_1, \dots, e_N\}$.

2. El método de aproximación no lineal: corresponde a minimizar

$$\left\| f - \sum_{\gamma \in A} a_{\gamma} e_{\gamma} \right\|, \quad \text{para todo } a_{\gamma} \in \mathbb{C}$$

y todo conjunto de índices $A \subset \mathbb{N}$ tal que $\text{Car}A \leq N$. Cuando $\{e_k\}$ es una base ortonormal, el mínimo es dado por

$$f_N = \sum_{r=1}^N \langle f, e_{k_r} \rangle e_{k_r},$$

donde los coeficientes se han ordenado en forma no creciente

$$|\langle f, e_{k_1} \rangle| \geq |\langle f, e_{k_2} \rangle| \geq \dots$$

A manera de motivación, considere el espacio de Hilbert $\mathcal{H} = L_2[0, b]$ y su base de Fourier asociada

$$e_n(t) = \frac{1}{\sqrt{b}} e^{2\pi i n t / b}, \quad n \in \mathbb{Z}.$$

Esta es una base ortonormal para $L_2[0, b]$, así que cualquier función f se puede representar en términos de su serie de Fourier

$$f(t) = \sum_{n \in \mathbb{Z}} \langle f, e_n \rangle e_n(t) = \sum_{n \in \mathbb{Z}} \frac{1}{\sqrt{b}} \hat{f}\left(\frac{2\pi n}{b}\right) e_n(t),$$

con la convergencia en el sentido de L_2 . La última identidad tiene sentido cuando f se extiende a cero fuera de $[0, b]$ y

$$\hat{f}(\omega) = \int_{\mathbb{R}} f(t) e^{-i\omega t} dt$$

es la transformada de Fourier de f . La mejor aproximación lineal para f con N coeficientes está dada por

$$f_N(t) = \sum_{|n| \leq N} \frac{1}{\sqrt{b}} \hat{f}\left(\frac{2\pi n}{b}\right) e_n(t),$$

y el error de aproximación es

$$E(N) = \|f - f_N\| = \left(\frac{1}{b} \sum_{|n| > N} \left| \hat{f}\left(\frac{2\pi n}{b}\right) \right|^2 \right)^{1/2}.$$

Ahora, la caracterización del conjunto de todas las señales $f \in L_2[0, b]$ para el cual el error de aproximación decae conforme $E(N) = O(N^{-\alpha})$, se hace teniendo en cuenta que la pequeñez de $E(N)$ está estrechamente relacionada al decaimiento rápido de $\hat{f}(\omega)$ cuando $|\omega| \rightarrow \infty$. Este decaimiento asintótico es de hecho equivalente a la regularidad de la función f , en virtud de la bien conocida identidad

$$\widehat{f^{(k)}}(\omega) = \int_{\mathbb{R}} f^{(k)}(t) e^{-i\omega t} dt = (i\omega)^k \hat{f}(\omega).$$

Estos hechos conducen a un candidato natural para el estudio de la aproximación lineal con Fourier: la familia de espacios de Sobolev (véase por ejemplo [21]).

Para $s \in \mathbb{R}$, el espacio de Sobolev $H^s(\mathbb{R})$ se define por

$$H^s(\mathbb{R}) := \left\{ f \in L_2(\mathbb{R}) : \int_{\mathbb{R}} (1 + |x|^2)^s |\hat{f}(x)|^2 dx < \infty \right\},$$

donde \hat{f} es la transformada de Fourier de f . $H^s(\mathbb{R})$ equipado con el producto interno

$$\langle f, g \rangle_{H^s} := \int_{\mathbb{R}} (1 + |x|^2)^s \hat{f}(x) \overline{\hat{g}(x)} dx$$

es un espacio de Hilbert. También se le asocia la norma

$$\|f\|_{H^s} := \left(\int_{\mathbb{R}} (1 + |x|^2)^s |\hat{f}(x)|^2 dx \right)^{1/2}.$$

En consecuencia, si $s > 0$ y $f \in L_2[0, b]$ y es de soporte compacto, entonces

$$f \in H^s(\mathbb{R}) \iff \sum_{N=1}^{\infty} N^{2s-1} E^2(N) < \infty.$$

En efecto,

$$\begin{aligned} \sum_{N=1}^{\infty} N^{2s-1} E^2(N) &= \sum_{N=1}^{\infty} N^{2s-1} \left(\frac{1}{b} \sum_{|n|>N} \left| \hat{f}\left(\frac{2\pi n}{b}\right) \right|^2 \right) \\ &= \frac{1}{b} \sum_{n \in \mathbb{Z}} \left(\sum_{N=1}^{|n|} N^{2s-1} \right) \left| \hat{f}\left(\frac{2\pi n}{b}\right) \right|^2 \\ &= \frac{C_s}{b} \sum_{n \in \mathbb{Z}} |n|^{2s} \left| \hat{f}\left(\frac{2\pi n}{b}\right) \right|^2 \\ &\approx \int_{\mathbb{R}} (1 + |\omega|^2)^s |\hat{f}(\omega)|^2 d\omega < \infty. \end{aligned}$$

En el último paso se utilizó el hecho plausible de ver la integral como una suma de Riemann.

Se puede concluir de lo anterior que la aproximación lineal con Fourier es muy buena en el análisis de señales con suavidad uniforme en todos los puntos $t \in \mathbb{R}$. Sin embargo, en el modelo de imágenes es mala, puesto que una sola discontinuidad en un punto se tornará en una baja sensible en la

suavidad global [33]. Nótese también que de la razón del decaimiento del error de aproximación, se puede estimar numéricamente el mejor s para el cual $f \in H^s$.

Para desarrollar la aproximación tanto lineal como no lineal usando wavelet, es necesario introducir algo de terminología matemática como la noción de espacio de Besov [19].

Dado $\alpha > 0$, $0 < p \leq \infty$, $0 < q \leq \infty$, y sea $r \in \mathbb{Z}$ con $q > \alpha \geq r - 1$. El espacio de Besov, $B_q^{\alpha,r}(L_p(Q))$, consiste de las funciones f para las cuales la norma $\|f\|_{B_q^{\alpha,r}(L_p(Q))}$ definida por

$$\|f\|_{B_q^{\alpha,r}(L_p(Q))} = \|f\|_{L_p(Q)} + \left(\int_0^\infty \left[\frac{\omega_r(f,t)_p}{t^\alpha} \right]^q \frac{dt}{t} \right)^{1/q} < \infty, \quad q < \infty$$

y

$$\|f\|_{B_\infty^{\alpha,r}(L_p(Q))} = \|f\|_{L_p(Q)} + \sup_{t>0} \left[\frac{\omega_r(f,t)_p}{t^\alpha} \right] < \infty, \quad q = \infty,$$

donde para cualquier $h \in \mathbb{R}^2$,

$$\omega_r(f,t)_p = \sup_{|h| \leq t} \left(\int_{Q_{rh}} |\Delta_h^r f(x)|^p dx \right)^{1/p}, \quad 0 \leq p \leq \infty,$$

es el r -módulo de continuidad en L_p , con el cambio usual del supremo esencial cuando $p = \infty$. Si se define $\Delta_h^0 f(x) = f(x)$, entonces

$$\Delta_h^r f(x) = \Delta_h^{r-1} f(x+h) - \Delta_h^{r-1} f(x), \quad r = 1, 2, \dots,$$

es la r -ésima diferencia hacia adelante de f , $Q_{rh} = \{x \in Q : x + rh \in Q\}$; recuerde que $Q = [0, 1] \times [0, 1]$.

Observación 3.1.1. 1) Si $0 < p < 1$ o $0 < q < 1$, entonces $\|\cdot\|_{B_q^{\alpha,r}(L_p(Q))}$ no satisface la desigualdad triangular. Si embargo, existe una constante C tal que para todo $f, g \in B_q^{\alpha,r}(L_p(Q))$,

$$\|f + g\|_{B_q^{\alpha,r}(L_p(Q))} \leq C(\|f\|_{B_q^{\alpha,r}(L_p(Q))} + \|g\|_{B_q^{\alpha,r}(L_p(Q))}).$$

2) Como, para cualquier $r > \alpha$, $r' > r$, $\|f\|_{B_q^{\alpha,r}(L_p(Q))}$ y $\|f\|_{B_q^{\alpha,r'}(L_p(Q))}$ son normas equivalentes, se denotará el espacio de Besov por $B_q^\alpha(L_p(Q))$ en vez de $B_q^{\alpha,r}(L_p(Q))$ para todo $r > \alpha$.

3) Para $p = q = 2$, $B_2^\alpha(L_2(Q))$ es el espacio de Sobolev $H^\alpha(L_2(Q))$.

4) Para $\alpha < 1$, $1 \leq p < \infty$, y $q = \infty$, $B_\infty^\alpha(L_p(Q))$ es el espacio de Lipschitz

$$\text{Lip}(\alpha, L_p(Q)) = \{f \in L_p(Q) : \|f(x+h) - f(x)\|_{L_p} \leq k|h|^\alpha, \quad k > 0 \text{ cte}\}.$$

5) $\|f\|_{B_2^\alpha(L_p(Q))}$ es equivalente a la norma $\left(\sum_k \sum_j 2^{\alpha k} |d_{j,k}|^q\right)^{1/q}$, donde $d_{j,k}$ son los coeficientes wavelet de f .

6) $\|f\|_{B_q^\alpha(L_p(Q))}$ es equivalente a la norma $\left(\sum_k \sum_j |d_{j,k}|^q\right)^{1/q}$, donde $\frac{1}{q} = \frac{\alpha}{2} + \frac{1}{2}$.

Compresión lineal

Sea ψ una wavelet y considere la imagen f dada por (3.1.5). Sea

$$f_N = \sum_{k < N} \sum_j d_{j,k} \psi_{jk}, \quad \text{donde } d_{j,k} = \langle f, \psi_{jk} \rangle$$

es decir, se incluye en la aproximación todos los coeficientes $d_{j,k}$ con frecuencia menor que 2^N , $N < m$ (f_N es la proyección sobre V_m). f_N se llama la aproximación wavelet para f . Entonces

$$\begin{aligned} \|f - f_N\|_{L_2(Q)}^2 &= \left\| \sum_{k \geq 0} \sum_j d_{j,k} \psi_{jk} - \sum_{0 \leq k < N} \sum_j d_{j,k} \psi_{jk} \right\|_{L_2(Q)}^2 \\ &= \left\| \sum_{k \geq N} \sum_j d_{j,k} \psi_{jk} \right\|_{L_2(Q)}^2 = \sum_{k \geq N} \sum_j \langle d_{j,k} \psi_{jk}, d_{j,k} \psi_{jk} \rangle_2 \\ &= \sum_{k \geq N} \sum_j d_{j,k} \overline{d_{j,k}} \langle \psi_{jk}, \psi_{jk} \rangle_2 \stackrel{*}{=} \sum_{k \geq N} \sum_j |d_{j,k}|^2 \\ &\leq \sum_{k \geq N} \sum_j \frac{2^{2\alpha k}}{2^{2\alpha N}} |d_{j,k}|^2 \leq 2^{-2\alpha N} \sum_k \sum_j 2^{2\alpha k} |d_{j,k}|^2 \\ &= 2^{-2\alpha N} \|f\|_{H^\alpha(L_2(Q))}^2 \quad (\text{por la observación (3.1.4)}). \end{aligned}$$

* Note que $\langle \psi_{jk}, \psi_{jk} \rangle_2 = \|\psi_{jk}\|^2 = 1$.

Por tanto,

$$\|f - f_N\|_{L_2(Q)} \leq 2^{-\alpha N} \|f\|_{H^\alpha(L_2(Q))}.$$

Observe que la aproximación lineal por wavelets produce resultados similares a la aproximación por Fourier [33].

Compresión no lineal

El propósito ahora es presentar la aproximación no lineal usando bases wavelets, más bien que la base de Fourier. Sea $\mathcal{H} = L_2(\mathbb{R}^2)$, y considere una base ortonormal de la forma $\{\psi_{jk}\}$. La meta es aprovechar la propiedad zoom de las wavelets para reproducir las singularidades esenciales de f en la imagen comprimida f_N . Recuerde que las singularidades producen coeficientes grandes de la wavelet, por lo tanto se espera que el error en

este caso, sea ciertamente una cantidad más pequeña que la del error de aproximación lineal.

Para una señal $f \in L_2(\mathbb{R})$, la aproximación no lineal con N coeficientes está dada por

$$f_N = \sum_{\gamma \in A_N} \langle f, \psi_\gamma \rangle \psi_\gamma,$$

donde $A_N = \{\gamma_1 = (j_1, k_1), \dots, \gamma_N = (j_N, k_N)\}$ y los coeficientes se escogen en orden no creciente:

$$|\langle f, \psi_{\gamma_1} \rangle| \geq |\langle f, \psi_{\gamma_2} \rangle| \geq \dots \quad (3.1.6)$$

El error de aproximación es la suma de los coeficientes restantes

$$\|f - f_N\|_{L_2(Q)}^2 = \sum_{\gamma \notin A_N} |\langle f, \psi_\gamma \rangle|^2.$$

Con el fin de minimizar este error, los índices en A_N deben corresponder a los N coeficientes grandes, pues éstos provienen de las singularidades de f .

En consecuencia, para el algoritmo de la compresión no lineal se toma

$$f_N = \sum_{k=1}^N |\langle f, \psi_{\gamma_k} \rangle| \psi_{\gamma_k}.$$

Ahora, suponiendo que $f \in B_q^\alpha(L_q(Q))$, $\frac{1}{q} = \frac{\alpha}{2} + \frac{1}{2}$, entonces el error mínimo en la aproximación no lineal es

$$\|f - f_N\|_{L_2(Q)}^2 = \sum_{k=N+1}^{\infty} |\langle f, \psi_{\gamma_k} \rangle|^2. \quad (3.1.7)$$

Como los coeficientes satisfacen (3.1.6), entonces

$$\begin{aligned} k |\langle f, \psi_{\gamma_k} \rangle|^q &= |\langle f, \psi_{\gamma_k} \rangle|^q + \dots + |\langle f, \psi_{\gamma_k} \rangle|^q \\ &\leq |\langle f, \psi_{\gamma_1} \rangle|^q + \dots + |\langle f, \psi_{\gamma_k} \rangle|^q \\ &= \sum_{n=1}^k |\langle f, \psi_{\gamma_n} \rangle|^q \leq \sum_{n=1}^{\infty} |\langle f, \psi_{\gamma_n} \rangle|^q \\ &= \|f\|_{\ell_q}^q \leq C \|f\|_{B_q^\alpha(L_q(Q))}^q. \end{aligned}$$

De donde

$$|\langle f, \psi_{\gamma_k} \rangle| \leq C k^{-1/q} \|f\|_{B_q^\alpha(L_q(Q))}. \quad (3.1.8)$$

Por tanto, al sustituir (3.1.8) en (3.1.7) se tiene

$$\|f - f_N\|_{L_2(Q)}^2 \leq C \|f\|_{B_q^\alpha(L_q(Q))}^2 \sum_{k=N+1}^{\infty} k^{-2/q},$$

y teniendo en cuenta que

$$\sum_{k=N+1}^{\infty} k^{-2/q} \leq \int_N^{\infty} x^{-2/q} dx = \frac{N^{-\alpha}}{\alpha}, \quad \alpha > 0$$

se deduce

$$\|f - f_N\|_{L_2(Q)} \leq CN^{-\alpha/2} \|f\|_{B_q^\alpha(L_q(Q))}.$$

Observación 3.1.2. 1. El análisis de arriba se puede aplicar a cualquier esquema de compresión que satisfaga

$$\tilde{f} = \sum \tilde{d}_{j,k} \psi_{jk}, \quad \tilde{d}_{j,k} = \langle \tilde{f}, \psi_{jk} \rangle$$

con $|d_{j,k} - \tilde{d}_{j,k}| \leq \gamma$ y $|d_{j,k}| < \gamma$ implicando $\tilde{d}_{j,k} = 0$.

2. Como se mencionó previamente, la cantidad $N^{-\alpha/2}$ se puede alcanzar por aproximación lineal para funciones que tienen α derivadas en L_2 , es decir, funciones en H^α . El aumento al conmutar a la aproximación no lineal está en que la clase $B_q^\alpha(L_q(Q))$ es más grande que H^α . En particular, $B_q^\alpha(L_q(Q))$ contiene funciones discontinuas para valores arbitrariamente grandes de α , mientras que las funciones en H^α son necesariamente continuas si $\alpha > 1/2$.

3.2. Introducción a la teoría de códigos

Una de las más importantes aplicaciones del álgebra moderna tiene que ver con el desarrollo algebraico de la teoría de codificación (ver [2], [40], [44] o [52]). La teoría de grupos ha influido enormemente en el avance de la teoría de códigos. Hay métodos algebraicos eficientes y confiables para la transmisión de información, estos métodos se llaman códigos algebraicos los cuales utilizan procedimientos de codificación y decodificación muy simples y fáciles de implementar.

La teoría matemática de comunicación originada en 1948 con los trabajos de Claude Shannon [46] y junto con los resultados de Richard Hamming (1950) [26] motivaron y desarrollaron esta teoría como respuesta a problemas prácticos de información, como por ejemplo, medir la cantidad de información presente en un archivo de datos o una cinta de texto o como codificar la información para almacenarse eficientemente o transmitirse confiablemente.

En consecuencia, la esencia del documento de Shannon tenía dos objetivos fundamentales: la transmisión de la máxima cantidad de información posible por un canal y, la detección y corrección de errores de transmisión debido a ruidos en dicho canal.

3.2.1. Elementos de la teoría de codificación

En general, una fuente de información es un conjunto finito

$$X = \{x_1, x_2, \dots, x_n\}$$

con probabilidades asociadas dadas por $p_i = P(X = x_i)$ para $1 \leq i \leq n$, donde se usa la notación $(X = x_i)$ para denotar el evento: la fuente produce el símbolo x_i . Acá $0 \leq p_i \leq 1$ y $\sum_{i=1}^n p_i = 1$. Se denota una fuente de información X por la expresión

$$X = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ p_1 & p_2 & \cdots & p_n \end{pmatrix}.$$

La cantidad promedio de información por símbolo de la fuente X se llama entropía, se denota por $H(X)$, y se define por

$$H(X) = - \sum_{i=1}^n p_i \log_2 p_i,$$

donde las unidades de $H(X)$ son los bits de información/símbolo fuente.

Se define $0 \log_2 0 = 0$ ya que $\lim_{x \rightarrow 0^+} x \log_2 x = 0$ y $\log_2 1 = 0$. Luego $H(X) = 0$ para tal fuente.

El mensaje es una unidad básica de información, y se define como una sucesión finita de símbolos o caracteres de un alfabeto finito. Un alfabeto es un conjunto finito de símbolos. Por ejemplo, se puede elegir un alfabeto como el conjunto $B = \{0, 1\}$. Cada símbolo que se quiera transmitir, una palabra, se representa como una sucesión de m elementos de B .

Si se define la operación $+$ mediante la siguiente tabla:

$+$	0	1
0	0	1
1	1	0

Tabla 1.

Entonces $B = \{0, 1\}$ es un grupo bajo esta operación binaria. Este grupo se denota por \mathbb{Z}_2 (ver [28]). Por \mathbb{Z}_2^n , se denota al conjunto de todas las n -adas

de números de \mathbb{Z}_2 , esto es, n -adas de ceros y unos. O sea, \mathbb{Z}_2^n es el conjunto de todas las palabras de longitud n .

La tarea básica en la transmisión de información es reducir la probabilidad de recibir una palabra diferente de la palabra enviada. Esto se hace siguiendo un modelo de canal simétrico binario, es decir, cuando un transmisor envía la señal 0 o 1 por el canal, a cada señal se le asocia una probabilidad p de transmisión incorrecta, como se ve en la Figura 3.2.1. Si la probabilidad es igual para las dos señales, el canal se llama simétrico.

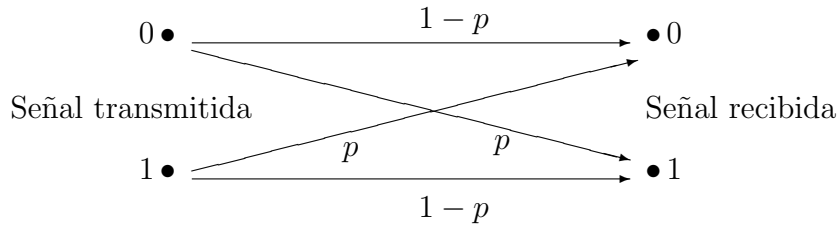


Figura 3.2.1

Si la entrada del canal es producida por alguna fuente de información X , entonces se puede describir la entrada del canal por la variable aleatoria X . El canal y la fuente se pueden pensar como otra fuente Y . Luego se puede describir la salida del canal por la variable aleatoria Y . En resumen, el canal simétrico binario se caracteriza por las siguientes probabilidades condicionales

$$\begin{aligned} P(Y = 0 | X = 0) &= 1 - p \\ P(Y = 0 | X = 1) &= p \\ P(Y = 1 | X = 0) &= p \\ P(Y = 1 | X = 1) &= 1 - p. \end{aligned}$$

Para cada valor diferente de p con $0 \leq p \leq 1$ se tiene un diferente canal simétrico binario, esto es, existe un número infinito de canales simétricos binarios.

En todo el análisis de la teoría de codificación, se supone que la transmisión de cualquier señal no depende de las transmisiones de las señales anteriores. En consecuencia, la probabilidad de que tengan lugar todos estos eventos independientes, se enuncia por el producto de probabilidades individuales. La probabilidad que en la transmisión de n símbolos se encuentren k errores es:

$$\binom{n}{k} p^k (1 - p)^{n-k}.$$

Ejemplo 3.2.1. Considere la palabra $c = 10110 \in \mathbb{Z}_2^5$. Si al transmitir cada bit de c por el canal simétrico binario se supone que la probabilidad de transmisión incorrecta es $p = 0,05$, entonces la probabilidad de que el receptor del mensaje de cinco bits reciba la palabra $r = 00110 \in \mathbb{Z}_2^5$ es $(0,05)(0,95)^4 = 0,041$. De igual manera, la probabilidad de recibir $r = 00100$ es $\binom{5}{2}(0,05)^2(0,95)^3 = 0,0021$.

Para mejorar la precisión de transmisión en un canal simétrico binario se pueden utilizar determinados esquemas de codificación en los que se presentan señales adicionales, para ello se darán algunas definiciones y propiedades sobre la teoría de codificación.

Definición 3.2.1. A la función uno a uno

$$h : M \subseteq \mathbb{Z}_2^m \rightarrow \mathbb{Z}_2^n, \quad \text{para } m, n \in \mathbb{N},$$

con $m < n$, se llama función codificadora o una codificación (m, n) . El conjunto $M \neq \emptyset$ consta de los mensajes a transmitir. El conjunto

$$C = \{h(c), c \in M\}$$

se llama un código y $h(c)$ es la palabra codificada. Los ceros y unos adicionales pueden proporcionar el medio para detectar o corregir los errores producidos en el canal de transmisión.

Las tres características básicas de un código son:

- a) Un conjunto de mensajes.
- b) Un método de codificación de los mensajes.
- c) Un método de decodificación de los mensajes recibidos.

La meta de la teoría de códigos es de disponer de métodos de codificación y decodificación que sean confiables, eficientes y fáciles de implementar.

El alfabeto para los códigos algebraicos son los elementos del campo finito de Galois, $\mathbb{F}_q = GF(q)$, el cual es isomorfo a \mathbb{Z}_q , donde q es un número primo. $\mathbb{F}_q^n = \langle \mathbb{F}_q^n, + \rangle$ representa el espacio vectorial sobre \mathbb{F}_q . Cuando \mathbb{F}_q es \mathbb{Z}_2 , el código se llama binario.

Definición 3.2.2. (distancia y peso de Hamming) Sean

$$X = x_1x_2 \dots x_n \in \mathbb{F}_q^n \quad \text{y} \quad Y = y_1y_2 \dots y_n \in \mathbb{F}_q^n$$

la distancia de Hamming, denotada $d(X, Y)$, se define como el número de componentes en que difieren los vectores X y Y .

El peso de Hamming, denotado $\omega(X)$, se define como el número de componentes x_i no nulas. Esto es, $\omega(X) = d(X, 0)$, donde $0 = 00 \dots 0 \in \mathbb{F}_q^n$.

Es claro que $d(X, Y) = d(X - Y, 0) = \omega(X) = \omega(X - Y)$. Se puede verificar que la distancia de Hamming cumple las propiedades de una métrica en \mathbb{F}_2^n .

Ejemplo 3.2.2. Sean $X = 01001$ $Y = 11101$ elementos de \mathbb{F}_2^5 , entonces $\omega(X) = 2$, $\omega(Y) = 4$. Como $X + Y = 10100$, entonces $\omega(X + Y) = 2 = d(X, Y)$.

Definición 3.2.3. Sea la función codificadora $h : \mathbb{F}_q^m \rightarrow \mathbb{F}_q^n$. El código

$$C = h[\mathbb{F}_q^m] = \{h(c) : c \in \mathbb{F}_q^m\}$$

se llama un código de grupo si C es un subgrupo de \mathbb{F}_q^n . A los códigos de grupo también se llaman códigos lineales.

Ejemplo 3.2.3. Considere la función codificadora $h : \mathbb{F}_2^3 \rightarrow \mathbb{F}_2^6$ definida por

$$\begin{aligned} h(000) &= 000000, & h(001) &= 001100, \\ h(011) &= 011111, & h(100) &= 100101, \\ h(110) &= 110110, & h(111) &= 111010, \\ h(010) &= 010011, & h(101) &= 101001. \end{aligned}$$

Es un código de grupo. Basta probar que el conjunto de palabras codificadas

$$C = \{000000, 001100, 011111, 100101, 110110, 111010, 010011, 101001\}$$

es un subgrupo de \mathbb{F}_2^6 . Primero observe que la identidad de \mathbb{F}_2^6 pertenece a C . Luego se verifica que si u y v son elementos de C , entonces $u + v \in C$ y así, C es un subgrupo de \mathbb{F}_2^6 y la función codificadora es un código de grupo.

Si un mensaje $c = c_1c_2 \dots c_n$ es codificado como $X = x_1x_2 \dots x_n$ y transmitido a través de un canal con ruido, y la palabra codificada X puede ser recibida como la palabra alterada $Y = y_1y_2 \dots y_n$, entonces la palabra error $Y - X$, se llama patrón de error. El siguiente criterio es utilizado para detectar errores.

En un código de grupo C , un error puede ser detectado si sólo si el patrón de error no es una palabra codificada en C .

En efecto, sea X la palabra codificada y $\varepsilon = Y - X \neq 0$ el patrón de error o $X + \varepsilon = Y$, donde Y es la palabra recibida. Si $\varepsilon \in C$, entonces $X + \varepsilon \in C$ y el patrón de error ε resulta ser una palabra codificada, por tanto la presencia de Y no es detectada. Recíprocamente, si $\varepsilon \notin C$ entonces $Y - X \notin C$, luego $Y \notin C$ y así, no es una palabra codificada.

Nota 3.2.1. Para decodificar el vector recibido Y como la palabra codificada más próxima a X , con respecto a la distancia de Hamming, se escoge un vector ε de peso mínimo. Este método de decodificación se llama de máxima verosimilitud, si la probabilidad de transmisión correcta $1 - p$ es mayor que $1/2$.

3.2.2. Matriz generadora y de verificación de paridad

Definición 3.2.4. Para $m < n$, sea $h : \mathbb{F}_q^m \rightarrow \mathbb{F}_q^n$ una función codificadora definida por $h(u) = uG$ con $u \in \mathbb{F}_q^m$ y G una matriz de orden $m \times n$ (con entradas en \mathbb{F}_q). La matriz G se llama matriz generadora para el código y está dada por:

$$G = (I_k : A) = \begin{bmatrix} 1 & 0 & \dots & 0 & \vdots & a_{11} & a_{12} & \dots & a_{1(n-m)} \\ 0 & 1 & \dots & 0 & \vdots & a_{21} & a_{22} & \dots & a_{2(n-m)} \\ \cdot & \cdot & \cdot & \cdot & \vdots & & & & \\ \cdot & \cdot & \cdot & \cdot & \vdots & & & & \\ \cdot & \cdot & \cdot & \cdot & \vdots & & & & \\ 0 & 0 & \dots & 1 & \vdots & a_{m1} & a_{m2} & \dots & a_{m(n-m)} \end{bmatrix} .$$

Dada la matriz G , el código generado por esta matriz puede ser obtenido formando todas las q^m posibles combinaciones lineales de las m filas de G . Una manera de alterna de describir un código de grupo C , es por medio de la matriz H de orden $(n - m) \times n$, llamada matriz de verificación de paridad, la cual satisface $HG^t = 0$.

Esto es, el espacio fila de H es el complemento ortogonal del espacio fila de G . Por tanto, cualquier palabra recibida V es una palabra codificada si y sólo si $HV^t = 0$. Al elemento HV^t se le llama síndrome de V .

En general, el conjunto mensaje consiste de todas las posibles q^m m -adas de \mathbb{F}_q^m . Si cada m -ada (vector fila m -dimensional) $u = \sigma_1\sigma_2 \dots \sigma_m$ es codificada con el vector n -dimensional uG donde G es la matriz generadora de orden $m \times n$, el resultado es un vector de la forma:

$$\begin{aligned} uG &= (\sigma_1\sigma_2 \dots \sigma_n)(I_m : A) & (*) \\ &= (\sigma_1\sigma_2 \dots \sigma_n b_1 b_2 \dots b_{n-m}) \end{aligned}$$

donde I_m es la matriz identidad de orden m , $b_j = \sum_{i=1}^m \sigma_i c_{ij}$ y $j = 1, 2, \dots, n - m$.

Estas ecuaciones se llaman ecuaciones de verificación de paridad. Cuando esto se hace con todas las posibles m -adas u , el resultado es un código de grupo

generado por G . Por tanto, con el esquema de codificación descrito por (*), cada palabra codificada en el código generado por G , tiene la propiedad atractiva de que el mensaje se obtiene de los primeros m símbolos. Un código con esta estructura, se llama código sistemático. Esto es, un código sistemático tiene matriz generadora de la forma $(I_m : A)$.

Ejemplo 3.2.4. Considere la función codificadora $h : \mathbb{F}_2^2 \rightarrow \mathbb{F}_2^5$ con matriz generadora

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

Determine el código de grupo C . En efecto, $\mathbb{F}_2^2 = \{00, 01, 10, 11\}$

$$h(00) = [00]G = 0000$$

$$h(01) = (01) \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} = 01011$$

$$h(10) = (10) \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} = 10110$$

Análogamente se obtiene : $h(11) = 11101$. Luego el código de grupo es

$$C = \{00000, 01011, 10110, 11101\}.$$

La matriz de verificación de paridad es

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Observe que si $u = \sigma_1\sigma_2 \in \mathbb{F}_2^2$ y $X = h(u) = \sigma_1\sigma_2\sigma_3\sigma_4\sigma_5 \in \mathbb{F}_2^5$, entonces

$$\begin{aligned} X &= uG = \sigma_1\sigma_2 \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} \\ &= (\sigma_1\sigma_2\sigma_1\sigma_1 + \sigma_2\sigma_2) \\ &= \sigma_1\sigma_2\sigma_1(\sigma_1 + \sigma_2)\sigma_2 \end{aligned}$$

de esto se tiene

$$\begin{aligned} \sigma_3 &= \sigma_1 \\ \sigma_4 &= \sigma_1 + \sigma_2 \\ \sigma_5 &= \sigma_2 \end{aligned}$$

estas son las ecuaciones de verificación de paridad. Ahora, como $\sigma_i \in \mathbb{F}_2$ para $1 \leq i \leq 5$ y $\sigma_i = -\sigma_i$ entonces estas ecuaciones pueden ser escritas como:

$$\begin{aligned}\sigma_1 + \sigma_3 &= 0 \\ \sigma_1 + \sigma_2 + \sigma_4 &= 0 \\ \sigma_2 + \sigma_5 &= 0.\end{aligned}$$

Luego,

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

o $HX^t = 0$. En consecuencia, $V = b_1b_2b_3b_4b_5 \in \mathbb{F}_2^5$ se puede identificar como una palabra codificada pues $HV^t = 0$.

La matriz de verificación de paridad H , proporciona un esquema de decodificación que corrige errores simples en las transmisiones; el procedimiento es el siguiente:

1. Para cualquier palabra recibida V , calcular HV^t .
2. Si $HV^t = 0$, no hay error en la transmisión.
3. Si HV^t es la i -ésima columna de H , entonces hubo un error simple en la transmisión y se cambia el i -ésimo componente de V para obtener la palabra codificada C . Aquí, si el código es binario, los primeros m componentes de C generan el mensaje inicial. Si el código no es binario, HV^t es r veces la i -ésima columna de H y suponiendo que la palabra fue $V - (00 \dots r0 \dots 0)$ donde $r \in \mathbb{F}_q$ ocurre en la i -ésima componente.
4. Si HV^t , no tiene la forma 2 o 3, entonces hubo más de un error en la transmisión y no se puede codificar.

Ejemplo 3.2.5. Suponga que se recibe $V = 11001$ que es el código del ejemplo 3.2.4, entonces

$$HV^t = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

Se presenta el caso (3), se cambia la tercera componente de V y se tiene $HV^t = 0$.

3.2.3. Códigos Huffman

Considere la fuente de información

$$X = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ p_1 & p_2 & \cdots & p_n \end{pmatrix}, \quad n \geq 2$$

para la codificación de la fuente X con entropía $H(X)$ se necesita, en promedio, un mínimo de $H(X)$ dígitos binarios por mensaje [46]. El número de dígitos en la palabra código es la longitud de la palabra. De esta manera, la longitud media de palabra de un código óptimo es $H(X)$. El código de Huffman es un procedimiento para encontrar el código de fuente óptimo, en el sentido que el promedio de una palabra codificada es lo más pequeño posible [29]. A continuación se describe el esquema de Huffman. Dada una fuente de información

$$X = \begin{pmatrix} x_1 & x_2 & \cdots & x_q \\ p_1 & p_2 & \cdots & p_q \end{pmatrix} \quad q \geq 2.$$

1. Si $q = 2$, sea $f(x_1) = 0$ $f(x_2) = 1$.
2. De otro lado, reordene X si es necesario tal que $p_1 \geq p_2 \geq \cdots \geq p_q$ y define la nueva fuente por

$$X' = \begin{pmatrix} x_1 & x_2 & \cdots & x_{q-2} & x' \\ p_1 & p_2 & \cdots & p_{q-2} & p_{q-1} + p_q \end{pmatrix}.$$

3. Aplique el algoritmo de Huffman a X' , obteniendo el esquema f' dado por

$$f'(x_1) = c_1, \dots, f'(x_{q-2}) = c_{q-2}, f'(x') = d,$$

donde c_i y d son cadenas de ceros y unos.

4. Defina el esquema de codificación, f , para X por

$$f(x_1) = c_1, \dots, f(x_{q-2}) = c_{q-2}, f(x_{q-1}) = d_0, f(x_q) = d_1.$$

El siguiente ejemplo ilustra el procedimiento.

Ejemplo 3.2.6. Considere la fuente de información

$$X = \begin{pmatrix} A & B & C & D & E & F \\ 0,30 & 0,25 & 0,15 & 0,12 & 0,10 & 0,08 \end{pmatrix}.$$

1) Crear una lista con tantos nodos como palabras se vayan a codificar. En cada nodo se almacena la probabilidad de la palabra a codificar en orden descendente.

2) Extraer de la lista los dos nodos con menor probabilidad (si hay más de dos, la elección es arbitraria) y cree un nuevo nodo con la suma de estas dos probabilidades. Este nodo no representa a una palabra en concreto.

3) Repita los pasos 1) y 2) en el próximo nivel.

Concretando, se extraen los nodos con probabilidades 0.10 y 0.08 y se forma el nodo con probabilidad 0.18. Luego se extraen los nodos con probabilidades 0.12 y 0.15 y se forma el nodo con probabilidad 0.27, a continuación se toman los nodos con probabilidades 0.18 y 0.25 y se forma el nodo con probabilidad 0.43 y se continúa de esta manera.

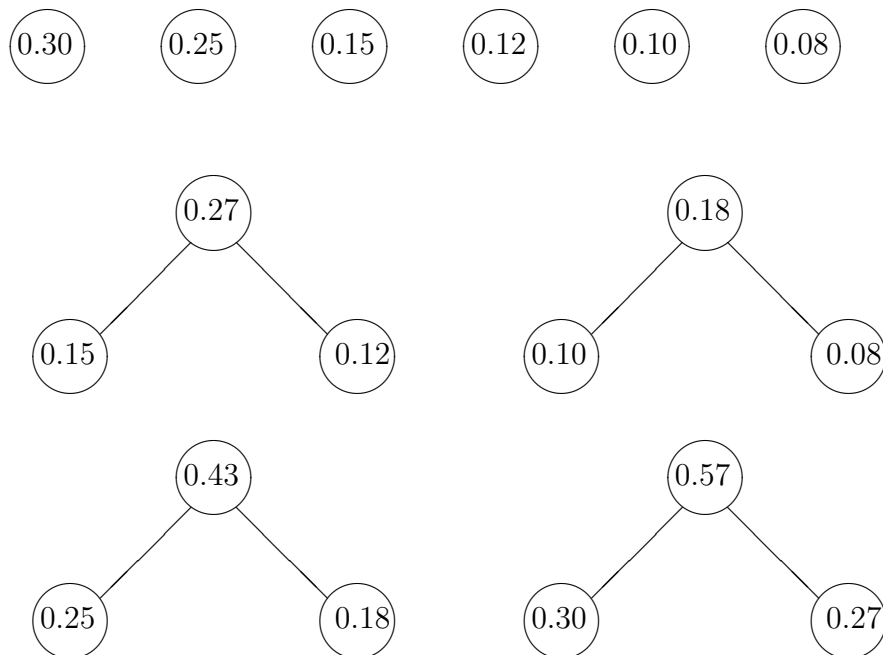


Figura 3.2.2

Ahora, asigne a las palabras codificadas en cada nodo empezando desde la parte superior del árbol, los dígitos 0 y 1 y a cada nodo dividido, al de la

izquierda, 0 y al de la derecha 1. De este modo se obtiene el árbol binario del código de Huffman.

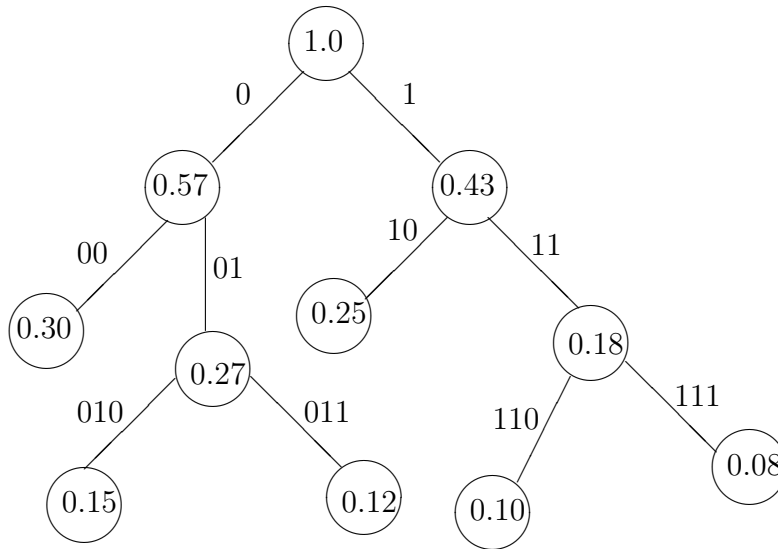


Figura 3.2.3

El código óptimo (Huffman) que se obtiene de este modo tiene una longitud media $L = \sum_{i=1}^n p_i l_i$, donde l_i es la longitud de la palabra i . En este caso la longitud del código es

$$L = \sum_{i=1}^n p_i l_i = 2,45 \text{ dígitos binarios.}$$

La entropía $H(X)$ de la fuente es

$$H(X) = - \sum_{i=1}^n p_i \log_2 p_i = 2,418 \text{ bits.}$$

El mérito de cualquier código se mide mediante su longitud media en comparación con $H(X)$. Se define la eficiencia del código η como

$$\eta = \frac{H(X)}{L},$$

donde L es la longitud media del código. En este ejemplo, $\eta = \frac{2,418}{2,45} = 0,976$.

3.3. Algoritmo de compresión

La transformada wavelet es una herramienta muy poderosa para resolver problemas que tienen que ver con la compresión de una señal y que otros métodos, como el análisis de Fourier tienen limitaciones bien conocidas, tal es el caso de las señales cuyo espectro varía con el tiempo, así como también las imágenes, ya que cuando se aplica el algoritmo de análisis a una imagen se pierde mucha información al realizarse la antitransformación en un intervalo finito. Esto último, es un problema que la transformada wavelet resuelve de manera satisfactoria y eficiente, ya que una imagen, utilizando este método, puede ser comprimida gastando menos recursos, por ejemplo de tipo computacionales, y minimizando la pérdida de información que se produce en todo proceso de compresión. En este punto, emerge la siguiente pregunta: En la práctica, ¿cómo se comprime una imagen usando la transformada wavelet?

3.3.1. Discretización de una imagen

Lo primero que se debe discutir es ¿cómo se representa digitalmente una imagen?, o lo que es lo mismo, ¿cómo se representa la información de una imagen en un computador? Físicamente, Una imagen es el mapeo de un conjunto de puntos luminosos de un objeto situado en una región del espacio a puntos en otra región del espacio, mediante los fenómenos de reflexión o refracción. Cada punto de la imagen tiene asociado un conjunto de propiedades matemáticas entre las que se destacan la intensidad, que es la cantidad de energía radiante que llega a cada punto de la imagen, y el color, que está asociado con las características espectrales de la misma. La Figura 3.3.1, por ejemplo, es la imagen digitalizada de un río en Indonesia; cada punto de la imagen se asocia con la información del objeto que se está observando. los puntos oscuros representan la información asociada con el curso del río, mientras que los lugares brillantes están asociados con la vegetación a las orilla del mismo. De igual forma el color brinda información muy importante del objeto; en este caso, la vegetación, está asociada con la longitud de onda correspondiente al color verde.

Ambas propiedades, “la intensidad y el color” pueden cuantificarse definiendo por cada longitud de onda del espectro electromagnético una función de variable real definida sobre un plano, es decir definiendo una representación matemática. Acorde con esto, se define una imagen, plana y rectangular, como una función $G_\lambda(x, y)$, donde λ es una longitud de onda dada, $x \in [a_x, b_x]$, $y \in [a_y, b_y]$ con $[a_x, b_x]$ y $[a_y, b_y]$ las fronteras de la imagen rectangular. $G_\lambda(x, y)$ da el nivel de intensidad de la imagen en el modo λ . Al

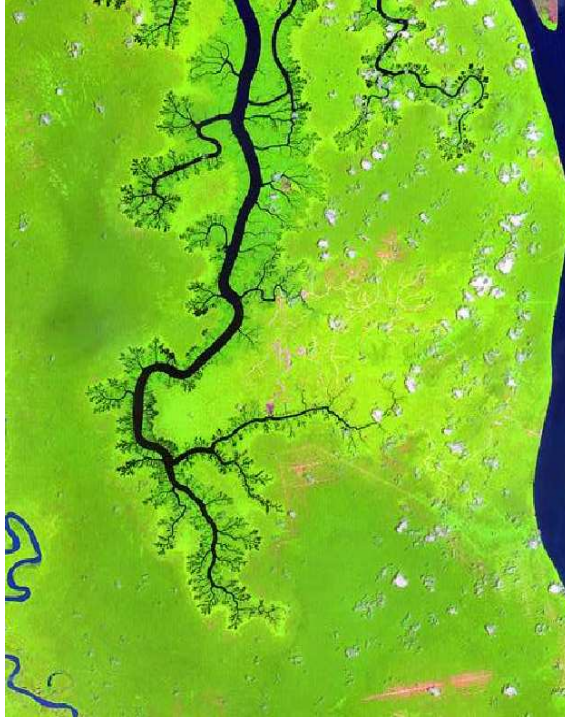


Figura 3.1: .

tratar de almacenar esa información, aparece una situación que tiene una alta complejidad tecnológica y la cual está asociada con el hecho que la imagen es la superposición de infinitos puntos, cada uno de los cuales está representado por una función de la forma $G_\lambda(x, y)$

Con el fin de ir reduciendo toda esta información a niveles “más manejables”, se procede a realizar un conjunto de discretizaciones, las cuales se describen a continuación.

Primera discretización (cuantización)

Se puede decir que es aquí donde comienza el proceso de compresión de una imagen. Cuando el mundo se discretiza, la información se empieza a reducir, y esto conlleva a una pérdida de una parte de la misma, con el propósito de poder manipularla. La primera discretización es la manipulación que se hace sobre el espectro, ya que toda la información se restringe al intervalo de longitudes de onda que es observable por el ojo humano. Sin embargo como aún la información a manipular, asociada con las longitudes de onda, es demasiado grande, entonces se discretiza todavía más definiendo unos subintervalos de longitudes de onda. En los esquemas de discretización

del espectro se definen las llamadas bandas espectrales, de las cuales la más usada a nivel de tecnología son las bandas (R, G, B) , correspondientes a las longitudes de onda del rojo (Red), el verde (Green) y el azul (Blue). Con esto, se han reducido el infinito de las funciones de longitud de onda λ a tres funciones $G_R(x, y)$, $G_G(x, y)$ y $G_B(x, y)$, las cuales representan la intensidad integrada sobre cada una de las bandas del espectro.

Al realizar esta primera discretización, la imagen que en principio presentaba una gran riqueza se ha reducido a tres funciones que contienen toda la información, ya que en cierta forma lo que se ha perdido en este primer paso son las tonalidades más finas de la imagen.

Segunda discretización

Pese a que con la primera discretización la información de la imagen se ha reducido, todavía es de difícil manipulación ya que está compuesta por infinitos puntos. El paso a seguir consiste en escoger un conjunto finito de puntos distribuidos sobre la imagen, en una malla rectangular de puntos bien espaciados y sobre cada punto (llamado punto de muestreo), tomar el vecino del punto y promediar la intensidad en cada banda que llega de la luz allí. El resultado es la reducción en la resolución espacial de la imagen. Cuando se trabaja con la imagen en puro, esta tiene una resolución infinita; cuando se discretiza espacialmente, la imagen comienza a perder nivel de detalle.

Esta discretización, lo que hace es reducir la imagen y las funciones que dan la intensidad integrada sobre las bandas, serán matrices de la forma $(G_R)_{mn}$, $(G_G)_{mn}$ y $(G_B)_{mn}$, que representan la intensidad media alrededor de ese punto de imagen (pixel).

Tercera discretización (codificación)

Hasta aquí la imagen se ha reducido a un conjunto de tres matrices $M \times N$ que contienen la información de cada pixel. Sin embargo, aún prevalece un continuo que se debe discretizar y son los posibles valores que puede adoptar la intensidad.

La intensidad es una variable física y puede adoptar un valor real con todas las posibles cifras decimales. En un dispositivo de almacenamiento, como un computador, no se pueden manipular números reales de precisión arbitraria y esto induce a una nueva discretización en la que los valores de la intensidad en una imagen, que conforman un continuo, se discretizan en valores bien definidos. Estos valores, que permiten valorar la intensidad de la luz en una imagen, reciben el nombre de niveles de intensidad.

El procedimiento consiste en redondear los valores del nivel de intensidad,

que como se dijo antes son números reales, lo que se consigue cortando los valores en algún punto, por ejemplo en la cuarta cifra decimal e imaginando un intervalo de intensidad entre 0 y 1 de tal manera que este queda reducido en un intervalo de diez mil valores posibles de intensidad; esto significa que hay una discretización en donde se convierten números reales en un intervalo de números enteros y cada número entero es una etiqueta que se asocia con cada nivel de intensidad. Normalmente ese conjunto de números enteros que representan la imagen deben ser almacenados en un mecanismo de almacenamiento digital el cual utiliza un sistema de codificación binario, prendido - apagado, cargado - descargado, que corresponde a un sistema de codificación de la información propio de los sistemas electrónicos. Un número delimitado de valores en el sistema binario corresponde a dígitos binarios con un número limitado de dígitos; es decir, si s representa un número de s dígitos entonces se pueden almacenar números entre 0 y $2^s - 1$; esto es, si i representa el número de enteros que se pueden almacenar, entonces, $i \in [0, 2^s - 1]$. Por ejemplo, si se tiene un número binario con 8 dígitos, entonces $s = 8$ y se pueden almacenar números $i \in [0, 255]$.

Un paso muy importante en el proceso de discretización de una imagen y de alguna manera en el proceso de compresión de la información que hay en ella, es tratar de acomodar los niveles de intensidad que se definen a una escala de valores relacionadas con un múltiplo de dos. Cuando la información de una imagen llega a este nivel, en donde se codifica la intensidad de la imagen en números con una longitud de s bits, se dice que la imagen está siendo codificada con una profundidad de color de s bits; por ejemplo, una imagen codificada a 8 bits es una imagen en la que hay 255 niveles de intensidad para representar el continuo de los valores de esta intensidad. Luego, el resultado final de lo que es inicialmente una imagen física son tres matrices $[I_R]_{mn}$, $[I_G]_{mn}$ y $[I_B]_{mn}$ de números enteros con cada uno de ellos representando un nivel de intensidad.

Cuando se llega a este punto, se puede afirmar que la imagen es de fácil manipulación digital, ya que se tienen unos dígitos binarios que se pueden almacenar, por ejemplo, en una USB. A este nivel, la imagen a color codificada en (R, G, B) , en un computador ocupa un espacio que es igual a s bits, que es la cantidad de bits necesaria para almacenar el entero que indica el nivel de intensidad en un pixel, multiplicado por el número de pixeles de la imagen MN , multiplicado por el número de canales, 3; luego, el tamaño de una imagen con las condiciones especificadas es:

$$T = 3MN s \quad (3.3.1)$$

Por ejemplo, el tamaño de una imagen que tiene una profundidad de 8 bits

de 400 por 400 pixel, tiene un tamaño de aproximadamente 469 kb, que corresponde a una imagen a color bastante sencilla.

En muchas situaciones no se requieren de los tres canales para conseguir la información a estudiar y posteriormente a almacenar; un primer ejemplo de esto son las huellas dactilares, de las cuales solo se requieren los surcos y estructuras de la piel de la mano, es decir, se necesita únicamente, en una visión por computadora, de la forma general o la silueta de la imagen. Este tipo de imágenes se les denomina imágenes binarias y es el tipo de imagen más simple y solo pueden tomar dos valores discretos, blanco y negro, en donde el negro es representado por el valor 0.

Otra situación en la que la consideración de los tres canales no es muy relevante son las imágenes satelitales para ciertos estudios de tipo Geológico, como lo es el estudio de la cuenca de un río, en donde la información más relevante y de interés no es la vegetación a la orilla sino el curso de agua. Para esto solo se requiere, al igual que en el caso anterior, de un solo canal espectral. Para este tipo de imágenes se requieren diferentes niveles de brillantez y son conocidas como imágenes a escala de grises, llamadas también monocromáticas o imágenes de un solo color, contienen datos de $8\text{bits}/\text{pixel}$, lo que significa que se pueden tener 256 (0 a 255) niveles de brillantez, donde 0 representa el negro y 255 el blanco.

Cuando se reduce a un solo canal, se dice que se ha hecho una conversión a escala de grises, lo que produce normalmente una reducción en un factor de tres en el tamaño de la imagen. Se podría pensar que ya se ha reducido en su totalidad la imagen. Sin embargo, se puede almacenar una imagen en escala de grises usando menos espacio, lo que se logra reduciendo la profundidad de cada pixel. La profundidad de cada pixel, como se dijo antes, es de 8 bit, lo que implica que los niveles de intensidad están entre 0 y 255. Ahora, en muchos casos, como por ejemplo en una huella dactilar, no se requieren tantos niveles de grises y se puede entonces reducir la profundidad a 4 bits, con lo cual se han reducido los niveles de intensidad a 32 niveles de grises. Reduciendo a la mitad la profundidad de color, se hace una reducción adicional en un factor de 2, y hasta aquí vamos por un factor de 6. Luego, al ir reduciendo los niveles de grises hasta que se observe que se está perdiendo información relevante, garantiza una “economía” en la imagen.

Todavía se puede reducir aún más la imagen. Piense en lo siguiente, cuando se tiene la imagen de un tablero de ajedrez, se observa que es muy poca la información útil, ya que la imagen puede ser reconstruída con muy poca información, pues es mucha la información redundante. Es más, el cerebro, que es uno de los mecanismos más sofisticado y complejo que existe para el procesamiento de señales, no toma toda la información que una persona observa, toma algunos elementos del paisaje y utilizando información

previa reconstruye la imagen que se está observando. Por lo tanto, la idea es obviar los detalles de la imagen y buscar los patrones, que es la información útil de la imagen; esta es la técnica que utilizan los métodos más sofisticados en matemáticas para comprimir imágenes.

La búsqueda de un patrón, implica la observación de una señal que, en general, es muy complicada; sin embargo, al muestrear la señal utilizando técnicas matemáticas, como por ejemplo análisis de Fourier, se identifica que esta es la superposición de un conjunto muy determinado de señales armónicas bien definidas. Al considerar una imagen como una señal, se hace algo parecido para comprimirla y que a continuación se entra a detallar.

3.3.2. Algoritmo para la compresión

Como se mencionó antes, una imagen, en blanco y negro, es una matriz de enteros $M \times N$ elegidos dentro de un rango específico, por ejemplo, entre 0 y $L - 1$. Cada elemento de la matriz es asociado a un elemento fotográfico, el cual es conocido como pixel y su valor está asociado con un matriz particular en la escala de grises. Es común asociar el color negro con el valor 0 y el valor $L - 1$ con el color blanco. Se supone que M es alguna potencia de dos, por lo general 256 ó 512. Si $M = 256$, lo que corresponde a 65536 pixeles, y $L = 256$, lo que corresponde a 8 bits por pixel, los requerimientos de almacenamiento para una imagen serán $256 \times 256 \times 8 = 524288$ bits. La meta en una compresión de imagen es aprovechar la estructura oculta de la imagen con el fin de reducir los requerimientos de almacenamiento. Esta técnica se aplica para el almacenamiento y transferencia de imágenes y además permite la extracción de características de la imagen.

Las siguientes propiedades están asociadas con un eficiente sistema compresor de imágenes:

1. Reducción significativa del número de bits utilizados en el almacenamiento.
2. Para el ojo humano, la pérdida debe ser insignificante en la calidad de la imagen. De igual manera, cuando se utiliza para aplicaciones en visión artificial, no suponer pérdida de información en las características más relevantes de la imagen.
3. Rapidez de cálculo para la compresión y para la descompresión.
4. El formato de salida debe permitir su almacenamiento y transferencia.

En general, un sistema de compresión de imágenes conta de tres etapas: La transformación, la cuantización y la codificación, con la característica que

los datos de salida de cada etapa son los datos de operación de la siguiente.

A continuación se exponen las características de cada una de estas etapas.

La etapa de transformación

Etapa en la cual se utiliza una función que transforma el conjunto de datos que componen la imagen original en un nuevo conjunto de datos, caracterizados por la eliminación de la información redundante, evitando pérdidas de información. En este paso la transformada wavelet juega un papel muy importante, ya que permite decorrelacionar o encontrar en la imagen toda la información “escondida” o niveles de resolución que con otros mecanismos, como por ejemplo la transformada de Fourier, no se pueden encontrar.

Se parte de un conjunto de funciones que definen una señal de comparación, que en el caso de análisis de Fourier son funciones armónicas, mientras que en análisis wavelet son un conjunto de bases distintas construídas a partir de la wavelet madre. Estas bases wavelet son una representación eficiente de funciones que se caracterizan por su uniformidad excepto en un pequeño conjunto de discontinuidades. Una imagen que tenga una región muy amplia en escala de grises constante se puede representar con una base wavelet.

Para una imagen dada, es posible encontrar el mejor paquete de bases wavelet y utilizar la expansión en esa base como la transformación. La ventaja de esto, es que los coeficientes que resultan se optimizan a alguna medida de eficiencia relativamente apropiada [56] y la desventaja, que la hay, es que la mejor base depende de la imagen; por lo tanto, para una imagen $M \times M$ hay $2^{\frac{M^2}{2}}$ paquetes de bases wavelet y para especificar la transformación que se usa se requieren $\frac{M^2}{2}$ bits, con lo cual hay un aumento de por lo menos 0.5 bit por pixel en el costo del gasto general.

Una solución, a lo planteado anteriormente, muy efectiva en un proceso de compresión de un gran número de imágenes con características similares, es calcular una base particular muy ajustada al conjunto. Esto se hace identificando un subconjunto representativo de imágenes a ser comprimidas. Para un costo funcional dado, se elige una base B que minimice $\sum_{i=1}^s M(f_i, B)$ y así B es “el mejor grupo de bases” para el subconjunto y se usa para especificar la transformación utilizada en la compresión. Un ejemplo de este criterio, es la compresión de huellas dactilares, ya que los surcos que presenta la huella se traducen en valores pixel que oscilan rápidamente y por esto, una base wavelet estándar no es una representación óptima.

En la elección de “el mejor grupo de bases” se debe tener en cuenta el

filtro wavelet que se debe utilizar. La recomendación son los filtros simétricos ya que muchos de los coeficientes que provienen de la periodización se pueden omitir. Como los filtros ortogonales (excepto los de Haar) no pueden ser simétricos, los filtros biortogonales siempre son los más usados en la compresión de imágenes.

Como la representación debe ser muy eficiente, se requieren de filtros con un gran número de momentos desvanecidos de tal forma que las partes suaves de una imagen produzcan coeficientes wavelets muy pequeños. Debido a las consideraciones de simetría, solo interesan wavelets biortogonales lo que se traduce en un número diferente de momentos desvanecidos en el filtro de análisis y en el filtro de reconstrucción. En el filtro de análisis los momentos desvanecidos dan como resultado pequeños coeficientes en la transformación, mientras que en el filtro de reconstrucción dan como resultado un número menor de bloques artificiosos en la imagen comprimida. También se busca que el filtro de análisis y el de reconstrucción sean tan cortos como sea posible. Cuantos más momentos desvanecidos tenga un filtro, mayor debe ser este; luego, existe una especie de “cambalache” [56] entre porciones que tienen momentos desvanecidos y filtros cortos. El filtro par $\frac{2}{7}$ es un buen compresor y es el más usado en la compresión de huellas dactilares.

La etapa de cuantización

Después de que la imagen ha sido transformada, se parte de un arreglo de $M \times M$ de coeficientes que pueden ser números de puntos flotantes de alta precisión, los cuales se deben cuantizar de tal forma que se tomen un número relativamente pequeño de valores. La cuantización se lleva a cabo por medio de un mapa de cuantización, Q , que es una función escalón de valor entero.

A continuación se describe un esquema de cuantización simple, llamado cuantización escalar.

Se comienza asumiendo que todos los coeficientes en el arreglo caen en un rango $[-\Lambda, \Lambda]$ y que se especifica el número de niveles cuantizados mediante un número entero q ; se particiona, en q subintervalos iguales el intervalo $[-\Lambda, \Lambda]$; se define un mapa de cuantización como se ve en la figura (2) izquierda; se observa que el rango de Q es el conjunto de $q - 1$ enteros $\{-q(q - 2)/2, \dots, (q - 2)/2\}$. Por último, se especifica una función de descuantización, Q^{-1} , como se ve en la Figura 3.3.2 derecha; Se observa que cada uno de los valores enteros en el rango de Q es mapeado al centro del intervalo que corresponde en la partición, excepto $Q^{-1}(0) = 0$

La impronta de buena calidad de una transformación efectiva para una codificación de imagen es que el mayor número de coeficientes sean pequeños

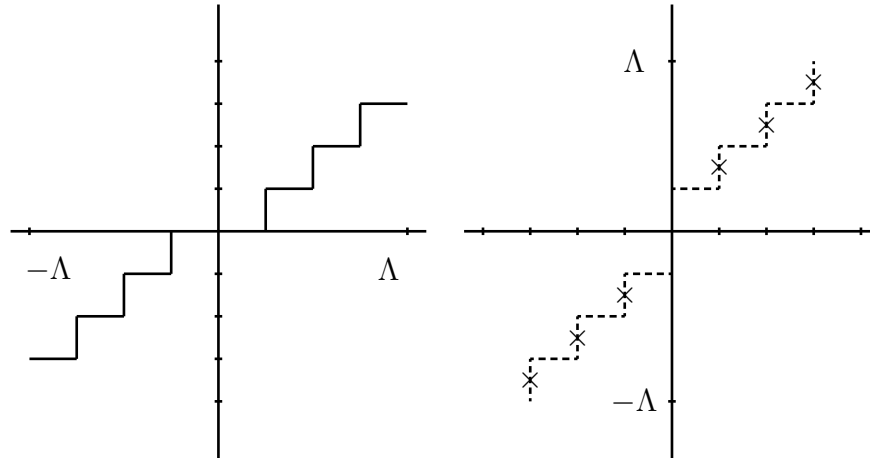


Figura 3.2: Izquierda: $Q(x)$, Derecha: $Q^{-1}(x)$.

y por lo tanto sean cuantizados a cero. Esto es muy útil para especificar un parámetro independiente o umbral (thresholding) $\lambda > 0$, tal que todos aquellos coeficientes menores que λ en valor absoluto sean cuantizados a cero. Hay dos tipos de umbral, el fuerte o duro y el suave o blando; la diferencia entre ellos radica en la forma en que se manipulan los coeficientes mayores que λ en valor absoluto. En el umbral fuerte, esos valores se dejan solos, mientras que en el umbral suave esos valores se reducen si λ es positivo e incrementan si λ es negativo. Un par de funciones umbral se definen como:

$$T_{hard}(x) = \begin{cases} 0, & \text{si } |x| \leq \lambda, \\ x, & \text{si } |x| > \lambda, \end{cases}$$

$$T_{soft}(x) = \begin{cases} 0, & \text{si } |x| \leq \lambda, \\ x - \lambda, & \text{si } x > \lambda, \\ x + \lambda, & \text{si } x < -\lambda \end{cases}$$

Estas funciones se muestran en la Figura 3.3.3. Con el uso de una función umbral, el mapa de cuantización tiene la forma Q o $T(x)$, donde T es la función umbral fuerte o la función umbral suave.

La etapa de codificación

Hasta aquí se tiene una imagen transformada, $M \times M$, que ha sido cuantizada de tal forma que el dato a ser comprimido consiste en una fila de M^2 enteros entre 0 y $r - 1$, para algún entero positivo r . Lo que sigue ahora

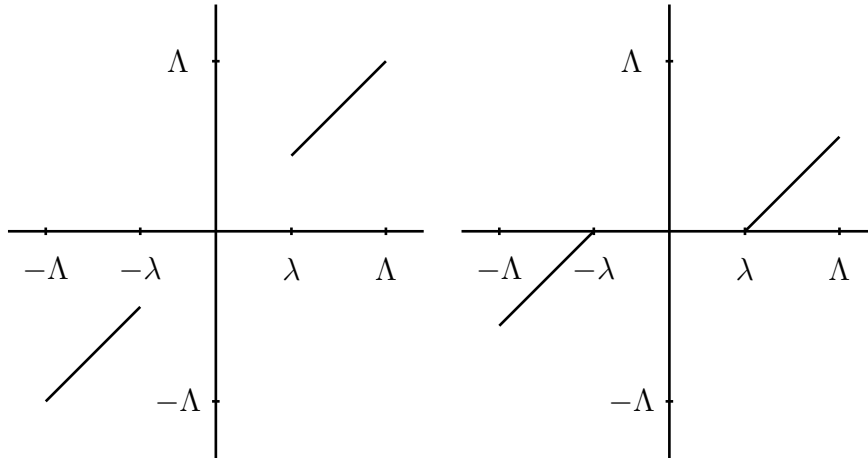


Figura 3.3: Izquierda: $T_{\text{hard}}(x)$, Derecha: $T_{\text{soft}}(x)$.

es codificar esta fila de números y aprovechar las redundancias con el fin de disminuir el número de bits que se requieren para almacenar la fila. Como un ejemplo de esto, considere la imagen de los coeficientes cuantizados como una señal que es una cadena de letras A, B, C, D , las cuales se van a almacenar como variables de dos bits en la memoria; de aquí, $r = 4$ y $M^2 = 16$ y el dato comprimido es escrito como:

AABCDAAABBADAAAA.

Este dato consta de cuatro símbolos diferentes y es posible codificarlos con dos dígitos binarios de la siguiente forma:

$$A \longrightarrow 00$$

$$B \longrightarrow 01$$

$$C \longrightarrow 10$$

$$D \longrightarrow 11.$$

El dato debe leerse como:

00000110110000000101001100000000

lo que requiere un total de 32 bits. Ahora, se observa que el símbolo A es el que más aparece, seguido del símbolo B y los símbolos C y D son los que menos aparecen. Entonces, el símbolo que más aparece se puede representar con un

número menor de bits y usar un número mayor de bits para los símbolos de menor frecuencia. En este caso, se puede utilizar el siguiente código:

$$A \longrightarrow 0$$

$$B \longrightarrow 10$$

$$C \longrightarrow 111$$

$$D \longrightarrow 110$$

con lo cual el código debe leerse como:

0010111110000101001100000

luego, el dato pasa de 32 a 25 bits, lo que representa un ahorro del 25 %.

CAPÍTULO 4

Manual del usuario y anexos

ImageZip

Codificación y comprensión de imágenes usando transformada wavelet

Versión 1, Revisión 0

Manual de uso

ImageZip es una herramienta de software desarrollada sobre Matlab y diseñada para ofrecer servicios básicos de compresión de imágenes usando transformada Wavelet. En su estado actual la herramienta tiene un propósito fundamentalmente educativo permitiendo a quien la usa explorar el efecto que distintas estrategias y técnicas tienen sobre la calidad y efectividad del proceso de compresión de una imagen.

La herramienta ha sido desarrollada en el marco del proyecto *Compresión de Imágenes con Wavelet* en la Universidad de San Buenaventura y como parte del trabajo de Maestría en Matemáticas Aplicadas de Hernán Salazar y Gloria Puetamán¹. En su desarrollo participó Jorge Zuluaga, profesor investigador adscrito al Instituto de Física de la Universidad de Antioquia.

ImageZip ha sido desarrollada usando GUIDE la herramienta para la creación de interfaces gráficas en Matlab. **ImageZip** requiere Matlab 7.0 para su exploración y ejecución.

El presente manual se estructura de la siguiente manera:

¹Hernan.Salazar@usbmed.edu.co Gloria.puetaman@usbmed.edu.co

- Sección 1. Instalación e Invocación
- Sección 2. Estructura de la interface gráfica
- Sección 3. Manejo básico de la herramienta
- Sección 4. Ejemplos de uso

4.1. Instalación e Invocación

ImageZip es un paquete formado por las siguientes componentes básicas:

- El archivo-M `imgzip.m`, donde se codifican todas las instrucciones utilizadas por el programa para realizar las tareas básicas de manipulación y representación de las imágenes.
- El archivo-fig `imgzip.fig` que contiene la información requerida por Matlab para construir la interface gráfica del programa.
- Un conjunto de rutinas especiales que realizan tareas específicas relacionadas con la manipulación y representación de las imágenes (`image*.m`) y otras que realizan las tareas propias del proceso de codificación y compilación.
- Imágenes de prueba que pueden ser utilizadas para reproducir los ejemplos presentados en este manual (directorio `images`)
- Manual del paquete **ImageZip.pdf**.

El paquete se distribuye para uso exclusivamente académico en la forma de un archivo-`zip` (`ImageZip-VV_vv.zip`) donde `VV_vv` es el número de versión (`VV`) y revisión de la herramienta (`vv`). La versión distribuida con este manual es la 1 con número de revisión 0, `ImageZip-1_0`.

La instalación de **ImageZip** es simple. En un directorio apropiado (se recomienda utilizar el directorio `MATLAB_INSTALLDIR/work`) se descomprime el paquete. Esta operación crea el directorio `ImageZip-VV_vv` que será utilizado como directorio base para la ejecución del programa.

Una vez instalado allí el programa puede ser invocado desde la línea de comando de Matlab con el siguiente procedimiento:

1. Abrir el entorno de trabajo de Matlab
2. Cambiar el directorio de trabajo (ver figura 4.1)

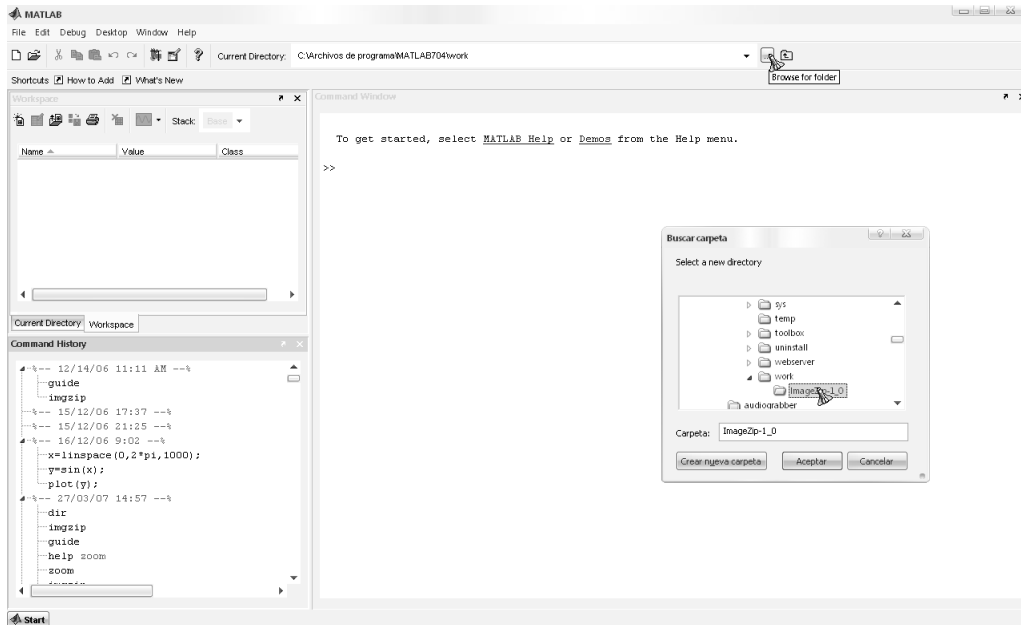


Figura 4.1: Cambio de directorio en Matlab

3. Escoger el directorio del paquete
4. Invocar la interface gráfica directamente desde la línea de comandos usando el comando `imgzip`

4.2. Estructura de la interface gráfica

La interface gráfica se estructura alrededor de 3 paneles: Parámetros de compresión (Panel 1), Área de Imágenes (Panel 2), Área de Estado (Panel 3).

A continuación se detallan las características y funciones de cada uno de los paneles y de sus componentes.

- **Panel 1. Parámetros de compresión.** Este panel contiene los formularios a través de los cuales se fijan los parámetros básicos del proceso de codificación y compresión. El panel incluye 4 formularios:
 - *Formulario de codificación básica.* Este formulario permite fijar el parámetro más básico de compresión: la profundidad de la imagen, es decir el número de bits con los que se codifican los niveles

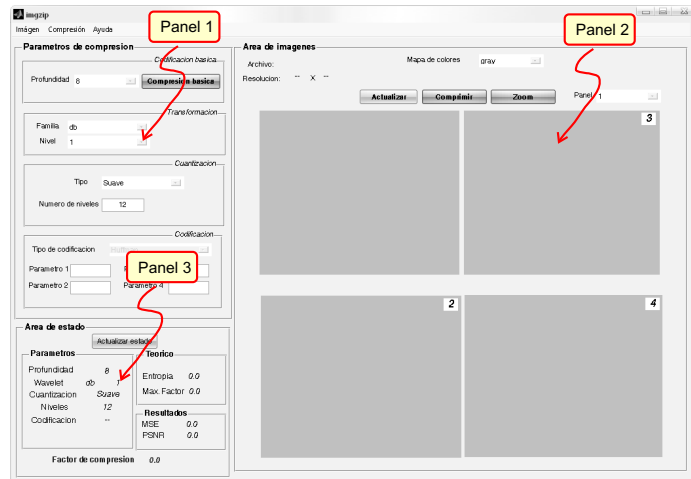


Figura 4.2: Interface gráfica de ImageZip. Identificación de los paneles

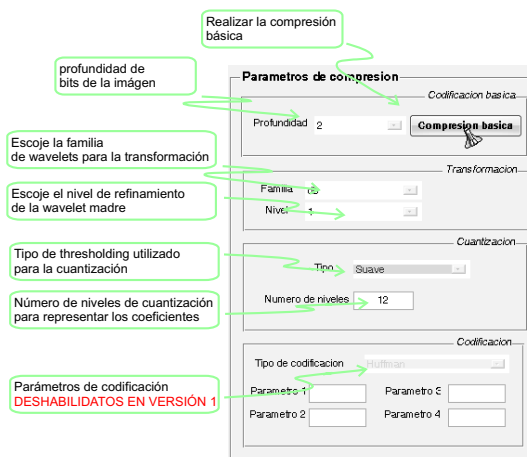


Figura 4.3: Panel 1. Parámetros de compresión

de intensidad en la imagen. En la presente versión se permite la elección de uno de 3 niveles posibles: 2, 4 y 8 bits por pixel (bpp).

- *Formulario de transformación.* Este formulario permite fijar los parámetros de la transformación wavelet a la que se somete la imagen para su posterior codificación. Dos parámetros se fijan en este punto: la familia wavelet que se quiere utilizar y el orden (Nivel) de la wavelet madre.
 - *Formulario de cuantificación.* En este formulario se escogen las características del proceso de cuantización (ver tesis). Es posible escoger el tipo de thresholding utilizado en la transformación de cuantización (hard thresholding o dura y soft thresholding o suave) y el número de niveles de cuantización que se utilizarán para codificar por niveles los valores de la transformada wavelet de la imagen.
 - *Formulario de codificación.* El algoritmo y las características del proceso de codificación son escogidas en este formulario. En la versión actual del programa (versión 1, revisión 0) este formulario esta inhabilitado. El programa utiliza el esquema natural de codificación binaria de los coeficientes wavelet pero para el cálculo del valor máximo de compresión se utilizan las características del esquema de codificación más eficiente teóricamente.
- **Panel 2. Área de imágenes.** Este panel esta reservado para desplegar las imágenes que resultan de la operación del programa. Esta formado por una área de información que despliega información sobre la imagen que esta siendo procesada, un menu desplegable para la elección del mapa de colores utilizado en la representación de las imágenes, unos botones de acción y un menú desplegable para la selección del área de imagen que desea utilizarse para presentar los resultados de una acción específica.
- El panel presenta 4 áreas de imagen distintas donde se pueden presentar los resultados de distintas operaciones sobre la imagen y compararlas (ver ejemplos de uso).
- **Panel 3. Área de estado.** Este panel esta reservado para la presentación de por un lado información sobre los parámetros elegidos para el proceso de codificación y compresión y de otro para la presentación de los resultados del proceso. El panel esta formado por 3 subpaneles:

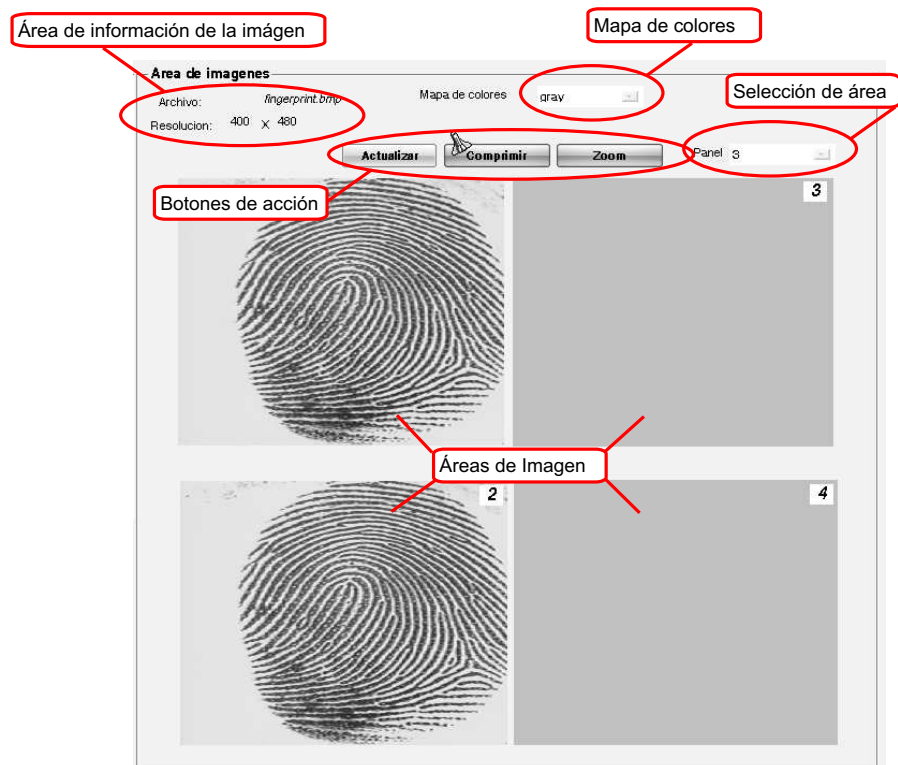


Figura 4.4: Panel 2. Área de imágenes

h

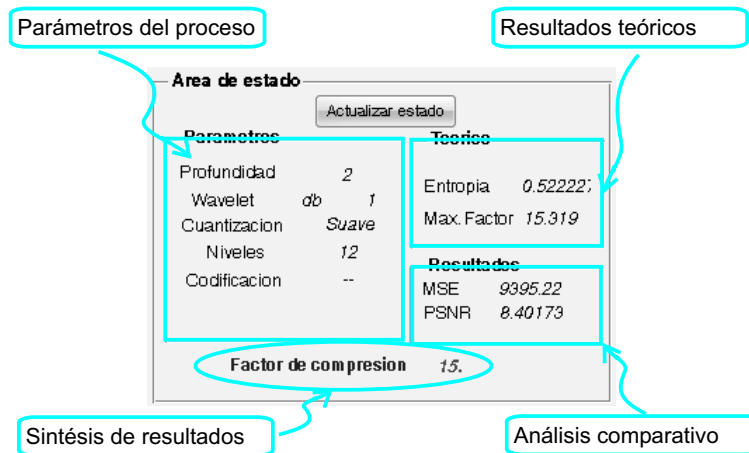


Figura 4.5: Panel 3. Área de estado

- *Parámetros*. Parámetros de codificación y compresión utilizados para la tarea que se ejecuta cuando se oprimen los botones de compresión básica o compresión (paneles 1 y 2).
- *Teórico*. Presenta 2 tipos de resultado. En primera instancia la entropía de la fuente de símbolos que genera la señal correspondiente a los coeficientes wavelet calculada después del proceso de cuantización. El segundo resultado presentado es el máximo factor de compresión que se conseguiría con un esquema de codificación máximamente eficiente. Este último parámetro sirve como referencia para evaluar la capacidad de la wavelet madre utilizada para representar las características de la imagen para comprimir.
- *Resultados*. Este subpanel presenta resultados de la evaluación comparativa de la imagen reconstruida a partir de los coeficientes wavelet cuantizados contra la imagen original. En el se presentan el valor del error cuadrático medio (Mean Square Error, MSE) y la razón señal pico a ruido (Peak Signal to Noise Ratio, PNSR).

Finalmente el panel de Área de Estado presenta como resumen del análisis de la imagen el valor del factor de compresión que se obtendría si se usará el más eficiente esquema de codificación.

4.3. Manejo básico de la herramienta

Se describen a continuación los procedimientos básicos requeridos para utilizar la herramienta, codificar y calcular las características básicas de compresión de una imagen.

4.3.1. Abrir Imagen

El primer paso consiste en cargar en memoria la imagen que se quiere procesar. Esta acción se realiza a través de la opción Abrir del menú Imagen. Por defecto el programa lista las imágenes con extensión bmp presentes en el directorio de instalación del paquete.

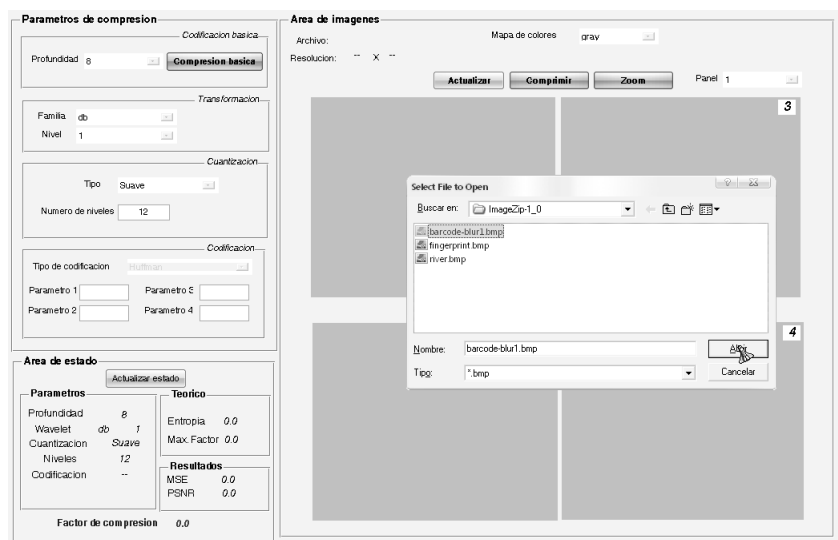


Figura 4.6: Abrir Imagen

Una vez la imagen es cargada en memoria el programa la coloca en la primera área de imagen del Panel 3. Cabe anotar que las imágenes procesadas por el paquete deben ser imágenes en escalas de grises.

4.3.2. Cambiar el mapa de colores

Para cambiar el mapa de colores utilizado en la representación de las imágenes se puede usar el menú desplegable del Panel 2 seguido del botón actualizar del mismo panel. El cambio del mapa de color permite en ocasiones mejorar las características visuales de una imagen o identificar patrones invisibles a simple vista.

4.3.3. Realizar una compresión básica

Es posible visualizar la imagen después de realizar una sencilla compresión consistente en cambiar la profundidad de bits (bits por pixel). La profundidad de bits determina el número de niveles de intensidad utilizados para representar la imagen.

Para visualizar el efecto de este procedimiento de compresión básica se debe proceder de la siguiente manera:

1. Abrir imagen
2. Seleccionar la nueva profundidad de bits
3. Presionar el botón de compresión básica

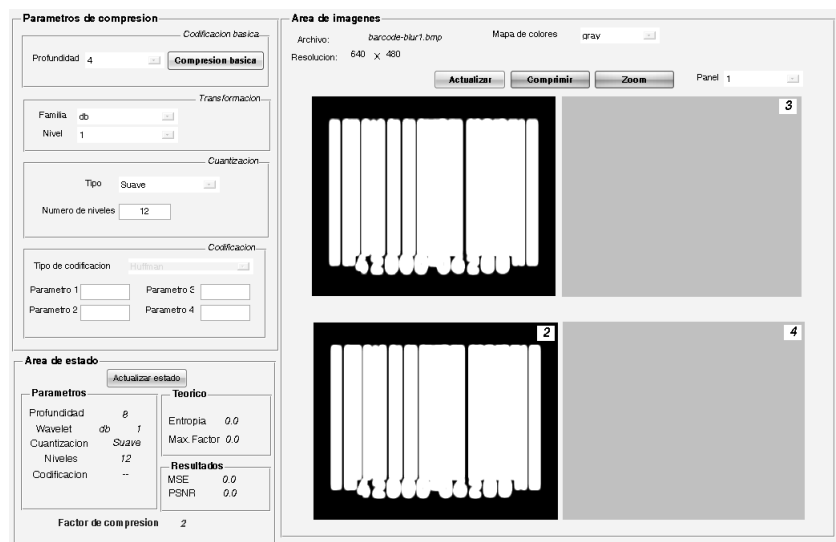


Figura 4.7: Compresión Básica

4.3.4. Realizar compresión

Para ejecutar la tarea básica de codificación y compresión se debe realizar el siguiente procedimiento:

1. Abrir imagen
2. Seleccionar la familia wavelet que se usará para la transformación



Figura 4.8: Compresión Completa

3. Seleccionar el orden de la wavelet
4. Escoger el tipo de thresholding a usarse
5. Escoger el número de niveles de cuantización
6. Escoger el panel donde se desea desplegar la imagen reconstruida después del proceso de compresión
7. Comprimir
8. Revisar los resultados

4.3.5. Zoom sobre las imágenes

Una vez realizada la compresión es posible realizar un zoom sobre las imágenes para comparar la calidad del proceso de codificación y compresión. El zoom se realiza presionando el botón correspondiente en el Panel 3. Una vez presionado el botón la acción de zoom se activa en todas las áreas de imagen permitiendo realizar la ampliación tanto en la imagen original como en las imágenes comprimidas.

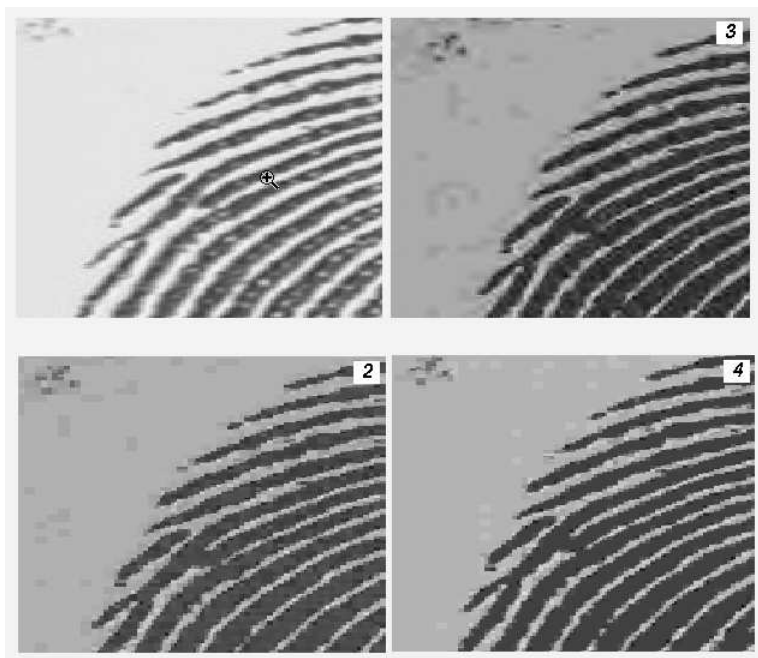


Figura 4.9: Resultados del zoom sobre la imagen original arriba a la izquierda y sobre las imágenes reconstruidas usando la familia de wavelets Daubechies de 3 órdenes distintos

4.4. Ejemplos de uso

En esta sección se ilustra la manera como la herramienta puede ser utilizada para estudiar el proceso de codificación y compresión de una imagen con propiedades espaciales distintas usando diversas familias de wavelets y distintos órdenes en cada familia.

En los ejemplos presentados se ilustra la aplicación de la herramienta para estudiar el proceso de compresión de tres imágenes distintas:

- Un código de barras (barcode.bmp) que representa una imagen con un patrón espacial repetitivo.
- Una huella digital (fingerprint.bmp) que si bien no tiene un patrón espacial tan simple como el de la imagen del código de barras presenta características geométricas distintas a las de imágenes más generales.
- Una imagen de satélite (river.bmp) del curso de un río y sus afluentes que tiene una complejidad espacial mucho mayor a la de las anteriores.

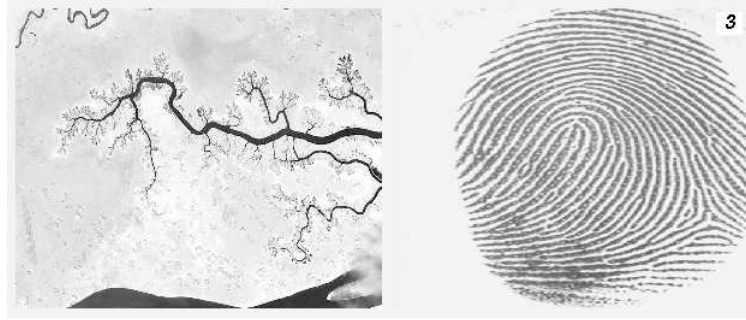


Figura 4.10: Dos de las imágenes utilizadas en los ejemplos, la huella digital (izquierda) y una imagen satelital de un río

Familia	db	db	bior 1.
Orden	1	3	1
Entropía	0.2	0.2	0.2
Máximo factor de compresión	40.1	40.1	40.0
MSE	873	2334	907
PNSR	18.8	14.5	18.6

Cuadro 4.1: Resultados de compresión de la imagen del código de barras usando diferentes familias de wavelets y distintos órdenes

4.4.1. Código de barras

Esta es una imagen muy simple de un código de barras en colores inversos. El código de barras representa un patrón geométrico repetitivo en una dirección específica de la imagen. Es de esperarse que dadas las propiedades espaciales “simples” de esta imagen su compresión pueda hacerse muy eficientemente con grandes factores de reducción en el tamaño.

En la tabla 4.1 se presentan los resultados de comprimir usando las familias disponibles en el paquete la imagen del código de barras

Se observa como el máximo valor de compresión no varía mucho al cambiar la familia. Sin embargo se ven variaciones apreciables al comparar las imágenes original y comprimida.

El factor de compresión es apreciable cuando se lo compara con el mismo factor para imágenes con propiedades espaciales menos regulares.

4.4.2. Huella digital

Una huella digital es una imagen que si bien no presenta propiedades espaciales tan simples como las de un código de barras la superposición de regiones claras y oscuras correspondientes a las hendiduras en la piel ofrecen una oportunidad para codificar eficientemente la información de la imagen a través de una transformación invertida como la transformada wavelet.



Figura 4.11: Imagen original de la huella digital (arriba a la izquierda) y sus reconstrucciones a partir de versiones comprimidas usando la wavelet Daubechies de orden 1 (abajo a la izquierda), 3 (arriba a la derecha) y 5

Al aplicar la herramienta a la imagen se obtuvieron factores de compresión entre 15 a 19 y entropías para la fuente de símbolos de los coeficientes de 0.5. Se observa la reducción en un factor de más de 2 en el factor máximo de compresión al compararlo con aquel obtenido para la imagen del código de barras, atribuible a las propiedades geométricas regulares de este último.

4.4.3. Imagen del río

La imagen del río es una imagen mucho mas compleja que las dos anteriores. Sin embargo al tratarla con el programa se obtienen factores máximos de compresión superiores a los de la huella digital

(entre 23 y 28). La razón de este hecho puede residir en que la imagen presenta zonas más amplias de intensidad aproximadamente constante que producen seguramente muchos más valores nulos de los coeficientes de la transformación.

4.5. Código Fuente

A continuación se presenta el código fuente contenido en el archivo-M del programa.

```
function varargout = imgzip(varargin)
% IMGZIP Comprime una imagen usando transformada Wavelet
%
%   IMGZIP es una interfaz grafica de usuario (GUI) creada para facili-
%   tar la compresión de imágenes usando transformada wavelet.
%
% See also: IMREAD, IMWRITE

% Last Modified by Jorge Zuluaga 13-Dec-2006 01:46:23

%=====
%INITIALIZACION - NO EDITE
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
gui_Singleton = 1; gui_State = struct('gui_Name',       mfilename,
...
                                     'gui_Singleton',  gui_Singleton, ...
                                     'gui_OpeningFcn', @imgzip_OpeningFcn, ...
                                     'gui_OutputFcn',  @imgzip_OutputFcn, ...
                                     'gui_LayoutFcn',   [] , ...
                                     'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end function imgzip_OpeningFcn(hObject, eventdata, handles,
varargin) handles.output = hObject; guidata(hObject, handles);
function varargout = imgzip_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%NO EDITE
%=====

%*****
%INITIALIZATION
%*****
set(handles.depth,'Value',3); set(handles.wavelet,'Value',1);
set(handles.wavelet_level,'Value',1);
set(handles.quantization,'Value',1);
set(handles.levels,'String',12); set(handles.grayscale,'Value',1);
set(handles.filename,'String',''); set(handles.panel,'Value',1);
update_state_Callback(hObject,eventdata,handles);
```



```

%*****
%CONTENIDO
%-SELECCION
%-RADIO
%-MENUS
%-BOTONES
%-RESTANTES
%*****

%*****
%SELECCION
%*****
%=====
%GRAYSCALE
%=====
%-----
function grayscale_Callback(hObject, eventdata, handles)
%-----
function grayscale_CreateFcn(hObject, eventdata, handles) if ispc
&& isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end set(hObject,'String',{'gray','pink','HSV'});

%=====
%WAVELET
%=====
%-----
function wavelet_Callback(hObject, eventdata, handles)
wavelet_index=get(hObject,'Value'); switch wavelet_index
    case 1
        set(handles.wavelet_level,'Enable','off');
    case 2
        set(handles.wavelet_level,'String',{'1','2','3'});
        set(handles.wavelet_level,'Enable','on');
end wavelet_string=get(hObject,'String');
set(handles.wavelet_val,'String',wavelet_string(wavelet_index));

%-----
function wavelet_CreateFcn(hObject, eventdata, handles) if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end set(hObject,'String',{'db','bior1.','rbio1.'});

%=====
%WAVELET LEVEL
%=====
%-----
function wavelet_level_Callback(hObject, eventdata, handles)
wavelet_level_index=get(hObject,'Value');
wavelet_level_string=get(hObject,'String');
set(handles.wavelet_level_val,'String',wavelet_level_string(wavelet_level_index));

%-----
function wavelet_level_CreateFcn(hObject, eventdata, handles) if
ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end set(hObject,'String',{'1','3','5'});

%=====

```

```

%DEPTH
%=====
%-----
function depth_Callback(hObject, eventdata, handles)
%-----
function depth_CreateFcn(hObject, eventdata, handles) if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end set(hObject,'String',{'2','4','8'});

%=====
%CODIFICACION
%=====
%-----
function coding_Callback(hObject, eventdata, handles)
%-----
function coding_CreateFcn(hObject, eventdata, handles) if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end set(hObject,'String',{'Huffman','Arithmetic'});
set(hObject,'Enable','off');

%=====
%QUANTIZACION
%=====
%-----
function quantization_Callback(hObject, eventdata, handles)
quantization_index=get(hObject,'Value');
quantization_string=get(hObject,'String');
set(handles.quantization_val,'String',quantization_string(quantization_index));

%-----
function quantization_CreateFcn(hObject, eventdata, handles) if
ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end set(hObject,'String',{'Suave','Dura'});

%=====
%PANEL
%=====
%-----
function panel_Callback(hObject, eventdata, handles)

%-----
function panel_CreateFcn(hObject, eventdata, handles) if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end set(hObject,'String',{'1','2','3','4'});

%*****
%RADIO
%*****

%*****
%BOTONES
%*****
%=====
%MUESTRA IMAGEN

```

```

%=====
function showImageMatrix(tarimage,depth,handles)
    %SELECCIONA EL MAPA DE COLORES
    colormap_sel_index=get(handles.grayscale,'Value');
    switch colormap_sel_index
    case 1;mapacolor=gray(2^depth);
    case 2;mapacolor=pink(2^depth);
    case 3;mapacolor=HSV(2^depth);
    end
    colormap(mapacolor);
    %SELECCIONA EL AXIS DONDE SE SITUARA LA IMAGEN
    panel_index=get(handles.panel,'Value');
    switch(panel_index)
    case 1;parentimg=handles.tarimage;
    case 2;parentimg=handles.zipimage2;
    case 3;parentimg=handles.zipimage3;
    case 4;parentimg=handles.zipimage4;
    end
    %SITUA LA IMAGEN
    zi=image(tarimage,'Parent',parentimg);
    %ELIMINA EL BORDE Y LOS TICS
    set(parentimg,'Visible','off');

function showImageFile(tarfile,handles)
    tarinfo=imfinfo(tarfile);
    depth=tarinfo.BitDepth;
    set(handles.depth_val,'String',depth);
    tarimage=imread(tarfile);
    showImageMatrix(tarimage,depth,handles)

%-----
function update_Callback(hObject, eventdata, handles)
    tarfile=get(handles.filename,'String');
    showImageFile(tarfile,handles);
    %=====
    %ZIP
    %=====
    %-----
function [name] = getWaveletName(handles)
    wavelet_index=get(handles.wavelet,'Value');
    wavelet_string=get(handles.wavelet,'String');
    wavelet=wavelet_string(wavelet_index);

    wavelet_level_index=get(handles.wavelet_level,'Value');
    wavelet_level_string=get(handles.wavelet_level,'String');
    level=wavelet_level_string(wavelet_level_index);
    name=strcat(wavelet,level);

%*****
%COMPRESSION
%*****
%-----
function zipbutton_Callback(hObject, eventdata, handles)
    tarfile=get(handles.filename,'String');
    %LEE IMAGEN
    tarimage=imread(tarfile); tarinfo=imfinfo(tarfile);
    %PROMEDIA IMAGEN
    tarimage=tarimage-mean(mean(tarimage));
    %CALCULA TRANSFORMADA WAVELET
    name=getWaveletName(handles)
    [C,S]=wavedec2(single(tarimage),2,char(name));
    %CALCULA PROPIEDADES CUANTIZACION

```

```

qmax=str2num(get(handles.levels,'String')); L=max(abs(C));
del=2*L/qmax; a=-L+(del/2):del:-(3/2)*del;
b=(3/2)*del:del:L-(del/2); z=[a 0 b]; w=[(del/2)*ones(1,length(a))
del (del/2)*ones(1,length(b))];
%CUANTIZA
for i=1:length(z)
    H(i)=sum(abs(C-z(i))<=w(i));
end s=tarinfo.BitDepth; eps=1e-17; x=H/length(C)+eps;
ent=sum(-x.*log2(x)); rmax=s/ent;
%DECUANTIZA
len=length(C); TMP=C; for i=1:len
    for j=1:length(z)
        if abs(TMP(i)-z(j))<=w(j) TMP(i)=z(j);
        end
    end
end
%RECONSTRUYE LA IMAGEN
zipimage=waverec2(TMP,S,char(name)); m=min(min(zipimage));
M=max(max(zipimage)); zipimage=(255*(zipimage-m)/(M-m))+1;
%CALCULA ERRORES
difimage=(zipimage-single(tarimage)).*(zipimage-single(tarimage));
sumimage=sum(sum(difimage)); sizeimage=size(tarimage);
MSqE=sumimage/(sizeimage(1)*sizeimage(1));
PeakNSR=20*log10(255/sqrt(MSqE));
%MUESTRA RESULTADO
showImageMatrix(zipimage,8,handles);
%CAMBIA VARIABLES DE ESTADO
set(handles.entropy,'String',ent);
set(handles.maxfactor,'String',rmax);
set(handles.mersq,'String',MSqE);
set(handles.psnr,'String',PeakNSR);
set(handles.factor,'String',rmax);

%=====
%BASIC COMPRESS
%=====
%-----
function bascompress_Callback(hObject, eventdata, handles)
tarfile=get(handles.filename,'String'); tarimage=imread(tarfile);
zi=image(tarimage,'Parent',handles.zipimage2);
set(handles.zipimage2,'Visible','off');
levels_text=get(handles.levels,'String');
set(handles.levels_val,'String',levels_text);
depth_index=get(handles.depth,'Value'); switch depth_index
    case 1;depth=2;
    case 2;depth=4;
    case 3;depth=8;
end tarinfo=imfinfo(tarfile); depth_ref=tarinfo.BitDepth;
mapacolor=gray(2^depth); colormap(mapacolor);
%CALCULO DEL FACTOR DE COMPRESION
factor=depth_ref/depth; set(handles.factor,'String',factor);

%=====
%UPDATE STATE
%=====
%-----
function update_state_Callback(hObject, eventdata, handles)
depth_index=get(handles.depth,'Value');
depth_string=get(handles.depth,'String');
set(handles.depth_val,'String',depth_string(depth_index));

wavelet_index=get(handles.wavelet,'Value');

```

```

wavelet_string=get(handles.wavelet,'String');
set(handles.wavelet_val,'String',wavelet_string(wavelet_index));

wavelet_level_index=get(handles.wavelet_level,'Value');
wavelet_level_string=get(handles.wavelet_level,'String');
set(handles.wavelet_level_val,'String',wavelet_level_string(wavelet_level_index));

quantization_index=get(handles.quantization,'Value');
quantization_string=get(handles.quantization,'String');
set(handles.quantization_val,'String',quantization_string(quantization_index));

levels_text=get(handles.levels,'String');
set(handles.levels_val,'String',levels_text);

%*****
%INPUT BOXES
%*****
%-----
function levels_CreateFcn(hObject, eventdata, handles) w=1
%-----
function levels_Callback(hObject, eventdata, handles)
set(handles.levels_val,'String',get(hObject,'String')); w=1

%*****
%MENU ACTIONS
%*****
% -----
function OpenItem_Callback(hObject, eventdata, handles) tarf =
uigetfile('images/*.bmp'); tarfile=strcat('images/',tarf); if
~isequal(tarfile, 0)
    showImageFile(tarfile,handles);
    tarinfo=iminfo(tarfile);
    set(handles.width,'String',tarinfo.Width);
    set(handles.height,'String',tarinfo.Height);
else
    errordlg('El archivo no existe!','Error');
end set(handles.filename,'String',tarfile);

% -----
function SaveReconstrItem_Callback(hObject, eventdata, handles)
helpdlg('La opción esta deshabilitada');

% -----
function ExitItem_Callback(hObject, eventdata, handles)

% -----
function ImageMenu_Callback(hObject, eventdata, handles)

% -----
function SaveCompressItem_Callback(hObject, eventdata, handles)
helpdlg('La opción esta deshabilitada');

% -----
function CompressionMenu_Callback(hObject, eventdata, handles)

% -----
function CompressItem_Callback(hObject, eventdata, handles)
helpdlg('La opción esta deshabilitada');

% -----
function Comprimir_Callback(hObject, eventdata, handles)
helpdlg('La opción esta deshabilitada');

```

```
% -----
function HelpItem_Callback(hObject, eventdata, handles)
helpdlg('La opción esta deshabilitada');

% -----
function HelpMenu_Callback(hObject, eventdata, handles)

% -----
function AboutItem_Callback(hObject, eventdata, handles)
helpdlg('Image Zip, Version 1, Revision 0');

%*****
%RESTANTES
%*****
function param1_CreateFcn(hObject, eventdata, handles) function
param2_CreateFcn(hObject, eventdata, handles) function
param3_CreateFcn(hObject, eventdata, handles) function
param4_CreateFcn(hObject, eventdata, handles) function
edit1_CreateFcn(hObject, eventdata, handles) function
edit1_Callback(hObject, eventdata, handles) function
edit2_CreateFcn(hObject, eventdata, handles) function
edit3_CreateFcn(hObject, eventdata, handles) function
edit4_CreateFcn(hObject, eventdata, handles)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%NEW CODE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% --- Executes on button press in zoom.
function zoom_Callback(hObject, eventdata, handles)
% hObject   handle to zoom (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
zoom
```

Bibliografía

- [1] Aldroubi, A. The wavelet transform: A surfing guide, pp 3-36 in *Wavelets in Medicine and Biology*, A. Audroubi, M. Unser (eds.), CRC Press, New York 1996.
- [2] Berlekamp, E. *Algebraic Coding Theory*. Mc Graw Hill, New York, 1968.
- [3] Billingsley, P. *Probability and Measure*, 3rd edition. John Wiley & Sons, Inc. New York, 1995.
- [4] Boggess, A., Narcowich, F. J. *A First Course in Wavelets with Fourier Analysis*. Prentice Hall, New Jersey, 2001.
- [5] Bonani, A., Durand, S. and Weiss, G. Wavelets obtained by continuous deformations of the Haar wavelet, *Revista Mat. Iberoamericana*, Vol. 12, No 1, 1996.
- [6] Burrus, C. S., Gopinath, R. A., Guo, H. *Introduction to Wavelets and Wavelets Transforms A Primer*. Prentice Hall, New Jersey, 1998.
- [7] Chambolle, A., DeVore, R. A., Lee, N. Y. and Lucier, B. J. Nonlinear Wavelet Image Processing: Variational Problems, Compression and Noise Removal through Wavelet shrinkage, *IEEE Trans. on Image Processing.*, vol 7, 1998, pp 319-335.
- [8] Chui, C. K. *An Introduction to Wavelets*. Academic Press, Boston, 1992.
- [9] Chui, C. K. *Wavelets: A Mathematical Tool for Signal Analysis*. SIAM Monographs on Mathematical Modeling and Computation, Philadelphia, 1997.

- [10] Chung, K. L. *Elementary Probability Theory with Stochastic Processes*. Springer-Verlag. New York, 1979.
- [11] Clarke, R. J. *Transform Coding of Images*. Academic Press, San Diego, CA, 1985.
- [12] Clarke, R. J. *Digital Compression of Still Images and Video*. Academic Press, San Diego, CA, 1995.
- [13] Cryer, J. D. *Time Series Analysis*, Duxbury Press. Boston, 1986.
- [14] Daubechies, I. Orthonormal bases of compactly supported wavelets, *Comm. Pure Appl. Math.*, 41 1988, 909-996.
- [15] Daubechies, I. *Ten Lectures on Wavelets*, CBMS Series 61, SIAM, Philadelphia 1992.
- [16] Daubechies, I. The wavelet transform, time-frequency localization and signal analysis, *IEEE Trans. Inform. Theory*, vol. 36 1990, 961-1005.
- [17] Delyon, B. and Juditsky, A. Estimating wavelets coefficients, pp 151-168 in *Wavelets and Statistics*, Springer-Verlag, Tokyo, 1995.
- [18] DeVore, R. A., Jawerth, B. and Lucier, B. J. Image Compression through Wavelet transform Coding, *IEEE Trans. Information Theory.*, vol 38, 1992, pp 719-746.
- [19] DeVore, R. A., Jawerth, B. and Popov, V. Compression of wavelet decomposition. *Americ. J. of Math.*, 114, 1992, pp 737-785.
- [20] Donoho, D. *Nonlinear Wavelet Methods for Recovery of Signals, Densities and Spectra from Indirect and Noisy Data*. "Different Perspectives on Wavelets", Proceedings of Symposia in Pure Math., AMS., I Daubechies, Edt., 47, 173-205, 1993.
- [21] Folland, G. B. *Real Analysis*, 2nd edition. John Wiley & Sons, Inc. New York, 1999.
- [22] Gabor, D. Theory of communications, *J. Inst. Elect. Eng. London*. 93(III) 1946, 429-457.
- [23] Gersho, A. and Gray R. M. *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, 1992.
- [24] Gonzalez, R. C. and Woods, R. E. *Digital Image Processing*. Addison-Wesley, New York, 1992.

-
- [25] Hamilton, J. D. *Time Series Analysis*. Princeton University Press, Princeton, New Jersey, 1994.
- [26] Hamming, R. W. Error detecting and error correcting codes, *Bell System Tech. J.*, vol. 29, 1950, 147-160.
- [27] Hernández, E. Weiss, G. *A First Course on Wavelets*. CRC Press, Boca Raton, FL, 1996.
- [28] Herstein, I. N. *Topics in Algebra*, 2nd edition, John Wiley & Sons, Inc. New York, 1975.
- [29] Huffman, D. A. A Method for Construction of Minimum Redundancy Codes, *Proc. IRE*, vol. 40, no. 10 1952, 1098-1101.
- [30] Irino, T., Kawahara, H. Signal reconstruction from modified auditory wavelet transform, *IEEE Trans. Signal Process.*, vol 41, 1993, 3549-3553.
- [31] Mallat, S. Multiresolution approximations and wavelet orthonormal bases for $L^2(\mathbb{R}^d)$, *Trans. of Amer. Math. Soc.* 315, 1989, 69-87.
- [32] Mallat, S. A theory for multiresolution signal decomposition: The wavelet representation, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, 1989, 674-693.
- [33] Mallat, S. *A Wavelets Tour of Signal Processing*. Academic Press, New York, 1998.
- [34] Meyer, Y. *Ondelettes et opérateurs, I: Ondelettes*. Herman, Paris, 1990.
- [35] Meyer, Y. Wavelets and functions with bounded variation from image processing to pure mathematics. *Rend. Mat. Acc. Lincei*. 9 fasc. spec. 2000, 77-105.
- [36] Miano, J. *Compressed Image File Formats: JPEG, PNG, GIF, XBM, BMP*. Addison-Wesley, New York, 1999.
- [37] Oppenheim, A., Shafer, R. *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [38] Percival, D. B., Walde, A. T. *Wavelet Methods for Time Series Analysis*, Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, 2000.
- [39] Pinsky, M. A. *Introduction to Fourier Analysis and wavelet*, Brooks/Cole, NJ, 2001.

- [40] Poli, A. and Llorenc, H. *Error Correcting Codes*. Prentice Hall Mason, Paris, 1992.
- [41] Proakis, J. G., Manolakis, D. G. *Signal Processing: Principles, Algorithms and Applications*, Macmilan, New York, 1992.
- [42] Quak, E. and Weyrich, N. Decomposition and reconstruction algorithms for spline wavelet on a bounded interval, *Appl. and Comp. Harmonic Anal.* (ACHA), 1 1994, 217-231.
- [43] Rioul, O., Duhamel, P. Fast algorithms for discrete and continuous wavelet transform, *IEEE Trans. Inform. Theory*, vol. IT-38 1992, 569-586.
- [44] Roman, S. *Introduction to Coding and Information Theory*. Springer - Verlag, Berlin, 1997.
- [45] Ross, S. M. *Introduction to Probability Models*, 8nd edition. Academic Press. New York, 2003.
- [46] Shannon, C. E. A mathematical theory of communication, *Bell System Tech. J.*, vol. 28, 1948, 379-423, 623-656.
- [47] Shapiro, J. M. Embedded image coding using zerotrees of wavelet coefficients, *IEEE Trans. Signal Proc.* 41 1993, 3445-3462.
- [48] Shen, X. A Galerkin-wavelet method for a singular convolution equation on the real line, *J. Int. Equa. Appl.* 12 2000, 157-176.
- [49] Strang, G. and Nguyen, T. *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Cambridge, MA, 1996.
- [50] Stremmler, F. G. *Introduction to Communication Systems*, 2nd edition. Addison-Wesley Publishing Company, Inc. Massachusetts, 1982.
- [51] Sun, W. and Zhou, X. Sampling theorem for wavelet subspaces: error estimate and irregular sampling, *IEEE Trans. Signal Process.*, vol. 48 2000, 223-226.
- [52] Togneri, R. and DeSilva, C. J. S. *Fundamentals of Information Theory and Coding Design*. Chapman & Hall/CRC Boca Raton, 2002.
- [53] Triebel, H., *Fourier Analysis and Function Spaces*. Teubner-Texte Math. 7, Leipzig: Teubner. 1977.

-
- [54] Unser, M. and Aldroubi, A. B-spline processing I: Theory and II: Efficient design and application, *IEEE Trans. Signal Process.*, vol 41 1993, 821-848.
- [55] Unser, M. A practical guide to implementation of the wavelet transform, pp 37-76 in *Wavelets in Medicine and Biology*, A. Audroubi, M. Unser (eds.), CRC Press, New York 1996.
- [56] Unser, M. and Blu, T. Mathematical Properties of the JPEG2000 Wavelet Filters, *IEEE Trans. Image Processing*, vol 12 2003, 1080-1090.
- [57] Walnut, D. *An Introduction to Wavelets Analysis*. Birkhäuser, Boston, 2002.
- [58] Walter, G. G. Approximation with impulse trains, *Results Math.* 34 1998, 185-196.
- [59] Walter, G. G. and Shen, X. A substitute for summability in wavelet expansions, *Appl.Numer. Harmon. Anal.*, Birkhäuser Boston, Boston, MA 1999, 51-63.
- [60] Walter, G. G. and Shen, X. Deconvolution using Meyer wavelets, *J. Integral Equations Appl.* 11 1999, 515-534.
- [61] Walter, G. G., Shen, X. *Wavelets and Other Orthogonal Systems*, 2nd edition. Chapman & Hall/CRC, Boca Raton, FL, 2001.
- [62] Wei, W. W. S. *Time Series Analysis*, 2nd edition. Addison-Wesley Publishing Company, Inc. Boston, 2006.
- [63] Yamada, K., Ohkitani, K. Orthonormal wavelet expansion and its application to turbulence, *Progr. Theoret. Phys.*, vol. 83-5, 1990, 819-823.
- [64] Yamada, K., Ohkitani, K. An identification of energy cascade in turbulence by orthonormal wavelet analysis, *Progr. Theoret. Phys.*, vol. 84-4, 1991, 799-815.
- [65] Yamada, K., Ohkitani, K. Orthonormal wavelet analysis of turbulence, *Fluid Dynam. Res.*, vol. 8, 1991, 101-115.

Índice alfabético

- alfabeto, 48
- algoritmo
 - de casacada, 35
 - de descomposición, 32
 - de reconstrucción, 36
 - piramidal, 35
- Análisis Multirresolución, 23
- código , 50
 - de grupo, 51
 - algebraico, 50
 - de Huffman, 55
 - sistemático, 53
- canal simétrico, 49
- convolución, 4
- downsampling, 34
- ecuación de escala, 27
- ecuaciones
 - de verificación de paridad, 52
- entropía, 48
- filtro , 8
 - paso bajo, 9
- Fourier
 - coeficientes de, 7
 - fórmula de inversión de, 5
 - serie de, 6
 - transformada de, 5
- fuelle de información, 48
- función
 - característica, 4
 - codificadora, 50
 - de escala, 23, 27
 - construcción de, 29
 - dilatada, 17
 - trasladada, 17
- funciones
 - cuadrado integrable, 3
 - ortogonales, 3
- Hamming
 - distancia, 50
 - peso, 50
- imagen observada, 40
- matriz
 - de verificación de paridad, 52
 - generadora, 52
- palabra codificada, 50
- Parseval
 - fórmula de, 6
- patrón de error, 51
- Plancharel
 - fórmula de, 6
- probabilidad , 10
 - espacio de, 11

señal, 7, 15

Shannon, 7

sistema lineal, 8

upampling, 36

wavelet , 16

coeficientes, 22

fórmula de inversión, 18

fórmula de Parseval, 19

fórmula de Plancherel, 17

serie, 21

transformada continua, 17

transformada discreta, 21