Universidad de Valladolid

FACULTAD DE CIENCIAS

TRABAJO FIN DE GRADO

Grado en Matemáticas

El sistema de Haar y sus aplicaciones

Autor: Carlos García-Lago Riego
Tutor: Ángel Durán Martín
2024-2025

A mi familia y amigos más cercanos, y a mi tutor, Ángel,
por apoyarme en todo momento.

Abstract

Las wavelets permiten descomponer una señal en sus diferentes componentes de frecuencias y estudiar cada uno de estos componentes con una resolución adecuada a su escala. Este trabajo construye la ondícula basada en el sistema de Haar. Se proporciona una explicación de los algoritmos de descomposición y de reconstrucción, que son la base de la transformada discreta de Haar, también explicada en detalle. En la última sección, se presentan varias implementaciones y ejemplos tanto en una como en dos dimensiones.

Wavelets allow to decompose a signal into different frequency components in order to study each of these components with a resolution according to its scale. This work constructs the wavelet based on the Haar system. It provides an explanation of the decomposition and reconstruction algorithms, which are the basis of the discrete Haar transformation, also explained in detail. In the last section, several implementations and examples in both one and two dimensions are presented.

Keywords

Sistema de Haar, Transformada discreta de Haar, Análisis Multiresolución

Haar system, Haar discrete transform, Multiresolution analysis

Índice general

Capítulo 1

Introducción:

1.1. Sistemas reproductores de señales

Este trabajo se enfoca en analizar la construcción del sistema de Haar y exponer algunas de sus aplicaciones en el proceso de señales. En esta introducción, basada en [8] y [9], presentamos el sistema, tanto en su contexto histórico como en el contexto de la teoría de representación de funciones mediante ondículas o wavelets.

Una de las herramientas para el procesado de señales consiste en la representación de funciones mediante partículas elementales o sistemas reproductores. Ejemplos de estos sistemas son los monomios $\{(x-a)^n, n = 0, 1, ..\}$, a partir de los cuales se representan los desarrollos de Taylor, o las funciones trigonométricas $\{e^{inx}\}_{n=-\infty}^{\infty}$, que sirven para construir los desarrollos de Fourier. En general, cualquier base ortonormal del correspondiente espacio de funciones es un sistema reproductor, utilizando los elementos de la base como partículas elementales.

Los primeros sistemas reproductores para el procesado de señales surgieron tomando como referencia la transformada de Fourier,

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(t)e^{-i\xi t}dt, \quad f \in L^1(\mathbb{R}), \tag{1.1}$$

que permite descomponer una señal f en cada una de las frecuencias $\hat{f}(\xi)$ que la componen. Para ello, hace uso de la transformada inversa

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{f}(\xi)e^{-i\xi t}d\xi. \tag{1.2}$$

La transformada de Fourier es la base del teorema de muestreo de Shannon, [14], que establece que toda función, o señal, f integrable en $\mathbb{R}$ y cuya transformada de Fourier tiene soporte en un intervalo del tipo $[-B\pi, B\pi]$, puede representarse como

$$f(t) = \sum_{k=-\infty}^{\infty} f(\frac{k}{B})\psi_k(t), \quad \psi_k(t) = \frac{sen\pi(Bt-k)}{\pi(Bt-k)}, \tag{1.3}$$

donde la convergencia de las k sumas parciales de la serie es en la norma de L^2. Bajo las hipótesis del teorema de Shannon, la expresión (1.3) permite representar f a partir de las partículas elementales del sistema reproductor

$$\{\psi_k(t) : k = 0, \pm 1, \pm 2, ...\}, \tag{1.4}$$

con los coeficientes dados por la muestra $f(\frac{k}{B})$ para $k = 0, \pm 1, \pm 2, ...$

La información que se obtiene aplicando la transformada de Fourier para señales f con frecuencia que cambia con el tiempo es limitada. Una alternativa para solventar este problema fue propuesta por Gabor. Esta alternativa consiste en utilizar la transformada de Fourier con ventana g,

$$S_g(f)(t,\xi) = \int_{-\infty}^{\infty} f(x)g(x-t)e^{-2\pi i x\xi}dx, \tag{1.5}$$

con $g \in L^2(\mathbb{R})$ bien localizada. El núcleo de integración de (1.5), $g_{t,\xi} = g(x-t)e^{-ix\xi 2\pi}$ está formado por una función de amplitud fija trasladada y modulada con funciones sínusoidales, por lo que no permite analizar variaciones más pequeñas que la amplitud de la ventana g.

El núcleo de integración puede discretizarse formando el sistema reproductor

$$\{g_{n,m}(x) = g(x-n)e^{2\pi i m x}, n, m \in \mathbb{Z}\}, \tag{1.6}$$

llamado sistema de Gabor, [6]. Cuando $g(x) = \chi_{[0,1]}(x)$, el sistema de Gabor es una base ortonormal de $L^2(\mathbb{R})$. Para que (1.6) sea un sistema reproductor, el teorema de Balian-Low, [1], impone una restricción sobre g,

$$(\int_{-\infty}^{\infty} x^2|g(x)|^2dx)(\int_{-\infty}^{\infty} |\hat{\xi}g(\hat{\xi})^2dx|) = \infty, \tag{1.7}$$

de manera que, o bien g, o bien su transformada $\hat{g}$, no pueden decaer más rápido que $\frac{1}{x^2}$. Esta es la razón por la que la ventana propuesta por Gabor, $g(x) = \frac{1}{\sqrt{\pi}}e^{-x^2}$, no genera un sistema reproductor, porque las integrales de (1.7) para g son ambas finitas.

Por otra parte, las singularidades de una función son más fácilmente detectables cuando se incorporan dilatadores a las ventanas. Este es el origen de la transformada continua de ondículas o transformada wavelet de una señal f,

$$W_\psi(f)(t,s) = \int_{-\infty}^{\infty} f(x)\psi_{s,t}(x)dx = \int_{-\infty}^{\infty} f(x)\sqrt{s}\psi(sx-t)dx, \tag{1.8}$$

con núcleo $\sqrt{s}\psi(sx-t)$, $t \in \mathbb{R}$, $s > 0$.

Ejemplos clásicos son las funciones de Morlet, que introdujo el concepto de transformada de ondículas y su formulación matemática,

$$\psi(x) = \pi^{-1/4}e^{-iw_0x}e^{-x^2/2}, \tag{1.9}$$

o el sombrero mejicano,

$$\psi(x) = \frac{2}{\sqrt{3}}\pi^{-1/4}(1-x^2)e^{-x^2/2}. \tag{1.10}$$

Para discretizar la transformada wavelet (1.8) se pueden usar traslaciones enteras y dilataciones de la forma 2^j, $\quad j \in \mathbb{Z}$, obteniéndose

$$W(\psi) = \{\psi_{j,k}(x) = 2^{j/2}\psi(2^jx-k), j,k \in \mathbb{Z}\}. \tag{1.11}$$

Cuando (1.11) es un sistema reproductor, se dice que la función ψ es una ondícula o wavelet ortonormal de $L^2(\mathbb{R})$. Los ejemplos más sencillos son

$$\psi = \chi_{[0,1/2)} - \chi_{[1/2,1)}, \tag{1.12}$$

dando lugar al sistema de Haar, [7], objeto de estudio de este trabajo, y el sistema de Shannon,

$$\hat{\psi}(\xi) = \chi_{[-1,-1/2)} - \chi_{[1/2,1)}, \tag{1.13}$$

relacionada con (1.3).

La idea de utilizar sistemas reproductores de la forma (1.11) fue inicialmente propuesta por Morlet en 1975. En 1985, el matemático francés Yves Meyer descubrió que podrían construirse sistemas reproductores de la forma (1.11) con $g = \psi$, no cumpliendo (1.7), las llamadas ondículas de Lemarié-Meyer, [10].

La construcción de los wavelets de Lemarié-Meyer fue generalizado por S.Mallat e Yves Meyer en el Análisis Multiresolución o AMR, quienes en 1989 publican esta estructura con la que se pueden obtener ondículas. Un AMR es una colección de subespacios lineales cerrados $\{V_j : j \in \mathbb{Z}\} \subset L^2(R)$ verificando las propiedades:

1. $V_j \subset V_{j+1}, \quad j \in \mathbb{Z}$.
2. $f \in V_j \Leftrightarrow f(2(.)) \in V_{j+1}, \quad j \in \mathbb{Z}$.
3. $\cap_{j\in\mathbb{Z}} V_j = \{0\}$.
4. $\overline{\cup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R})$.
5. Existe $\psi \in V_0$ (función de escala) tal que $\{\psi(. - k) : k \in \mathbb{Z}\}$ es una base ortonormal de V_0.

La construcción de una ondícula ψ a partir de un AMR se basa en la propiedad de que $\{\psi(. - k) : k \in \mathbb{Z}\}$ sea una base ortonormal de W_0, siendo W_0 el complemento ortogonal de V_0 en V_1, [7]. Para ello, es necesario utilizar la función de escala ψ y el llamado filtro de paso bajo m_0, definido a partir de la identidad $\hat{\psi}(2\xi) = m_0(\xi)\hat{\psi}(\xi)$. S. Mallat define ψ a partir de su transformada

$$\hat{\psi}(2\xi) = e^{i\xi/2}\overline{m_0(\xi/2 + \pi)}\hat{\psi}(\xi/2), \quad \xi \in \mathbb{R}. \tag{1.14}$$

Véase [11], [12], [3]. En el caso del sistema de Haar, V_j será el subespacio de $L^2(\mathbb{R})$ de funciones constantes en intervalos de la forma $[2^{-j}k, 2^{-j}(k + 1)], \quad k \in \mathbb{Z}$, con $\varphi(x) = \chi_{[0,1]}(x)$. En tal caso, puede comprobarse que

$$m_0(\xi) = \frac{1}{2}(1 + e^{i\xi}) \tag{1.15}$$

y $\psi(x) = \varphi(2x) - \varphi(2x - 1)$. Se tiene que (1.15) es un filtro de respuesta finita, es decir, su representación en serie de Fourier tiene un número finito de coeficientes. Esto lo hace preferible para la implementación y las aplicaciones, [7]. Pero la wavelet ψ no es continua, por lo que puede no representar bien señales regulares. La construcción de filtros de respuesta finita que producen ondículas suaves es debida a Ingrid Daubechies, [5]. La regularidad de las ondículas, que son de soporte compacto, aumenta con el número de coeficientes no nulos del filtro. Una variante de estas ondículas son las que se usan actualmente para la compresión de imágenes utilizando el algoritmo JPEG2000, [13], [15].

1.2. Estructura del TFG

El sistema de Haar, que se ha presentado en la sección anterior en el contexto del AMR, se va a describir en profundidad en esta memoria, aunque con un enfoque diferente. Se va seguir un enfoque más cercano al trabajo original de Haar, [7]. En el capítulo 2, se construye el sistema, primero en el intervalo $[0, 1]$, y luego se generaliza a $\mathbb{R}$, donde la estructura AMR se menciona implícitamente. Los algoritmos de descomposición y de reconstrucción que se describen en el capítulo 2, serán la base del capítulo 3, donde se analizará la transformada discreta de Haar en una y dos dimensiones. Finalmente, en el capítulo 4, se ilustran numéricamente varios ejemplos sobre algunos usos del sistema de Haar, y la transformada discreta de Haar, tanto en una como en dos dimensiones.

Capítulo 2

El sistema de Haar

2.1. Sistema de Haar en $\Omega = [0, 1]$

En esta sección, se va a formalizar el sistema de Haar, [7], en el intervalo $\Omega = [0, 1]$. El sistema de Haar es un sistema ortonormal en Ω. También, construiremos la base de Haar, que es el ejemplo más sencillo de una base ortonormal para wavelets. Las funciones de la base de Haar tienen soporte en subintervalos uniformes de Ω y son funciones escalonadas con discontinuidades. La importancia del sistema de Haar y la base de Haar radica en la representación de funciones con segmentos suaves y de variación lenta, interrumpidos por picos bruscos y discontinuidades. El primer objetivo será definir el sistema de Haar, así como la construcción de la base de Haar en el intervalo Ω, [4]. Durante dicha construcción, también introduciremos más conceptos necesarios para comprender el análisis multiresolución y la construcción de bases wavelet generales.

Definición 2.1. *Para $j \geq 0$ fijo, se define una partición uniforme Ω_j^k en el intervalo $\Omega = [0, 1]$ y con $k = 0, 1, ..., 2^j - 1$ donde*

$$\Omega_j^k = [2^{-j}k, 2^{-j}(k + 1)), \tag{2.1}$$

para $k = 0, 1, ..., 2^j - 1$. La longitud de los intervalos es 2^{-j}.

A continuación, se va a ilustrar gráficamente la generación de los intervalos definidos en (2.1). De forma gráfica, podemos apreciar cómo se construye la partición de $\Omega = [0, 1]$. La construcción de los intervalos representados en la figura 2.1 permite deducir las siguientes propiedades:

- $\Omega_j^k \cap \Omega_j^{k'} = \emptyset$ con $0 \leq j$, $k \neq k'$ y $k, k' = 0, 1, ..., 2^j - 1$.
- $\Omega_{j'}^k \subset \Omega_j^k$ con $0 \leq j < j'$ y $k = 0, 1, ..., 2^j - 1$.

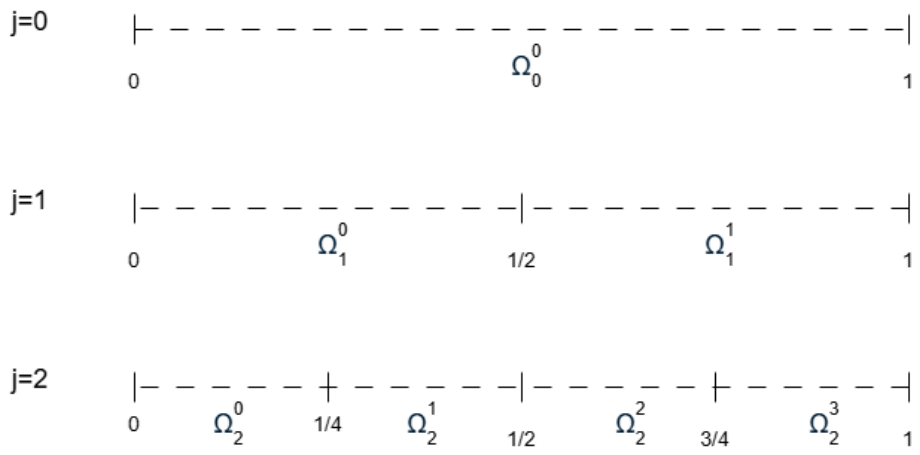

Figura 2.1: Se representa la generación de los intervalos 2.1 para diferentes valores de j, donde $\Omega_0^0 = [0, 1)$.

Proposición 2.1. *Sea $\phi = \chi_{[0,1]}$ la función característica de $\Omega = [0, 1]$. Entonces,*

$$\phi(x) = \phi(2x) + \phi(2x - 1). \tag{2.2}$$

Además, la función característica del intervalo $\Omega_j^k = [2^{-j}k, 2^{-j}(k + 1)]$,es

$$x \to \phi(2^j x - k)). \tag{2.3}$$

Demostración. La propiedad (2.2) se deduce del hecho de que $x \in [0, 1] \Leftrightarrow$ o bien $2x \in [0, 1]$, o bien $2x - 1 \in [0, 1]$

La propiedad (2.3) se deduce de que $x \in \Omega_j^k \Leftrightarrow 2^j x - k \in [0, 1]$. Es fácil de ver que para $x \in [2^{-j}k, 2^{-j}(k + 1)]$ tenemos que

$$0 = 2^{-j}2^j k - k \leq 2^j x - k \leq 2^j 2^{-j}(k + 1) - k = 1.$$

□

Ahora, queremos aproximar una función f por su proyección sobre el espacio de funciones que son constantes en los intervalos Ω_j^k.

Definición 2.2. *Sea $f \in L^1(\Omega)$ y $j \geq 0$. Se define*

$$P_j f(x) = \sum_{k=0}^{2^j-1} P_j^k f(x)\phi(2^j x - k), \tag{2.4}$$

para $x \in [0, 1]$, donde

$$P_j^k f = 2^j \int_{\Omega_j^k} f(t)dt, \tag{2.5}$$

para $k = 0, 1, ..., 2^j - 1$.

Como Ω es un conjunto acotado y $f \in L^1(\Omega)$, entonces, $f \in L^2(\Omega)$. Se dice entonces que (2.4) es la proyección de f sobre el siguiente espacio vectorial

$$V_j = \{f \in L^2(\Omega), f_{\Omega_j^k} \text{ es constante para } k = 0, 1, ..., 2^j - 1\}. \tag{2.6}$$

Además, (2.5) da el valor constante de $P_j f$ sobre Ω_j^k.

El sistema de Haar, [7], se define a través de las siguientes funciones

$$\phi_j^k(x) = 2^{j/2}\phi(2^j x - k), \tag{2.7}$$

$\forall x \in \Omega$ y para $k = 0, 1, ..., 2^j - 1$.

Proposición 2.2. *Sea $j \geq 0$. El espacio vectorial V_j definido en (2.6) tiene dimensión 2^j y las funciones $\phi_j^k(x)$, definidas en (2.7) para $k = 0, ..., 2^j - 1$, son una base ortonormal relativa al producto escalar de $L^2(\Omega)$,*

$$< f, g >= \int_\Omega f(t)g(t)dt. \tag{2.8}$$

Demostración. Vamos a ver que esta familia de funciones es una base de V_j. Primero, veamos que son vectores generadores de V_j, o, en otras palabras, que todo elemento de V_j se puede escribir como combinación lineal de las funciones ϕ_j^k.

Tomemos $f \in V_j$. Entonces f pertenece a $L^2(\Omega)$ tal que $f_{\Omega_j^k}$ es constante. Luego, se puede escribir

$$f(x) = \sum_{k=0}^{2^j-1} f(x)\phi(2^j x - k),$$

siendo $x \to \phi(2^j x - k)$ la función característica en el intervalo Ω_j^k. Como $f_{\Omega_j^k} = c_j^k$ es constante,

$$f(x) = \sum_{k=0}^{2^j-1} c_j^k \phi(2^j x - k).$$

Como $\phi_j^k(x) = 2^{j/2}\phi(2^j x - k)$,

$$f = \sum_{k=0}^{2^j-1} c_j^k / 2^{j/2}\phi_j^k(x) = \sum_{k=0}^{2^j-1} a_j^k \phi_j^k(x).$$

Por lo tanto, f se puede escribir como combinación lineal de $\{\phi_j^k\}$.

Veamos que $\{\phi_j^k\}$ son ortonormales respecto al producto escalar definido en (2.8). Tomemos $k, k' \in [0, 1, 2, ..., 2^j - 1]$ tal que $k \neq k'$. Sean Ω_j^k y $\Omega_j^{k'}$ los intervalos correspondientes a k y k', como $k \neq k'$, entonces $\Omega_j^k \neq \Omega_j^{k'}$. Por lo tanto, $< \phi_j^k, \phi_j^{k'} >= 0$ y el producto escalar en L^2 es

$$< \phi_j^k, \phi_j^{k'} >= \int_\Omega \phi_j^k(x)\phi_j^{k'}(x)dx = 0.$$

De forma análoga, si $k = k'$ entonces $\Omega_j^k = \Omega_j^{k'}$. Por lo tanto, $< \phi_j^k, \phi_j^{k'} >= |\phi_j^k|^2 = 1$ y

$$< \phi_j^k, \phi_j^{k'} >= \int_{\Omega_j^k} \phi_j^k(x)\phi_j^{k'}(x)dx = \int_{\Omega_j^k} |\phi_j^k|^2 = 1.$$

Por lo tanto, la familia de funciones $\{\phi_j^k\}$ es linealmente independiente para el producto escalar en $L^2(\Omega)$.

Además, la norma de ϕ_j^k es unitaria, ya que

$$||\phi_j^k||_2 =< \phi_j^k, \phi_j^k >= \int_{\Omega_j^k} \phi_j^k(x)\phi_j^k(x)dx = 1.$$

Concluimos que $\{\phi_j^k\}$ es una base de V_j. Y, como la base está compuesta por 2^j elementos, la dimensión de V_j es finita e igual a 2^j.

□

A continuación, vamos a ver que $P_j f$ es la proyección ortogonal de f sobre V_J. Para $x \in \Omega$, tenemos que

$$P_j f(x) = \sum_{k=0}^{2^j-1} P_j^k f \phi(2^j x - k) = \sum_{k=0}^{2^j-1} 2^j \int_{\Omega_j^k} f(t)dt \quad \phi(2^j x - k)$$

$$= \sum_{k=0}^{2^j-1} 2^{j/2} \int_{\Omega_j^k} f(t)dt \quad \phi_j^k(x) = \sum_{k=0}^{2^j-1} \int_{\Omega} f(t)\phi_j^k(t)dt \quad \phi_j^k(x)$$

$$= \sum_{k=0}^{2^j-1} < f, \phi_j^k > \phi_j^k(x) = \sum_{k=0}^{2^j-1} c_j^k \phi_j^k(x)$$

Luego tenemos que los coeficientes c_j^k son las componentes de $P_j f$ en la base $\{\phi_j^k\}$ y se calculan de la siguiente manera,

$$c_j^k =< f, \phi_j^k >= \int_{\Omega} f(t)\phi_j^k(t)dt = 2^{j/2} \int_{\Omega_j^k} f(t)dt. \tag{2.9}$$

Deducimos que P_j es la proyección ortogonal sobre V_j con el producto escalar de $L^2(\Omega)$. Luego la aproximación de f a una resolución 2^{-j} se define como una proyección ortogonal sobre el espacio V_j.

Lema 2.1. *Sea f continua en $\Omega = [0,1]$ y $\epsilon > 0$. Entonces, existe $J > 0$ y $g_J \in V_J$ tal que*

$$||f - g_J||_\infty = max_{x\in[0,1]}|f(x) - g_J(x)| < \epsilon \tag{2.10}$$

Demostración. Como f es continua en $\Omega = [0,1]$, un intervalo compacto, entonces f es uniformemente continua en Ω. Luego, existe un δ que depende de ϵ, $\delta(\epsilon)$, positivo tal que si $x, y \in [0,1]$ con $|x - y| < \delta$ entonces,

$$|f(x) - f(y)| < \epsilon.$$

Ahora, sea J suficientemente grande tal que $2^- J < \delta$, definimos $g_J = P_J f$. Si $x \in [0,1]$, entonces existe $k = 0, ..., 2^j - 1$ tal que $x \in \Omega_j^k$. Por tanto,

$$g_J(x) = P_J f(x) = P_j^k f = 2^j \int_{\Omega_j^k} f(t)dt,$$

y, deducimos que

$$|f(x) - g_J(x)| = |2^j \int_{\Omega_j^k} f(x) - f(t)dt|$$

$$\leq 2^j \int_{\Omega_j^k} f(x) - f(t)dt < \epsilon.$$

□

Ahora, vamos a estudiar la relación entre los espacios V_j.

Teorema 2.1. *Sean $f \in L^2(\Omega)$, $j' > j \geq 0$ fijos y sus respectivos espacios $V_{j'}$ y V_j. Entonces:*

1. $\|P_{j'}f - f\|_2 < \|P_j f - f\|_2$.
2. $\|P_j f - f\|_2 \to 0$ *si* $j \to \infty$.
3. *Si $f \in C(\Omega)$, entonces $\|P_j f - f\|_\infty \to 0$ cuando $j \to \infty$.*

Demostración. .

1. De la construcción de los intervalos Ω_j^k definidos en (2.1), se deduce que $V_j \subset V_{j'}$, Por otro lado, la caracterización de $P_j f$ como la proyección ortogonal de f sobre V_j implica que, como $P_j f \in V_j \subset V_{j'}$

$$||P_{j'}f - f||_2 = min_j\{||g - f||_2 : g \in V_{j'}\} < ||P_j f - f||_2.$$

2. Como el espacio de funciones continuas en $\Omega = [0,1]$ es denso en $L^2(\Omega)$, entonces bastará con comprobar la propiedad para f continua. En este caso, por el lema 1, existe $J > 0$ y $g_J \in V_J$ verificando (2.10). Por tanto,

$$||f - g_J||_2 \leq ||f - g_J||_\infty < \epsilon. \tag{2.11}$$

   Sea $j \geq J$, como $V_J \subset V_j$, entonces $g_J \in V_j$. Por lo tanto, $P_j g_J = g_J$.

   Usando esta propiedad junto con (2.11), se tiene que

$$\begin{aligned} ||P_j f - f||_2 &\leq ||P_j f - P_j g_J||_2 + ||P_j g_J - g_J||_2 + ||g_J - f||_2 \\ &= ||P_j(f - g_J)||_2 + ||g_J - f|| \leq 2||g_j - f||_2 < 2\epsilon. \end{aligned}$$

3. Siguiendo la demostración del Lema 1, sea $j \geq J$ con $2^{-J} < \delta(\epsilon)$. El mimso razonamiento prueba que si $x \in [0,1]$ existe $k = 0, 1, ..., 2^j - 1$ tal que $x \in \Omega_j^k$ y, por tanto, tenemos que

$$|P_j f(x) - f(x)| < \epsilon.$$

□

A continuación, vamos a ver la relación entre los coeficientes c_j^k y $c_{j+1}^{k'}$. Sea $j \geq 0$ arbitrario, consideramos los dos espacios V_j y V_{j+1}. Para todo $f \in L^2(\Omega)$ y para $k = 0, 1, ..., 2^j - 1$, tenemos que

$$\sqrt{2}\int f(t)\phi_j^k(t)dt = \int f(t)\phi_{j+1}^{2k}(t)dt + \int f(t)\phi_{j+1}^{2k+1}(t)dt.$$

Luego, para $k = 0, 1, ..., 2^j - 1$, obtenemos que

$$c_j^k = (c_{j+1}^{2k} + c_{j+1}^{2k+1})/\sqrt{2}.$$

El espacio V_j agrupa todas las posibles aproximaciones a la resolución 2^{-j}, siendo la resolución 2^0 el correspondiente a la base ϕ.

Vamos a ilustrar la aproximación por funciones constantes a trozos a través del sistema de Haar con un ejemplo. Se va a representar la función $f(x) = e^{-x}sen(4\pi x)$ en Ω en los intervalos de longitud 2^{-j} y su aproximaciones (2.4) para $j = 8$ y $j = 14$. La comparación de las figuras 2.2, 2.3 y 2.4 muestra el aumento de la precisión en la aproximación a f en medida que j crece. Este ejemplo ilustra el caso 3. del Teorema 1.

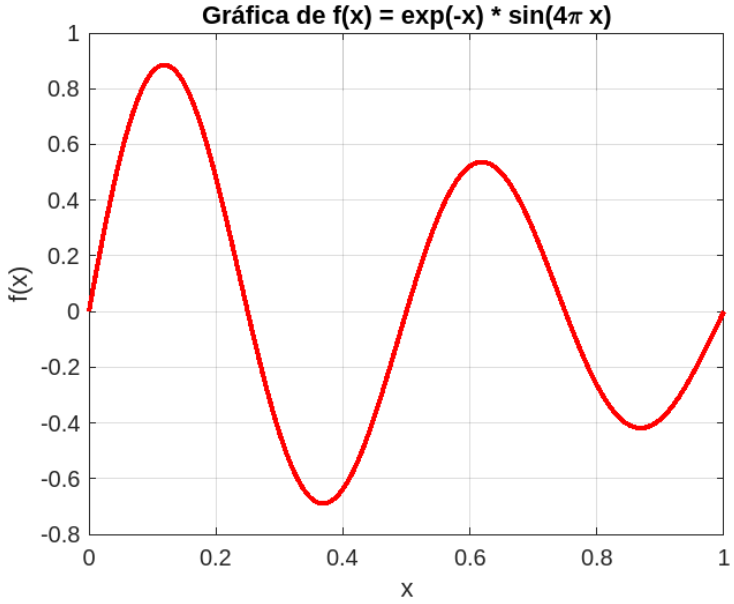

Figura 2.2: Función $f(x) = e^{-x}sen(4\pi x)$ en el intervalo $\Omega = [0, 1]$.

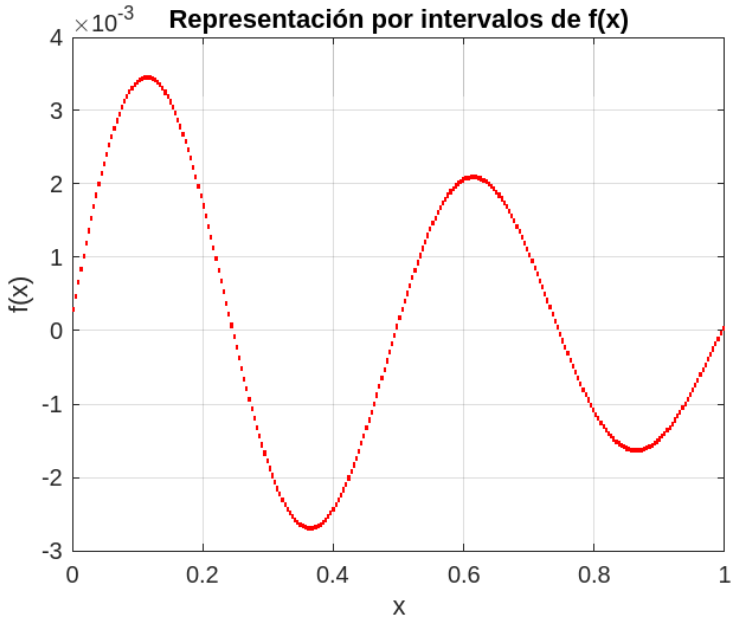

Figura 2.3: Aproximación de la fucnión de la figura 2.2 mediante $P_j f$ definida en (2.4) con $j = 8$.

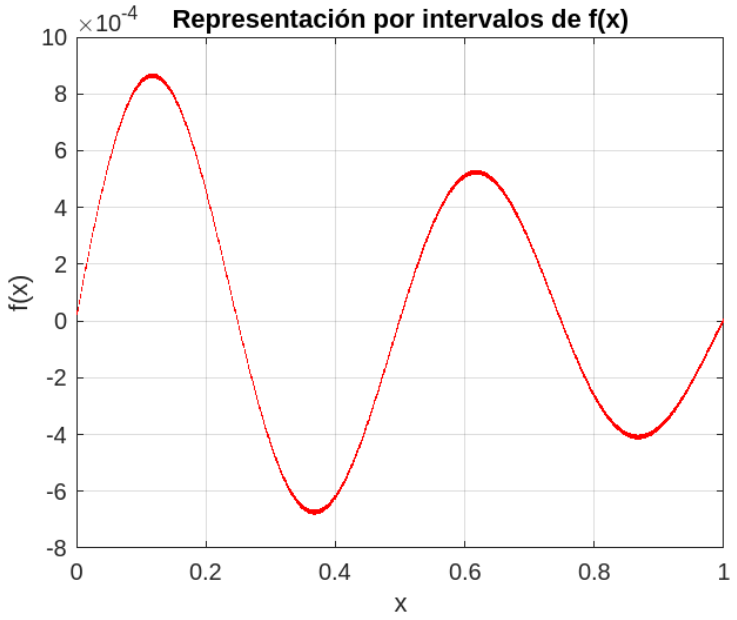

Figura 2.4: Aproximación de la fucnión de la figura 2.2 mediante $P_j f$ definida en (2.4) con $j = 14$.

2.2. Descomposición de los espacios de resolución

A continuación, se estudia la descomposición del espacio V_{j+1} de tal manera que la proyección ortogonal de una función $f \in L^2(\Omega)$ sobre V_{j+1} puede expresarse en términos de su proyección sobre V_j más un término de corrección, [4]. Para formalizar esta descomposición, se introduce el subespacio W_j, que representa la diferencia entre V_{j+1} y V_j. También, se define una familia de funciones , ψ_j^k, base de W_j, y se obtienen expresiones para los coeficientes c_j^k y d_j^k, que corresponden con la aproximación y la fluctuación de f con distintas resoluciones.

Sea $J \geq 0$ arbitrario y sea j tal que $J > j \geq 0$. De la definición 2.6 se deduce que los subespacios $V_j, V_{j+1}, ..., V_J$ satisfacen que

$$V_j \subset V_{j+1} \subset ... \subset V_J. \tag{2.12}$$

Dado $f \in L^2(\Omega)$, podemos escribir la proyección ortogonal $P_{j+1}f$ de f sobre V_{j+1}. Se puede escribir en términos de la proyección de f sobre V_j, P_jf, y un término de corrección,

$$P_{j+1}f = P_jf + (P_{j+1}f - P_jf) = P_jf + Q_jf, \tag{2.13}$$

donde Q_j es el operador definido como $Q_j = P_{j+1} - P_j$.

Podemos observar que cuando pasamos de V_j a V_{j+1} estamos añadiendo funciones de resolución 2^{j+1}.

Definición 2.3. *Sea W_j el complemento ortogonal de V_j en V_{j+1}, es decir,*

$$V_{j+1} = V_j \oplus W_j, \tag{2.14}$$

donde si $g \in V_j$ y $h \in W_j$, entonces $< g, h >= 0$.

La relación (2.12) establece que para $f \in L^2(\Omega)$, Q_jf es la proyección ortogonal de f sobre W_j. W_j contiene las funciones de resolución 2^{j+1} pero no las funciones de resolución 2^j.

Buscamos ahora una base ortonormal para W_j. Sea $\psi = \chi_{[0,1/2)} - \chi_{[1/2,1]}$. La demostración de la proposición 2.2 garantiza que ψ cumple la siguiente propiedad,

$$\psi(x) = \phi(2x) - \phi(2x - 1). \tag{2.15}$$

Proposición 2.3. *Sea $j \geq 0$. Las funciones ψ_j^k definidas como*

$$\psi_j^k(x) = 2^{j/2}\psi(2^jx - k), \tag{2.16}$$

para $k = 0, 1, ..., 2^j - 1$ y $x \in \Omega$ forman una base ortonormal de W_j con respecto al producto interno de L^2 dado por (2.8).

Demostración. La demostración va a seguir tres pasos:

1. Demostrar que $\{\psi_j^k\}_{k=0}^{2^j-1}$ forma un sistema ortonormal y, por lo tanto, son linealmente independientes.

2. Demostrar que $\psi_j^k \in W_j$ para $k = 0, ..., 2^j - 1$.

3. Demostrar que $\dim W_j = 2^j$.

1. Veamos que $\{\psi_j^k\}$ forma un sistema ortonormal. Tomemos $k, k' \in [0, 1, 2, ..., 2^j - 1]$ tal que $k \neq k'$. Sean Ω_j^k y $\Omega_j^{k'}$ los intervalos correspondientes a k y k'. Como $k \neq k'$, entonces $\Omega_j^k \cap \Omega_j^{k'} = \emptyset$. Por lo tanto, tenemos que

$$< \psi_j^k, \psi_j^{k'} >= \int_\Omega \psi_j^k(x)\psi_j^{k'}(x)dx = 0.$$

   Lo que implica que el sistema $\{\psi_j^k\}$ es ortogonal. De forma análoga, si $k = k'$ entonces $\Omega_j^k = \Omega_j^{k'}$. Por lo tanto,

$$< \psi_j^k, \psi_j^{k'} >= \int_{\Omega_j^k} \psi_j^k(x)\psi_j^{k'}(x)dx = \int_{\Omega_j^k} |\psi_j^k|^2 = 1.$$

2. Para demostrar que $\psi_j^k \in W_j$ para $k = 0, ..., 2^j - 1$ basta ver que

$$< \psi_j^k, \phi_j^{k'} >= \int_\Omega \psi_j^k(x)\phi_j^{k'}(x)dx = 0,$$

   con $k' = 0, ..., 2^j - 1$.

   Supongamos que $k \neq k'$, entonces $\Omega_j^k \cap \Omega_j^{k'} = \emptyset$. Por lo tanto, tenemos que

$$< \psi_j^k, \phi_j^{k'} >= \int_\Omega \psi_j^k(x)\phi_j^{k'}(x)dx = 0.$$

   Ahora, supongamos que $k = k'$. Calculemos el producto escalar de ψ_j^k y ϕ_j^k, utilizando sus definiciones. Entonces, tenemos

$$< \psi_j^k, \phi_j^{k'} >= \int_{\Omega_j^k} \psi_j^k(x)\phi_j^{k'}(x)dx$$

$$= \int_{\Omega_j^k} 2^j(\phi(2^{j+1}x - 2k) - \phi(2^{j+1}x - 2k - 1))(\phi(2^{j+1}x - 2k) + \phi(2^{j+1}x - 2k - 1))dx$$

$$= \int_{\Omega_j^k} 2^j(\phi(2^{j+1}x - 2k)^2 - \phi(2^{j+1}x - 2k - 1)^2)dx = 0.$$

3. Sabemos que la dimensión de V_{j+1} es $2^j + 1$ y que la dimensión de V_j es 2^j. Partiendo de (2.13), entonces

$$dim V_{j+1} = dim V_j + dim W_j.$$

   Luego, $dim \quad W_j = 2^j$.

□

A continuación, vamos a comparar las funciones ϕ_j^k y ψ_j^k. Para ello, vamos a ilustrar para $j = 2$, las funciones ϕ_2^1 y ψ_2^1 en el intervalo $[0, 1]$.

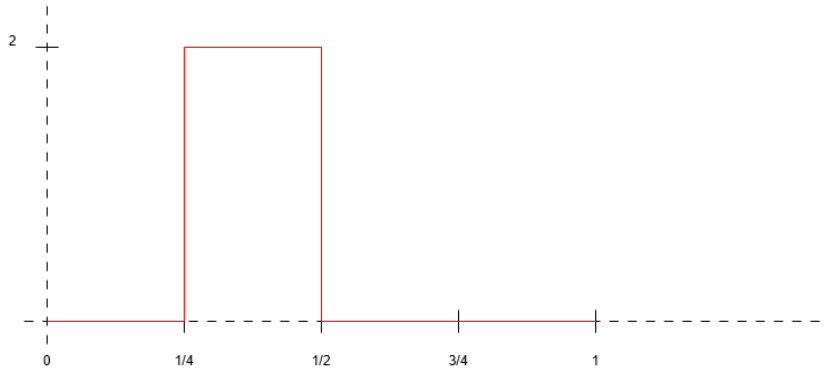

Figura 2.5: Representación de ϕ_2^1 en el intervalo $[0,1]$.

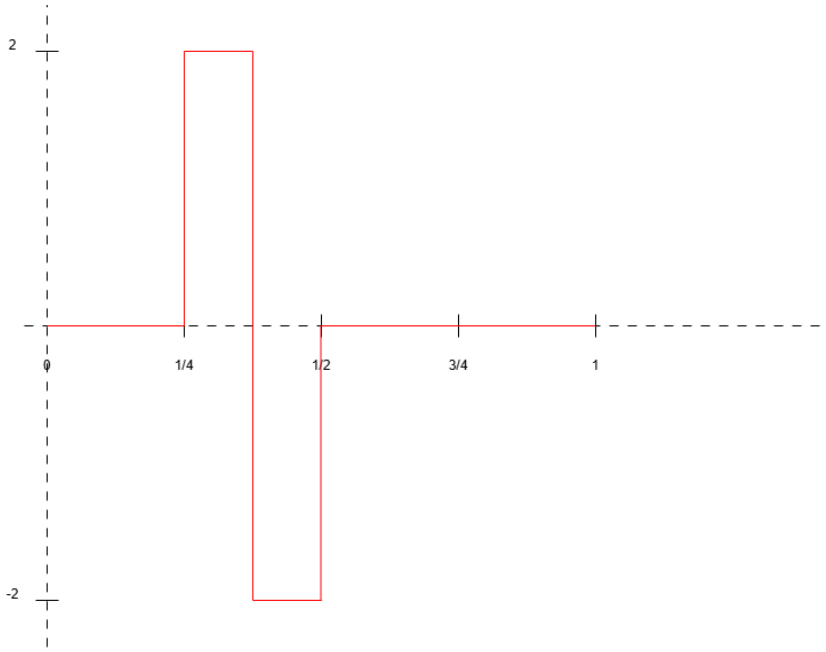

Figura 2.6: Representación de ψ_2^1 en el intervalo $[0,1]$.

Definición 2.4. *Sea $f \in L^2(\Omega)$ una función arbitraria. Definimos los coeficientes d_j^k como*

$$d_j^k =< f, \psi_j^k >= \int f(t)\psi_j^k(t)dt = 2^{j/2}\int_{\Omega_{j+1}^{2k}} f(t)dt - 2^{j/2}\int_{\Omega_{j+1}^{2k+1}} f(t)dt. \tag{2.17}$$

A estos coeficientes d_j^k se les conoce como la fluctuación de f en los intervalos Ω_j^k.

Utilizando la definición 2.5 tenemos la siguiente relación entre los coeficientes d_j^k y c_j^k, [2],

$$d_j^k = (c_{j+1}^{2k} - c_{j+1}^{2k+1})/\sqrt{2}, \tag{2.18}$$

y por la relación (2.1), se deduce

$$\begin{cases} \sqrt{2}c_{j+1}^{2k} = c_j^k + d_j^k & \text{para } k = 0, 1, ..., 2^j - 1 \\ \sqrt{2}c_{j+1}^{2k+1} = c_j^k - d_j^k & \text{para } k = 0, 1, ..., 2^j - 1. \end{cases} \tag{2.19}$$

A partir de estas dos relaciones se obtienen los algoritmos de descomposición y reconstrucción a escala j. Sea J un entero arbitrario y $f \in L^2(\Omega)$. Expresamos f en términos

de la base canónica de V_J con sus 2^J coeficientes c_J^k, $k = 0, ..., 2^{J-1}$.

El algoritmo de descomposición viene dado por (2.18) en la forma:

$$\begin{cases} c_j^k = (c_{j+1}^{2k} + c_{j+1}^{2k+1})/\sqrt{2}, \\ d_j^k = (c_{j+1}^{2k} - c_{j+1}^{2k+1})/\sqrt{2}, \end{cases} \tag{2.20}$$

para $k = 0, ..., 2^j - 1$.

Calculemos el coste computacional del algoritmo (2.20). Para el paso j, se realizan $2 * 2^{j-1}$ sumas y restas y $2 * 2^{j-1}$ divisiones. Esto hace un total de 2^{j+1} operaciones, [2]. Por lo tanto, el coste computacional del algoritmo completo es:

$$2^{J+1} + 2^J + ... + 2^2 + 1 = 2^{J+2} - 1.$$

Se puede considerar un coste óptimo, ya que $2^{J+2} = 4 * 2^J$, y, por lo tanto, el coste de las operaciones es del orden de $O(2^J)$.

Para el algoritmo de reconstrucción, asumimos que conocemos los coeficientes c_0^0 y d_j^k para $j = 0, ..., J-1$. Vamos a obtener los coeficientes c_j^k mediante el algoritmo de reconstrucción, que viene dado por (2.18) de la forma:

$$\begin{cases} c_{j+1}^{2k} = c_j^k + d_j^k/\sqrt{2}, \\ c_{j+1}^{2k+1} = c_j^k - d_j^k/\sqrt{2}, \end{cases} \tag{2.21}$$

para $k = 0, ..., 2^j - 1$.

Calculemos el coste computacional del algoritmo de reconstrucción. Para el paso j, se realizan $2 * 2^{j-1}$ sumas y restas y $2 * 2^{j-1}$ divisiones. Esto hace un total de 2^{j+1} operaciones, [2]. Por lo tanto, el coste computacional del algoritmo completo es:

$$2^{J+1} + 2^J + ... + 2^2 + 1 = 2^{J+2} - 1.$$

Por lo tanto, el coste de las operaciones es del orden de $O(2^J)$. Se puede considerar un coste óptimo, ya que $2^{J+2} = 4 * 2^J$, es el mínimo número de operaciones posibles. El coste de los dos algoritmos vemos que es el mismo.

Los pseudocódigos correspondientes al algoritmo de descomposición y al algoritmo de reconstrucción, así como su explicación, son los siguientes:

El algoritmo de descomposición (2.20) consiste en las operaciones necesarias para pasar de f expresada en la base canónica de V_J a la misma función f pero expresada en términos de la base de Haar. Por lo tanto, es una herramienta útil para realizar el cambio de base entre la base canónica y la base de Haar. Partimos de los coeficientes c_J^k, y, a partir de ellos, calculamos los coeficientes c_j^k y d_j^k.

El pseudocódigo correspondiente al algoritmo de descomposición es:

```
para j = J-1,...,0
    para k = 0,...,2^j-1
```

```
        c(k,j) = (c(2*k,j+1) + c(2*k+1,j+1)) / sqrt(2);
        d(k,j) = (c(2*k,j+1) - c(2*k+1,j+1)) / sqrt(2);
    end
end
```

El algoritmo empieza por el caso $j = J - 1$ hasta llegar al caso $j = 0$. En el paso j, se calcula c_{j-1}^k y d_{j-1}^k a partir de c_j^k. Este paso se suele representar de esta manera, [4]:

$$\begin{array}{ccc} & c_j^k & \\ & (0 \leq k < 2^j) & \\ & \swarrow \quad \searrow & \\ c_{j-1}^k & & d_{j-1}^k \\ (0 \leq k < 2^{j-1}) & & (0 \leq k < 2^{j-1}) \end{array}$$

Nótese que una vez calculados los coeficientes d_j^k, no se vuelven a utilizar en los siguientes pasos. Solo se usan en los pasos posteriores los 2^j coeficientes c_j^k para calcular c_{j-1}^k y d_{j-1}^k.

El algoritmo de reconstrucción (2.21) consiste en las operaciones necesarias para pasar de f expresada en la base de Haar de V_J a la misma función f pero expresada en términos de la base canónica. Luego, se utiliza para realizar el cambio de base entre la base de Haar y la base canónica. Partimos de los coeficientes d_j^k, y, a partir de ellos, calculamos los coeficientes c_j^k.

El pseudocódigo correspondiente al algoritmo de reconstrucción es:

```
para j = 0,...,J-1
    para k = 0,...,2^j-1
        c(2*k,j+1) = (c(k,j) + d(k,j)) / sqrt(2);
        c(2*k+1,j+1) = (c(k,j) - d(k,j)) / sqrt(2);
    end
end
```

El algoritmo empieza por el caso $j = 0$ hasta llegar al caso $j = J - 1$. En el paso j, se calcula c_{j+1}^{2k} y c_{j+1}^{2k+1} a partir de c_j^k y d_j^k. Este paso se suele representar de esta manera, [4]:

$$\begin{array}{ccc} c_{j-1}^k & & d_{j-1}^k \\ (0 \leq k < 2^{j-1}) & & (0 \leq k < 2^{j-1}) \\ & \searrow \quad \swarrow & \\ & c_j^k & \\ & (0 \leq k < 2^j). & \end{array}$$

Estas relaciones entre los coeficientes c_j^k y d_j^k se pueden escribir en forma matricial. El algoritmo de descomposición sería:

$$\begin{pmatrix} c_j(k) \\ d_j(k) \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} c_{j+1}(2k) \\ c_{j+1}(2k+1) \end{pmatrix}, k = 0, ..., 2^{J-1}.$$

El algoritmo de reconstrucción sería, [16]:

$$\begin{pmatrix} c_{j+1}(2k) \\ c_{j+1}(2k+1) \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} c_j(k) \\ d_j(k) \end{pmatrix}, k = 0, ..., 2^{J-1}.$$

Los argumentos anteriores permiten descomponer el espacio V_J de la siguiente forma,

$$V_J = V_{J-1} \oplus W_{J-1} = V_{J-2} \oplus W_{J-2} \oplus W_{J-1} = ... = V_0 \oplus W_0 \oplus ... \oplus W_{J-2} \oplus W_{J-1} \quad (2.22)$$

Como el sistema de funciones ϕ_j^k forma una base ortonormal de V_j, y el sistema de funciones ψ_j^k forma una base ortonormal de W_j, podemos definir dos bases ortonormales de V_J. De este modo, tenemos una base generada por las funciones

$$\{\phi_J^k\}_{k=0}^{2^J-1}, \quad (2.23)$$

definida en (2.8), llamada la base canónica. También podemos definir otra base utilizando la descomposición (2.20) y la proposición 3 generada por las funciones

$$\{\phi_0^0\} \cup \{\psi_j^k\}_{j=0,...,J-1, k=0,...,2^j-1}. \quad (2.24)$$

Es también una base de V_j y se llama la base de Haar.

Las correspondientes representaciones de la proyección de la función $f \in L^2(\Omega)$ en V_J en (2.23) y (2.24) son, respectivamente,

$$P_J f = \sum_{K=0}^{2^j-1} c_J^k \phi_J^k, \quad (2.25)$$

y,

$$P_J f = c_0^0 \phi_0^0 + \sum_{j=0}^{J-1} \sum_{K=0}^{2^j-1} d_j^k \psi_j^k, \quad (2.26)$$

donde el coeficiente $c_0^0 = \int_\Omega f(t)dt$ es la media de f en Ω.

Sea $J \geq 0$, el sistema de Haar de escala J en $[0, 1]$ es,

$$\{\phi_J^k(x) : 0 \leq k \leq 2^J - 1\} \cup \{\psi_j^k(x) : j \geq J,\ 0 \leq k \leq 2^j - 1\}. \quad (2.27)$$

La base de Haar en el intervalo $[0, 1]$ está compuesta por las funciones ψ_j^k asociadas a los intervalos Ω_j^k junto con la única función de escalado $\phi_0^0(x)$. Para un valor $J > 0$, el sistema de Haar de escala J en $[0, 1]$ incluye todas las funciones $\psi_j^k(x)$ correspondientes a los intervalos Ω_j^k tales que $j \geq J$ así como la función de escala J. Para $J = 0$, se dice que (2.27) es el sistema de Haar en $[0, 1]$.

2.3. El sistema de Haar en R

En esta sección, se va a describir el sistema de Haar en $\Omega = \mathbb{R}$, [16]. Se va a generalizar los conceptos del capítulo anterior a $\mathbb{R}$. Para cada par $j, k \in \mathbb{Z}$, definimos ahora los siguientes intervalos

$$\Omega_j^k = [2^{-j}k, 2^{-j}(k+1)), \quad (2.28)$$

estos intervalos se llaman intervalos diádicos.

Argumentos similares a los dados para los intervalos Ω_j^k del capítulo 2 para $\Omega = [0,1]$ nos llevan a las propiedades siguientes para (2.28).

Lema 2.2. *Dado $j_0, k_0, j_1, k_1 \in \mathbb{Z}$, con $j_0 \neq j_1$ ó $k_0 \neq k_1$, entonces se cumple una de las siguientes condiciones:*

1. $\Omega_{j_1,k_1} \cap \Omega_{j_0,k_0} = \emptyset$,
2. $\Omega_{j_1,k_1} \subseteq \Omega_{j_0,k_0}$,
3. $\Omega_{j_0,k_0} \subseteq \Omega_{j_1,k_1}$.

Definición 2.5. *Sea $\phi(x) = \chi_{[0,1)}(x)$, y para cada $j,k \in \mathbb{Z}$, definimos,*

$$\phi_j^k(x) = 2^{j/2}\phi(2^j x - k). \tag{2.29}$$

El conjunto $\{\phi_j^k\}_{j,k\in\mathbb{Z}}$ se llama sistema de Haar de funciones de escala. Para un $j \in \mathbb{Z}$ fijo, $\{\phi_j^k\}_{k\in\mathbb{Z}}$ es el sistema de Haar de funciones de escala a escala j.

Observación 2.1. *A partir de la definición, podemos observar que:*

1. *Para cada $j,k \in \mathbb{Z}$, se tiene que*

$$\phi_j^k(x) = 2^{j/2}\chi_{\Omega_j^k}(x).$$

Por lo que $\phi_j^k(x)$ tiene soporte en Ω_j^k y no se anula en dicho intervalo. Se dice que la función $\phi_j^k(x)$ está asociada al intervalo (2.28). Esto implica, en particular, que para cada $j \in \mathbb{Z}$, $\{\phi_j^k\}_{k\in\mathbb{Z}}$ es un sistema ortogonal.

2. *Para cada $j,k \in \mathbb{Z}$, tenemos que*

$$\int_{\mathbb{R}} \phi_j^k(x)dx = \int_{\Omega_j^k} \phi_j^k(x)dx = 2^{-j/2},$$

$$\int_{\mathbb{R}} |\phi_j^k(x)|^2 dx = \int_{\Omega_j^k} |\phi_j^k(x)|^2 dx = 1.$$

Definición 2.6. *Sea $\psi(x) = \chi_{[0,\frac{1}{2})}(x) - \chi_{[\frac{1}{2},1)}(x)$,. Para cada $j,k \in \mathbb{Z}$, definimos,*

$$\psi_j^k(x) = 2^{j/2}\psi(2^j x - k). \tag{2.30}$$

El sistema de funciones $\{\psi_j^k(x)\}_{j,k\in\mathbb{Z}}$ se llama el sistema de Haar en $\mathbb{R}$. Y, para un $j \in \mathbb{Z}$ dado, $\{\psi_j^k(x)\}_{k\in\mathbb{Z}}$ se conoce como el sistema de funciones de Haar a escala j.

Observación 2.2. *A partir de lo anterior deducimos que:*

1. *Para cada $j,k \in \mathbb{Z}$, se tiene que*

$$\psi_j^k(x) = 2^{j/2}(\chi_{\Omega_j^{k^+}}(x) - \chi_{\Omega_j^{k^-}}(x)) = 2^{1/2}(\chi_{\Omega_{j+1}^{2k}}(x) - \chi_{\Omega_{j+1}^{2k+1}}(x)).$$

donde $\Omega_j^{k^-}(x)$ corresponde a la mitad izquierda del intervalo Ω_j^k y $\Omega_j^{k^+}(x)$ corresponde a la mitad derecha del intervalo Ω_j^k. Luego, $\psi_j^k(x)$ tiene soporte en el intervalo Ω_j^k y no se anula en ese intervalo. Se dice que la función de Haar $\psi_j^k(x)$ está asociada al intervalo Ω_j^k.

2. *Para cada $j,k \in \mathbb{Z}$, $\psi_j^k(x)$ es una función escalón diádica de escala $j+1$. Una función escalonada diádica se define como una función $f(x)$ que, para un $j \in \mathbb{Z}$, es constante en todos los intervalos diádicos Ω_j^k, donde $k \in \mathbb{Z}$. En este caso, se dice que f es una función escalonada diádica a escala j. En particular, $\{\psi_j^k\}_{k\in\mathbb{Z}}$ es un sistema ortogonal.*

3. *Para cada $j,k \in \mathbb{Z}$,*
$$\int_{\mathbb{R}} \psi_j^k(x)dx = \int_{\Omega_j^k} \psi_j^k(x)dx = 0,$$
$$\int_{\mathbb{R}} |\psi_j^k(x)|^2 dx = \int_{\Omega_j^k} |\psi_j^k(x)|^2 dx = 1.$$

Teorema 2.2. *El sistema de Haar definido en $\mathbb{R}$ es un sistema ortonormal en $L^2(\mathbb{R})$.*

Demostración. Primero, demostremos la ortonormalidad para una escala fija. Sea $j \in \mathbb{Z}$ fijo y $k,k' \in \mathbb{Z}$. Por el Lema 2 tenemos que,

- Si $k \neq k'$, entonces los soportes son disjuntos, $\Omega_j^k \cap \Omega_j^{k'} = \emptyset$, y, por lo tanto,
$$\langle \psi_j^k, \psi_j^{k'} \rangle = \int_{\mathbb{R}} \psi_j^k(x)\psi_j^{k'}(x)dx = 0.$$

- Si $k = k'$, entonces
$$\langle \psi_j^{k'}, \psi_j^k \rangle = \int_{\mathbb{R}} |\psi_j^k(x)|^2 dx = 1.$$

Ahora, probemos la ortogonalidad entre diferentes escalas. Sean $j \neq j'$. Supongamos que $j > j'$, y $k,k' \in \mathbb{Z}$. Podemos tener dos casos posibles:

1. Si $\Omega_j^k \cap \Omega_{j'}^{k'} = \emptyset$, entonces el producto escalar es cero ya que,
$$\psi_j^k(x)\psi_{j'}^{k'}(x) = 0.$$

2. Si $\Omega_j^k \subset \Omega_{j'}^{k'}$, entonces $\psi_{j'}^{k'}(x)$ es constante en Ω_j^k, y como ψ_j^k tiene integral 0 por la observación 2, el producto escalar también es 0.

□

Definición 2.7. *Para cada $j \in \mathbb{Z}$, definimos P_j como el operador de aproximación de $f(x)$ en $L^2(\mathbb{R})$,*
$$P_j f(x) = \sum_k < f, \phi_j^k > \phi_j^k(x), \tag{2.31}$$
donde $\phi_j^k(x)$ son las funciones de escala definidas en (2.29), y $< f, \phi_j^k >$ es el producto escalar en $L^2(\mathbb{R})$ dado por (2.8) con $\Omega = \mathbb{R}$.

Definición 2.8. *Para cada $j \in \mathbb{Z}$, definimos el espacio V_j por*
$$V_j = \overline{span\{\phi_j^k(x)\}_{k\in\mathbb{Z}}}. \tag{2.32}$$

El espacio V_j también puede definirse de otra manera. Para cada $j \in \mathbb{Z}$, V_j se define como la colección de todas las funciones escalonadas diádicas a escala j. Luego, en el caso de $\mathbb{R}$, V_j es el espacio que contiene a las funciones escalonadas diádicas en $L^2(\mathbb{R})$ en los intervalos diádicos de $\mathbb{R}$, o, de otra manera,
$$V_j = \{f \in L^2(\mathbb{R}), f_{\Omega_j^k} \text{ es constante con } k \in \mathbb{Z}\}. \tag{2.33}$$

Observación 2.3. *Como* $\phi_j^k(x) = 2^{j/2}\chi_{\Omega_j^k}(x)$,

$$< f, \phi_j^k > \phi_j^k(x) = (2^j \int_{\Omega_j^k} f(t)dt)\chi_{\Omega_j^k}. \tag{2.34}$$

Esto quiere decir que, en el intervalo Ω_j^k, $P_j f(x)$ *es el valor promedio de* $f(x)$ *en* Ω_j^k. *En otras palabras, al pasar de* $f(x)$ *a* $P_j f(x)$, *el comportamiento de* $f(x)$ *en el intervalo* Ω_j^k *se reduce a un solo número, el valor promedio de* $f(x)$ *en* Ω_j^k. *Por lo tanto, cualquier dato sobre la variación a pequeña escala de* $f(x)$ *en* Ω_j^k *se pierde.*

En este sentido, $P_j f(x)$ *puede presentarse como una versión desenfocada de* $f(x)$ *a escala* 2^{-j}, *ya que los detalles en* $f(x)$ *de tamaño menor que* 2^{-j} *son invisibles en la aproximación* $P_j f(x)$, *pero las características de tamaño mayor que* 2^{-j} *si son visibles en* $P_j f(x)$.

Los operadores P_j tienen las siguientes propiedades:

Lema 2.3. .

1. *Para cada* $j \in \mathbb{Z}$, P_j *es un operador lineal,*

$$P_j(\alpha f + \beta g) = \alpha P_j(f) + \beta P_j(g), \quad f, g \in L^2(\Omega), \alpha, \beta \in \mathbb{C}.$$

2. *Para cada* $j \in \mathbb{Z}$ *y* $f \in L^2(\Omega)$,

$$P_j P_j f = P_j f.$$

3. *Si* $g \in V_j$ *y* $j \leq j'$, *entonces* $P_{j'} g = g$.

4. *Para cada* $j \in \mathbb{Z}$ *y* $f \in L^2(\Omega)$,

$$||P_j f||_2 \leq ||f||_2.$$

Demostración. .

1. Es inmediato de la fórmula (2.31).

2. Se deduce de la observación 2.1, al tener ϕ_j^k soporte en Ω_j^k.

3. Se obtiene de que $V_j \subset V_{j'}$ y de que si $g \in V_j$, entonces $P_j g = g$.

4. Por último, como $\{\phi_j^k\}_{k\in\mathbb{Z}}$ es un sistema ortonormal, si $f \in L^2(\Omega)$, entonces

$$||P_j f||_2^2 = \sum_k | < f, \phi_j^k > |^2 = \sum_k |2^{j/2} \int_{\Omega_k^j} f(x)dx|^2.$$

Y por la desigualdad de Cauchy-Schwarz, obtenemos

$$|2^{j/2} \int_{\Omega_k^j} f(x)dx|^2 \leq (\int_{\Omega_k^j} 2^j dx)(\int_{\Omega_k^j} |f(x)|^2 dx) = \int_{\Omega_k^j} |f(x)|^2 dx.$$

Por lo tanto,

$$||P_j f||_2^2 \leq \sum_k \int_{\Omega_k^j} |f(x)|^2 dx = \int_{\mathbb{R}} |f(x)|^2 dx = ||f||_2^2.$$

□

Lema 2.4. *Sea f continua y de soporte compacto en $\mathbb{R}$. Entonces*

a) $\lim_{j\to\infty} ||P_j f - f||_2 = 0$,

b) $\lim_{j\to\infty} ||P_{-j} f||_2 = 0$.

Demostración. .

a) Supongamos que f tiene soporte en un intervalo de la forma $[-2^N, 2^N]$ para algún entero N. Utilizando la demostración del lema 2.1, existe un entero J y $g \in V_J$ tal que

$$||f - g|| < \frac{\epsilon}{\sqrt{2^{N+3}}}.$$

Si $j \geq J$ por el lema 2.3, tenemos que $P_j g = g$. Entonces, también por el lema 2.3,

$$\begin{aligned} ||P_j f - f||_2 &\leq ||P_j f - P_j g||_2 + ||P_j g - g||_2 + ||g - f||_2 \\ &= ||P_j(f - g)||_2 + ||g - f||_2 \\ &\leq 2||g - f||_2. \end{aligned}$$

Como

$$\begin{aligned} ||g - f||_2^2 &= \int_{-2^N}^{2^N} |g(x) - f(x)|^2 \, dx \\ &\leq \int_{-2^N}^{2^N} \frac{\epsilon^2}{\sqrt{2^{N+3}}} \, dx \\ &= \frac{\epsilon^2}{4}. \end{aligned}$$

se tiene que

$$||P_j f - f||_2 \leq 2||g - f||_2 < \epsilon.$$

b) De la defínción 2.31, tenemos

$$||P_{-j} f||_2^P 2 \leq 2^{-j} \sum_k (\int_{\Omega_k^j} |f(x)|dx)^2,$$

donde, como f tiene soporte compacto, el sumatorio del lado derecho de la desigualdad está acotado por una constante independiente de j.

□

Definición 2.9. *Para cada $j \in \mathbb{Z}$, definimos el operador Q_j sobre las funciones $f(x)$, L^2 en $\mathbb{R}$, como*

$$Q_j f(x) = P_{j+1} f(x) - P_j f(x). \tag{2.35}$$

Definición 2.10. *Definimos el espacio W_j, llamado espacio Wavelet. Para cada $j \in \mathbb{Z}$,*

$$W_j = span\{\psi_j^k(x)\}_{k\in\mathbb{Z}}. \tag{2.36}$$

Dada la interpretación anterior de $P_j f(x)$ como la versión desenfocada de $f(x)$ a escala 2^{-j}, también podemos interpretar $Q_j f(x)$ como el operador que contiene los detalles de $f(x)$ que son de tamaño menor que 2^{-j} pero mayor que 2^{-j-1}. Es decir, $Q_j f(x)$ contiene aquellos detalles invisibles para la aproximación $P_j f(x)$ pero visibles para la aproximación $P_{j+1} f(x)$.

Lema 2.5. .

1. *Para cada $j \in \mathbb{Z}$, Q_j es un operador lineal,*

$$Q_j(\alpha f + \beta g) = \alpha Q_j(f) + \beta Q_j(g), \quad f, g \in L^2(\Omega), \alpha, \beta \in \mathbb{C}.$$

2. *Para cada $j \in \mathbb{Z}$ y $f \in L^2(\Omega)$,*

$$Q_j Q_j f = Q_j f.$$

3. *Si $g \in W_j$ y $j' \neq j$, entonces $Q_{j'} g = 0$.*

4. *Para cada $j \in \mathbb{Z}$ y $f \in L^2(\Omega)$,*

$$||Q_j f||_2 \leq ||f||_2.$$

Demostración. Todas las propiedades se deducen del lema 2.3. □

Lema 2.6. *Dado $j \in \mathbb{Z}$ y una función $f(x)$ continua y de soporte compacto en $\mathbb{R}$, se tiene que*

$$Q_j f(x) = \sum_k < f, \psi_j^k > \psi_j^k(x), \tag{2.37}$$

donde la suma es finita.

Demostración. Sea $j \in \mathbb{Z}$ dado y sea f una función continua y de soporte compacto en $\mathbb{R}$. Sea $x \in \mathbb{R}$, entonces existe $k \in \mathbb{Z}$ tal que $x \in \Omega_j^k$. Calculamos $Q_j f(x)$. Tenemos que

$$P_{j+1} f(x) = \begin{cases} 2^{j+1} \int_{\Omega_j^{k-}} f(t)\, dt & \text{si } x \in \Omega_j^{k-}, \\ 2^{j+1} \int_{\Omega_j^{k+}} f(t)\, dt & \text{si } x \in \Omega_j^{k+}, \end{cases}$$

y que

$$P_j f(x) = 2^j \int_{\Omega_j^k} f(t)\, dt \quad \text{si } x \in \Omega_j^k.$$

Ahora, para $x \in \Omega_j^{k-}$, se tiene que

$$\begin{aligned} Q_j f(x) &= P_{j+1} f(x) - P_j f(x) \\ &= 2^j \left(2 \int_{\Omega_j^{k-}} f(t)\, dt - \int_{\Omega_j^{k-}} f(t)\, dt - \int_{\Omega_j^{k+}} f(t)\, dt \right) \\ &= 2^j \left(\int_{\Omega_j^{k-}} f(t)\, dt - \int_{\Omega_j^{k+}} f(t)\, dt \right) \\ &= 2^{j/2} \langle f, \psi_j^k \rangle. \end{aligned}$$

y en Ω_j^{k+}, se tiene que

$$\begin{aligned} Q_j f(x) &= 2^j \left(- \int_{\Omega_j^{k-}} f(t)\, dt + \int_{\Omega_j^{k+}} f(t)\, dt \right) \\ &= -2^{j/2} \langle f, \psi_j^k \rangle. \end{aligned}$$

Además, como

$$\psi_j^k(x) = \begin{cases} 2^{j/2} & \text{si } x \in \Omega_j^k-, \\ -2^{j/2} & \text{si } x \in \Omega_j^k+, \\ 0 & \text{en otro caso.} \end{cases},$$

obtenemos que

$$Q_j f(x) =< f, \psi_j^k > \psi_j^k(x).$$

y se deduce (2.37). □

Ahora, veamos cómo sería la base de Haar para el sistema de Haar en $\mathbb{R}$. Sea $J \in \mathbb{Z}$ fijo. El sistema de Haar de escala J es,

$$\{\phi_J^k(x),\ \psi_j^k(x) : j \geq J,\ k \in \mathbb{Z}\}. \tag{2.38}$$

Teorema 2.3. *El sistema de Haar en $\mathbb{R}$ a escala J es un sistema ortonormal completo en $\mathbb{R}$.*

Demostración. La ortonormalidad se deduce de a partir de los argumentos del teorema 2.2, de las observaciones 2.1 y 2.2 y de las definciones de ϕ_j^k y ψ_j^k. Para probar la completitud, como el espacio de funciones continuas de soporte compacto es denso en $L^2(\Omega)$, basta con probar que, si f es una función continua de soporte compacto, entonces

$$f \in \overline{span\{\phi_J^k(x),\ \psi_j^k(x) : j \geq J,\ k \in \mathbb{Z}\}}.$$

Luego, sea $\epsilon > 0$ y sea $f(x)$ continua de soporte compacto Ω. Por el Lema 2.2, existe un entero N tal que $\|P_N f - f\|_2 < \varepsilon$. Tomamos $N > J$. Por (2.35),

$$\sum_{j=J}^{N-1} Q_j f(x) = \sum_{j=J}^{N-1} (P_{j+1} f(x) - P_j f(x)) = P_N f(x) - P_J f(x).$$

Por lo tanto, usando el Lema 2.3, tenemos

$$P_N f(x) = \sum_{j=J}^{N-1} \sum_k \langle f, \psi_j^k \rangle \psi_j^k(x) + \sum_k \langle f, \phi_J^k \rangle \phi_J^k(x),$$

donde el sumatorio en k es finito ya que el soporte de $f(x)$ es compacto.

Dado que $P_N f(x) \in \text{span}\{\phi_J^k(x), \psi_j^k(x) \mid j \geq J, k \in \mathbb{Z}\}$ y $\|P_N f - f\|_2 < \varepsilon$, se prueba la completitud. □

Teorema 2.4. *El sistema de Haar en $\mathbb{R}$, $\{\psi_j^k(x) : j \in \mathbb{Z},\ k \in \mathbb{Z}\}$ es un sistema ortonormal completo en $\mathbb{R}$.*

Demostración. Por el Teorema 2.2, el sistema de Haar en $\mathbb{R}$ es ortonormal en $\mathbb{R}$. Para demostrar la completitud, es suficiente mostrar que, dada $f(x)$ continua con soporte compacto, entonces

$$f(x) \in \overline{\text{span}\{\psi_j^k(x) \mid j, k \in \mathbb{Z}\}}.$$

Ahora, sea $\epsilon > 0$ y sea f continua de soporte compacto Ω. Para cualquier $J \in \mathbb{N}$, tenemos que

$$\sum_{j=-J}^{J-1} Q_j f(x) = \sum_{j=-J}^{J-1} (P_{j+1} f(x) - P_j f(x)) = P_J f(x) - P_{-J} f(x).$$

Ahora, usando el Lema 2.2, tenemos que

$$\lim_{J\to\infty} \|P_J f - f\|_2 + \|P_{-J} f\|_2 = 0$$

Finalmente, utilizando la desigualdad de Minkowski, obtenemos

$$\lim_{J\to\infty} \left\| f - \sum_{j=-J}^{J-1} Q_j f \right\|_2 = \lim_{J\to\infty} \|f - P_J f + P_{-J} f\|_2 \leq lim_{J\to\infty} \|P_J f - f\|_2 + \|P_{-J} f\|_2 = 0.$$

Además, por el Lema 2.3 tenemos que

$$\sum_{j=-J}^{J-1} Q_j f(x) = \sum_{j=-J}^{J-1} \sum_{k} \langle f, h_{j,k} \rangle h_{j,k}(x),$$

donde el sumatorio en k es finito con f de soporte compacto. Como

$$\sum_{j=-J}^{J-1} Q_j f(x) \in \text{span}\{h_{j,k}(x) \mid j, k \in \mathbb{Z}\},$$

se concluye que el sistema de Haar es completo. □

Capítulo 3

La transformada discreta de Haar

3.1. La transformada discreta de Haar en una dimensión

Para $J \geq 0$, los resultados del capítulo 2 nos permiten representar una función $f \in L^2([0,1])$ en términos de la base de Haar,

$$f(x) = \sum_{j \geq J} \sum_{k=0}^{2^J-1} d_j^k \psi_j^k(x) + \sum_{k=0}^{2^J-1} c_j^k \phi_j^k(x), \tag{3.1}$$

con $x \in [0,1]$, [16].

Los algoritmos de reconstrucción (2.21) y de descomposición (2.20) motivan la formulación de una versión discreta del desarrollo (3.1). Parece natural buscar la aproximación a partir de los operadores P_N con $N \geq J$.

$$f(x) \approx P_j f(x) = \sum_{k=0}^{2^N-1} < f, \phi_N^k > \ \phi_N^k(x). \tag{3.2}$$

De esta manera, los coeficientes en la base de Haar de $f(x)$ pueden ser aproximados por los coeficientes de $P_N f(x)$

$$(f, \psi_j^k) \approx (P_N f, \psi_j^k) \quad (f, \phi_J^k) \approx (P_N f, \phi_J^k). \tag{3.3}$$

Supongamos que tenemos una secuencia finita de longitud 2^N para algún $N \in \mathbb{Z}$, $\{c_0(k)\}_{k=0}^{2^N-1}$. Asumimos que existe cierta función f que cumple que

$$c_0(k) = (f, \phi_N^k).$$

Fijamos $J \in \mathbb{N}$ con $J < N$, y para cada $1 \leq j \leq J$, definimos:

$$c_j(k) =< f, \phi_{N-j}^k > \quad d_j(k) =< f, \psi_{N-j}^k > . \tag{3.4}$$

Observamos que, de las definiciones (2.32) y (2.33), se tiene que

$$\phi_0^0(x) = 2^{-j/2}\phi_1^0(x) + 2^{-j/2}\phi_1^1(x),$$

$$\psi_0^0(x) = 2^{-j/2}\phi_1^0(x) - 2^{-j/2}\phi_1^1(x).$$

De aquí, se deduce que para cada $l, k \in \mathbb{Z}$,

$$\phi_l^k(x) = \frac{1}{\sqrt{2}}(\phi_{l+1}^{2k}(x) + \phi_{l+1}^{2k+1}(x)), \tag{3.5}$$

$$\psi_l^k(x) = \frac{1}{\sqrt{2}}(\phi_{l+1}^{2k}(x) - \phi_{l+1}^{2k+1}(x)), \tag{3.6}$$

Entonces, a partir de (3.4), (3.5) y (3.6), obtenemos

$$c_j = \frac{1}{\sqrt{2}}(c_{j-1}(2k) + c_{j-1}(2k+1)), \tag{3.7}$$

$$d_j = \frac{1}{\sqrt{2}}(c_{j-1}(2k) - c_{j-1}(2k+1)). \tag{3.8}$$

Y, en forma matricial,

$$\begin{pmatrix} c_j(k) \\ d_j(k) \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} c_{j-1}(2k) \\ c_{j-1}(2k+1) \end{pmatrix},$$

o bien,

$$\begin{pmatrix} c_{j-1}(2k) \\ c_{j-1}(2k+1) \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} c_j(k) \\ d_j(k) \end{pmatrix},$$

Definición 3.1. *Dado $J, N \in \mathbb{N}$ con $J < N$ y $c_0 = \{c_0(k)\}_{k=0}^{2^N-1}$, la transformada discreta de Haar (DHT) de c_0 se define como,*

$$\{d_j(k) : 1 \leq j \leq J; 0 \leq k \leq 2^{N-j} - 1\} \cup \{c_J(k) : \cdot\, 0 \leq k \leq 2^{N-J} - 1\},$$

donde los coeficientes c_j y d_j son,

$$c_j(k) = \frac{1}{\sqrt{2}} \left(c_{j-1}(2k) + c_{j-1}(2k+1)\right),$$

$$d_j(k) = \frac{1}{\sqrt{2}} \left(c_{j-1}(2k) - c_{j-1}(2k+1)\right).$$

La transformada inversa DHT se define como,

$$c_{j-1}(2k) = \frac{1}{\sqrt{2}}(c_j(k) + d_j(k)),$$

$$c_{j-1}(2k+1) = \frac{1}{\sqrt{2}}(c_j(k) - d_j(k)).$$

La Transformada Discreta de Haar (DHT), al igual que su inversa, puede interpretarse como una transformación lineal en un espacio de dimensión finita. Por lo tanto, puede expresarse mediante producto de matrices, de la siguiente manera:

Definición 3.2. *Dado $L \in \mathbb{N}$ par, definimos las matrices:*

- *H_L: matriz de dimensiones $\frac{L}{2} \times L$ definida como,*

$$H_L = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & & \dots & & 0 \\ 0 & 0 & 1 & 1 & 0 & \dots & 0 \\ & & & & \vdots & & \\ 0 & & \dots & & 0 & 1 & 1 \end{bmatrix}.$$

- *G_L: matriz de dimensiones $\frac{L}{2} \times L$ y definida como,*

$$G_L = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 & 0 & & ... & & 0 \\ 0 & 0 & 1 & -1 & 0 & ... & 0 \\ & & & & \vdots & & \\ 0 & & ... & & 0 & 1 & -1 \end{bmatrix}.$$

- *W_L: matriz de dimensiones $L \times L$ tal que,*

$$W_L = \begin{pmatrix} H_L \\ G_L \end{pmatrix}. \tag{3.9}$$

La matriz H_L se llama la matriz de aproximación, la matriz G_L se conoce como la matriz de detalle y W_L como la matriz wavelet.

De la definición (3.2) y de la forma de las matrices H_L, G_L, se tiene que las columnas de W_L son ortogonales, por lo que W es una matriz ortogonal. Si I_L es la matriz identidad de $L \times L$ y W_L^* denota la matriz adjunta de W_L, entonces

$$I_L = W_L^* W_L = H_L^* H_L + G_L^* G_L. \tag{3.10}$$

Teorema 3.1. *Sea $J, N \in \mathbb{N}$, con $J < N$, y un vector*

$$c_0 = (c_0(0), c_0(1), \ldots, c_0(2^N - 1)),$$

de longitud 2^N. La DHT de c_0 es entonces,

$$(d_1 \quad d_2 \quad ... \quad d_J \quad c_J),$$

donde

$$\begin{pmatrix} c_j \\ d_j \end{pmatrix} = \begin{pmatrix} H \\ G \end{pmatrix} c_{j-1}, \quad \textit{para } 1 \leq j \leq J,$$

con $H = H_L, G = G_L, L = 2^N$. La transformada inversa viene dada por,

$$c_{j-1} = H^* c_j + G^* d_j.$$

Demostración. Sea c_0

$$c_0 = \left(c_0(0),\, c_0(1),\, \ldots,\, c_0(2^N - 1)\right),$$

y, para $1 \leq j \leq J$, sea

$$c_j = \left(c_j(0),\, c_j(1),\, \ldots,\, c_j(2^{N-j} - 1)\right), \quad d_j = \left(d_j(0),\, d_j(1),\, \ldots,\, d_j(2^{N-j} - 1)\right).$$

Entonces, de las definiciones (3.1) y (3.2) con $L = 2^N$ y $W = W_L$, la DHT de c_0 está dada por

$$\begin{pmatrix} c_j \\ d_j \end{pmatrix} = W c_{j-1} = \begin{pmatrix} H \\ G \end{pmatrix} c_{j-1},$$

o, lo que es lo mismo,

$$c_j = H c_{j-1}, \quad d_j = G c_{j-1},$$

donde H y G son matrices de dimensión $2^{N-j} \times 2^{N-j+1}$.

Ahora, veamos cómo sería para la transformada inversa de Haar. Esta se obtiene mediante,

$$c_{j-1} = W^* \begin{pmatrix} c_j \\ d_j \end{pmatrix} = \begin{pmatrix} H^* \\ G^* \end{pmatrix} \begin{pmatrix} c_j \\ d_j \end{pmatrix},$$

luego,

$$c_{j-1} = H^* c_j + G^* d_j.$$

Entonces el algoritmo de la Transformada Discreta de Haar (DHT) se reduce a una multiplicación matricial. □

Proposición 3.1. *Dado $N \in \mathbb{N}$, la DHT de un vector de longitud 2^N puede computarse con a lo sumo 2^{N+1} multiplicaciones.*

Demostración. Sea c_0 el vector,

$$c_0 = \Big(c_0(0),\, c_0(1),\, 0,\, 0,\, \ldots,\, c_0(2^N - 1)\Big).$$

c_1 se calcula de la siguiente manera,

$$c_1(k) = \frac{1}{\sqrt{2}} \left(c_0(2k) + c_0(2k+1)\right).$$

Esto requiere una multiplicación por cada $0 \leq k \leq 2^{N-1} - 1$.

Asimismo, d_1 se calcula como sigue,

$$d_1(k) = \frac{1}{\sqrt{2}} \left(c_0(2k) - c_0(2k+1)\right).$$

Lo que también requiere una multiplicación por cada $0 \leq k \leq 2^{N-1} - 1$. Por lo tanto, el total de multiplicaciones para obtener c_1 y d_1 es 2^N.

De forma análoga, el cálculo de c_2 y d_2 requiere 2^{N-1} multiplicaciones. Por lo tanto, el número total de multiplicaciones necesarias está dado por,

$$\sum_{j=1}^{J} 2^{N-j+1} = 2^{N+1} \sum_{j=1}^{J} 2^{-j} = 2^{N+1} \left(1 - \frac{1}{2^J}\right) < 2^{N+1}.$$

□

3.2. La transformada de Haar en dos dimensiones

Ahora nos vamos a centrar en la Transformada discreta de Haar en dos dimensiones, [16]. Es especialmente útil para el procesamiento de imágenes, ya que las señales que se analizan se representan de forma matricial, al igual que las imágenes.

Por lo tanto, vamos a trabajar con matrices de tamaño $L \times M$ de la forma:

$$c = \{c(n,m) : 0 \leq n \leq L-1,\ 0 \leq m \leq M-1\}.$$

Teniendo como objetivo de esta sección definir una generalización de la Transformada Discreta de Haar (DHT) para matrices.

3.2.1. Matrices de aproximación y detalle

Para poder realizar esta generalización de la DHT a dos dimensiones, primero tenemos que aplicar por separado la transformada de Haar a las filas y columnas de la señal.

Sea $L \in \mathbb{N}$ un número par, sean H y G las matrices de tamaño $\frac{L}{2} \times L$ definidas en la definición 3.2 y dada c una matriz de tamaño $M \times L$ tal que

$$c = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{M-1} \end{pmatrix}. \tag{3.11}$$

Definimos la matriz de aproximación por filas de c, $H^{row}c$, como la matriz de tamaño $M \times \frac{L}{2}$ definida por,

$$H^{row}c = \begin{pmatrix} Hc_0 \\ Hc_1 \\ ... \\ Hc_{M-1} \end{pmatrix} \tag{3.12}$$

También, podemos construirla matriz de aproximación por filas de c de la siguiente manera, según (3.7),

$$(H_{\text{row}}c)(n,m) = \frac{1}{\sqrt{2}}\, c(n,2m) + \frac{1}{\sqrt{2}}\, c(n,2m+1). \tag{3.13}$$

De la misma manera, definimos la matriz de detalle por filas de c, $G_{\text{row}}c$, como,

$$G^{row}c = \begin{pmatrix} Gc_0 \\ Gc_1 \\ ... \\ Gc_{M-1} \end{pmatrix}. \tag{3.14}$$

Y, por (3.7), podemos construirla de la siguiente manera,

$$(G_{\text{row}}c)(n,m) = \frac{1}{\sqrt{2}}\, c(n,2m) - \frac{1}{\sqrt{2}}\, c(n,2m+1). \tag{3.15}$$

Podemos observar que $H_{\text{row}}c$ se obtiene al hacer el producto de cada fila de c por la matriz H. Y, de manera análoga, $G_{\text{row}}c$ al hacer el producto de cada fila de c por la matriz G.

Ahora, vamos a calcular la DHT para las columnas. Sea $L \in \mathbb{N}$ par y c una matriz de tamaño $L \times M$ de la forma

$$c = (c_0,\ c_1,\ \ldots,\ c_{M-1})\,. \tag{3.16}$$

Definimos la matriz de aproximación por columnas de c, $H_{\text{col}}c$, como la matriz de tamaño $\frac{L}{2} \times M$ dada por,

$$H^{col}c = (Hc_0 \quad Hc_1 \quad ... \quad Hc_{M-1}), \tag{3.17}$$

y, podemos construir tambíen de la siguiente manera, utilizando (3.7) de nuevo,

$$(H_{\text{col}}c)(n,m) = \frac{1}{\sqrt{2}}\, c(2n,m) + \frac{1}{\sqrt{2}}\, c(2n+1,m). \tag{3.18}$$

A continuación, definimos la matriz de detalle por columnas, $G_{\text{col}}c$, definida como sigue,

$$G^{col}c = (Gc_0 \quad Gc_1 \quad ... \quad Gc_{M-1}), \tag{3.19}$$

y, podemos escribirla de la siguiente manera mediante (3.7),

$$(G_{\text{col}}c)(n,m) = \frac{1}{\sqrt{2}}\, c(2n,m) - \frac{1}{\sqrt{2}}\, c(2n+1,m). \tag{3.20}$$

De forma análoga al calculo realizado con las filas, podemos observar que $H_{\text{col}}c$ se obtiene al multiplicar cada columna de c por la matriz H, y que $G_{\text{col}}c$ se obtiene multiplicando cada columna por G.

Con los conceptos de la sección anterior, podemos definir la transformada discreta de Haar para matrices a partir de las matrices de aproximación y detalle.

Durante esta sección, asumimos que las matrices son cuadradas, y que el número de filas y columnas de las matrices que se van a tratar son potencias de dos. Dicho de otra manera, c es siempre una matriz de tamaño $2^N \times 2^N$ para algún $N \in \mathbb{N}$.

Definición 3.3. *Dados $J, N \in \mathbb{N}$ con $J < N$, y una matriz $c_0 = \{c(n,m)\}_{n,m=0}^{2^N-1}$, definimos las matrices de tamaño $2^{N-j} \times 2^{N-j}$ para $1 \leq j \leq J$:*

$$c_j = H^{col}H^{row}\, c_{j-1},$$

$$d_j^{(1)} = G^{col}H^{row}\, c_{j-1},$$

$$d_j^{(2)} = H^{col}G^{row}\, c_{j-1},$$

$$d_j^{(3)} = G^{col}G^{row}\, c_{j-1},$$

donde $H^{col}, G^{col}, H^{row}, G^{row}$ son las matrices de dimensión $2^{N-j} \times 2^{N-j+1}$ definidas en (3.17), (3.19), (3.12) y (3.14) respectivamente.

Entonces, la DHT de c_0 se define como la colección de matrices,

$$\left\{d_j^{(1)},\, d_j^{(2)},\, d_j^{(3)}\right\}_{j=1}^{J} \cup \{c_J\}. \tag{3.21}$$

A continuación, veamos cómo es la inversa de la DHT para matrices. Esta se obtiene a partir de los adjuntos de las matrices H y G por filas y por columnas.

Primero veamos los cálculos por columnas. Dado $L \in \mathbb{N}$ par, y sean H' y G' las matrices inversas de H y G, respectivamente. Dado c una matriz $M \times \frac{L}{2}$ tal que,

$$c = \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{M-1} \end{pmatrix}. \tag{3.22}$$

donde c_j corresponde a la fila $j+1$-ésima, con $j = 0, .., M-1$.

Definimos las siguientes matrices,

- La aproximación adjunta de c por filas:

$$H^{\text{row}^*}c = \begin{pmatrix} H^*c_0 \\ H^*c_1 \\ \vdots \\ H^*c_{M-1} \end{pmatrix}. \tag{3.23}$$

- El detalle adjunto de c por filas:

$$G^{\text{row}^*}c = \begin{pmatrix} G^*c_0 \\ G^*c_1 \\ \vdots \\ G^*c_{M-1} \end{pmatrix}. \tag{3.24}$$

De forma análoga, ahora para filas, dada una matriz de tamaño $\frac{L}{2} \times M$

$$c = (c_0 \quad c_1 \quad ... \quad c_{M-1}) \tag{3.25}$$

donde c_j corresponde a la columna $j+1$-ésima, con $j = 0, .., M-1$. Se define:

- La aproximación adjunta de c por columnas como:

$$H^{\text{col}^*}c = \begin{pmatrix} H^*c_0 & H^*c_1 & \cdots & H^*c_{M-1} \end{pmatrix}. \tag{3.26}$$

- El detalle adjunto de c por columnas como:

$$G^{\text{col}^*}c = \begin{pmatrix} G^*c_0 & G^*c_1 & \cdots & G^*c_{M-1} \end{pmatrix}. \tag{3.27}$$

Combinando (3.23) y (3.24), tenemos que,

$$H^{\text{row}^*}H^{\text{row}}\, c = (H^*Hc_0 \quad H^*Hc_1 \quad ... \quad H^*Hc_{M-1}), \tag{3.28}$$

$$G^{\text{row}^*}G^{\text{row}}\, c = (G^*Gc_0 \quad G^*Gc_1 \quad ... \quad G^*Gc_{M-1}), \tag{3.29}$$

Como $H^*H + G^*G = I$, siendo I la matriz identidad, entonces

$$H^{\text{col}^*}H^{\text{col}}c + G^{\text{col}^*}G^{\text{col}}\, c = c, \tag{3.30}$$

$$H^{\text{row}^*}H^{\text{row}}\, c + G^{\text{row}^*}G^{\text{row}}\, c = c. \tag{3.31}$$

Teorema 3.2. *La inversa de la DHT para matrices está dada por,*

$$c_{j-1} = H^{row\prime}H^{col\prime}c_j + H^{row^*}G^{col^*}d_j^{(1)} + G^{row^*}H^{col^*}d_j^{(2)} + G^{row^*}G^{col^*}d_j^{(3)}, \tag{3.32}$$

donde las matrices $H^{col}, G^{col}, H^{row}, G^{row}$ son matrices de tamaño $2^{N-j-2} \times 2^{N-j-1}$ definidas en (3.17), (3.19), (3.12) y (3.14) respectivamente.

Demostración. Fijemos j tal que $1 \leq j \leq J$. Entonces

$$H^{\text{row}^*} H^{\text{col}^*} c_j + H^{\text{row}^*} G^{\text{col}^*} d_j^{(1)} =$$

$$H^{\text{row}^*} H^{\text{col}^*} H^{\text{col}} H^{\text{row}} c_{j-1} + H^{\text{row}} G^{\text{col}} G^{\text{col}} H^{\text{row}} c_{j-1} = \tag{3.33}$$

$$H^{\text{row}^*} (H^{\text{col}^*} H^{\text{col}} + G^{\text{col}} G^{\text{col}}) H^{\text{row}} c_{j-1} =$$
$$H^{\text{row}^*} H^{\text{row}} c_{j-1}.$$

De forma similar,

$$G^{row*} H^{col*} d_j^{(2)} + G^{row*} G^{col*} d_j^{(3)} = G^{row*} G^{row} c_{j-1}. \tag{3.34}$$

Entonces, como,

$$H^{\text{row}^*} H^{\text{row}} c_{j-1} + G^{\text{row}^*} G^{\text{row}} c_{j-1} = c_{j-1}.$$

se tiene que (3.33) y (3.34) implican (3.32). □

Capítulo 4

Implementación y ejemplos

En este capítulo, vamos a presentar un ejemplo sobre la utilización de los algoritmos de descomposición (2.20) y de reconstrucción (2.21) y su computación en Matlab, [4]. A mayores, se presentarán otros ejemplos relacionados con la transformada de Haar en dos dimensiones, [16]. El capítulo tiene como objetivo que el lector entienda y vea los usos del sistema de Haar y su transformada discreta.

4.1. Implementación de los algoritmos de descomposición y reconstrucción

Comenzamos describiendo la implementación de los algoritmos (2.20) y (2.21).

En una primera etapa, se calculan de los coeficientes c_J^k siguiendo los pasos descritos en el algoritmo de descomposición (2.20). Los valores resultantes se almacenan en el array c_J, que contiene estos 2^J coeficientes.

Una vez se ha completado esta etapa, aplicamos el algoritmo de descomposición (2.20) para calcular los coeficientes d_j^k y c_j^k a partir de los coeficientes c_J. Los $2^J - 1$ coeficientes se almacenan en el array d_J, que servirá posteriormente para la reconstrucción.

Finalmente, si se desea recuperar la información original contenida en los coeficientes c_J^k, se puede utilizar el algoritmo de reconstrucción 2.21. Este procedimiento que se realiza a partir de los coeficientes almacenados en el array d_J, permite reconstruir los coeficientes c_j^k, hasta computar los coeficientes c_J. De esta manera, es posible descomponer y reconstruir una señal o función de manera precisa. Además, los coeficientes c_J^k obtenidos mediante el algoritmo de reconstrucción son prácticamente idénticos a los originales.

4.1.1. Ejemplo 1:

Consideramos la función $f(x) = e^{-x} sen(4\pi x)$ definida en el intervalo $\Omega = [0, 1]$ y tomamos $J = 10$. A continuación, se presenta el código correspondiente. Después, se comentará.

```
f= @(x) exp(-x).*sin(4 * pi * x);
J=10;
c=zeros(1, 2^J);
```

```
for k = 1:2^J
    a = 2^(-J)*k;
    b = 2^(-J)*(k+1);
    c(k) = integral(f, a, b);

end


function [d,d_0,C_0] = descomposicion(J, c)

    C=zeros(J,2^J);
    d=zeros(J,2^J);

    for k=1:2^10
            C(10,k)=c(k);
    end

    for j=9:-1:1
        for k=1:2^j
            C(j,k)=(C(j+1,2*k-1)+C(j+1,2*k))/sqrt(2);
            d(j,k)=(C(j+1,2*k-1)-C(j+1,2*k))/sqrt(2);
        end
    end
    C_0=(C(1,1)+C(1,2))/sqrt(2);
    d_0=(C(1,1)-C(1,2))/sqrt(2);

end



[d,d_0,C_0] = descomposicion(J, c)

function C = reconstruccion(d,d_0,C_0, J)

    C=zeros(J,2^J);

    C(1,1)=(C_0+d_0)/sqrt(2);
    C(1,2)=(C_0-d_0)/sqrt(2);

    for j=1:9
        for k=1:2^j
            C(j+1,2*k-1)=(C(j,k)+d(j,k))/sqrt(2);
            C(j+1,2*k)=(C(j,k)-d(j,k))/sqrt(2);
        end
    end

end
```

```
C = reconstruccion(d,d_0,C_0, J)



K = 2^J;
x = 2^(-J) * (0:K);

figure;
for k = 1:K
   x_interval = [x(k), x(k+1)];
   y_value = [C(10,k), C(10,k)];
   plot(x_interval, y_value, 'r', 'LineWidth', 2);
   hold on;

end

grid on;
xlabel('x');
ylabel('f(x)');
title('Representación por intervalos de los coefientes');
```

Como se puede observar, el código empieza definiendo la función f y el valor de J dado. A continuación, calcula los coeficientes c_J^k. La función 'descomposición' se encarga de implementar el algoritmo de descomposición (2.20) y la función 'reconstrucción', el algoritmo de reconstrucción (2.21). La función de 'descomposición' toma como parámetros los coeficientes c_J^k y J y calcula los coeficientes de la función f en la base de Haar, y los almacena en el vector d. En el proceso, se calculan el resto de coeficientes c_j así como los coeficientes c_0 y d_0. Después, la función 'reconstrucción' toma como parámetros los coeficientes en la base de Haar, d_j^k, así como los coeficientes c_0 y d_0, y calcula los coeficientes de la función f en la base canónica. Finalmente, se representa en la figura 4.1 tanto la función original como los coeficientes c_j obtenidos a través de la función 'reconstrucción' figura 4.2. El resultado son unos coeficientes prácticamente idénticos a los originales.

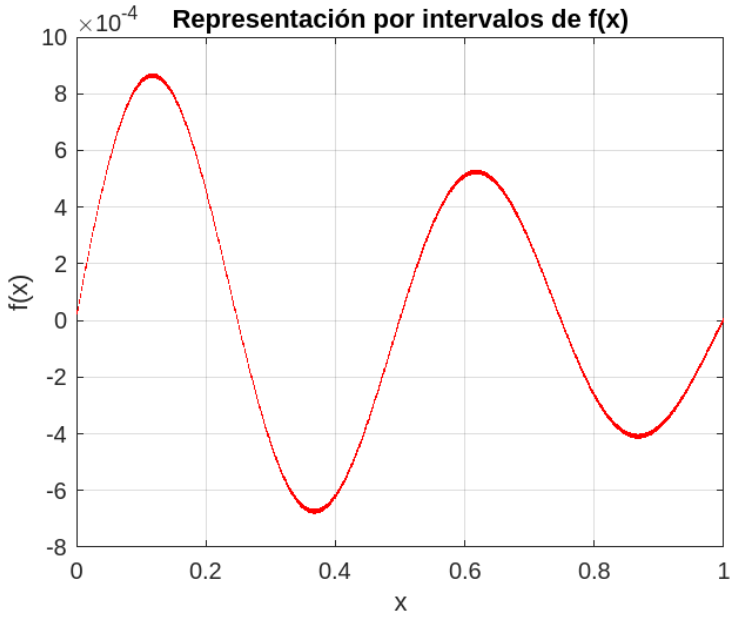

Figura 4.1: Representación de la función $f(x) = e^{-x}sen(4\pi x)$ en el intervalo $[0, 1]$ con $J = 10$.

Figura 4.2: Representación de los coeficientes c_J de la función $f(x) = e^{-x}sen(4\pi x)$ en el intervalo $[0, 1]$ con $J = 10$ obtenidos a través del algoritmo de descomposición y posteriormente del algoritmo de reconstrucción.

La versión anterior del código plantea un cálculo matricial de los coeficientes en ambas funciones. Ahora, se presentan las funciones 'descomposicion' y 'reconstruccion' que, funcionalmente, implementan el mismo uso, pero reducen los cálculos matriciales a cálculos vectoriales. De esta manera, se reduce a vectores el almacenamiento de los coeficientes.

Empezamos con el paso J del algoritmo en que $[c_J] = [d_J]$. En el paso siguiente, $J-1$, los 2^{J-1} coeficientes $c_{J-1}^{k'}$ se guardan en la primera mitad del vector $[d_J]$, en los componentes 1 hasta 2^{J-1}, mientras que los 2^{J-1} coeficientes $d_{J-1}^{k'}$ se guardan en la segunda mitad del vector, en los componentes 2^{J-1} hasta 2^J. En el paso $J-2$, los 2^{J-2} coeficientes $c_{J-2}^{k''}$ se guardan en el primer cuarto del vector $[d_J]$, en los componentes 1 hasta 2^{J-2}, eliminando los coeficientes $c_{J-1}^{k'}$ para $k' = 1, 2, ..., 2^{J-2}$. Después, los 2^{J-2} coeficientes $d_{J-2}^{k''}$ se guardan en el segundo cuarto del vector $[d_J]$, en los componentes $2^{J-2}+1$ hasta 2^{J-1}, eliminando entonces los coeficientes $c_{J-1}^{k'}$ para $k' = 2^{J-2}+1, ..., 2^{J-1}$. Nótese que en este paso, los componentes $2^{J-1}+1$ hasta 2^J del vector $[d_J]$, que corresponden con los coeficientes $d_{J-1}^{k'}$ no se modifican.

Realizando el resto de pasos hasta llegar a $j = 0$, obtenemos el siguiente vector $[d_J]$:

$$[d_J] = [c_0^0, d_0^0, d_1^0, d_1^1, ..., d_j^0, d_j^1, ..., d_j^{2^j-1}, ..., d_{J-1}^0, ..., d_{J-1}^{2^{J-1}-1}]$$

La forma de este vector está relacionada con la descomposición del espacio V_J. Veamos una ilustración de dicha descomposición,

$$V_J = \begin{cases} W_{J-1} \\ \\ \\ V_{J-1} \end{cases} = \begin{cases} W_{J-2} \\ \\ \\ V_{J-2} \end{cases} = \begin{cases} ... \\ \\ ... \\ \\ \\ \\ \end{cases} = \begin{cases} W_0 \\ \\ V_0 \end{cases}$$

El código correspondiente se presenta a continuación:

```
function d = descomposicion(c, J)
    d = c;

    for j = J:-1:1

        daux = zeros(1, 2^j);
```

```
        for k = 0:(2^j/2)-1
            c1 = d(2*k+1);
            c2 = d(2*k+2);
            daux(k+1) = (c1 + c2) / sqrt(2);
            daux(k+1 + 2^j/2) = (c1 - c2) / sqrt(2);
        end


        d(1:2^j) = daux;
    end
end

d = descomposicion(c, J)



function c = reconstruccion(d, J)
    c = d;

    for j = 1:J
        caux = zeros(1, 2^j);

        for k = 0:(2^j/2)-1
            d1 = c(k+1);
            d2 = c(k+1 + 2^j/2);
            caux(2*k+1) = (d1 + d2) / sqrt(2);
            caux(2*k+2) = (d1 - d2) / sqrt(2);
        end

        c(1:2^j) = caux;
    end
end



c = reconstruccion(d, J)
```

4.1.2. Ejemplo 2:

Continuamos con el ejemplo anterior, aunque ahora se va a analizar cómo influye en la aproximación la eliminación de coeficientes muy pequeños. En concreto, se van a eliminar los coeficientes del array d_J que sean menores que un valor muy pequeño ϵ.

Primero, se calcula el nuevo array d_J^ϵ, que es una copia del array original d_J, pero modificando los coeficientes cuyo valor absoluto es menor que ϵ por cero. Sobre este array d_J^ϵ, se utiliza el algoritmo de reconstrucción (2.21) antes programado, para calcular de nuevo los coeficientes c_J^ϵ. Al código utilizado en el ejemplo anterior, se le añade lo siguiente:

```
D=d;
```

```
epsilon=2^(-J/2)*10^(-3);
D(abs(D) < epsilon) = 0;
```

Este código toma los coeficientes d_j^k calculados por la función 'descomposición' y sustituye los coeficientes menores que el valor ϵ calculado por 0. Después, se pasa a la función 'reconstruccion' este nuevo vector D calculado como parámetro. Veamos ahora gráficamente la representación de los coeficientes c_J calculados con el algoritmo de reconstrucción sin la eliminación de los coeficientes menores que ϵ y la representación eliminando los coeficientes menores que ϵ. Esto viene representado en las figuras 4.3, 4.4 y 4.5, respectivamente.

Figura 4.3: Representación de los coeficientes c_J de la función $f(x) = e^{-x}sen(4\pi x)$ en el intervalo $[0, 1]$ con $J = 10$ obtenidos a través del algoritmo de descomposición y posteriormente del algortmto de reconstrucción.

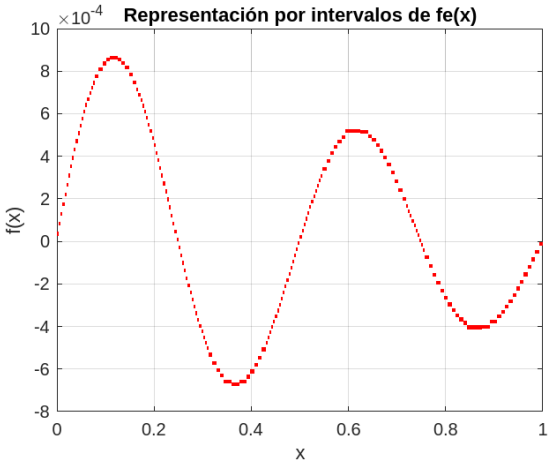

Figura 4.4: Representación de los coeficientes c_J de la función $f(x) = e^{-x}sen(4\pi x)$ en el intervalo $[0, 1]$ con $J = 10$ obtenidos a través del algoritmo de descomposición y posteriormente del algoritmo de reconstrucción y eliminando los coeficientes menores que $\epsilon = 2^{-J/2}10^{-3}$.

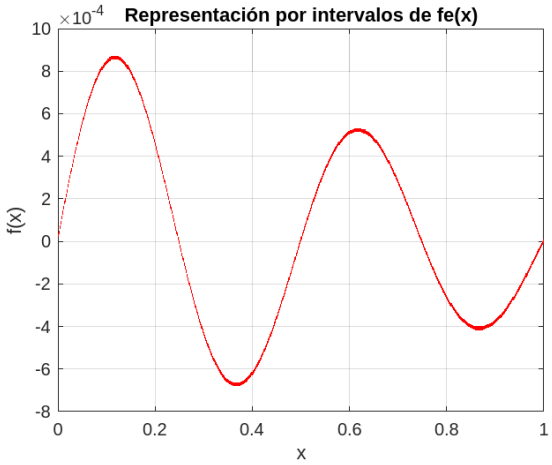

Figura 4.5: Representación de los coeficientes c_J de la función $f(x) = e^{-x}sen(4\pi x)$ en el intervalo $[0, 1]$ con $J = 10$ obtenidos a través del algoritmo de descomposición y posteriormente del algoritmo de reconstrucción y eliminando los coeficientes menores que $\epsilon = 2^{-J/2}10^{-4}$.

En la figura 4.4, se han eliminado 860 coeficientes, mientras que en la figura 4.5, solo 285 coeficientes, debido al ϵ más pequeño. Como se puede apreciar en ambas figuras, la pérdida de detalles no afecta a la forma de la función. Esto se corresponde con una técnica de compresión de datos, en la que eliminamos coeficientes sin perder la información esencial de la función original.

4.2. Implementación de imágenes

Los siguientes ejemplos tienen por objetivo describir cómo se puede utilizar la transformada discreta de Haar, c_j, $d_j^{(1)}$, $d_j^{(2)}$ y $d_j^{(3)}$, presentada en la sección 3.2.

4.2.1. Ejemplo 3:

Se va a describir cómo se puede utilizar la transformada discreta de Haar en la aproximación de imágenes en escala de grises, a partir de las matrices de la DHT c_j, $d_j^{(1)}$, $d_j^{(2)}$ y $d_j^{(3)}$.

Partimos de una imagen en blanco y negro que se almacena como una matriz de números $\{c(n, m)\}$. Cada elemento de la matriz se corresponde con un píxel de la imagen. Como la imagen está en escala de grises, suponemos que cada valor de $c(n, m)$ es un entero no negativo que indica el color del píxel al que corresponde. Los valores de los píxeles van de 0 hasta M, un número suficientemente grande. 0 significa que el píxel es de color negro, y un valor de M significa que es de color blanco. Los valores intermedios entre 0 y M corresponden a los valores de la escala de grises.

Nuestro objetivo es interpretar las matrices de aproximación c_j producidas. Veamos primero la matriz de aproximación c_1 de dimensiones $2^{N-1} \times 2^{N-1}$. Dado que $c_1 = H_{2^N}^{\text{col}} H_{2^N}^{\text{row}} c_0$, se tiene que para cualquier $0 \leq n, m < 2^{N-1}$:

$$c_1(n,m) = \frac{1}{2}\left(c_0(2n,2m) + c_0(2n,2m+1) + c_0(2n+1,2m) + c_0(2n+1,2m+1)\right)$$

De este cálculo, deducimos que los valores de los cuatro píxeles en $(2n, 2m)$, $(2n, 2m+1)$, $(2n+1, 2m)$ y $(2n+1, 2m+1)$ son reemplazados por un único valor $c_1(n, m)$. Este valor se corresponde con el doble del promedio de esos cuatro píxeles. Esto significa que cualquier variación en píxeles dentro de ese bloque de 2×2 se pierde. En otras palabras, c_1 representa solo aquellas características de la imagen que existen a escala 2 o mayor.

Ahora, veamos cómo sería la matriz c_2. c_2 implica tomar el doble del promedio de los cuatro valores $c_1(2n, 2m)$, $c_1(2n+1, 2m)$, $c_1(2n, 2m+1)$ y $c_1(2n+1, 2m+1)$. Dado que cada uno de estos números se calculan como un promedio multiplicado por 2 de un bloque de 2×2 píxeles, cada elemento de c_2 es un promedio multiplicado por 4 de un bloque de 4×4 píxeles. Luego, la variación en los píxeles dentro de ese bloque de 4×4 se pierde. Es decir, c_2 representa solo aquellas características de la imagen que existen a escala 4 o mayor.

De forma análoga para el resto de escalas, se deduce que la matriz c_j representa aquellas características de la imagen que existen a escala 2^j o mayor.

Para observar este efecto vamos a poner un ejemplo práctico. Tomamos una imagen de 256×256 y computamos,

- la matriz c_1 de 128×128,
- la matriz c_2 de 64×64,
- la matriz c_3 de 32×32.

Veamos el código que implementa este ejemplo:

```
img = imread('imagenprueba.jpg');

if size(img, 3) == 3
    img = rgb2gray(img);
end

N = 8;
tamano = 2^N;
img = imresize(img, [tamano, tamano]);
img = double(img);

figure, imshow(uint8(img)), title('Imagen original');

c1 = calculoCj(img);
c2 = calculoCj(c1);
c3 = calculoCj(c2);
```

```
figure, imshow(uint8(c1)), title('Aproximación c1');
figure, imshow(uint8(c2)), title('Aproximación c2');
figure, imshow(uint8(c3)), title('Aproximación c3');

function cj = calculoCj(c0)
    [filas, columnas] = size(c0);


    cj = zeros(filas / 2, columnas / 2);

    for n = 1:(filas / 2)
        for m = 1:(columnas / 2)
            i = 2 * n - 1;
            j = 2 * m - 1;

            cj(n, m) = (c0(i, j) + c0(i, j + 1) + c0(i + 1, j) + ...
            ... + c0(i + 1, j + 1))/2;
        end
    end
end
```

El código empieza leyendo una imagen cualquiera, en nuestro caso 'imagenprueba.jpg'. A continuación, transforma la imagen a escala de grises, en el caso de que no estuviera en escala de grises. Tomamos el valor de $N = 8$ que se corresponde con una imagen de 2^Nx$2^N = 256$x256 píxeles. Después, convertimos la imagen a una matriz cuadrada de tamaño 256x256 en la que se normalizan los valores para realizar un mejor análisis. Se imprime la imagen original. A continuación, se calculan las matrices c_1, c_2 y c_3 y se imprimen por pantalla. Los coeficientes se calculan utilizando la función 'calculoCj'. Esta función toma la imagen que se pasa como parámetro y calcula la nueva aproximación de la siguiente manera,

$$\begin{aligned} c_j(n,m) &= \frac{1}{\sqrt{2}}\Big(H_{2^N}^{\text{row}} c_0(2n,m) + H_{2^N}^{\text{row}} c_0(2n+1,m)\Big) \\ &= \frac{1}{2}\big(c_{j-1}(2n,2m) + c_{j-1}(2n,2m+1) + c_{j-1}(2n+1,2m) + c_{j-1}(2n+1,2m+1)\big). \end{aligned} \tag{4.1}$$

Ahora, ilustremos gráficamente la imagen original y sus aproximaciones c_1, c_2 y c_3, veamos las figuras (4.6)-(4.9):

Figura 4.6: Representación imagen 'imagenprueba.jpg'.

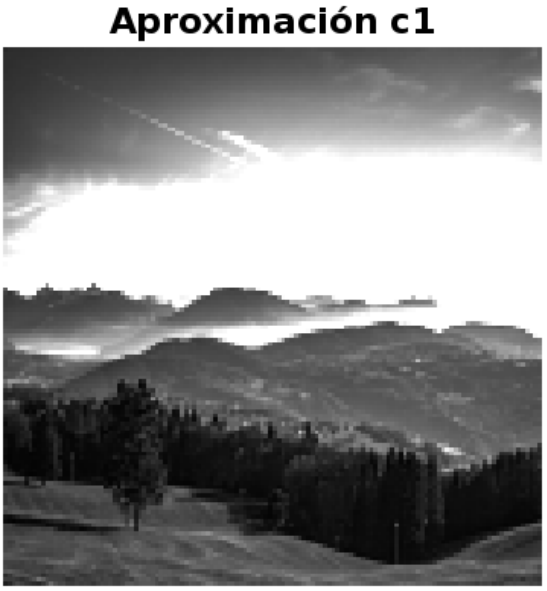

Figura 4.7: epresentación de la aproximación c_1 de la imagen 'imagenprueba.jpg'.

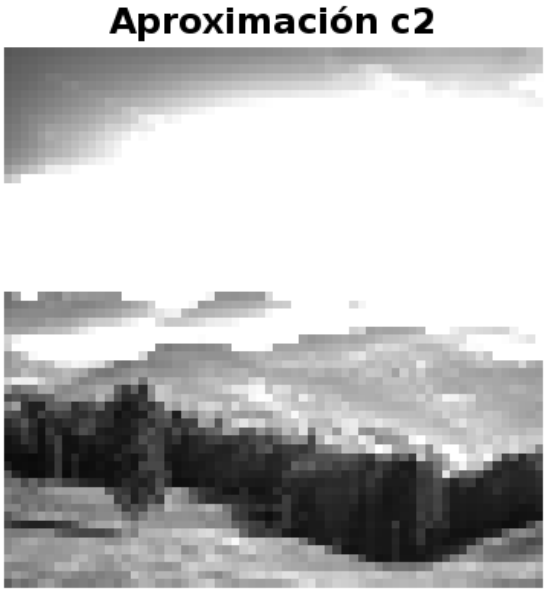

Figura 4.8: Representación de la aproximación c_2 de la imagen 'imagenprueba.jpg'.

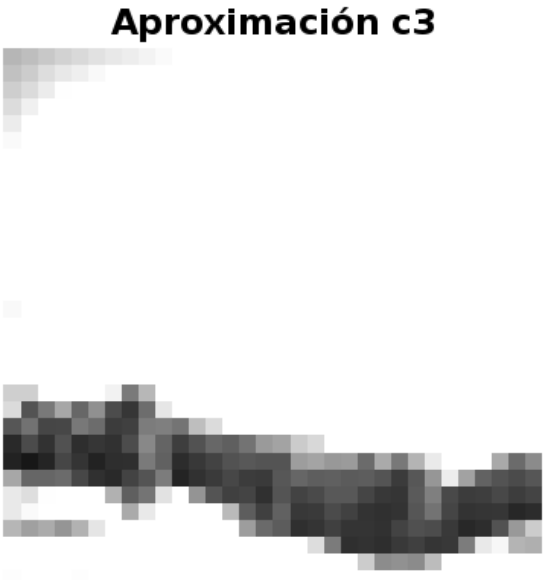

Figura 4.9: Representación de la aproximación c_3 de la imagen 'imagenprueba.jpg'.

Se puede observar cómo en cada aproximación, los píxeles se hacen más grandes y se va perdiendo detalle, hasta en la aproximación c_3 obtener una imagen casi irreconocible.

Continuamos con el ejemplo anterior, aunque ahora enfocándonos en los detalles de la imagen. En el procesamiento de imágenes, identificar cambios en la intensidad de los píxeles es fundamental para identificar bordes dentro de una imagen. Este proceso puede realizarse analizando las variaciones en los valores de los píxeles en horizontal, vertical y diagonal. La Transformada Discreta de Haar aplicada a matrices de píxeles permite

detectar estos bordes mediante las matrices $d_j^{(1)}$, $d_j^{(2)}$ y $d_j^{(3)}$.

Antes de comenzar con el análisis de la DHT, es importante entender qué es un borde. Un borde es un punto donde existe una gran variación en el valor del píxel, es decir, si el valor de un píxel es significativamente diferente del valor de sus vecinos. Cada píxel en una imagen tiene ocho píxeles vecinos: dos en la dirección horizontal, dos en la dirección vertical y cuatro en las dos diagonales. Si en un píxel la variación con los píxeles vecinos es pequeña en la dirección vertical pero grande en la dirección horizontal, entonces ese píxel es un punto de borde vertical de la imagen. De manera análoga, si la variación es pequeña en la dirección horizontal pero grande en la vertical, entonces ese píxel se corresponde con un punto de borde horizontal. Finalmente, si la variación es grande en ambas direcciones, horizontal y vertical, entonces el píxel es un punto de borde diagonal.

Puesto que la DHT para matrices calcula promedios y diferencias de valores de píxeles vecinos en diversas combinaciones, podemos interpretar los coeficientes de la DHT como identificadores de puntos de borde dentro de la imagen. Consideremos la matriz $2^{N-1} \times 2^{N-1}$, $d_1^{(1)}$, derivada de una imagen c_0. Por la definición 3.3, $d_1^{(1)} = G^{row}H^{col}c_0$. Tomando $0 \leq n, m \leq 2^{N-1} - 1$, tenemos

$$\begin{aligned} d_1^{(1)}(n,m) &= \frac{1}{\sqrt{2}}H^{\text{row}}c_0(2n,m) - \frac{1}{\sqrt{2}}H^{\text{row}}c_0(2n+1,m) \\ &= \frac{1}{2}\Big(c_0(2n,2m) + c_0(2n,2m+1)\Big) - \frac{1}{2}\Big(c_0(2n+1,2m) + c_0(2n+1,2m+1)\Big) \\ &\quad + \frac{1}{2}\Big(c_0(2n,2m) - c_0(2n+1,2m)\Big) + \frac{1}{2}\Big(c_0(2n,2m+1) - c_0(2n+1,2m+1)\Big). \end{aligned} \tag{4.2}$$

Interpretemos esta fórmula. Si $(2n, 2m)$ es un punto de un borde horizontal de la imagen c_0, entonces las diferencias

$$c_0(2n,2m) - c_0(2n+1,2m) \quad \text{y} \quad c_0(2n,2m+1) - c_0(2n+1,2m+1),$$

tienden a ser grandes debido a la gran variación en los valores de píxeles en la dirección vertical. Si $(2n, 2m)$ es un punto de un borde vertical, estas diferencias tenderán a ser cercanas a cero. Y, si $(2n, 2m)$ es un punto de borde diagonal, los valores tenderán a ser similares en alguna de las direcciones diagonales, es decir, al menos una de las siguientes diferencias tiende a cero:

$$c_0(2n,2m) - c_0(2n+1,2m+1) \quad \text{ó} \quad c_0(2n,2m+1) - c_0(2n+1,2m).$$

Por lo tanto, si $(2n, 2m)$ es un punto de borde horizontal tenemos que $d_1^{(1)}(n, m)$ tenderá a ser mayor que si $(2n, 2m)$ es un borde vertical o diagonal. El mismo argumento se aplica si el punto de borde está en $(2n, 2m + 1)$, $(2n + 1, 2m)$, o $(2n + 1, 2m + 1)$. Análogamente, $d_1^{(2)}(n, m)$ será máximo si alguno de los puntos mencionados es un borde vertical, y $d_1^{(3)}(n, m)$ será máximo si alguno es un borde diagonal. Veamos cómo son $d_1^{(2)}(n, m)$ y

$d_1^{(3)}(n,m)$:

$$\begin{aligned} d_1^{(2)}(n,m) &= H^{col}G^{row}c_0 \\ &= \frac{1}{2}\left(c_0(2n,2m) - c_0(2n,2m+1)\right) + \frac{1}{2}\left(c_0(2n+1,2m) - c_0(2n+1,2m+1)\right) \\ d_1^{(3)}(n,m) &= G^{col}G^{row}c_0 \\ &= \frac{1}{2}\left(c_0(2n,2m) - c_0(2n,2m+1)\right) - \frac{1}{2}\left(c_0(2n+1,2m) - c_0(2n+1,2m+1)\right) \end{aligned}$$

Estos argumentos se extienden a cualquier escala a nivel j. Por lo tanto, las matrices $d_j^{(1)}$, $d_j^{(2)}$ y $d_j^{(3)}$ se interpretan como identificadores de los bordes horizontales, verticales y diagonales, respectivamente.

Veamos el código asociado a este ejemplo:

```
img = imread('imagenprueba3.jpg');

if size(img, 3) == 3
    img = rgb2gray(img);
end

N = 8;
tamano = 2^N;
img = imresize(img, [tamano, tamano]);
img = double(img);


[c1, d1_h, d1_v, d1_d] = trasformada(img);


img_blur = inversa(c1, zeros(size(d1_h)), zeros(size(d1_v)), zeros(size(d1_d)));
img_horiz = inversa(c1, d1_h, zeros(size(d1_v)), zeros(size(d1_d)));
img_vert = inversa(c1, zeros(size(d1_h)), d1_v, zeros(size(d1_d)));
img_diag = inversa(c1, zeros(size(d1_h)), zeros(size(d1_v)), d1_d);
img_recon = inversa(c1, d1_h, d1_v, d1_d);


figure, imshow(uint8(img)), title('Imagen original');
figure, imshow(uint8(img_blur)), title('Imagen Borrosa (solo c1)');
figure, imshow(uint8(img_horiz)), title('Solo bordes horizontales (d1^1)');
figure, imshow(uint8(img_vert)), title('Solo bordes verticales (d1^2)');
figure, imshow(uint8(img_diag)), title('Solo bordes diagonales (d1^3)');
figure, imshow(uint8(img_recon)), title('Reconstrucción completa');


function [c, d1, d2, d3] = trasformada(img)
    [filas, columnas] = size(img);
    mitadfilas = filas / 2;
    mitadcolumnas = columnas / 2;
```

```
    c  = zeros(mitadfilas, mitadcolumnas);
    d1 = zeros(mitadfilas, mitadcolumnas);
    d2 = zeros(mitadfilas, mitadcolumnas);
    d3 = zeros(mitadfilas, mitadcolumnas);

    for n = 1:mitadfilas
        for m = 1:mitadcolumnas
            i = 2*n - 1;
            j = 2*m - 1;

            a = img(i, j);
            b = img(i, j+1);
            c_ = img(i+1, j);
            d = img(i+1, j+1);

            c(n, m)  = (a + b + c_ + d)/2;
            d1(n, m) = (a + b - c_ - d)/2;
            d2(n, m) = (a - b + c_ - d)/2;
            d3(n, m) = (a - b - c_ + d)/2;
        end
    end
end

function img = inversa(c, d1, d2, d3)
    [filas, columnas] = size(c);
    img = zeros(2*filas, 2*columnas);

    for n = 1:filas
        for m = 1:columnas
            avg = c(n,m);
            dh = d1(n,m);
            dv = d2(n,m);
            dd = d3(n,m);

            img(2*n-1, 2*m-1) = avg + dh + dv + dd;
            img(2*n-1, 2*m)   = avg + dh - dv - dd;
            img(2*n,   2*m-1) = avg - dh + dv - dd;
            img(2*n,   2*m)   = avg - dh - dv + dd;
        end
    end
end
```

El algoritmo toma la imagen y la convierte en una imagen cuadrada de 256x256 píxeles a escala de grises. Posteriormente, utiliza la función 'trasnformada' para calcular las matrices c_1, $d_1^{(1)}$,$d_1^{(2)}$ y $d_1^{(3)}$, utilizando las fórmulas anteriores. Una vez calculadas, se llama a la función 'inversa' para reconstruir las imágenes utilizando la inversa de Haar, definida en

(3.32). De esta definición, se deduce que

$$\begin{aligned} c_{j-1}(2n-1,2m-1) &= c_j(n,m) + d_j^{(1)}(n,m) + d_j^{(2)}(n,m) + d_j^{(3)}(n,m), \\ c_{j-1}(2n-1,2m) &= c_j(n,m) + d_j^{(1)}(n,m) - d_j^{(2)}(n,m) - d_j^{(3)}(n,m), \\ c_{j-1}(2n,2m-1) &= c_j(n,m) - d_j^{(1)}(n,m) + d_j^{(2)}(n,m) - d_j^{(3)}(n,m), \\ c_{j-1}(2n,2m) &= c_j(n,m) - d_j^{(1)}(n,m) - d_j^{(2)}(n,m) + d_j^{(3)}(n,m). \end{aligned}$$

Posteriormente, se imprimen por pantalla la imagen original, la aproximación c_1, la aproximación junto con los bordes horizontales, la aproximación junto con los bordes verticales, la aproximación con los bordes diagonales, y finalmente, la aproximación con todos sus bordes. Las imágenes se muestran en las figuras 4.10-4.15.

Figura 4.10: Representación de la imagen 'imagenprueba.jpg'.

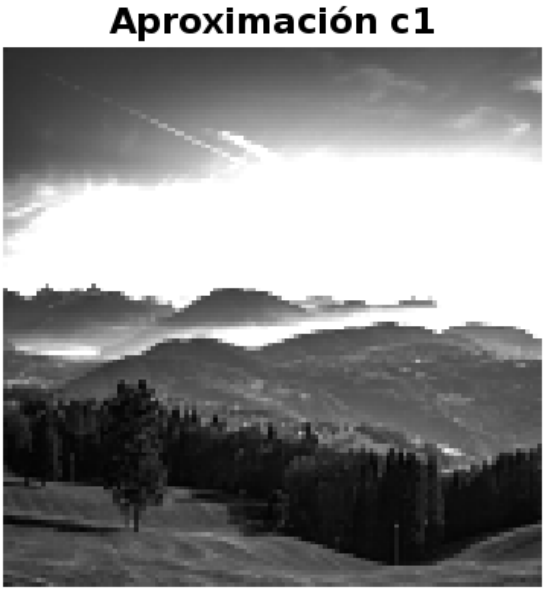

Figura 4.11: Representación de la aproximación c_1 de la imagen 'imagenprueba.jpg'.

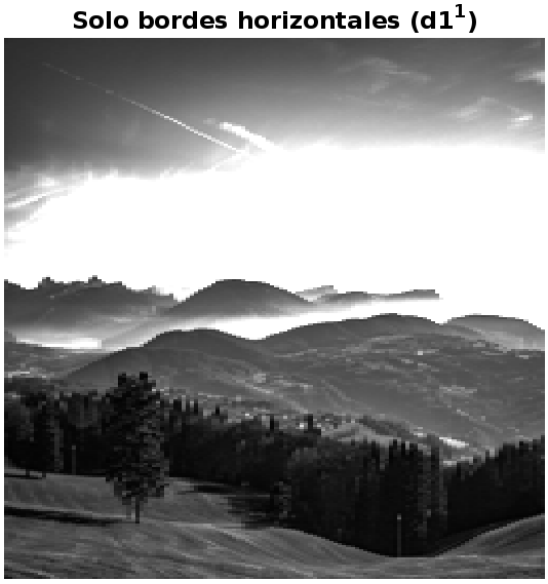

Figura 4.12: Representación de la aproximación c_1 y la matriz $d_1^{(1)}$ de la imagen 'imagenprueba.jpg'.

Figura 4.13: Representación de la aproximación c_1 y la matriz $d_1^{(2)}$ de la imagen 'imagenprueba.jpg'.

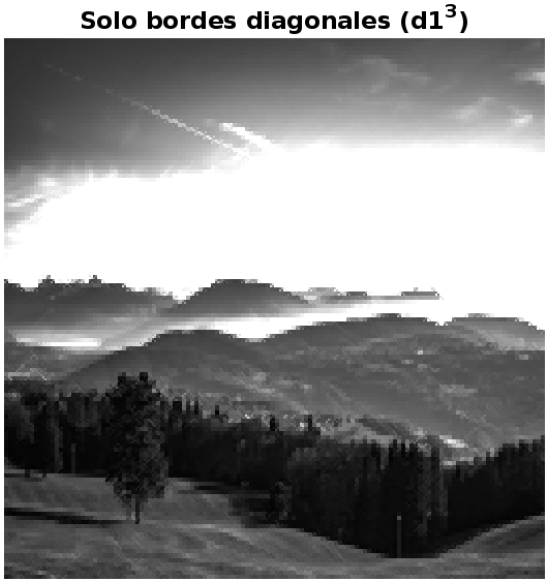

Figura 4.14: Representación de la aproximación c_1 y la matriz $d_1^{(3)}$ de la imagen 'imagenprueba.jpg'.

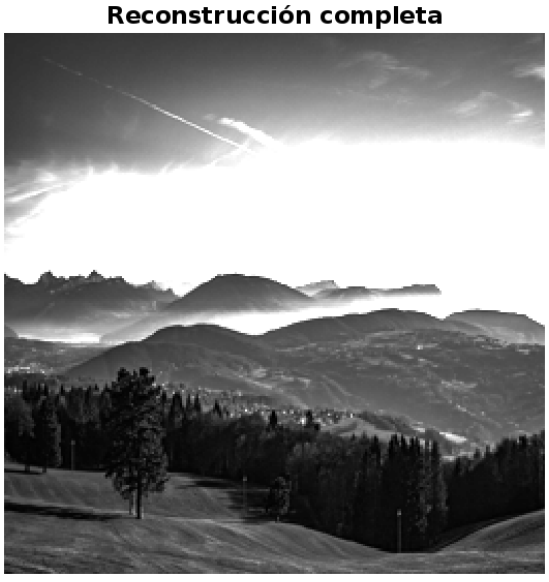

Figura 4.15: Representación de la aproximación c_1 y las matrices $d_1^{(1)}$,$d_1^{(2)}$ y $d_1^{(3)}$ de la imagen 'imagenprueba.jpg'.

Podemos observar los detalles que añaden las matrices $d_1^{(1)}$,$d_1^{(2)}$ y$d_1^{(3)}$ sobre los bordes horizontales, verticales y diagonales, respectivamente, mejorando así la aproximación de la imagen.

En un último ejemplo se va a hacer uso de la DHT para la compresión de imágenes. Vamos a ver cómo permite representar una imagen con una cantidad menor de información sin afectar visiblemente a la calidad de la imagen. Se va a establecer un umbral y la posterior eliminación de coeficientes menores que ese umbral, y se va a comprobar que la mayoría de las características visuales importantes no se pierden.

La clave para una buena compresión de imágenes es encontrar una representación de la imagen con la menor cantidad de números posible. Los coeficientes pequeños van a ser sustituidos por cero sin afectar significativamente a la calidad de la imagen. La idea central de este ejemplo es que al descomponer la matriz c_0 en las cuatro matrices c_1, $d_1^{(1)}$, $d_1^{(2)}$ y $d_1^{(3)}$, hemos separado las partes suaves de las partes no suaves (los bordes) de la imagen. Si la imagen consiste en grandes áreas de intensidad constante separadas por bordes, las matrices de detalle contendrán muchos elementos que son prácticamente cero.

Se va a elegir un umbral, es decir, números fijos por debajo de los cuales los coeficientes se ponen en cero, y se reconstruyen las imágenes correspondientes. El código se presenta a continuación, y se añade al código presentado en el anterior ejemplo.

```
[c1, d1_h, d1_v, d1_d] = trasformada(img);
[c1, d1_h, d1_v, d1_d] = compresion(c1,d1_h,d1_v,d1_d, 0.8)


img_blur = inversa(c1, zeros(size(d1_h)), zeros(size(d1_v)), zeros(size(d1_d)));
img_horiz = inversa(c1, d1_h, zeros(size(d1_v)), zeros(size(d1_d)));
img_vert = inversa(c1, zeros(size(d1_h)), d1_v, zeros(size(d1_d)));
img_diag = inversa(c1, zeros(size(d1_h)), zeros(size(d1_v)), d1_d);
img_recon = inversa(c1, d1_h, d1_v, d1_d);

figure, imshow(uint8(img)), title('Imagen original');
figure, imshow(uint8(img_blur)), title('Imagen Borrosa (solo c1)');
figure, imshow(uint8(img_horiz)), title('Solo bordes horizontales (d1^1)');
figure, imshow(uint8(img_vert)), title('Solo bordes verticales (d1^2)');
figure, imshow(uint8(img_diag)), title('Solo bordes diagonales (d1^3)');
figure, imshow(uint8(img_recon)), title('Reconstrucción completa');



function [ c_nuevo, d1_h_nuevo, d1_v_nuevo,
d1_d_nuevo] = compresion(c, d1, d2, d3, porcentaje)

    transf = [c(:); d1(:); d2(:); d3(:)];

    valorabs = abs(all_coeffs);
    coeffsord = sort(valorabs);
    valorumbral = floor(porcentaje * numel(coeffsord));
```

```
    umbral = coeffsord(valorumbral);

    c_nuevo  = c;
    d1_h_nuevo = d1;
    d1_v_nuevo = d2;
    d1_d_nuevo = d3;

    c_nuevo(abs(c_nuevo) < umbral)   = 0;
    d1_h_nuevo(abs(d1_h_nuevo) < umbral) = 0;
    d1_v_nuevo(abs(d1_v_nuevo) < umbral) = 0;
    d1_d_nuevo(abs(d1_d_nuevo) < umbral) = 0;

end
```

El código calcula las nuevas matrices c_1, $d_1^{(1)}$, $d_1^{(2)}$ y $d_1^{(3)}$ a través de la función 'compresion'. Esta función se encarga de eliminar el porcentaje dado de los coeficientes más pequeños. Para ello, toma el valor absoluto de los coeficientes, los ordena de menor a mayor, calcula el número de coeficientes a eliminar en función del porcentaje, y toma el valor siguiente como umbral. Finalmente, de cada matriz, elimina los coeficientes menores que este umbral. Veamos ahora gráficamente cómo funciona este algoritmo de compresión con los porcentajes 70 %,80 % y 95 %. La información viene representada en las figuras 4.16-4.20.

Podemos observar cómo en la compresión al 70 %, la imagen es prácticamente idéntica. Sin embargo, a medida que comprimimos un mayor porcentaje se pierde más detalle.

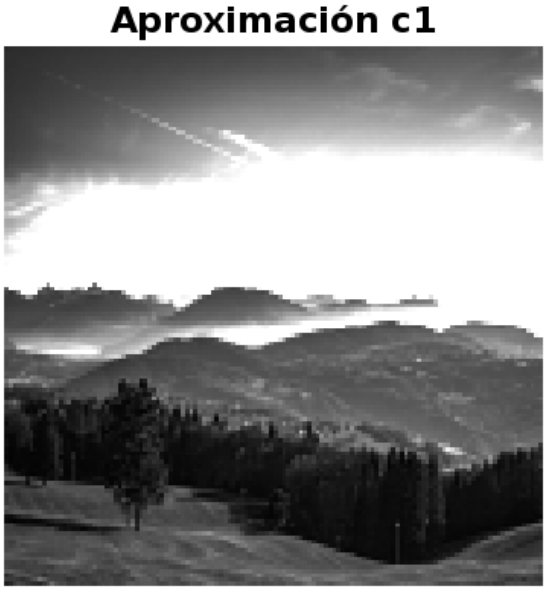

Figura 4.16: Representación de la aproximación c_1 de la imagen 'imagenprueba.jpg'.

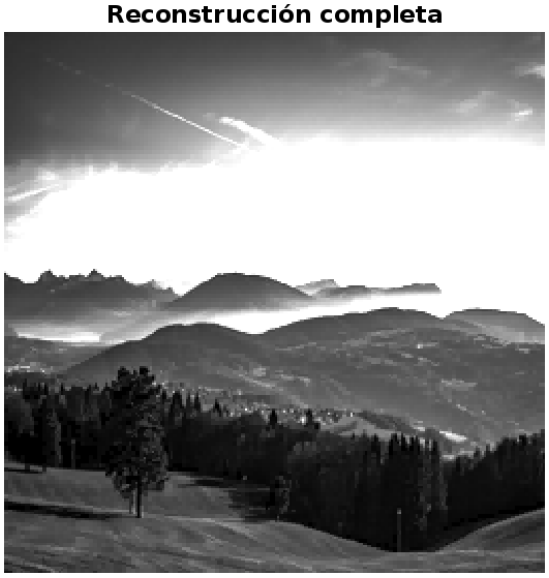

Figura 4.17: Aproximación c_1 de la imagen 'imagenprueba.jpg' comprimiendo la imagen un 70 %.

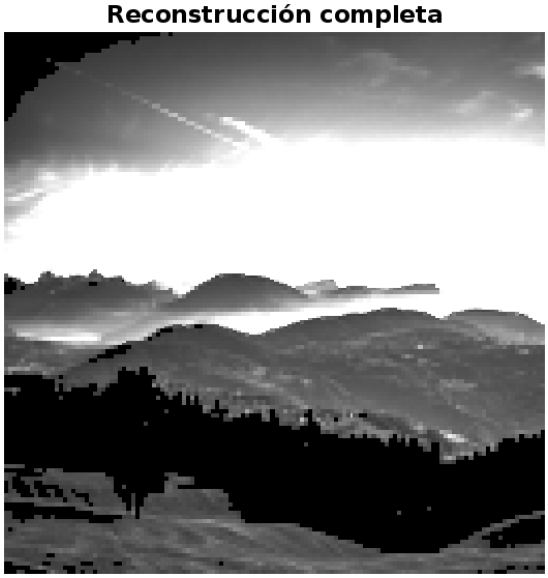

Figura 4.18: Aproximación c_1 de la imagen 'imagenprueba.jpg' comprimiendo la imagen un 80 %.

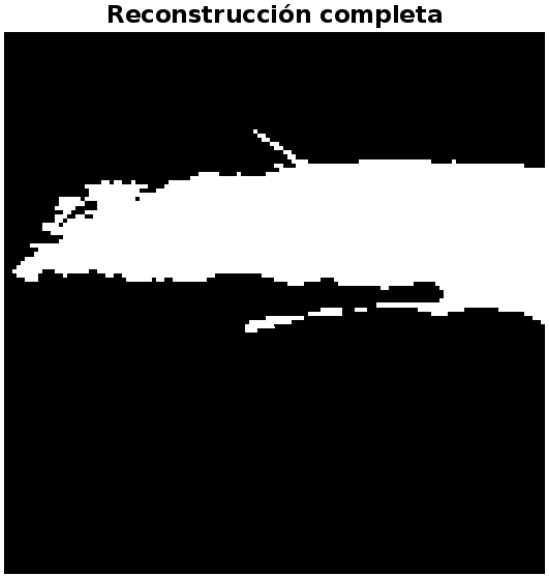

Figura 4.19: Aproximación c_1 de la imagen 'imagenprueba.jpg' comprimiendo la imagen un 95 %.

Bibliografía

[1] R. Balian, *Un principe d´incertitude fort en theorie du siyinal ov en mézcniqve quentique*, C.R. Acad, Sci. Paris Ser. II 292 (1981), 13571362.

[2] G. Beylkin, R. Coifman, V. Rokhlin, *Fast Wavelet Transforms and Numerical Algorithms I*, Comm. Pure Appl. Math., 44 ,1991 141-183.

[3] C. K. Chui, *Wavelets, A Mathematical Tool for Signal Processing*, SIAM, Philadelphia, 1997.

[4] I.Danaila, P.Joly, S.M.Kaber, M.Postel, *An Introduction to Scientific Computing*, Springer, 2023.

[5] I. Daubechies, *Ten Lectures on Wavelets*, CBS-NSF Regional conferencies in applied mathematics, 61, SIAA, 1992.

[6] A. Gabor, *Theory of Communication*, J. Inst. Elect. Eng. London 93(III). 429-457, 11946.

[7] A. Haar, *Zur theorie der orthogonalen Funktionensysteme*, Math. Ann. 69 (1910), 331-371.

[8] E. Hernández, G. Weiss, *A first Course on Wavelets*, CRC Press, 1997.

[9] E. Hernández, *Ondiculas: historia, teoría y aplicación*, Gaceta de le RSME, Vol. 21 (2018), 275-299.

[10] P.G. Lemarié, I. Meyer, *Ondelettes et bases hilbertiennes*, Rev. Mat. Iberoamericans, 2(1986) 1-18.

[11] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic press, 1998.

[12] I.Meyer, *Wavelets and Operators*, Cambridge Univ. Press (1942).

[13] T. Nguyen, G. Strang, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, 1996.

[14] C.E. Shannon, *Communications in The Presence of Noise*, Proceedings of the I.R. E., 37 (1944), 10-21.

[15] J. S. Walker, *A Primer on Wavelets and Their Scientific Aplications*, 2nd ed., Chapman and Hall/CRC, 2008.

[16] D.F.Walnut, *An Introduction to Wavelet Analysis*, Birthhäuser, 2002.