

Estudio de los módulos máximos de la transformada wavelet “à trous” y sus aplicaciones en la detección de bordes

Joaquín Marcos Javier Núñez Cortés

Directora: Dra. Ana María C. Ruedin

29 de agosto de 2005

Resumen

En este trabajo utilizamos la transformada wavelet “à trous” para procesar señales e imágenes. Nuestro objetivo es generar detectores de bordes utilizando la información de multirresolución generada por esta transformada. De esta forma se logra identificar los bordes más importantes y descartar los menos significativos.

Para facilitar la comprensión del funcionamiento de esta transformada se la compara con la transformada wavelet tradicional remarcando similitudes y diferencias.

En la implementación se realizó un análisis cuidadoso y detallado de las convoluciones sucesivas generadas, para lograr un buen centrado de la transformada.

Se extiende la antitransformada wavelet “à trous” a 2 dimensiones, permitiendo de esta forma reconstruir imágenes sin perder información.

Proponemos también un nuevo algoritmo de detección de módulos máximos sobre imágenes, basado en combinaciones lineales entre píxeles, que permite obtener una mayor exactitud en la detección.

Mediante el estudio de la evolución de los módulos máximos a través de las escalas de la transformada se obtiene una imagen binaria con los bordes de la imagen. Nuestro método es comparable al de otros detectores de bordes como Canny, Sobel, Prewitt o LoG. En las zonas donde hay profusión de altas frecuencias, nuestro método ofrece la ventaja de producir una imagen más sencilla y de mayor calidad visual.

Al final del trabajo se implementa un algoritmo iterativo para la reconstrucción de una imagen a partir de los módulos máximos de su transformada.

A mis padres, por apoyarme siempre...

A mi esposa Cinthia y a mi hijo Sebastián, por acompañarme siempre...

Índice general

1. Introducción	7
1.1. Objetivos	8
1.2. Detección de bordes	8
1.3. Análisis wavelet y sus aplicaciones	9
2. Señales	11
2.1. Introducción	11
2.2. ¿Qué es una señal?	11
2.3. Procesamiento de señales	12
2.3.1. Transformadas	12
2.3.2. Transformada de Fourier	13
2.3.3. Importancia del Espacio de Frecuencias	15
2.3.4. Filtrado o convolución	15
2.3.5. Filtrado de señales finitas	15
2.3.6. Convolución circular	16
3. Wavelets	17
3.1. Introducción	17
3.2. ¿Qué es una wavelet?	17
3.3. Transformada wavelet de señales discretas	18
3.3.1. Transformada wavelet “à trous”	19
3.3.2. Transformada “à trous” en una dimensión	19
3.3.3. Comparación entre transformadas “à trous” y tradicional	23
3.3.4. Centrado de la transformada “à trous”	24
3.3.5. Transformada “à trous” en dos dimensiones (imágenes)	31
4. Detección de los módulos máximos	37
4.1. Introducción	37
4.2. Módulos máximos en una dimensión	37
4.3. Módulos máximos en dos dimensiones (imágenes)	39
4.4. Combinación lineal de los píxeles vecinos	44
4.5. Obtención rápida de los módulos máximos	47
5. Detección de bordes	50
5.1. Esquema general de detección de bordes en imágenes	50
5.2. Multirresolución y wavelets	51
5.2.1. Elección de la transformada “à trous”	54

5.3. Métodos alternativos de detección de bordes	54
5.3.1. Método de la propagación entre escalas	55
5.3.2. Método de la evolución de los coeficientes MM	62
5.4. Comparación con los métodos clásicos	68
6. Reconstrucción usando los módulos máximos	76
6.1. Introducción	76
6.2. Reconstrucción en 1D	76
6.2.1. Frames	77
6.2.2. Definición del operador L en 1D	79
6.2.3. Obtención de una aproximación de la señal original	80
6.2.4. Ejemplos de reconstrucción en 1D	81
6.3. Reconstrucción en Imágenes	81
6.3.1. Ejemplos de reconstrucción en imágenes	86
7. Conclusiones	93

Capítulo 1

Introducción

El sistema de visión humano es un sistema complejo que nos permite reconocer el entorno en donde nos encontramos interpretando las variaciones de luz del mismo.

En esta interpretación entran en juego varios subsistemas. El ojo transforma los estímulos luminosos en impulsos eléctricos que llegan hasta el cerebro. Allí son interpretados y asociados con formas. Estas formas son relacionadas entre sí y con otras almacenadas en la memoria para armar la escena, etc. Es un proceso del cual no nos percatamos, pero que al ser analizado en profundidad revela una gran complejidad [19, 26].

Una característica interesante es la capacidad de asociación que tiene el ser humano al ver una imagen. Si vemos varias fotografías de árboles o flores por ejemplo, aunque no las hayamos visto antes, muy probablemente las asociaremos con lo que son, árboles o flores. Esta capacidad de asociación es particularmente notable con los bordes o contornos. Nos es fácil reconocer estructuras por sus bordes. Si le pedimos a un niño de 5 años que dibuje un árbol, seguramente esbozará algunas líneas que (seguramente) tanto nosotros como él interpretaremos como un árbol.

Esto nos lleva a pensar que los bordes de una imagen contienen un alto porcentaje de la información que tiene la misma.

Por otra parte, al mirar una imagen automáticamente realizamos un proceso de segmentación, identificando distintas zonas con distintos conceptos. En este proceso de segmentación, nuevamente lo más importante de la imagen son sus bordes, y no tanto sus texturas.

Siguiendo estos hechos, en el campo de procesamiento de señales e imágenes, se han desarrollado una gran cantidad de técnicas que nos permiten detectar bordes en imágenes, segmentarlas, identificar contornos que nos permitan determinar objetos, etc.

Sin embargo, si por un minuto nos detenemos a analizar lo que vemos tomaremos conciencia de las diferentes escalas que hay a nuestro alrededor. Por ejemplo, si usted está leyendo este trabajo impreso, la escala más importante en este momento serán los contornos que forman las letras del texto. Pero si se aleja un poco de la hoja de papel, verá que los diferentes párrafos forman estructuras independientes. Un poco más allá está el borde de la hoja, etc.

En definitiva, a cada momento nos encontramos con una gran cantidad de escalas conviviendo juntas ante nuestra vista.

Nuestro sistema visual tiene la capacidad de ajustarse a la escala más apropiada dependiendo de la acción que queremos realizar o a los objetos a analizar. De esta forma, dependiendo de la escala en la que nos situemos la escena cambia, y por lo tanto, cambian los bordes de la misma.

El análisis wavelet nos permite dividir una señal o una imagen en diferentes escalas o resoluciones.

En este trabajo se pretende **utilizar la capacidad de análisis de multirresolución wavelet para la detección de bordes**, y de esta manera poder identificar los más importantes.

En este capítulo se describen los objetivos del trabajo y se comentan brevemente los aportes realizados. También se hace una pequeña introducción a la detección de bordes y al procesamiento wavelet. En el capítulo 2 se hace una introducción general a señales, su procesamiento y al concepto de transformadas.

En el capítulo 3 se definen las distintas transformadas wavelet y las relaciones entre ellas. Se realiza

una comparación entre la transformada wavelet “à trous” (la utilizada en este trabajo) y la tradicional, enumerando sus similitudes y diferencias, y justificando la elección de la primera. En este capítulo se explican los diferentes métodos desarrollados para ubicar en las posiciones correctas a los coeficientes de la transformada, algo indispensable para el correcto funcionamiento del resto de las técnicas desarrolladas.

En este capítulo se extiende a imágenes la antitransformada wavelet “à trous” definida por Mallat en [16]. Para ello, se utiliza toda la representación “à trous”, incluyendo los detalles diagonales (ver 3.3.5). Esta extensión nos permite analizar una imagen usando su transformada “à trous”, y reconstruirla sin pérdida.

El capítulo 4 describe el concepto de módulos máximos en general y sobre la transformada wavelet “à trous” en particular. Primero se lo define para señales de una dimensión, y luego se lo extiende a imágenes. En imágenes, se explica la forma de obtención clásica y un método original desarrollado en esta tesis. Este método se basa en la combinación lineal de los vecinos de cada píxel, y nos permite realizar un análisis más riguroso en cuanto a la localización de los módulos máximos detectados.

En el capítulo 5 se presenta el esquema básico de detección de bordes y se describen los métodos desarrollados en esta tesis para la detección de bordes utilizando el análisis de multirresolución que provee la transformada wavelet. Se muestran varios ejemplos para cada método desarrollado, y se realizan comparaciones con los métodos clásicos de detección de bordes.

El capítulo 6 describe un método de reconstrucción de imágenes a partir de sus bordes propuesto por Mallat en [18] y se muestran los resultados obtenidos con la implementación propia que se realizó del mismo.

Finalmente, en el capítulo 7 se enuncian brevemente las conclusiones obtenidas durante el proceso de investigación y desarrollo de este trabajo.

1.1. Objetivos

El principal objetivo del presente trabajo es explorar las capacidades del análisis de multirresolución wavelet en el campo de la detección de bordes.

Para lograrlo, se estudiaron algunas de las propiedades de las distintas transformadas wavelet, como costo de obtención de cada una de ellas, invarianza frente a los desplazamientos y eficiencia en la detección de singularidades tanto en señales unidimensionales como en imágenes.

Finalmente se intenta obtener detectores de bordes que utilicen la información que contiene la imagen en sus diferentes escalas, tratando de lograr identificar los bordes más importantes de la misma, pero sin tener que recurrir a la elección manual de un valor de umbralado.

En base al estudio realizado sobre las transformadas wavelet disponibles, se optó por utilizar la llamada “à trous”. Una vez obtenida esta transformada, es necesario obtener un método que permita identificar correctamente las posiciones de máxima variación de los coeficientes de la misma aplicada a imágenes.

Un objetivo secundario del trabajo es analizar la posibilidad de reconstruir imágenes a partir de los bordes seleccionados, para lo cual se estudia un algoritmo iterativo propuesto por Mallat en [18].

1.2. Detección de bordes

La detección de bordes es fundamental en la visión por computadora, porque usualmente es la primera etapa de un largo proceso de interpretación de las imágenes. El objetivo de la detección de bordes es localizar los píxeles de una imagen que se corresponden con los bordes de los objetos existentes en la misma. Usualmente ésto se realiza mediante una aproximación de la primera o segunda derivada de la imagen, para luego aplicar un test que nos indique si un píxel pertenece a un borde o no [7].

Los bordes caracterizan límites de estructuras que proveen información para identificar objetos en una imagen. La detección de bordes es un paso fundamental para la mayoría de las aplicaciones en visión por computadora como la utilización de sensores remotos, el análisis de huellas digitales, la citología, el reconocimiento de caracteres y la inspección industrial. Una buena imagen de bordes también puede reducir el espacio y el tiempo de procesamiento requerido en los siguientes pasos del análisis de la

imagen utilizando métodos como *labeling*, la transformada de Hough, etc [24].

El sitio *wikipedia.org* define al objetivo de la detección de bordes como:

[...] identificar los puntos de una imagen en los cuales la intensidad de la misma cambia bruscamente. Los cambios bruscos de las propiedades de una imagen usualmente reflejan eventos importantes. La detección de bordes es un importante campo de investigación dentro de la visión por computadora así como en la extracción de características (feature extraction).

Las propiedades de un buen detector de bordes son [3]:

- Buena detección. Debe haber la menor cantidad posible de falsos positivos y falsos negativos.
- Buena localización. El borde se debe indicar lo más cercano posible a su posición correcta.
- Debe indicarse sólo una respuesta para un borde simple.

Históricamente la detección de bordes es un campo de investigación que ha recibido una gran cantidad de aportes de varios campos de la ciencia. En las últimas tres décadas han habido importantes avances [4, 3], pero lo sorprendente es que aún hoy siguen apareciendo trabajos sobre este tema [30, 31, 22, 5, 1, 21, 6, 8] por citar sólo algunas de las publicaciones consultadas durante la realización de este trabajo.

Seguramente la razón de esta investigación continua es que se trata todavía de un problema no resuelto para todas las posibles aplicaciones que surgen día a día. Esta es una de las razones que motivaron este trabajo: la obtención de detectores de bordes que permitan obtener los más importantes de una imagen real, utilizando para ello la información de multirresolución wavelet.

1.3. Análisis wavelet y sus aplicaciones

Las wavelets son funciones que satisfacen ciertos requerimientos. Reciben el nombre *wavelets* (onditas) porque se pide como requerimiento que deben tener integral nula, deben “ondear” hacia arriba y hacia abajo del eje X . La connotación diminutiva de ondita sugiere que la función tiene que estar bien localizada [32].

La idea principal en wavelets es el concepto de *escala*. En la escala más fina se realizan sumas y restas ponderadas entre vecinos. Avanzando hacia una escala mayor se toman sumas y restas ponderadas nuevamente, pero entre elementos más distanciados entre sí. Esto se realiza en forma recursiva, aplicamos la misma transformación en una nueva escala. Así obtenemos una descripción en **multirresolución** de la señal original. En las diferentes escalas se trabaja con aproximaciones y detalles [29].

El análisis wavelet es una herramienta del procesamiento de señales que nos permite analizar la información contenida en una señal o una imagen dividiéndola en diferentes escalas o frecuencias. A diferencia del análisis de Fourier cuyas bases tienen perfecta localización frecuencial pero ninguna localización espacial (o temporal), las wavelets tienen buena localización frecuencial y muy buena localización espacial. Esto permite un análisis más fino, utilizando escalas o resoluciones gruesas en donde hay bajas frecuencias, y escalas más finas en las secciones con altas frecuencias. Es decir que se logra utilizar la escala adecuada en cada porción de la señal que se analiza.

Para una introducción más profunda sobre wavelets se pueden revisar [32, 10, 15, 25].

Las wavelets han sido utilizadas en una gran cantidad de campos de investigación y desarrollo de tecnologías. Se las aplica para analizar señales médicas, geológicas, económicas, etc. Se las utiliza para comprimir señales e imágenes. Permiten detectar singularidades. Su campo de aplicaciones es extremadamente amplio.

La aplicación concreta del análisis wavelet se realiza mediante el uso de transformadas. Existen varios tipos de transformadas, entre ellas podemos nombrar la continua, que se define para señales continuas y es muy útil en los desarrollos teóricos o cuando es necesario desglosar las señales en una cantidad arbitraria de escalas. La transformada más utilizada es la diádica discreta con submuestreos, que se define para señales discretas, y permite una representación muy compacta. En este trabajo se utiliza

la transformada “à trous”, porque brinda un balance apropiado entre información obtenida y tiempo de cálculo.

En esta tesis se la utiliza para extraer la información de las singularidades que presentan distintos tipos de señales de una y dos dimensiones.

Capítulo 2

Señales

2.1. Introducción

En este capítulo se introduce la noción de señal en base al concepto clásico de función. Se establece también la notación que se utilizará en el resto del trabajo. Además se introducen los conceptos básicos del procesamiento de señales como el filtrado de señales y transformadas. No se pretende dar una explicación profunda de estos conceptos sino presentar la idea de los mismos y sus definiciones para poder utilizarlos en los capítulos posteriores. Para consultar un desarrollo en mayor profundidad se recomienda consultar el capítulo 5 de [13] y también [28].

La finalidad del procesamiento de señales como disciplina es tomar información del mundo real, representarla de manera que pueda manipularse fácilmente y procesarla mediante varios métodos para obtener los datos necesarios para un determinado fin. Por lo tanto, lo más importante es la información subyacente. Durante el proceso, esta información puede tomar diferentes representaciones, pero sigue siendo la misma (o una buena aproximación). Es importante tener este concepto en mente.

2.2. ¿Qué es una señal?

Una forma posible y bastante usada de representar la información es mediante funciones:

$$f : C_E \rightarrow C_S$$

En el contexto del procesamiento de señales, C_E suele representar un espacio dimensional. Habitualmente C_E es el tiempo (espacio unidimensional), el plano (espacio bidimensional), o un espacio tridimensional. De esta forma C_S representa las mediciones realizadas sobre el espacio C_E .

Llamaremos *señal* a una función que represente un fenómeno de la realidad.

Por ejemplo, la temperatura promedio dentro de una habitación se puede representar mediante una función $temp(t)$, que toma como entrada un instante de tiempo t , y entrega como salida el valor de temperatura promedio del lugar. Claramente esta función tiene entradas reales y salidas reales $temp : \mathbb{R} \rightarrow \mathbb{R}$, debido a que entre dos instantes de tiempo t_0 y t_1 cualesquiera, existe un t_2 tal que $t_0 < t_2 < t_1$. Esto mismo sucede con el rango de temperaturas entregadas, ya que el valor promedio puede estar entre 10 y 20 grados centígrados por ejemplo, pero puede ser 15, 15.1, 15.01, 15.001, etc.

Otra función a considerar es $Im : \mathbb{R}^2 \rightarrow \mathbb{R}$, que nos permite representar una fotografía en blanco y negro. Así, $Im(x, y)$ representará la intensidad o luminancia de la posición (x, y) . Por supuesto, tanto en este último ejemplo como en cualquier otro caso, es necesario establecer un sistema de referencia adecuado.

En los ejemplos anteriores se trabaja sobre espacios continuos. Sin embargo, debido a las limitaciones técnicas tanto de almacenamiento como de procesamiento, en las aplicaciones prácticas debemos trabajar con espacios discretos. De esta forma, tendremos una función:

$$f : D_E \rightarrow D_S$$

con D_E y D_S espacios discretos. Habitualmente se considera que el fenómeno real es una función continua, y luego a partir de ésta se obtiene una función discreta con la cual se trabaja. A este proceso se lo llama discretización o digitalización, y habitualmente se trata de tomar muestras de la función original respetando ciertas normas. El caso básico es una función continua de reales en reales (como la del ejemplo anterior de la temperatura promedio de una habitación). De esta forma, dada:

$$f_C : \mathbb{R} \rightarrow \mathbb{R}$$

obtenemos

$$f_D : \mathbb{Z} \rightarrow \mathbb{R}$$

donde

$$f_D[n] = f_C(nT)$$

siendo T el período de muestreo.

Notar que para funciones continuas se utilizan paréntesis (), y para funciones discretas corchetes [].

A las funciones discretas se las suele representar en forma vectorial. Para funciones unidimensionales tendremos:

$$f = [\dots f[-1] \quad f[0] \quad f[1] \dots]$$

siendo $f[n]$ la n -ésima componente de f . Estos vectores pueden ser finitos o infinitos. En dimensiones mayores las funciones toman la forma de matrices:

$$g = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & g[-1, -1] & g[-1, 0] & g[-1, 1] & \dots \\ \dots & g[0, -1] & g[0, 0] & g[0, 1] & \dots \\ \dots & g[1, -1] & g[1, 0] & g[1, 1] & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

Si una señal es finita, entonces posee una primera y una última componente. Por convención $f[0]$ corresponde a la primer componente y $f[N-1]$ a la última. De esta forma la señal tendrá N componentes, y se la representa de la siguiente forma:

$$f = [f[0] \ f[1] \dots f[N-1]]$$

2.3. Procesamiento de señales

Se entiende por procesamiento de señales al conjunto de técnicas y procedimientos que se realizan sobre las mismas. Estos pueden aplicarse para lograr diferentes objetivos, como ser realzar determinadas características de la señal (o atenuarlas), obtener información que en primera instancia no es visible, comprimir la señal, dividirla, combinarla con otras, etc.

Dentro del marco del procesamiento de señales existen varias herramientas básicas que nos permiten realizar tareas más complejas. Una de las herramientas fundamentales son las transformadas.

2.3.1. Transformadas

Como se aclaró en el principio del capítulo, las funciones se utilizan para expresar la información. Si mediante un proceso T transformamos una función en otra, y luego mediante el proceso inverso T^{-1} obtenemos nuevamente la función original:

$$\begin{array}{ccc} & \xrightarrow{T} & \\ f_1 & & f_2 \\ & \xleftarrow{T^{-1}} & \end{array}$$

con

$$f_1 : E_1 \rightarrow S_1 \quad y \quad f_2 : E_2 \rightarrow S_2$$

Podemos afirmar que la información representada o almacenada en f_1 no se ha perdido, es decir que f_2 representa la misma información que f_1 pero en un sistema de referencia distinto. Es decir, que se hizo

un *cambio de base* o *transformación*. En el procesamiento de señales, estas transformaciones tienen como finalidad expresar la información en la base adecuada para hacer notoria cierta información de la misma.

Por lo general, las señales se expresan en las bases canónicas. Estas bases son las más comunes o naturales para un determinado espacio.

Para señales continuas $f : \mathbb{R} \rightarrow \mathbb{R}$, se puede utilizar una simplificación que hará más claro este concepto.

Si $\delta(t)$ es la delta de Dirac y cumple:

$$\delta(t) = 0 \text{ para } t \neq 0 \quad y \quad \int_{-\infty}^{+\infty} \delta(t) dt = 1$$

Luego, las funciones se pueden expresar como combinaciones de las translaciones de $\delta(t)$:

$$f(x) = \int_{-\infty}^{+\infty} f(t) \delta(x - t) dt$$

Para más detalles consultar el Apéndice I de [16].

Análogamente las señales discretas $s : \mathbb{Z} \rightarrow \mathbb{R}$ son combinaciones lineales de las translaciones discretas del impulso unitario:

$$s[n] = \sum_{i=-\infty}^{+\infty} s[i] \delta[n - i]$$

donde ahora $\delta[k]$ es la delta de Kronecker y vale:

$$\delta[k] = \begin{cases} 0 & \text{si } k \neq 0 \\ 1 & \text{si } k = 0 \end{cases}$$

Notar que para las señales continuas se utilizan integrales, mientras que para señales discretas se utilizan sumatorias.

Se dice que cuando las señales están expresadas en estas bases se encuentran en el dominio *temporal* o *espacial*, porque se define a la señal indicando cuales son sus valores para cada instante t o posición x .

En el caso de señales discretas finitas de largo N , éstas pertenecen a un espacio de dimensión N . La base de este espacio está formada por un conjunto de vectores de dimensión N :

$$\begin{aligned} v_0 &= [1 \ 0 \ \dots \ 0]^t \\ v_1 &= [0 \ 1 \ \dots \ 0]^t \\ &\vdots \\ v_{N-1} &= [0 \ 0 \ \dots \ 1]^t \\ B_N &= < v_0, v_1, \dots, v_{N-1} > \end{aligned}$$

Luego, las señales de este espacio se pueden escribir como:

$$s = \sum_{i=0}^{N-1} s[i] v_i$$

2.3.2. Transformada de Fourier

En la sección anterior se expresó a las señales en el dominio del tiempo, indicando el valor de las mismas para cada instante t . Existe otro dominio muy utilizado para expresar la información almacenada en una señal, el de las *frecuencias*, también llamado dominio de *Fourier* [11].

En el dominio de frecuencias, las bases son funciones sinusoidales asociadas a una frecuencia determinada. Para simplificar la notación estas funciones se expresan mediante exponenciales complejas de la forma:

$$\begin{aligned} f_\omega : \mathbb{R} &\rightarrow \mathbb{C} \\ f_\omega(x) &= e^{i\omega x} \\ &= \cos(\omega x) + i \sin(\omega x) \end{aligned}$$

La última se llama *Fórmula de Euler*.

El valor ω (omega) especifica la frecuencia con la que oscila la función, y es fijo para cada exponencial compleja.

Se define:

$$L^1(\mathbb{R}) = \left\{ f : \mathbb{R} \rightarrow \mathbb{R} \text{ tal que } \int_{-\infty}^{+\infty} |f(x)| dx < \infty \right\}$$

al espacio de las funciones integrables.

Asimismo, se define:

$$L^2(\mathbb{R}) = \left\{ f : \mathbb{R} \rightarrow \mathbb{R} \text{ tal que } \int_{-\infty}^{+\infty} |f(x)|^2 dx < \infty \right\}$$

al espacio de las funciones de cuadrado integrable o energía finita.

La *transformada de Fourier* primero se define sobre $L^1(\mathbb{R})$, y luego se la extiende a $L^2(\mathbb{R})$ [16].

Finalmente tenemos que dada $f \in L^2(\mathbb{R})$, su *transformada de Fourier* se define como:

$$\begin{aligned} \hat{f} : \mathbb{R} &\rightarrow \mathbb{C} \\ \hat{f}(\omega) &= \int_{-\infty}^{+\infty} f(x) e^{-i\omega x} dx \end{aligned} \tag{2.1}$$

Siendo $\hat{f}(\omega)$ una función perteneciente al dominio de las frecuencias, que nos permite analizar la información de $f(x)$ desde otra perspectiva. Para volver al dominio del tiempo se define la *transformada inversa de Fourier* como:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega) e^{i\omega x} d\omega \tag{2.2}$$

De la misma forma, definimos:

$$l^2(\mathbb{Z}) = \left\{ f : \mathbb{Z} \rightarrow \mathbb{R} \text{ tal que } \sum_n f[n]^2 < \infty \right\}$$

Como el espacio de las señales discretas de energía finita.

Luego dada una señal: $f \in l^2(\mathbb{Z})$ finita de N componentes se define su *transformada de Fourier discreta* o *respuesta en frecuencias* como:

$$\begin{aligned} \hat{F} : [-\pi, \pi] &\rightarrow \mathbb{C} \\ \hat{F}(\omega) &= \sum_{k=0}^{N-1} f[k] e^{-i\omega k} \end{aligned} \tag{2.3}$$

En este caso también existe una forma de retornar al dominio temporal, usando la *transformada discreta inversa*:

$$f = \frac{1}{N} \sum_{k=0}^{N-1} \hat{F}(\omega) e^{i\omega k} \tag{2.4}$$

2.3.3. Importancia del Espacio de Frecuencias

El espacio de frecuencias nos permite analizar la información de una señal desde un enfoque diferente, permitiéndonos separar las diferentes frecuencias que la componen.

Las frecuencias puras son las funciones sinusoidales (por eso la transformada de Fourier expresa a la función como combinaciones de exponenciales complejas, que representan funciones sinusoidales).

Las bajas frecuencias se asocian con señales en donde los cambios son suaves. Las señales con las más bajas frecuencias son las constantes, por ej $f = [1 \ 1 \ 1 \ 1 \ \dots]$. Estas frecuencias se encuentran alrededor del cero en el espacio de frecuencias.

A la inversa, las altas frecuencias se asocian con las señales en donde los cambios son abruptos. La señal discreta que posee la más alta frecuencia es $f = [1 \ -1 \ 1 \ -1 \ \dots]$. Estas frecuencias se encuentran alrededor de π y $-\pi$ en el espacio de frecuencias.

Aplicando estos conceptos a imágenes se asocian las bajas frecuencias con las zonas de transiciones suaves, como sectores de un tono constante o con poca variación. En cambio las altas frecuencias se encuentran en los bordes, en las transiciones bruscas, como puede ser un límite entre sectores de diferente color. Este hecho nos permite la detección de bordes dentro de una imagen, realzando las zonas de altas frecuencias y anulando (llevando a cero) las zonas de frecuencias bajas. Para realizar esta operación se utilizan filtros.

2.3.4. Filtrado o convolución

El filtrado (o convolución) es una herramienta del procesamiento de señales que nos permite tomar como entrada una señal f_e , aplicarle un filtro h y obtener como salida una señal f_s :

$$f_s = f_e * h$$

$$f_s[n] = \sum_p f_e[n-p] \cdot h[p] \quad (2.5)$$

La señal de salida presentará sólo algunas características de la señal de entrada, dependiendo de las propiedades del filtro.

Existen filtros *pasa bajos* que sólo dejan pasar las bajas frecuencias de la señal de entrada y anulan las altas. Filtros *pasa altos*, que permiten el paso de las altas frecuencias y anulan las bajas, y filtros *pasa banda*, que permiten el paso de un rango de frecuencias específico, anulando las frecuencias que se encuentran por debajo o por encima de ese rango.

2.3.5. Filtrado de señales finitas

En la ecuación 2.5 se considera que tanto f_e como h (y por lo tanto también f_s) son infinitos y de soporte compacto. De esta forma, no es necesario definir los límites de la sumatoria.

Al trabajar con señales finitas tenemos que dados:

$$\begin{aligned} f_e[n] & \text{ con } 0 \leq n < N \\ h[m] & \text{ con } 0 \leq m < M \end{aligned}$$

se define su *convolución finita* como:

$$f_s = f_e * h$$

$$f_s[n] = \sum_{p=0}^{M-1} f_e[n-p] \cdot h[p] \quad (2.6)$$

En este caso, la señal de salida f_s tendrá $N + M - 1$ componentes.

2.3.6. Convolución circular

La mayoría de las veces se requiere que la señal de salida posea la misma cantidad de componentes que la señal de entrada. De esta forma, dados:

$$\begin{aligned} f_e[n] & \text{ con } 0 \leq n < N \\ h[m] & \text{ con } 0 \leq m < M \end{aligned}$$

se define su *convolución circular* como:

$$\begin{aligned} f_s &= f_e \circledast h \\ f_s[n] &= \sum_{p=0}^{M-1} f_e[(n-p) \bmod N] \cdot h[p] \end{aligned} \tag{2.7}$$

En este caso, la señal de salida f_s tendrá la misma cantidad de coeficientes que f_e , es decir, N .

Capítulo 3

Wavelets

3.1. Introducción

En este capítulo se introduce el concepto de wavelet que se usará a lo largo de todo el trabajo. Se definen las *transformadas wavelet continua* y *discreta* para señales unidimensionales y para imágenes.

3.2. ¿Qué es una wavelet?

Una wavelet es una función $\psi \in L^2(R)$ tal que:

$$\int_{-\infty}^{+\infty} \psi(t) dt = 0 \quad (3.1)$$

A esta función ψ se la normaliza para tener $\|\psi\| = 1$, y se la centra alrededor de $t = 0$.

Hay dos operaciones fundamentales que se realizan sobre las wavelets: escalado y translación.

Aplicándolas sobre una wavelet ψ se genera toda una familia de wavelets derivadas de ésta y se las nota:

$$\psi_{u,s}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right) \quad (3.2)$$

Cada una de estas nuevas wavelets se encuentra escalada (dilatada o contraída) con la escala s y trasladada en u .

Sobre estas nuevas funciones se define la *transformada wavelet continua* de una función $f \in L^2(R)$ como:

$$Wf(u, s) = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right) dt \quad (3.3)$$

El efecto que se logra al aplicar la transformada a una función f es capturar las frecuencias que ésta contiene dependiendo de la escala s que se tome. Como las wavelets tienen buena localización en tiempo y en frecuencias, permiten un análisis preciso de una señal a diferentes resoluciones (o escalas, o frecuencias).

Al igual que con la transformada de Fourier, con la wavelet se puede ir del espacio temporal al de la transformada wavelet y luego volver nuevamente al espacio temporal usando la antitransformada wavelet. En este caso no se detalla la expresión de la antitransformada continua dado que no es necesaria para el presente trabajo. Su definición puede encontrarse en [16, 29].

En la figura 3.2 se muestra una función continua unidimensional, y la representación gráfica de su transformada wavelet, calculada usando la wavelet *Mexican Hat* (figura 3.1), que corresponde a la derivada segunda de una función Gaussiana, y cuya expresión analítica es:

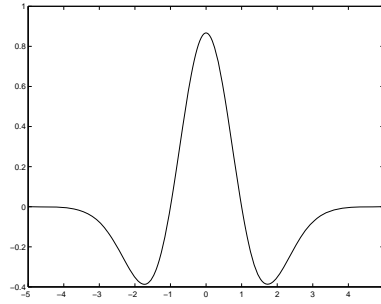


Figura 3.1: Wavelet Mexican Hat

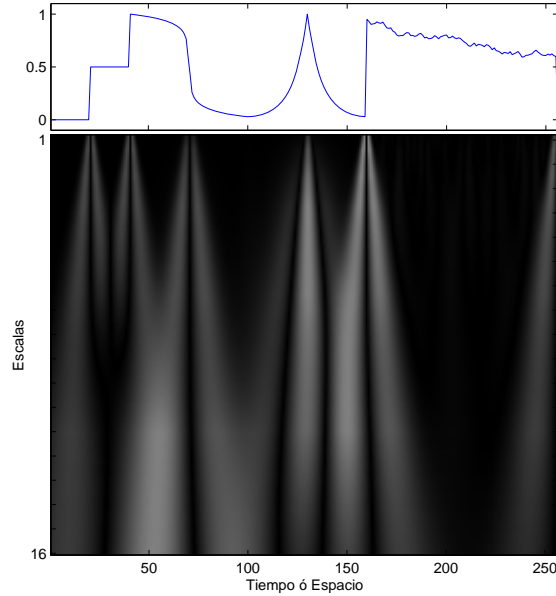


Figura 3.2: Transformada wavelet continua.

$$\psi(t) = \frac{2}{\pi^{1/4} \sqrt{3\sigma}} \left(\frac{t^2}{\sigma^2} - 1 \right) \exp\left(\frac{-t^2}{2\sigma^2}\right)$$

Como se puede ver en la figura anterior, la transformada presenta un comportamiento especial en las singularidades de la función. Estas singularidades pueden ser saltos, picos, ruido u otros cambios bruscos.

Fundamentalmente ésto sucede porque las wavelets son sensibles a las altas frecuencias de la función que están analizando, es decir que se comportan como un filtro, realzando las altas frecuencias o singularidades de la función.

Asociada a la función wavelet ψ existe una función de escala ϕ . Esta función de escala es la contraparte en cuanto a frecuencias de la wavelet, debido a que toma las bajas frecuencias de la función que analiza.

3.3. Transformada wavelet de señales discretas

La transformada wavelet discreta surge de una discretización en el tiempo y en la escala.

Las wavelets utilizadas habitualmente¹ tienen asociados filtros discretos y finitos (en inglés FIR – Finite Impulse Response– Filters) que nos permiten calcular las transformadas de señales discretas en

¹A estas wavelets se las llama *ortogonales* y *biortogonales*. Se puede obtener una definicin formal de las mismas en [16, 29].

forma eficiente mediante convoluciones discretas.

En la transformada wavelet discreta se calculan solamente las escalas que son potencias de dos, por eso se la llama *diádica*. Las ventajas de utilizar solamente estas escalas son principalmente dos:

1. Se puede obtener cada una de las escalas utilizando un algoritmo recursivo en donde cada paso es una convolución discreta.
2. La transformada que se obtiene es completa, es decir que no se pierde información al descartar las escalas intermedias, y se puede reconstruir la señal original a partir de los datos que se retienen.

Una ventaja adicional es que el algoritmo de reconstrucción es muy parecido al de descomposición.

Principalmente existen dos transformadas wavelet (discretas y diádicas) diferentes. Una es la *tradicional*. Esta transformada no es redundante, y se implementa mediante convoluciones seguidas por submuestreos.

La otra es la transformada “à trous”, que es redundante y se obtiene filtrando sin submuestreos. Posee muchos más coeficientes que la anterior, pero se comporta mucho mejor en el análisis de singularidades, ya que es invariante frente a las translaciones, es decir, que si se desplaza la señal de entrada, la salida sólo se modificará en un desplazamiento proporcional.

3.3.1. Transformada wavelet “à trous”

El algoritmo que se utiliza en este trabajo para obtener la transformada wavelet discreta se llama “à trous”, que en francés significa “con agujeros”. Esto es porque para calcular las distintas escalas de la transformada se agregan ceros (agujeros) dentro de los filtros que se utilizan. Este algoritmo fue introducido por *Holschneider, Kronland-Martinet, Morlet y Tchamitchian* en [12].

Se utilizan dos filtros para la transformada y otros dos para la antitransformada. En cada uno de estos pares tenemos un filtro de aproximación y uno de detalle. Los llamaremos h , g y \tilde{h} , \tilde{g} . Los dos primeros son los utilizados al transformar y los segundos al antitransformar. Los h son de aproximación y los g de detalle.

En [18] se establece la relación entre la transformada wavelet continua y la transformada discreta “à trous”. Lo que allí se muestra es que la transformada discreta es igual a la continua en determinados puntos, que dependen del muestreo al discretizar y de un corrimiento dependiente del filtro.

3.3.2. Transformada “à trous” en una dimensión

Antes de dar la definición concreta del algoritmo, es necesario introducir algunas de definiciones auxiliares:

Definimos el **sobremuestreo** de una señal S por un factor p cómo:

$$(S \uparrow p)[k] = \begin{cases} S[r] & \text{si } k = pr \\ 0 & \text{en otro caso} \end{cases}$$

Definimos el **submuestreo** de una señal S por un factor p cómo:

$$(S \downarrow p)[k] = S[pk]$$

Dado un filtro F , se nota $(F \uparrow 2^j)$ al filtro obtenido al insertar $2^j - 1$ ceros entre las componentes de F .

Dado F , definimos su reverso \bar{F} como: $\bar{F}[n] = F[-n]$.

Luego, la **transformada wavelet “à trous”** de una señal a_0 de N componentes² se define como el conjunto de vectores:

² N es diádico, es decir que $N = 2^k$ con $k \in \mathbb{N}$.

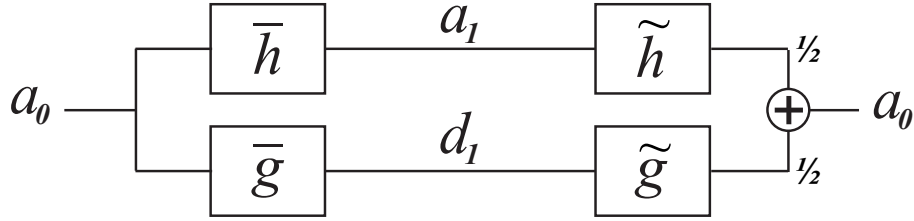


Figura 3.3: Obtención de la primera escala de la transformada wavelet “à trous” de la señal a_0 , y reconstrucción de a_0 a partir de su transformada.

$$\{d_1, d_2, \dots, d_J, a_J\} \quad \text{con } J = \log_2(N) \quad (3.4)$$

donde definimos:

$$\begin{aligned} a_{j+1} &= a_j \otimes (\bar{h} \uparrow 2^j) \\ d_{j+1} &= a_j \otimes (\bar{g} \uparrow 2^j) \end{aligned}$$

como igualdad entre vectores para $0 \leq j < J$ [16, 18].

A los coeficientes a_j se los llama de *aproximación*, porque suavizan la señal del nivel anterior, y a los coeficientes d_j se los llama de *detalle*, porque representan los detalles descartados en la señal suavizada.

Como se puede ver en 3.4, la transformada wavelet “à trous” de a_0 se define como el conjunto de coeficientes de detalle $\{d_j\}$ hasta la escala 2^J , más el residuo de bajas frecuencias a_J .

La última señal de aproximación, a_J es una señal constante, con valor:

$$C_{a_0} = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} a_0[n]$$

En todos los casos se utilizan convoluciones circulares (representadas por el símbolo \otimes), centrando el resultado de forma tal que la morfología de la señal sea lo más parecida de una escala a la siguiente, es decir, que no se presenten desplazamientos entre escalas, estos desplazamientos son intrínsecos del proceso de filtrado. En la sección 3.3.4 se presentan las técnicas utilizadas para centrar la transformada.

De esta forma se obtienen $\log_2(N)$ señales de N componentes, cada una representando una escala de la transformada, más un coeficiente de aproximación.

Para recuperar la señal original a_0 a partir de los coeficientes de la transformada wavelet “à trous”, se utiliza la **antitransformada wavelet “à trous”** definida por:

$$a_j = \frac{1}{2} \left(a_{j+1} \otimes (\tilde{h} \uparrow 2^j) + d_{j+1} \otimes (\tilde{g} \uparrow 2^j) \right) \quad (3.5)$$

Para que valga la fórmula 3.5, se debe cumplir que:

$$\bar{h} * \tilde{h} + \bar{g} * \tilde{g} = 2\delta_0$$

lo cual se verifica para los filtros de la tabla 3.1.

A modo de ejemplo, en las figuras 3.3 y 3.4 se muestran los pasos a seguir para obtener las escalas 2^1 ($[d_1, a_1]$) y 2^2 ($[d_1, d_2, a_2]$) respectivamente, y cómo reconstruir la señal original a_0 a partir de los coeficientes de la transformada.

La transformada wavelet “à trous” divide la información de la señal sobre la que se la aplica en diferentes escalas. Cada una de estas escalas (d_1, d_2, \dots, d_J) retiene una porción específica del espectro de frecuencias de la señal original. De esta forma, analizando las distintas escalas de la transformada, se analizan las distintas frecuencias que componen la señal original. En la figura 3.5 se esquematiza este proceso de selección de frecuencias. La figura muestra el espectro de frecuencias de alguna señal, en el centro se encuentran las bajas frecuencias, y hacia los lados las altas frecuencias. Los tres bloques muestran las porciones de frecuencias retenidas por d_1 , d_2 y d_3 .

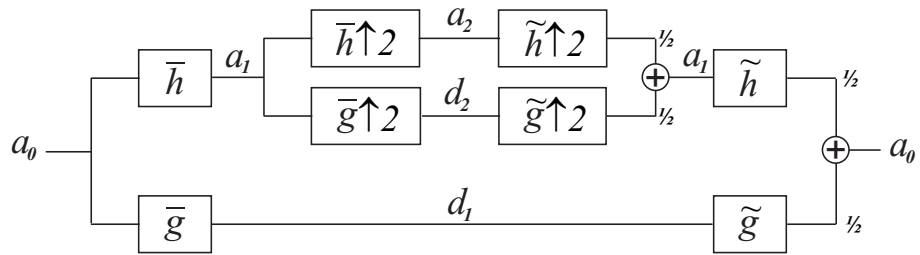


Figura 3.4: Obtención de la primera y segunda escala de la transformada wavelet “à trous” de la señal a_0 , y reconstrucción de a_0 a partir de su transformada.

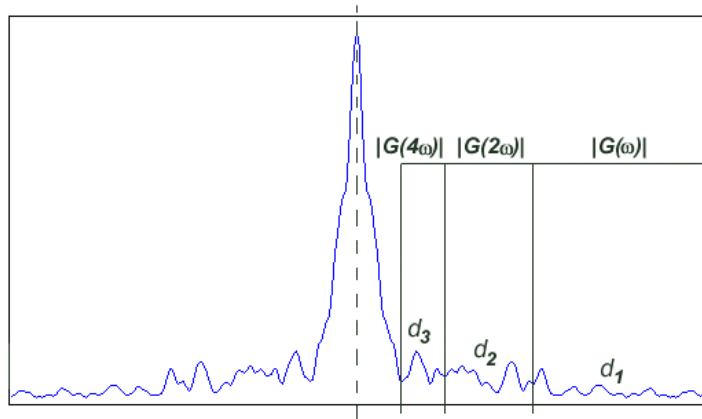


Figura 3.5: Espectro de frecuencias de una señal dada. Los bloques señalan las porciones de frecuencias correspondientes a las primeras tres escalas de la transformada “à trous” de la señal.

$h[n]$	$\sqrt{2} \cdot [0,125 \ 0,375 \ 0,375 \ 0,125]$
$g[n]$	$\sqrt{2} \cdot [-0,5 \ 0,5]$
$\tilde{h}[n]$	$\sqrt{2} \cdot [0,125 \ 0,375 \ 0,375 \ 0,125]$
$\tilde{g}[n]$	$\sqrt{2} \cdot [-0,03125 \ -0,21875 \ -0,6875 \ 0,6875 \ 0,21875 \ 0,03125]$

Tabla 3.1: Filtros asociados a las wavelets biortogonales propuestas por Mallat

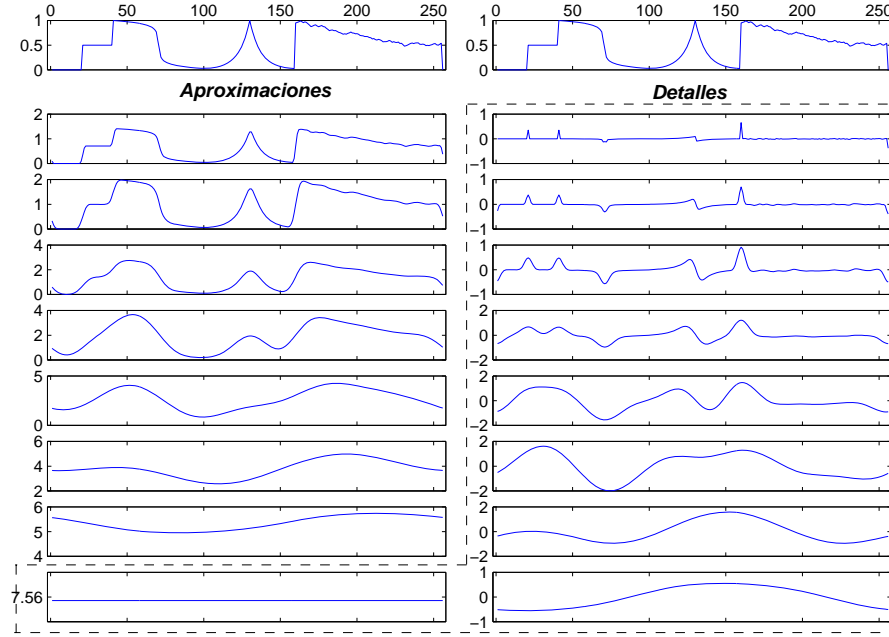


Figura 3.6: Transformada wavelet “à trous” de una señal de 256 componentes. La línea punteada contiene los coeficientes de la transformada.

Ejemplo de la transformada wavelet “à trous”

Para ilustrar el funcionamiento de esta transformada, se procesaron dos señales artificiales y se graficaron todos los pasos de la misma.

Los filtros utilizados para realizar el proceso son los definidos por Mallat en [18], y su respuesta al impulso se da en la tabla 3.1.

Las figuras 3.6 y 3.7 tienen dos columnas, en la primera fila de cada columna se encuentra la señal original a modo de contraste con las filas inferiores. La columna de la izquierda muestra las aproximaciones obtenidas en cada escala, es decir $\{a_j\}_{1 \leq j \leq \log_2(N)}$, mientras que la columna de la derecha muestra los detalles, es decir $\{d_j\}_{1 \leq j \leq \log_2(N)}$.

Se podría finalizar la transformada en cualquier nivel intermedio, y la representación que se obtendría serían todos los detalles de la derecha, más la última aproximación de la izquierda.

En realidad, las aproximaciones intermedias no son parte de la transformada y sólo se utilizan para poder calcular los detalles, ya que en cada nivel d_{j+1} se calcula en función de a_j .

En la figura 3.6 se dibujó una línea punteada que abarca los valores de la transformada final. Estos valores incluyen a todos los coeficientes de detalle de la columna derecha, más el valor de aproximación 7,56 de la izquierda, que es el valor constante de la última aproximación, la del último nivel, en este caso el octavo, debido a que la señal original tiene 256 componentes, y $8 = \log_2(256)$.

Comentarios sobre las figuras anteriores Las aproximaciones de la columna izquierda claramente tienen una forma similar a la señal original, pero se van haciendo más suaves a medida que el

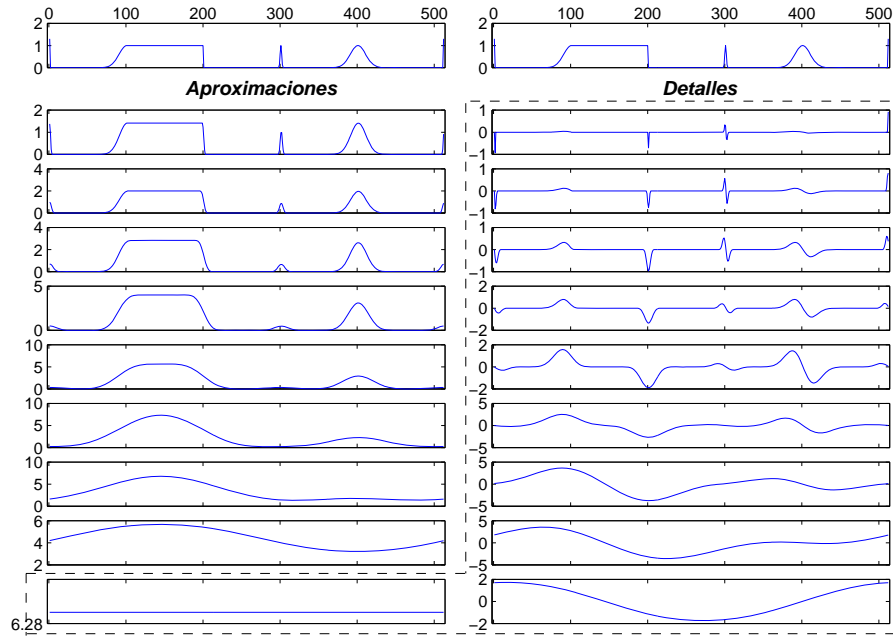


Figura 3.7: Transformada wavelet “à trous” de una señal de 512 componentes. La línea punteada contiene los coeficientes de la transformada.

nivel avanza, hasta llegar a ser una señal constante. Hay que notar, que en cada paso la escala de las señales se incrementa. Por ejemplo en la figura 3.6 la señal original tiene una escala que va del 0 al 1, mientras que su primera aproximación va desde 0 a 2, y la tercera de 0 a 4. Esto se debe a que durante las convoluciones la suma de los valores del filtro de aproximación h es mayor a 1, y por lo tanto, los valores de las señales resultantes de la convolución crecen.

Por otro lado, es interesante notar que en las señales de detalle (columna de la derecha) las singularidades de las mismas aparecen en las posiciones en donde existen cambios abruptos en la señal original, y que a través de las escalas, estas singularidades van evolucionando, en algunos casos crecen, y en otros disminuyen. El tipo de singularidad se debe al tipo de cambio brusco que existe en la señal original, y el estudio de la evolución de la transformada nos permite clasificar las singularidades de la señal original. Se volverá a tratar este tema en el capítulo 5, en la sección 5.3.2 cuando se explique su utilidad en la detección de bordes.

3.3.3. Comparación entre transformadas “à trous” y tradicional

Como se indicó anteriormente, existen dos transformadas wavelets discretas, y si bien en el presente trabajo se utiliza exclusivamente la “à trous”, es interesante compararla con la que probablemente es la más común, a la que llamaremos *tradicional*. La transformada tradicional se calcula utilizando el algoritmo de Cascada, en el que para una señal de entrada de N componentes se obtiene una de salida también de N componentes.

La transformada tradicional se define de la siguiente forma:

$$\begin{aligned} a_{j+1} &= (a_j \otimes \bar{h}) \downarrow 2 \\ d_{j+1} &= (a_j \otimes \bar{g}) \downarrow 2 \end{aligned} \quad (3.6)$$

y la antitransformada:

$$a_j = (a_{j+1} \uparrow 2) \otimes \bar{h} + (d_{j+1} \uparrow 2) \otimes \bar{g} \quad (3.7)$$

Las formulas anteriores corresponden a una transformada wavelet biortogonal, diádica con submuestreo (llamada tradicional). En el caso ortogonal, $\tilde{h} = h$.

La transformada wavelet “à trous” también es una transformada biortogonal diádica pero sin submuestreo.

En cada paso del algoritmo se submuestra el resultado de la convolución, haciendo que la cantidad de componentes obtenidas se reduzca a la mitad.

Esta transformada también es completa y reversible (se puede obtener la señal original a partir de ella), pero contiene muchos menos valores que la transformada “à trous”. Esto nos indica que la transformada “à trous” es redundante, ya que utiliza más componentes para almacenar la misma cantidad de información. Esta redundancia es fundamental para el análisis de singularidades, ya que las mismas son mucho más claras en la transformada “à trous” que en la tradicional.

Pese a que estas transformadas parecen ser totalmente diferentes, se puede ver que si se separan los componentes de la transformada tradicional en forma diádica, y se los translada de forma adecuada, coinciden con los componentes de la transformada “à trous”. La figura 3.8 muestra esta comparación de forma clara.

3.3.4. Centrado de la transformada “à trous”

A partir de este momento, siempre que se haga referencia a la transformada wavelet se tratará de la “à trous”.

Antes de avanzar al caso bidimensional de la transformada, estudiaremos un detalle que hace que los resultados obtenidos sean claros y que además nos permite utilizar esta técnica para detectar bordes en imágenes. Este detalle es el centrado de la transformada.

Al convolucionar una señal con un filtro, se produce un desplazamiento de la señal resultante, debido al funcionamiento del operador de convolución. Este desplazamiento es directamente proporcional al largo del filtro que estemos utilizando, y en el caso de la transformada “à trous”, como los filtros crecen considerablemente en los niveles más altos (por el agregado exponencial de ceros intermedios), los desplazamientos son tan grandes que hacen irreconocible la señal resultante.

Antes de continuar se definen los conceptos de *largo* y *largo efectivo* de una señal:

Dada la señal S :

Diremos que su *largo* es la cantidad de componentes que la integran (s_0, s_1 , etc) y lo notaremos como $largo(S)$ o $largo_S$.

Diremos también que su *largo efectivo* es la cantidad de componentes que tiene la señal sin contar los ceros del principio o del final, y lo notaremos como $largoEfectivo(S)$ o $largoEfectivos$.

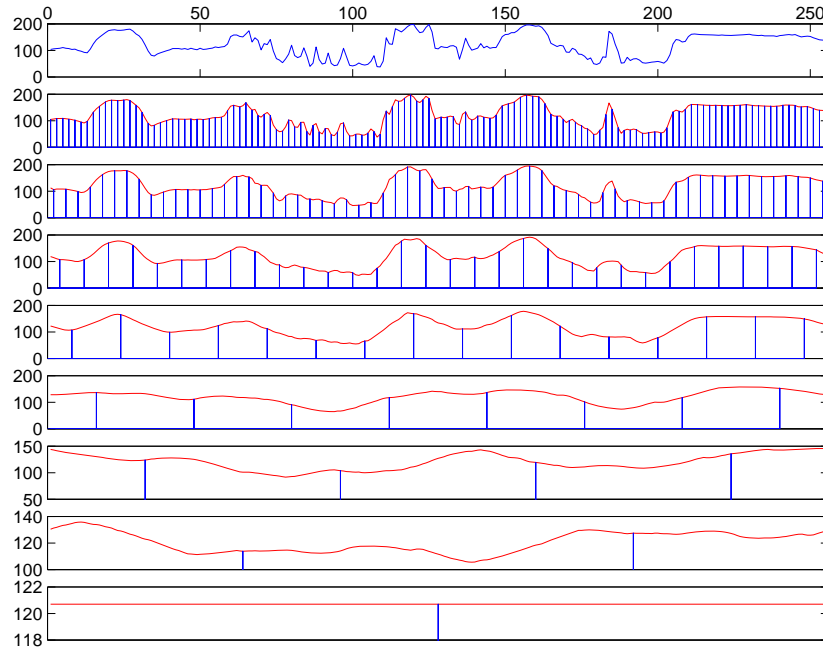
Algunos ejemplos:

$$\begin{aligned} largo([x0x]) &= 3 & , & \quad largoEfectivo([x0x]) = 3 \\ largo([xx000]) &= 5 & , & \quad largoEfectivo([xx000]) = 2 \\ largo([00x0xx]) &= 6 & , & \quad largoEfectivo([00x0xx]) = 4 \end{aligned}$$

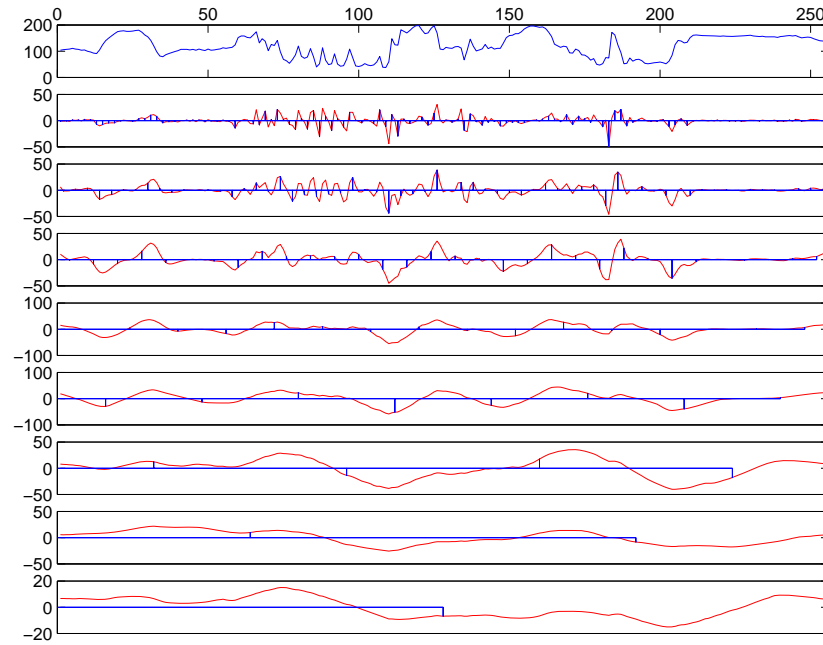
donde x representa un valor real cualquiera.

En el trabajo se desarrollaron 3 técnicas de centrado sobre señales unidimensionales.

En todos los casos se supone que las componentes de las señales se indexan a partir del cero (s_0, s_1 , etc.). En algunos paquetes de software, en cambio, se utiliza como convención indexarlas a partir del 1. Si ese es el caso, los desarrollos siguientes continúan siendo válidos, sólo es necesario tener en cuenta ese detalle al indexar.



(a) Aproximaciones



(b) Detalles

Figura 3.8: Transformadas wavelet “à trous” y tradicional. La gráfica continua pertenece a la transformada “à trous”, mientras que los coeficientes de la tradicional se indican con marcadores verticales.

Centrado Simple

Esta técnica es la que se utilizó para obtener las figuras 3.6 y 3.7. La misma consiste en establecer los centros tanto de la señal como del filtro, y luego, en base a ellos calcular el centro de la señal resultante en cada convolución.

La señal a convolucionar se nota con S y el filtro con que se convoluciona se nota con F .

Los pasos a seguir son los siguientes:

- a) Preparar la entrada para simular una convolución circular.

Para ello se repite la señal S a los lados de ella misma tantas veces como sea necesario dependiendo del tamaño de F .

$$S' = \dots S S S \dots$$

- b) Ubicar el centro de S' y de F .

La idea es ubicar los centros justo en la mitad de las señales:

$$S'_{centro} = \frac{largoEfectivo_{S'}}{2} - 0,5$$

$$F_{centro} = \frac{largoEfectivo_F}{2} - 0,5$$

Se resta 0,5 para que el centro sea exacto. Si la señal tiene 8 componentes (largo efectivo), el centro está entre las componentes 3 y 4. Si en cambio tiene 7 componentes, el centro está exactamente en la componente 3.

- c) Convolucionar.

Se realiza la convolución clásica mediante dos ciclos anidados.

$$R' = S' * F$$

- d) Calcular el centro de R' .

$$R'_{centro} = \left\lfloor S'_{centro} + F_{centro} \right\rfloor$$

- e) Armar el resultado R .

Para ello se toman $largoEfectivo_S$ componentes de R' dejando como centro a R'_{centro} .

Este método, pese a su simplicidad, es muy ineficiente y no se utiliza en la práctica.

Las principales causas de su ineficiencia son:

- El proceso de replicación de la señal original es altamente costoso.
- Se convoluciona una señal mucho más grande que la original, sobre todo en las escalas más gruesas donde los filtros son más largos.
- Los ceros insertados en el filtro al avanzar en las escalas son tratados como el resto de las componentes del mismo, cuando en realidad, ya de antemano se sabe que no es necesario multiplicar por cero.

Utilizando *aConv()*

En Wavelab 8.02 (un paquete dedicado a wavelets), existe una función llamada *aConv* que implementa una convolución circular, pero que utiliza fuertemente una función precompilada para realizar los cálculos, la función *filter*. Esta última función, no sólo es rápida por estar implementada en código de base, sino también porque utiliza la transformada *FFT*, logrando una complejidad del orden de $O(N \log(N))$, siendo $N = \max(\text{largoEfectivo}_S, \text{largoEfectivo}_F)$, con S la señal de entrada y F el filtro.

La utilización de *aConv* nos obliga a desarrollar un método de centrado diferente, dado que entrega como salida una señal desplazada hacia la izquierda. Este desplazamiento es proporcional al largo efectivo del filtro, y el filtro crece en cada paso de la transformada “à trous”.

Al agregar ceros entre las componentes originales de un filtro, su largo y su largo efectivo varían de la siguiente forma:

$$\begin{aligned} F = (F \uparrow 2^0) &= [xxx] \\ (F \uparrow 2^1) &= [x0x0x0] \\ (F \uparrow 2^2) &= [x000x000x000] \\ &\dots \end{aligned}$$

Por lo tanto:

$$\text{largo}(F \uparrow 2^j) = 2^j \text{largo}(F) \quad (3.8)$$

$$\text{largoEfectivo}(F \uparrow 2^j) = 2^j (\text{largo}(F) - 1) + 1 \quad (3.9)$$

Para calcular el factor de corrección necesario se realizó el siguiente experimento.

La idea es tratar de observar cómo se va desplazando la señal a medida que avanzamos en los niveles de la transformada, para ello podemos generar una señal testigo y aplicarle a ella la transformada:

$$S[n] = \begin{cases} 0 & \text{si } 0 \leq n < 256 \text{ y } n \neq 128 \\ 1 & \text{si } n = 128 \end{cases}$$

Esta señal tiene 256 componentes, todas ellas cero, salvo la componente central que es un uno. Las figuras 3.9, 3.10 y 3.11 presentan las convoluciones obtenidas con los siguientes filtros respectivamente:

$$\begin{aligned} F^{(2)} &= 2^{-1} [1 \ 1] \\ F^{(4)} &= 4^{-1} [1 \ 1 \ 1 \ 1] \\ F^{(8)} &= 8^{-1} [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1] \end{aligned}$$

Nota: Si bien el desarrollo se realiza sobre los filtros $F^{(2)}$, $F^{(4)}$ y $F^{(8)}$, el mismo se aplica directamente a cualquier filtro de largo efectivo igual a 2, 4 u 8 respectivamente. Esto es así porque al efectuar el centrado lo importante es el largo efectivo del filtro y no sus componentes. Por otra parte, los resultados pueden extenderse fácilmente a filtros de otras longitudes.

En cada figura, se puede observar que el valor central de la señal original se comienza a dilatar nivel a nivel, y se desplaza hacia la izquierda. El último nivel de la figura 3.9 es como siempre una constante. En la figura 3.10 se pueden notar dos particularidades. Por un lado, en su sexto nivel se ve que la dilatación y el desplazamiento se hacen en forma circular, ya que al desplazarse hacia la izquierda y llegar al borde, la señal continúa desde el borde derecho. Por otro lado, se ve que en este caso los últimos dos niveles son constantes. En la figura 3.11 hay tres niveles constantes, ésto se debe a la longitud inicial del filtro, y a que éste en ese nivel ya ha crecido demasiado y se llega al nivel constante más rápido.

En cada nivel no constante se indica el rango de valores donde se encuentra la dilatación del pico original, y (luego de la flecha) el largo de la misma, estos valores están resumidos en la tabla 3.2.

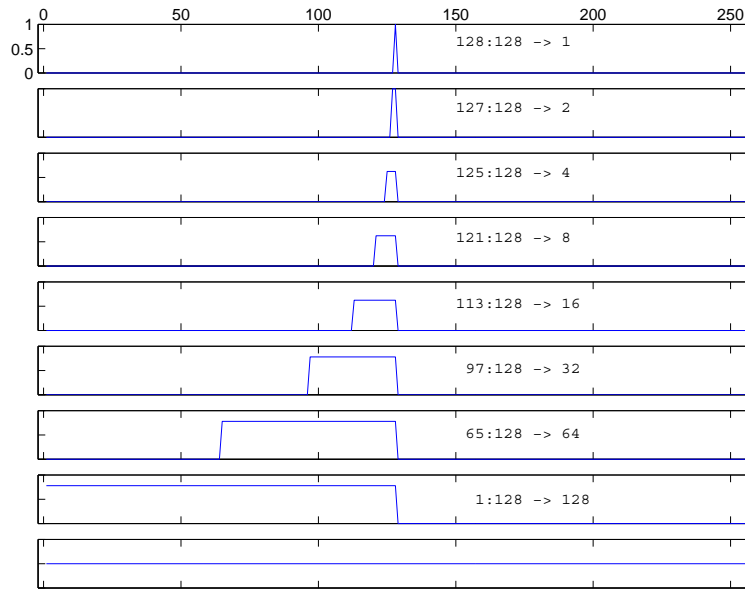


Figura 3.9: Aproximaciones de S calculadas con $F^{(2)}$. Primer renglón $a_0 = S$. Sigüientes renglones $a_{j+1} = a_j \otimes (F^{(2)} \uparrow 2^j)$ para $0 \leq j \leq 7$.

	$F^{(2)}$		$F^{(4)}$		$F^{(8)}$	
Nivel	Ancho	Incremento	Ancho	Incremento	Ancho	Incremento
0 (Señal)	1		1		1	
1	2	1	4	3	8	7
2	4	2	10	6	22	14
3	8	4	22	12	50	28
4	16	8	46	24	106	56
5	32	16	94	48	218	112
6	64	32	190	96		
7	128	64				

Tabla 3.2: Resumen de Dilataciones

Como se muestra en la tabla, el incremento, nivel a nivel, es igual al largo efectivo del filtro utilizado en ese nivel menos uno. Por ejemplo, usando el filtro $F^{(4)}$, al pasar del nivel 2 al 3 el filtro es :

$$F^{(4)} \uparrow 2^2 = \begin{bmatrix} \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 \end{bmatrix}$$

con $\text{largoEfectivo}(F^{(4)} \uparrow 2^2) = 13$, y como se ve en la tabla el incremento es de 12. Este resultado es consistente con la fórmula del incremento de longitud en una convolución:

Si

$$\text{largoEfectivo}(S) = N \quad \text{y} \quad \text{largoEfectivo}(F) = M$$

entonces

$$\text{largoEfectivo}(S * F) = N + M - 1 \quad (3.10)$$

En este caso estamos hablando de una convolución no circular de señales finitas.

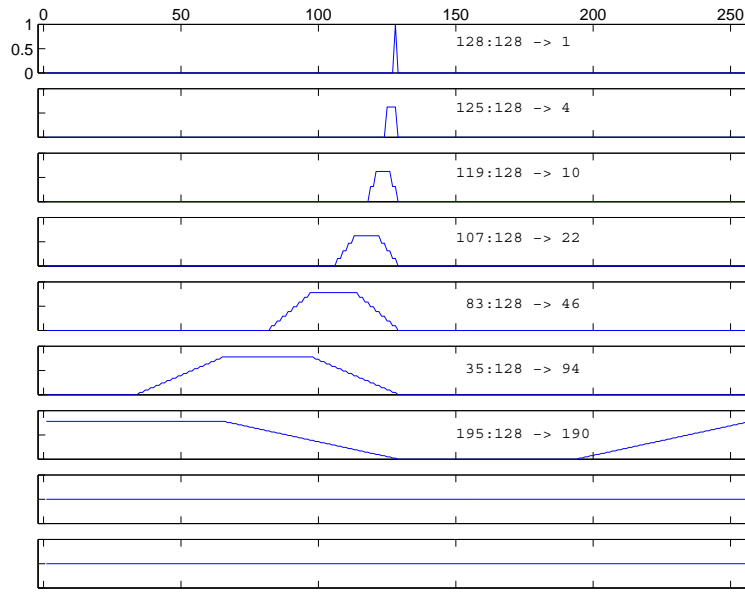


Figura 3.10: Aproximaciones de S calculadas con $F^{(4)}$

La idea utilizada en este método de centrado, es lograr un factor de desplazamiento D para cada filtro, basándonos en su largo efectivo. Luego este factor será utilizado para calcular la cantidad de posiciones en que debe ser desplazada la señal (hacia la derecha y en forma circular). Repitiendo este procedimiento para cada nivel.

Así tenemos:

$$PosicionesADesplazar_{j+1} = D_F \cdot largo(F \uparrow 2^j) \quad (3.11)$$

Es decir, la cantidad de posiciones a desplazar en el nivel $j + 1$ depende del factor de desplazamiento del filtro utilizado (D_F) y de su largo en el nivel j , que es el que se utiliza para generar el nivel $j + 1$. Notar que el factor D_F no varía dependiendo del nivel, y que será el mismo para cualquier filtro F' de la misma longitud que F .

Despejando de 3.11, tenemos que:

$$D_F = \frac{PosicionesADesplazar_{j+1}}{largo(F \uparrow 2^j)} \quad (3.12)$$

Si suponemos que la señal se encuentra centrada en el nivel j , y que al pasar al nivel $j + 1$ se dilata hacia la izquierda en k posiciones, para centrarla basta con desplazarla hacia la derecha $k/2$ posiciones. Como la dilatación entre niveles es igual al largo efectivo del filtro en el nivel origen menos uno (ecuación 3.10), tenemos que:

$$PosicionesADesplazar_{j+1} = \frac{largoEfectivo(F \uparrow 2^j) - 1}{2} \quad (3.13)$$

Combinando 3.12 y 3.13 obtenemos:

$$D_F = \frac{largoEfectivo(F \uparrow 2^j) - 1}{2 \cdot largo(F \uparrow 2^j)} \quad (3.14)$$

Este resultado es similar para todos los niveles, dado que el factor de desplazamiento depende del filtro original y no del nivel en cuestión (ver tabla 3.3).

De esta forma, podemos utilizar el nivel 0 y obtener finalmente:

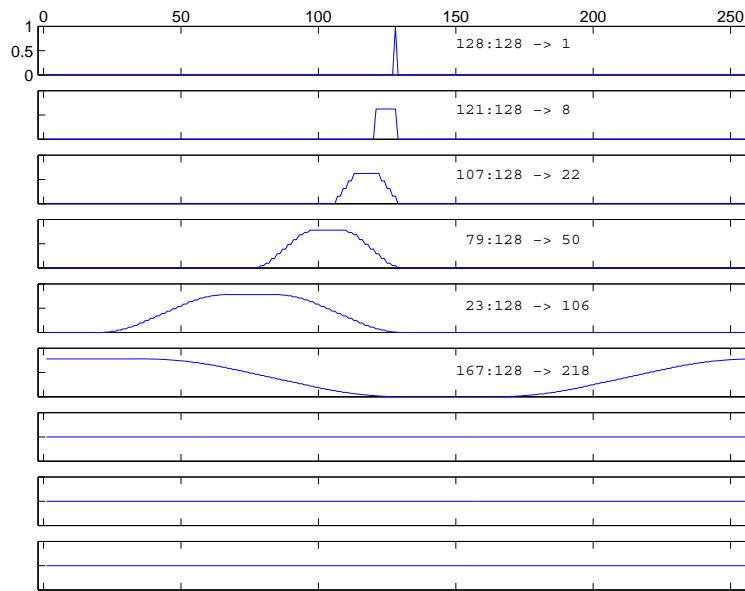


Figura 3.11: Aproximaciones de S calculadas con $F^{(8)}$

	$F^{(2)}$	$F^{(4)}$	$F^{(8)}$
$j = 0$	$\frac{2-1}{2 \cdot 2} = \frac{1}{4}$	$\frac{4-1}{2 \cdot 4} = \frac{3}{8}$	$\frac{8-1}{2 \cdot 8} = \frac{7}{16}$
$j = 1$	$\frac{3-1}{2 \cdot 4} = \frac{1}{4}$	$\frac{7-1}{2 \cdot 8} = \frac{3}{8}$	$\frac{15-1}{2 \cdot 16} = \frac{7}{16}$
$j = 2$	$\frac{5-1}{2 \cdot 8} = \frac{1}{4}$	$\frac{13-1}{2 \cdot 16} = \frac{3}{8}$	$\frac{29-1}{2 \cdot 32} = \frac{7}{16}$
...

Tabla 3.3: Valores de D_F para $F^{(2)}$, $F^{(4)}$ y $F^{(8)}$ (calculados utilizando 3.14). Se observa que se obtiene el mismo valor sin importar en que nivel se lo calcule.

$$D_F = \frac{\text{largo}(F) - 1}{2 \cdot \text{largo}(F)}$$

Si bien este método es mucho mejor que el anterior, todavía sigue teniendo una falencia muy importante. Siguen sin diferenciarse las componentes nulas que se agregan al filtro en cada nivel de la transformada. Este parece un tema menor a primera vista, pero como el agregado de ceros es exponencial con respecto al nivel de la transformada, el filtro crece exponencialmente, haciendo los cálculos realmente lentos.

El siguiente método ataca este problema.

Fórmula eficiente para el cálculo de $S \circledast (F \uparrow 2^j)$

En este caso se utiliza la definición de *convolución circular*:

$$\begin{aligned}
 &S[n] \text{ con } 0 \leq n < N \\
 &F[m] \text{ con } 0 \leq m < M \\
 &(S \circledast F)[n] = \sum_{p=0}^{M-1} S[(n-p) \bmod N] \cdot F[p]
 \end{aligned}$$

Si en lugar de F ahora convolucionamos con $F \uparrow 2^j$, tenemos que evitar multiplicar por los ceros que se agregaron entre las componentes de F . O lo que es lo mismo, sólo deberíamos multiplicar por las componentes de $F \uparrow 2^j$ que se encuentran en las posiciones que son múltiplos de 2^j .

$$S \circledast (F \uparrow 2^j)[n] = \sum_{p \text{ múltiplo de } 2^j} S[(n - p) \bmod N] \cdot (F \uparrow 2^j)[p]$$

Por otro lado, si p es múltiplo de 2^j se cumple la siguiente igualdad:

$$(F \uparrow 2^j)[p] = F\left[\frac{p}{2^j}\right] \quad \text{siendo } p = 2^j k$$

Y se puede expresar la convolución anterior como:

$$S \circledast (F \uparrow 2^j)[n] = \sum_{p=2^j k} S[(n - p) \bmod N] \cdot F[k]$$

Finalmente, haciendo un cambio de variables llegamos a la fórmula definitiva:

$$S \circledast (F \uparrow 2^j)[n] = \sum_{p=0}^{M-1} S[(n - 2^j p) \bmod N] \cdot F[p] \quad (3.15)$$

La señal producida por esta convolución no está centrada, sino que se encuentra desplazada en forma circular hacia la derecha. La magnitud del desplazamiento es proporcional al largo efectivo del filtro sobremuestreado, más exactamente:

$$\text{posicionesDesplazadasHaciaLaDerecha} = \text{largoEfectivo}(F \uparrow 2^j) - 1$$

Si utilizamos la relación entre el largo efectivo y el largo de un filtro de la ecuación 3.9 tenemos:

$$\begin{aligned} \text{posicionesDesplazadasHaciaLaDerecha} &= 2^j(\text{largo}(F) - 1) + 1 - 1 \\ &= 2^j(M - 1) \end{aligned}$$

Por lo tanto, para centrar la señal obtenida simplemente debemos desplazarla hacia la izquierda (en forma circular) en la mitad de esta magnitud, es decir:

$$\text{posicionesADesplazar} = \frac{2^j(M - 1)}{2} = 2^{(j-1)}(M - 1) \quad (3.16)$$

3.3.5. Transformada “à trous” en dos dimensiones (imágenes)

Para extender la definición del algoritmo de una a dos dimensiones se procede de forma vectorial, aplicando la versión unidimensional en forma separada a las filas y/o a las columnas de la imagen, considerando en cada caso a cada una de éstas como una señal unidimensional.

En la figura 3.13 se muestra un paso en el proceso de análisis de la transformada.

Partiendo de la imagen I_0 se obtienen 4 imágenes, una versión suavizada de I_0 que se genera al aplicar el filtro pasa bajos h sobre ella (primero sobre sus filas, y luego sobre sus columnas) y a la que llamaremos I_1 , y 3 imágenes de detalles DV_1 , DH_1 e DD_1 (detalles verticales, horizontales y diagonales respectivamente), que se generan aplicando el filtro pasa altos g (respectivamente sobre las filas, sobre las columnas, y sobre las filas y columnas).

En la figura las cajas representan una convolución sobre filas ($x \text{ fils}$) o sobre columnas ($x \text{ cols}$), utilizando el filtro que se indica en su interior.

Como puede verse, las imágenes DD_1 e I_1 pueden obtenerse de dos maneras distintas, primero calculando sobre filas y luego sobre columnas o viceversa. Ambas opciones generan el mismo resultado, por lo tanto, la sección punteada muestra las convoluciones que no se realizan al calcular la transformada.

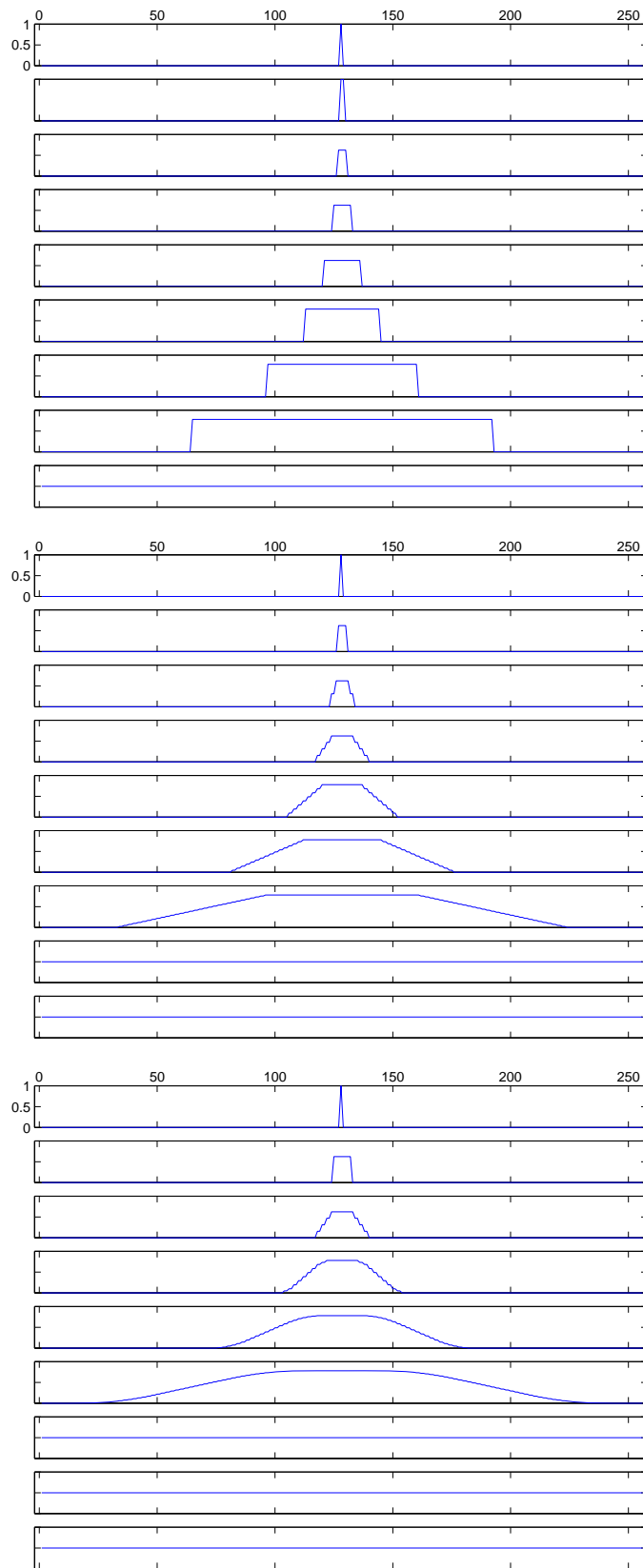


Figura 3.12: Transformadas luego del centrado. Utilizando $F^{(2)}$ arriba, $F^{(4)}$ en el medio y $F^{(8)}$ abajo. Todos los métodos generan la misma salida.

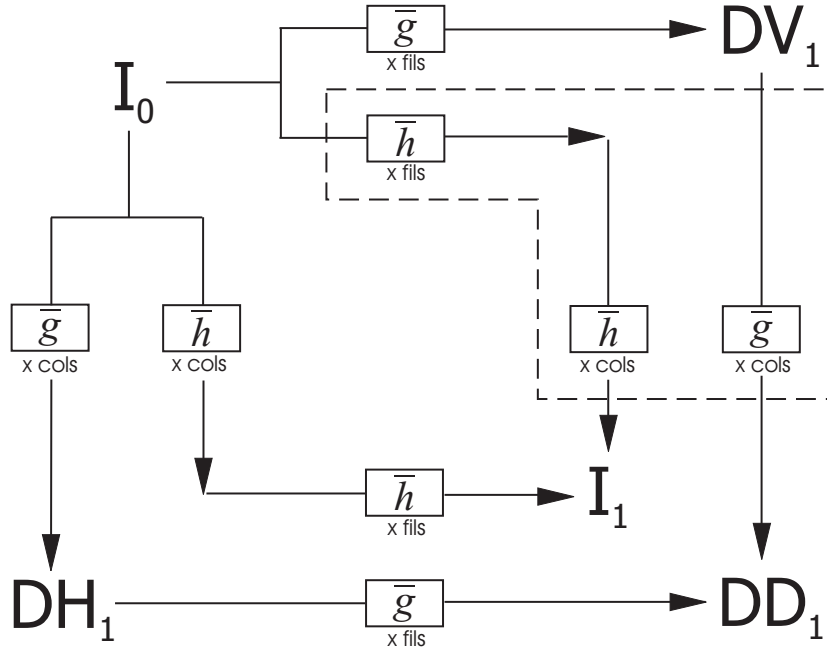


Figura 3.13: Un paso de la transformada “à trous” en 2D

Esta transformada es redundante: a partir de 1 imagen de $N \times N$ se obtienen 4 imágenes de $N \times N$ en cada escala.

En base a las 4 imágenes generadas por el proceso anterior se puede obtener nuevamente la imagen original I_0 . El proceso se muestra en la figura 3.14. En este caso se emplean los filtros duales \tilde{h} y \tilde{g} .

Sobre la imagen de aproximación I_1 se aplica el filtro \tilde{h} dos veces, primero por filas, y luego por columnas. Sobre la imagen de detalles horizontales DH_1 se aplica el filtro \tilde{g} sólo por filas, y sobre la de detalles verticales DV_1 se aplica el mismo filtro sólo por columnas. Sobre la imagen de detalles diagonales DD_1 se aplica el filtro \tilde{g} dos veces, primero por filas, y luego por columnas.

Una vez que se aplicaron los filtros se ponderan los resultados obtenidos por las constantes indicadas en la figura y por último se suman todas las imágenes ponderadas para obtener nuevamente la imagen original I_0 .

Las constantes que se utilizan para ponderar las imágenes luego de haberles aplicado los filtros son $1/4$, $1/2$, $1/2$ y $-1/4$, para la aproximación, los detalles horizontales, los verticales y los diagonales respectivamente. Notar que sobre la imagen de aproximación y sobre la de detalles diagonales se emplean constantes con módulo $1/4$, mientras que para las de detalles horizontales y verticales se emplean constantes con módulo $1/2$. Esto se debe a que en las primeras se aplican los filtros dos veces, y en las segundas sólo una vez.

Otra cosa a notar es que la constante de la imagen de detalles diagonales es negativa, ésto se debe a que la información que ésta contiene, también se encuentra en las imágenes de detalles horizontales y verticales, y por lo tanto se debe quitar al sumarlas.

Para obtener la transformada completa se efectúa el proceso descrito en la figura 3.13 $\log_2(N)$ veces (siendo N el largo del lado de la imagen), usando en cada iteración la imagen de aproximación obtenida en el paso anterior.

Así se obtiene:

$$\left[\{DH_j, DV_j, DD_j\}_{1 \leq j \leq J}, I_J \right] \quad \text{con } J = \log_2(N) \quad (3.17)$$

En la figura 3.15 se muestra la transformada completa de una imagen artificial, y en la figura 3.16 la de una imagen natural.

La primera fila sólo contiene la imagen original, las siguientes filas son los distintos niveles de la

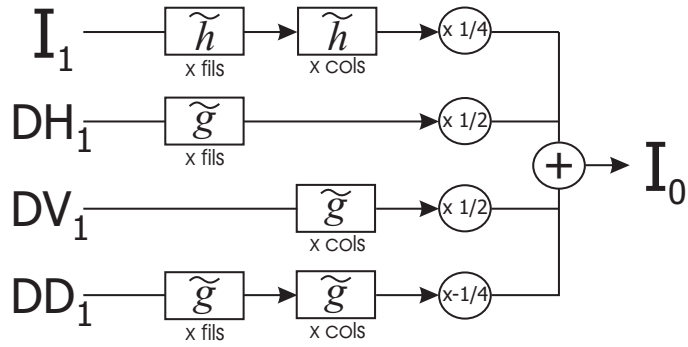


Figura 3.14: Un paso de la transformada inversa “à trous” en 2D

transformada. En cada fila de la transformada tenemos cuatro columnas. La primera son las *aproximaciones*, la segunda los *detalles horizontales*, la tercera los *detalles verticales*, y por último, la cuarta los *detalles diagonales*.

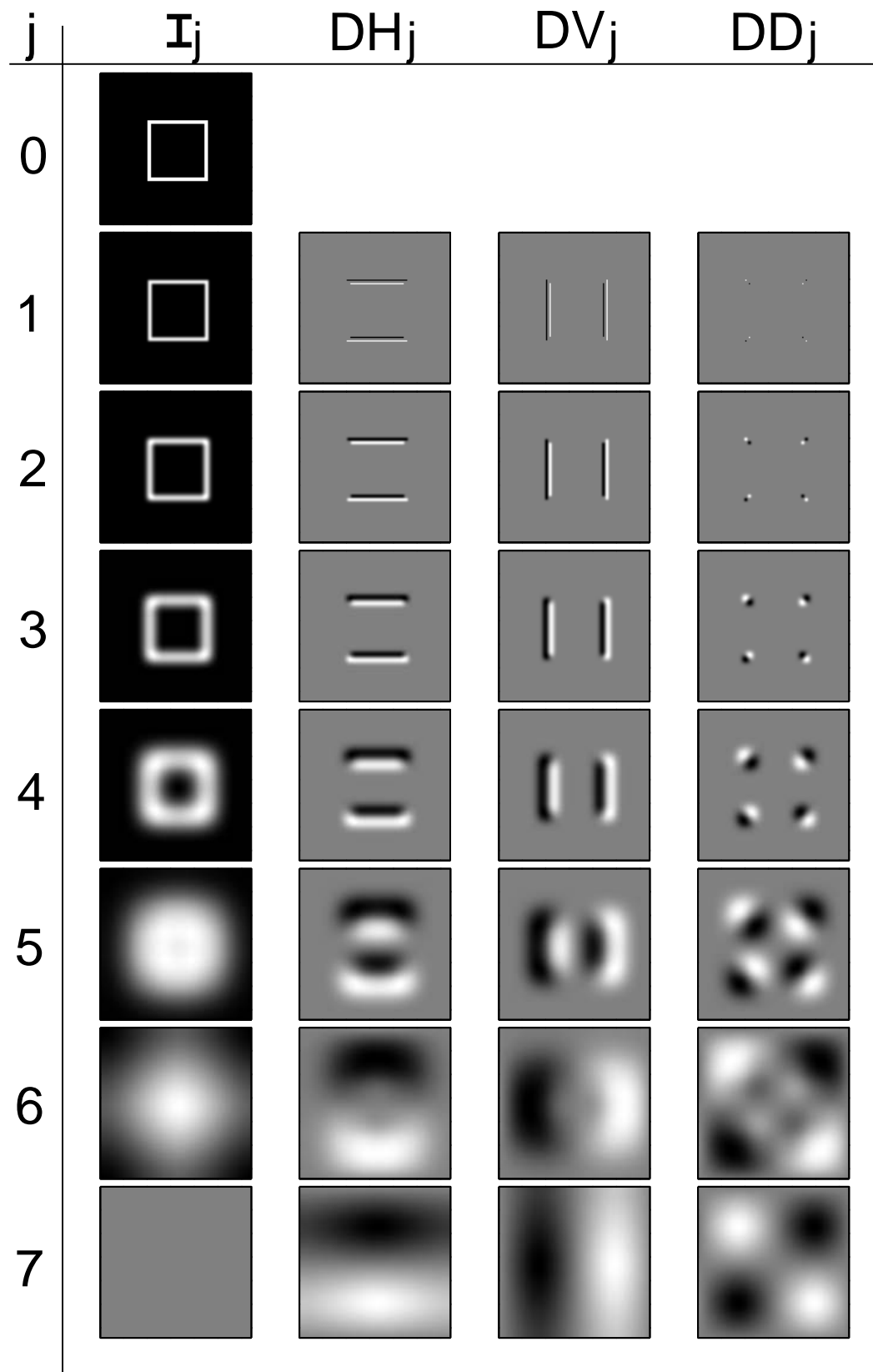


Figura 3.15: Transformada completa 2D (Im. Artificial)

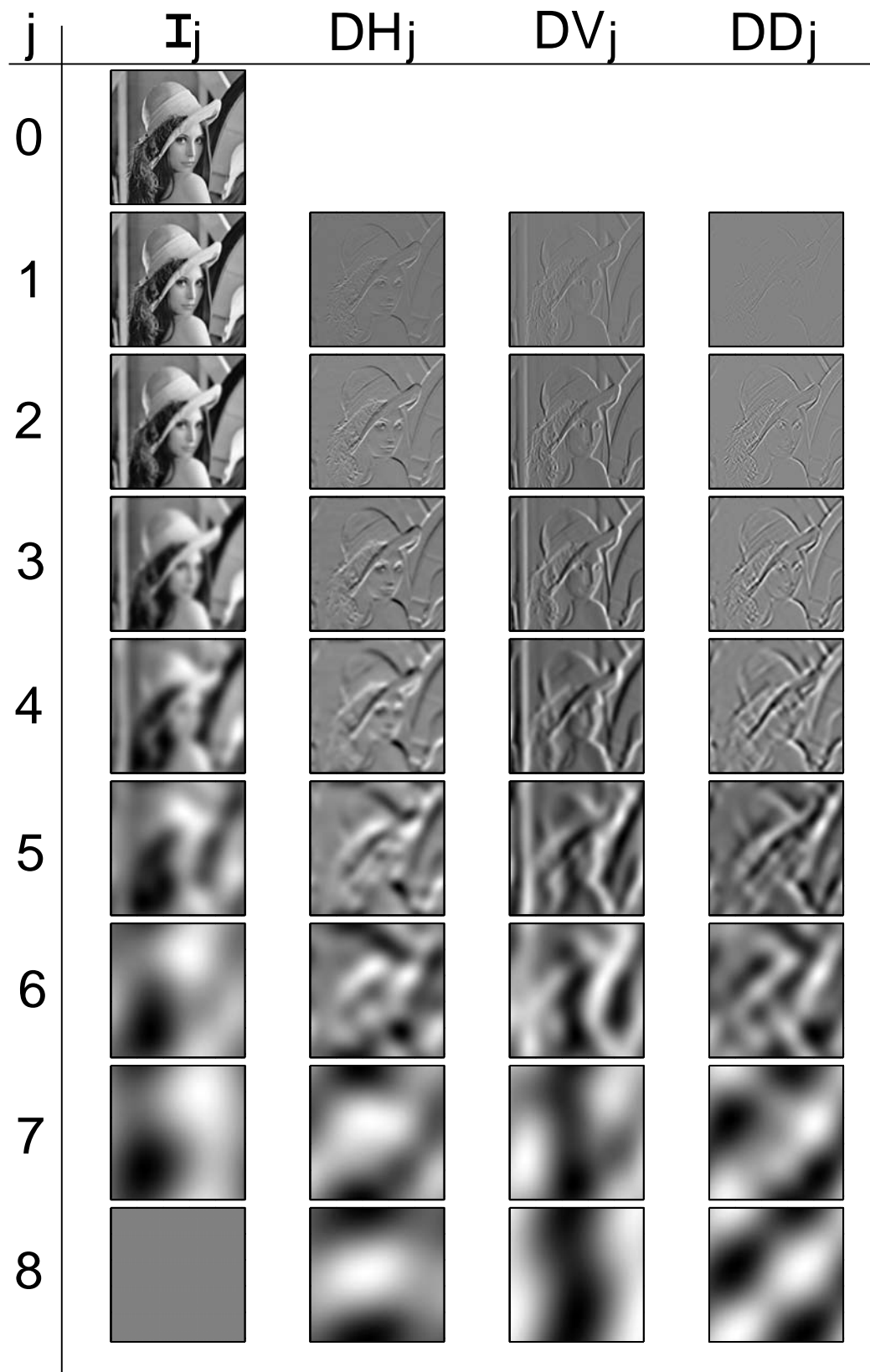


Figura 3.16: Transformada completa 2D (Im. Natural)

Capítulo 4

Detección de los módulos máximos

4.1. Introducción

En este capítulo explicamos cómo se realiza el cálculo de los módulos máximos (MM) de la transformada wavelet “à trous”.

Se busca calcular los máximos locales (en valor absoluto) de los coeficientes de esta transformada a distintas escalas.

Dada una señal de entrada con valores reales, se calculan los coeficientes de la transformada wavelet a varias escalas o resoluciones. Para cada una de las escalas se obtiene una señal de salida sólo con unos y ceros (o verdadero y falso) en donde se indica para cada coeficiente si es un módulo máximo o no.

La razón para concentrar nuestro análisis en los coeficientes wavelets cuyos módulos son máximos locales es la siguiente: los módulos de los coeficientes wavelets se intensifican en las porciones de la señal en donde se encuentran las altas frecuencias (bordes, picos, discontinuidades, etc), ésto es así porque las wavelets funcionan como un filtro pasa altos. Por otra parte, intuitivamente la mayor cantidad de información que contiene una señal ¹ se encuentra justamente en sus transiciones bruscas (altas frecuencias) dado que éstas nos muestran sectores en donde la información cambia y no puede estimarse con la información circundante. De esta forma, al seleccionar los coeficientes de máximo cambio (los que son máximos locales) estamos reteniendo sólo las porciones de la señal con mayor información.

4.2. Módulos máximos en una dimensión

En la figura 4.1 se muestra el proceso simplificado de detección en una dimensión: se toma la función, se calcula su módulo y se identifican los máximos locales del mismo.

Como nuestro análisis se realiza sobre los coeficientes wavelets de la función original, en el caso continuo deberíamos analizar la transformada wavelet continua en cada escala s del plano escala/tiempo. La figura 4.2 muestra la misma transformada que la de la figura 3.2, pero en este caso se indica con blanco las posiciones donde se encuentran localizados los módulos máximos.

Sobre la transformada discreta este análisis es diádico, porque la misma no posee escalas intermedias entre cada nivel. Sin embargo, al detectar los MM sobre la transformada continua vemos más claramente un fenómeno interesante, el de la formación de líneas que atraviesan las escalas en forma vertical a las que llamaremos *líneas de MM*.

Estas líneas de MM se producen por la aparición de una discontinuidad en la señal original y se propagan a través de las escalas. Dependiendo del tipo de discontinuidad, los coeficientes wavelet sobre estas líneas tienen propiedades diferentes, y estudiándolas se pueden clasificar las discontinuidades en diferentes tipos, permitiendo de esta forma un análisis de bordes. Esta evolución de los coeficientes wavelet fue estudiada por Stéphane Mallat: [16, 18, 17, 15], entre otros.

¹Al hablar de información de la señal lo estamos haciendo en el sentido de la teoría de la información: la información es mayor si no es predecible.

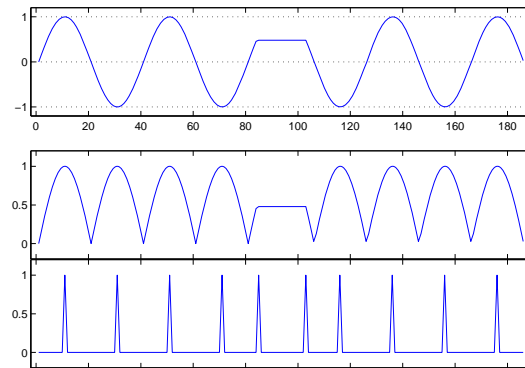


Figura 4.1: Módulos máximos de una función unidimensional de variable continua. Arriba se encuentra la función original, $f(x)$. En el medio sus módulos, $|f(x)|$. Finalmente, abajo se se indican las posiciones de los máximos locales de $|f(x)|$.

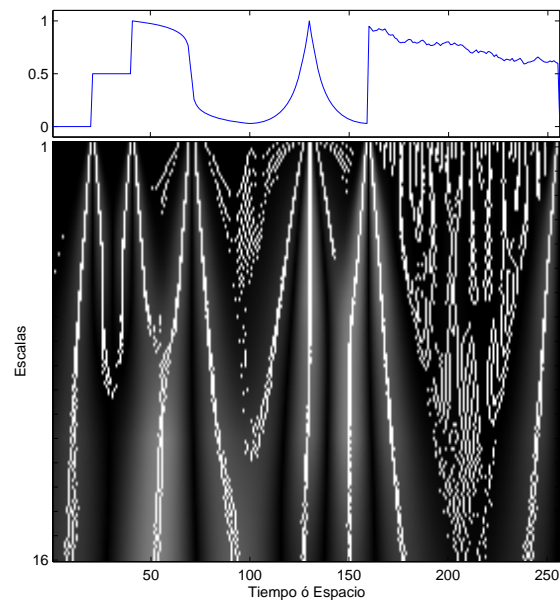


Figura 4.2: Módulos máximos sobre una transformada wavelet continua

Cabe aclarar que si dos o más discontinuidades se encuentran lo suficientemente cerca en la señal original, las líneas de MM que producen interferirán unas con otras al aumentar el factor de escala y, por lo general, tenderán a convertirse en una sola línea de MM. Un ejemplo de este comportamiento se puede observar en la figura 4.2, en las discontinuidades alrededor de la abscisa 50 y en el sector derecho, en la zona de altas frecuencias.

En el caso discreto se define que un valor $f[n]$ es de módulo máximo si:

$$\begin{aligned} &|f[n]| \geq |f[n-1]| \\ &\quad y \\ &|f[n]| \geq |f[n+1]| \\ &\quad y \\ &(|f[n]| > |f[n-1]| \text{ o } |f[n]| > |f[n+1]|) \end{aligned}$$

Es decir, si $|f[n]|$ es mayor o igual al módulo de sus dos vecinos y al menos mayor estricto que alguno de ellos. La restricción de mayor estricto se establece para evitar detectar como módulos máximos a todos los valores de una porción constante de la señal.

Para aplicar este proceso de detección de MM a la transformada wavelet discreta, se utiliza la transformada “à trous” porque es invariante frente a los desplazamientos y nos brinda una mayor cantidad de información de la señal en cada banda de frecuencia que la transformada tradicional (con decimación).

Dada la señal discreta S , de N componentes, su transformada “à trous” se nota $\{d_1, d_2, \dots, d_J, a_J\}$, donde $J = \log_2(N)$. Al detectar los valores de módulo máximo sobre ella, obtenemos un conjunto de señales:

$$\{P_j\}_{1 \leq j \leq J} \quad \text{de } N \text{ componentes cada una} \quad (4.1)$$

que indican las posiciones de los módulos máximos detectados.

Estas señales están compuestas sólo de ceros y unos. Un uno indica que en esa posición de la transformada se encuentra un MM:

$$P_j[n] = \begin{cases} 1 & \text{si } d_j[n] \text{ es MM} \\ 0 & \text{si no} \end{cases} \quad (4.2)$$

En la figura 4.3 se muestra en la columna de la izquierda los detalles de la transformada, y en la columna de la derecha las posiciones donde se encuentran los MM como líneas verticales. Cada línea está escalada según el valor de la transformada en esa posición.

Antes de pasar al caso bidimensional, mencionaremos una pequeña salvedad que presentan los módulos máximos. Debido a que éstos se corresponden con los máximos locales del módulo de una señal f , si f presenta un mínimo local con valor negativo en $f[n]$, este valor es detectado como un módulo máximo. Sin embargo, si f presenta un mínimo local en $f[n]$, pero su valor es positivo, este valor NO es detectado como un módulo máximo. La figura 4.4 muestra un ejemplo que clarifica la explicación anterior. La misma situación se presenta entre los máximos locales de f con valor positivo, que sí son detectados como MM, y los máximos locales de f con valor negativo, que NO son detectados como MM.

En este trabajo se detectan los módulos máximos de la transformada “à trous” con el objetivo de identificar los bordes de la señal original. Como se explicará en el capítulo 5, dado que esta transformada se comporta como un operador de derivación, no nos interesa detectar las salvedades mencionadas, es decir, mínimos locales positivos y máximos locales negativos, ya que no se corresponden con los bordes de la señal original.

4.3. Módulos máximos en dos dimensiones (imágenes)

La idea de la detección de MM en 1 dimensión es seleccionar los valores que se encuentran en la posición de mayor variación de la señal. En 2 dimensiones la idea es la misma: un valor es MM si en esa posición la imagen tiene la mayor variación de su entorno.

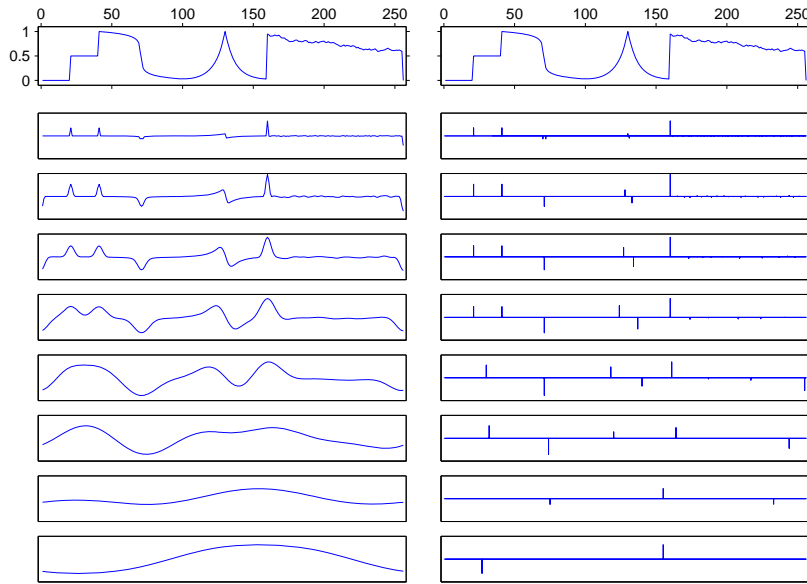
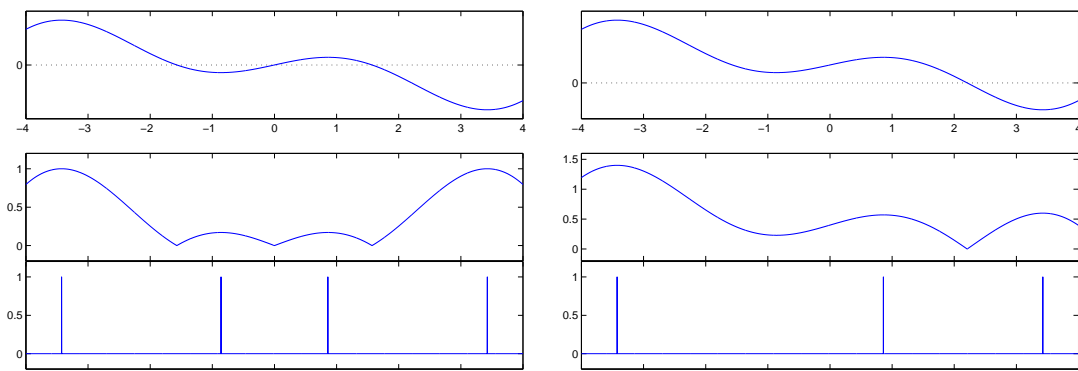


Figura 4.3: Izquierda: Transformada “à trous”. Derecha: Módulos máximos escalados y con su signo.



(a) Los mínimos locales negativos son detectados como módulos máximos. (b) Los mínimos locales positivos NO son detectados como módulos máximos.

Figura 4.4: Diferencia entre mínimos locales negativos y positivos en la detección de módulos máximos

De la misma forma que la detección de MM en 1 dimensión se realiza comparando cada valor con sus dos vecinos, en 2 dimensiones también se compara cada valor con sus vecinos. La diferencia es que en este caso cada píxel tiene 8 vecinos a su alrededor, por lo que se presenta el problema de la elección de los vecinos contra los que comparar.

En el presente trabajo, nos interesa calcular los MM de la transformada “à trous”, por lo que el proceso de detección se realiza sobre las imágenes generadas por el algoritmo descrito en la sección 3.3.5. Este algoritmo genera tres imágenes de detalles DH_j , DV_j y DD_j , por cada nivel j de la transformada.

Para calcular los MM en 2 dimensiones es conveniente pensar a la transformada como una representación vectorial, teniendo un vector por cada píxel de la imagen original y por cada escala de la transformada wavelet.

Para cada posición $[n, m]$ y cada escala j de la transformada, se tiene un coeficiente de detalle horizontal $DH_j[n, m]$, y un coeficiente de detalle vertical $DV_j[n, m]$. Con esta información se obtiene un vector de detalle para esa posición².

La figura 4.5 muestra la primera escala de la transformada para una imagen sencilla con 4 líneas, una horizontal, otra vertical y dos diagonales.

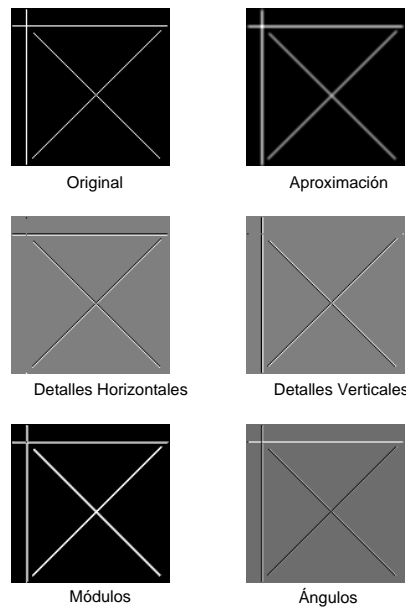


Figura 4.5: Primeras 4 figuras: figura original y un paso de la transformada. Últimas 2 figuras: detalles en coordenadas polares. El negro siempre es el valor más bajo y el blanco el más alto. En las imágenes con fondo gris el negro representa valores negativos, el blanco positivos y el gris al cero.

En la fila del medio se encuentran los detalles horizontales y verticales.

Puede verse que la línea horizontal de la imagen original sólo genera detalles horizontales.

Asimismo, la vertical sólo genera detalles verticales.

Por otro lado, las diagonales generan tanto detalles verticales como horizontales.

Esto muestra que en una representación vectorial de la transformada, los detalles horizontales corresponden al eje X , mientras que los verticales al eje Y (ver figura 4.6³). Esto es coherente con el hecho de que las wavelets se comportan como filtros pasa altos, obteniendo al aplicarlas una aproximación de las dos componentes del gradiente de la imagen [9].

Los vectores de la transformada pueden representarse mediante sus dos componentes canónicas X e

²Se hicieron varias pruebas para tratar de incluir los detalles diagonales en la detección en dos dimensiones, pero en ninguna de ellas se obtuvieron resultados que justifiquen la obtención de estos detalles. El principal problema es que cada matriz DD_j se obtiene con un filtrado adicional sobre DH_j o DV_j . Este filtrado adicional genera un desplazamiento en los detalles diagonales haciendo que éstos no coincidan exactamente con la intersección de los horizontales y verticales.

³Para simplificar la notación en las figuras, en algunos lugares se indexan las matrices utilizando la letra p , que hace referencia a la posición del píxel, es decir $[n, m]$

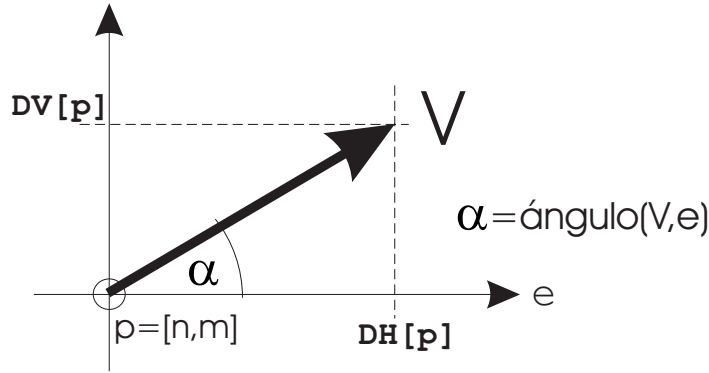


Figura 4.6: Vector de borde generado a partir de los detalles horizontales y verticales del píxel p .

Y , o mediante coordenadas polares (módulo y ángulo). Esta última representación presenta varias ventajas:

- Permite identificar en forma eficiente los MM, porque éstos se encuentran en los máximos locales del módulo.
- Permite obtener una imagen de bordes clara y definida (notar que en la figura 3.16 no están claramente definidos los bordes de Lena).
- Permite identificar los bordes más significativos de la imagen.
- Tiene una relación directa con un método clásico de detección de bordes (Canny).

Trabajaremos con las matrices M_j de módulos, y A_j de ángulos⁴, definiéndolas como:

$$M_j[n, m] = \sqrt{(DH_j[n, m])^2 + (DV_j[n, m])^2} \quad (4.3)$$

$$A_j[n, m] = \arctan\left(\frac{DV_j[n, m]}{DH_j[n, m]}\right) \quad (4.4)$$

para obtener la matriz de posiciones de los módulos máximos:

$$P_j[n, m] = \begin{cases} 1 & \text{si } M_j[n, m] \text{ es máximo} \\ 0 & \text{si no} \end{cases} \quad (4.5)$$

Es importante tener en cuenta que si un píxel en la posición $[n, m]$ de la escala j , pertenece a un borde se verifica que:

- $M_j[n, m]$, el módulo, corresponde a la magnitud o intensidad del borde en ese píxel.
- $A_j[n, m]$, el ángulo, corresponde a la dirección que toma el borde al pasar por ese píxel.

Para ubicar las posiciones MM se busca al píxel con mayor módulo entre sus vecinos en la dirección de mayor variación. Esta dirección es la que atraviesa al borde, es decir la perpendicular a $A_j[n, m]$. Esto se nota muy claramente cuando los bordes son horizontales, verticales o forman un ángulo de 45 grados, como los de la figura 4.7.

Los problemas aparecen cuando el ángulo en un determinado píxel no pertenece exactamente a una de estas categorías, como por ejemplo en la figura 4.8.

⁴Las matrices de ángulos A_j se calculan utilizando la función arco tangente, pero modificada de forma tal que su imagen sea el rango $[0, 2\pi]$, y que no se indetermine cuando $DH_j[n, m]$ vale cero.

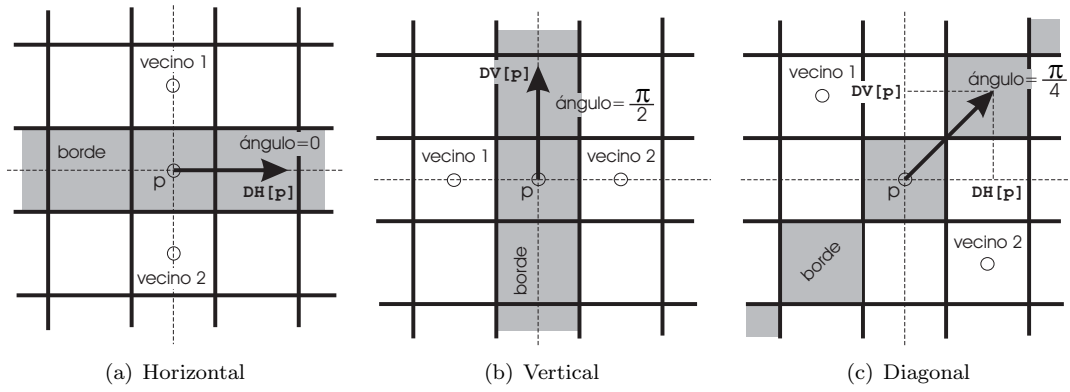


Figura 4.7: Bordes donde la elección de los vecinos a revisar es sencilla

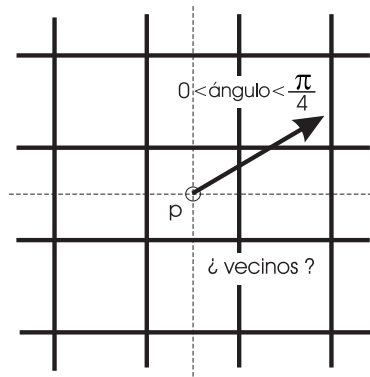


Figura 4.8: Borde con un ángulo entre la horizontal y la diagonal.

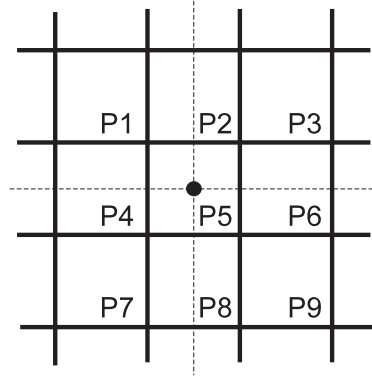


Figura 4.9: Vecinos alrededor del píxel central, identificado como $P5$

Ángulo ($\alpha = A[P5]$)	Píxel del Primer Vecino	Píxel del Segundo Vecino
0	P2	P8
$1/4\pi$	P1	P9
$2/4\pi$	P4	P6
$3/4\pi$	P7	P3
π	P8	P2
$5/4\pi$	P9	P1
$6/4\pi$	P6	P4
$7/4\pi$	P3	P7
2π	P2	P8

Tabla 4.1: Vecinos a utilizar cuando el ángulo del borde pertenece a una de las categorías simples. Recordar que siempre se utilizan los vecinos en la dirección perpendicular al ángulo del borde.

4.4. Combinación lineal de los píxeles vecinos

Cuando el ángulo del borde no pertenece a una de las categorías simples (horizontal, vertical o diagonales) no es fácil obtener los vecinos para comparar.

Una forma de obtenerlos es realizando una combinación lineal entre los vecinos más cercanos.

Para simplificar la explicación del método usado para obtener estas combinaciones lineales, le pondremos nombres a los píxeles en un entorno de 3x3 (ver figura 4.9). Estos nombres van de $P1$ a $P9$, y nos ayudarán a identificar fácilmente a los píxeles. El píxel por el que pasa el borde es el central, identificado como $P5$.

Como este razonamiento es el mismo sin importar la escala en la que estemos trabajando, se omiten los subíndices j .

Para hacer más clara la lectura, llamaremos $\alpha = A[P5]$.

La tabla 4.1 nos muestra los vecinos a utilizar para las categorías simples dependiendo del ángulo del borde.

Ejemplo:

Si el ángulo está entre 0 y $1/4\pi$, el primer vecino se tiene que obtener combinando linealmente el valor del píxel $P2$ con el valor del píxel $P1$:

$$\begin{aligned}\alpha = 0 &\implies \text{Vecino1} = M[P2] \\ \alpha = 1/4\pi &\implies \text{Vecino1} = M[P1]\end{aligned}$$

$$0 < \alpha < 1/4\pi \implies Vecino1 = CombinacionLineal(M[P2] , M[P1])$$

Para obtener esta combinación lineal calculamos:

$$Vecino1 = a \cdot \alpha + b \quad (4.6)$$

donde a y b son la pendiente y el desplazamiento de la recta que representa la combinación lineal.

Esta recta tiene que pasar por los puntos:

$$(0, M[P2]) \quad y \quad (1/4\pi, M[P1])$$

del plano (ángulo, módulo).

Por lo tanto, utilizando la ecuación de la recta y teniendo como valores conocidos dos puntos por los que ésta pasa obtenemos que:

$$Vecino1 = \frac{4(M[P1] - M[P2])}{\pi} \cdot \alpha + M[P2] \quad si \quad 0 \leq \alpha < 1/4\pi$$

Análogamente se puede obtener el $Vecino2$ y ambos vecinos para el resto de los ángulos intermedios.

Expresión general:

Dado un ángulo α , tal que:

$$0 \leq \alpha < 2\pi \quad (4.7)$$

consultando la tabla 4.1, se lo puede acotar entre dos ángulos de categorías simples consecutivas:

$$\alpha_I \leq \alpha < \alpha_F \quad con \quad \alpha_I \text{ y } \alpha_F \text{ consecutivos} \quad (4.8)$$

Estos ángulos de categorías simples tienen asociados los valores de sus vecinos:

$$\begin{aligned} \alpha = \alpha_I &\implies Vecino = M[P_I] \\ \alpha = \alpha_F &\implies Vecino = M[P_F] \end{aligned}$$

Por lo tanto la recta que representa la combinación lineal de estos valores debe pasar por los siguientes puntos del plano (ángulo, módulo):

$$(\alpha_I, M[P_I]) \quad y \quad (\alpha_F, M[P_F])$$

Utilizando nuevamente la ecuación de la recta que pasa por dos puntos obtenemos:

$$Vecino = a \cdot \alpha + b$$

donde

$$\begin{aligned} a &= \frac{M[P_I] - M[P_F]}{\alpha_I - \alpha_F} \\ b &= \frac{\alpha_I M[P_F] - \alpha_F M[P_I]}{\alpha_I - \alpha_F} \end{aligned}$$

En la figura 4.10 se muestra este proceso de detección de MM sobre los tres primeros niveles de la transformada de Lena. En la columna de la izquierda se muestra la imagen M de módulos, en la central se muestran en negro las posiciones donde se encuentran los MM. Por último en la columna de la derecha se muestran los coeficientes de la imagen M sólo en las posiciones donde están los MM.

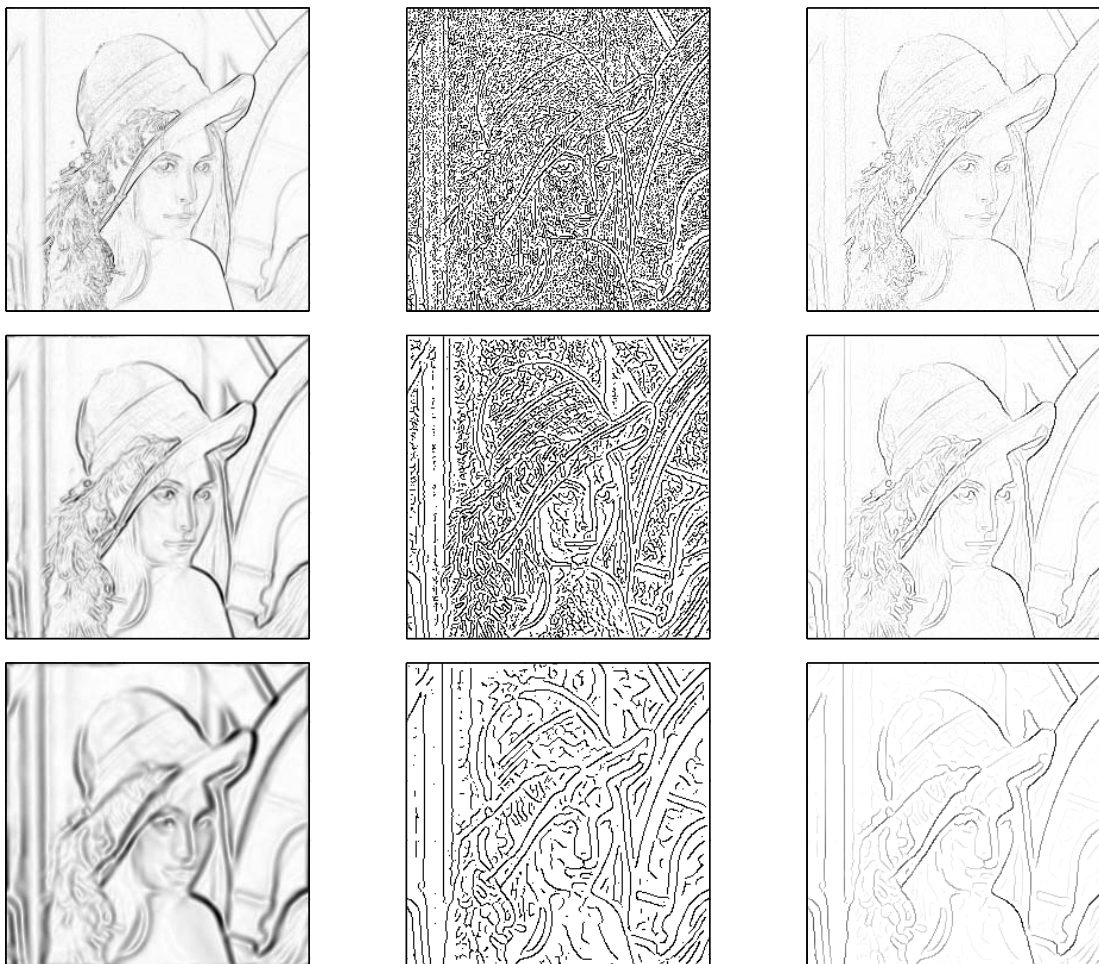


Figura 4.10: Detección de MM interpolando vecinos sobre los 3 primeros niveles de la transformada de Lena. A la izquierda se muestran los módulos de la transformada. En el centro las posiciones de los MM. A la derecha los módulos de la transformada en las posiciones de los MM.

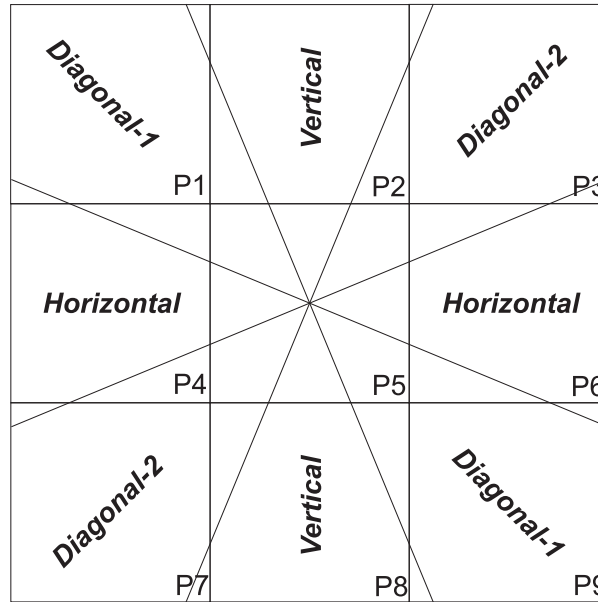


Figura 4.11: División sencilla de los vectores de dirección en cuatro categorías iguales

Ángulo (α)		Categoría	Vecino 1	Vecino 2
Desde (incl.)	Hasta (excl.)			
0	$1/8\pi$	Horizontal	P2	P8
$1/8\pi$	$3/8\pi$	Diagonal-2	P1	P9
$3/8\pi$	$5/8\pi$	Vertical	P4	P6
$5/8\pi$	$7/8\pi$	Diagonal-1	P3	P7
$7/8\pi$	$9/8\pi$	Horizontal	P2	P8
$9/8\pi$	$11/8\pi$	Diagonal-2	P1	P9
$11/8\pi$	$13/8\pi$	Vertical	P4	P6
$13/8\pi$	$15/8\pi$	Diagonal-1	P3	P7
$15/8\pi$	2π	Horizontal	P2	P8

Tabla 4.2: Categorías según α y sus vecinos asociados

4.5. Obtención rápida de los módulos máximos

El problema del método anterior es que insume demasiado tiempo de procesamiento en el cálculo de los dos promedios ponderados necesarios para determinar los vecinos.

Una solución a este problema es definir las categorías como un rango de ángulos, y asociar a cada una de estas categorías un par de vecinos. De esta forma, identificando a que categoría pertenece el ángulo del borde, automáticamente se identifican los vecinos para comparar.

En la figura 4.11 se muestran las categorías propuestas.

En este caso se divide el espacio de vecinos en sólo cuatro categorías de igual tamaño: *Horizontal*, *Vertical*, *Diagonal-1* y *Diagonal-2*.

En la tabla 4.2 se muestran las categorías en las que se divide la vecindad de píxeles. Para cada una se muestra los ángulos que la delimitan y los vecinos que tiene asociados.

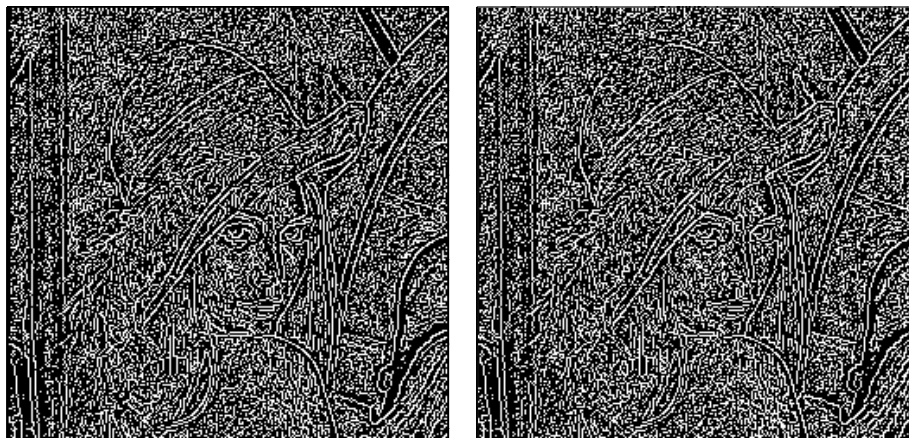
Los resultados obtenidos mediante este método son muy similares (casi indistinguibles a simple vista) a los obtenidos con el método de combinaciones lineales.

En realidad, al comparar dos detecciones (una realizada con el método de combinaciones lineales y la otra con el método simplificado) se percibe que la detección por combinaciones lineales genera un resultado un poco más limpio que el simplificado. En la imagen del método simplificado aparecen

algunos puntos dispersos que en el primero no son detectados como MM.

En la figura 4.12 se muestra la comparación de los 3 primeros niveles de MM calculados sobre Lena. Puede notarse que prácticamente no existen diferencias a simple vista.

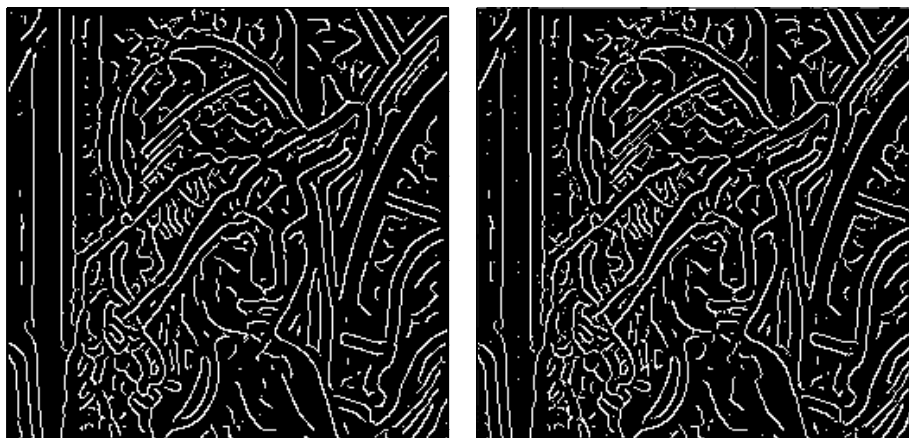
Otra ventaja del método simplificado es que el algoritmo para su obtención puede vectorizarse, obteniendo así una eficiencia mucho mayor en paquetes de software especializados en el tratamiento de matrices.



(a) Primer nivel



(b) Segundo nivel



(c) Tercer nivel

Figura 4.12: Comparación de los tres primeros niveles de MM para Lena. A la izquierda los detectados con el método de combinaciones lineales, y a la derecha los detectados con el método simplificado

Capítulo 5

Detección de bordes

5.1. Esquema general de detección de bordes en imágenes

La detección de bordes es un área del procesamiento de imágenes que ha recibido una gran cantidad de aportes a lo largo del tiempo, y como consecuencia se han desarrollado diferentes métodos para resolver esta tarea en forma eficiente.

La razón por la que se ha convertido en un tema central en el procesamiento de imágenes es porque los bordes tienen aplicaciones muy diversas, como ser la detección de patrones, la visión por computadora, la compresión de imágenes, la segmentación de las mismas, el almacenamiento y búsqueda en bases de datos de imágenes, etc.

Esta gran cantidad de aportes, la especialización de detectores de bordes para ambientes específicos, y la complejidad inherente del problema entre otros factores, hacen que en la actualidad existan diferentes definiciones del concepto de bordes en imágenes.

En el presente trabajo se estudian imágenes monocromáticas, que representan la información mediante variaciones en el nivel de gris (o contraste). En este contexto, se puede definir en forma general a un borde como la frontera entre dos regiones con propiedades de nivel de gris relativamente distintas entre sí.

Básicamente, la idea subyacente en la mayoría de las técnicas de extracción de bordes es el cálculo de un operador local de derivación ¹. La figura 5.1 esquematiza este concepto.

En la columna de la izquierda se trabaja con una imagen con fondo negro y una banda blanca, mientras que en la columna de la derecha se trabaja con la inversa de la imagen anterior. En la segunda fila se muestra el perfil del nivel de gris (tomado en forma perpendicular a las bandas) de las imágenes de la fila superior.

En los perfiles se puede observar que las transiciones de negro a blanco y viceversa no son abruptas, sino que se realizan de forma gradual. Este modelo refleja el hecho de que los bordes de las imágenes digitales están, en general, ligeramente difusos a causa del muestreado que se realiza para obtenerlas. Este efecto gradual también se produce en forma natural en las imágenes, por ejemplo en el caso de las sombras difusas.

La tercera y cuarta fila representan la primera y segunda derivada del perfil respectivamente.

Claramente se nota que tanto la primera derivada como la segunda sólo presentan valores distintos de cero en las posiciones donde se encuentran las transiciones de gris. Puede notarse también que el módulo de la primera derivada llega a sus máximos locales en las posiciones centrales de los cambios de contraste, mientras que en la segunda derivada se observa un paso por cero en esas mismas posiciones.

Por lo tanto, los bordes se encuentran en:

- Los máximo y mínimos locales de f'
- Los cambios de signo de f''

¹Enriquece el análisis suponer que la imagen a analizar I proviene de una función $f(x, y)$ de variable continua que fue discretizada.

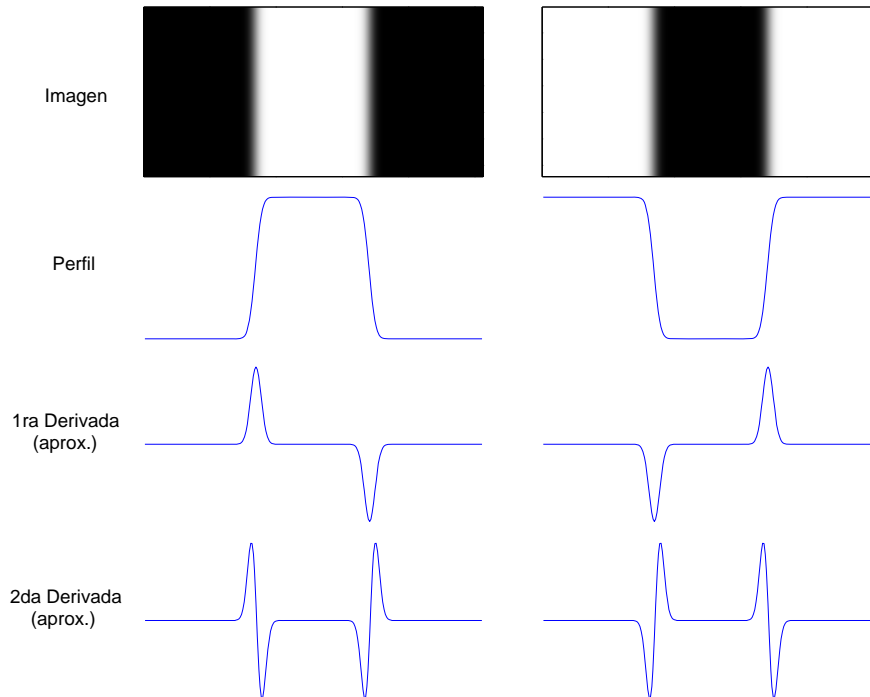


Figura 5.1: Esquema general de detección de bordes mediante operadores locales de derivación. En la primera fila, la imagen. En la segunda, un perfil perpendicular a las bandas. En las últimas dos las estimaciones de la primera y segunda derivadas del perfil.

El razonamiento anterior se puede extender fácilmente a un borde con cualquier orientación dentro de una imagen. Simplemente se define un perfil perpendicular a la dirección del borde en el punto que se desee y se interpretan los resultados siguiendo el mismo razonamiento.

La forma clásica de obtener la primera derivada de una imagen es mediante el gradiente de la misma, mientras que para la derivada segunda se utiliza el laplaciano.

5.2. Multirresolución y wavelets

En la sección anterior se introdujo el esquema general de detección de bordes en una imagen. Este esquema presenta varios inconvenientes, principalmente que es muy sensible al ruido, y que no realiza ninguna distinción entre los bordes que delimitan grandes estructuras, de aquellos que se generan debido a pequeños detalles de la imagen.

Una forma de atacar estos inconvenientes es realizar la detección de bordes a distintas resoluciones o escalas. En una escala gruesa se detectarán bordes que delimitan grandes estructuras, mientras que en una escala fina aparecerán los bordes de los pequeños detalles.

El concepto de diferentes resoluciones está directamente relacionado con el concepto de escalas en la transformada wavelet. Para fines prácticos la escala más fina de la que se dispone es la que se encuentra en la imagen original, y a medida que vamos realizando aproximaciones (o suavizados) sobre la imagen, vamos obteniendo las escalas más gruesas hasta llegar a una imagen constante.

Para mayor claridad de esta tesis transcribimos en esta sección el trabajo de Mallat y Zhong [18], adaptándolo a la notación que venimos usando.

En este trabajo se presenta formalmente la relación que existe entre la transformada wavelet de una señal unidimensional y las derivadas de esta señal.

Previamente necesitamos definir algunos conceptos:

Función de suavizado

Llamaremos *función de suavizado* a cualquier función $\theta(x)$, tal que su integral sea igual a 1, y tienda a 0 en el infinito.

Una función que cumple estas condiciones, y que es muy usada en la detección de bordes es la función gaussiana:

$$g_\sigma(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{t^2}{2\sigma^2}\right)$$

Wavelets como derivadas de θ

Supondremos que $\theta(t)$ es dos veces derivable y definimos $\psi^a(t)$ y $\psi^b(t)$ como la primera y segunda derivada de $\theta(t)$:

$$\psi^a(t) = \frac{-d\theta(t)}{dt} \quad (5.1)$$

$$\psi^b(t) = \frac{d^2\theta(t)}{dt^2} \quad (5.2)$$

Por definición, las funciones $\psi^a(t)$ y $\psi^b(t)$ son wavelets porque su integral es 0 (ver ecuación 3.1).

Dilataciones y desplazamientos

Dada la función $\psi(t)$, recordemos la definición de translación en u y dilatación en s de la ecuación 3.2:

$$\psi_{u,s}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right)$$

Además definimos:

$$\bar{\psi}_s(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{-t}{s}\right) \quad (5.3)$$

Producto escalar de funciones reales

Dadas $f, g : \mathbb{R} \rightarrow \mathbb{R}$, definimos su producto escalar como:

$$\langle f, g \rangle = \int f(t)g(t)dt \quad (5.4)$$

Convolución de funciones reales

Dadas $f, g : \mathbb{R} \rightarrow \mathbb{R}$, definimos su convolución como:

$$(f * g)(u) = \int f(t)g(u-t)dt \quad (5.5)$$

Transformada wavelet continua

Dada la función $f(t)$, recordemos la definición de la transformada wavelet continua dada en 3.3:

$$Wf(u, s) = \int_{-\infty}^{+\infty} f(t) \frac{1}{\sqrt{s}} \psi\left(\frac{t-u}{s}\right) dt$$

Con las nuevas definiciones podemos notar como:

$$W^a f(u, s) = \langle f, \psi_{u,s}^a \rangle \quad y \quad W^b f(u, s) = \langle f, \psi_{u,s}^b \rangle$$

a las transformadas wavelet de f con respecto a ψ^a y ψ^b respectivamente.

Transformada wavelet como operador de derivación

Queremos ver que la transformada wavelet se comporta como un operador de derivación.

Trabajaremos sólo sobre la wavelet ψ^a , dado que el desarrollo para ψ^b es análogo.

Queremos probar que:

$$W^a f(u, s) = s \frac{d}{du} (f * \bar{\theta}_s)(u) \quad (5.6)$$

Es decir, que la transformada wavelet de f es la derivada de la función f suavizada con la función de suavizado θ .

Demostración:

Observemos que:

$$\begin{aligned} \frac{d}{du} \left[\theta\left(\frac{t-u}{s}\right) \right] &= \frac{-1}{s} \frac{d\theta}{du} \left(\frac{t-u}{s}\right) \\ &= \frac{1}{s} \psi^a\left(\frac{t-u}{s}\right) \end{aligned}$$

Calculando la convolución tenemos:

$$\begin{aligned} (f * \bar{\theta}_s)(u) &= \int f(t) \bar{\theta}_s(u-t) dt \\ &= \int f(t) \frac{1}{\sqrt{s}} \theta\left(\frac{t-u}{s}\right) dt \end{aligned}$$

Derivando:

$$\begin{aligned} s \frac{d}{du} (f * \bar{\theta}_s)(u) &= s \frac{d}{du} \left[\int f(t) \frac{1}{\sqrt{s}} \theta\left(\frac{t-u}{s}\right) dt \right] \\ &= s \int f(t) \frac{1}{\sqrt{s}} \frac{d}{du} \left[\theta\left(\frac{t-u}{s}\right) \right] dt \\ &= s \int f(t) \frac{1}{\sqrt{s}} \frac{1}{s} \psi^a\left(\frac{t-u}{s}\right) dt \\ &= \int f(t) \psi_{u,s}^a(t) dt \\ &= \langle f, \psi_{u,s}^a \rangle \\ &= W^a f(u, s) \end{aligned}$$

Este razonamiento muestra que detectar los módulos máximos de la transformada wavelet, con los métodos propuestos en el capítulo 4, es similar a detectar las posiciones de mayor valor absoluto de la derivada de la señal suavizada $\langle f, \theta_{u,s} \rangle$.

Por lo tanto, la detección de módulos máximos de la transformada wavelet es en sí un método de detección de bordes.

El razonamiento anterior se extiende fácilmente a imágenes, ya que como se menciona en los capítulos anteriores, el cálculo de la transformada wavelet “à trous” sobre imágenes se hace aplicando la transformada sobre las filas y las columnas de la misma, y así la transformada wavelet de una imagen corresponde al gradiente de la misma imagen suavizada a diferentes escalas.

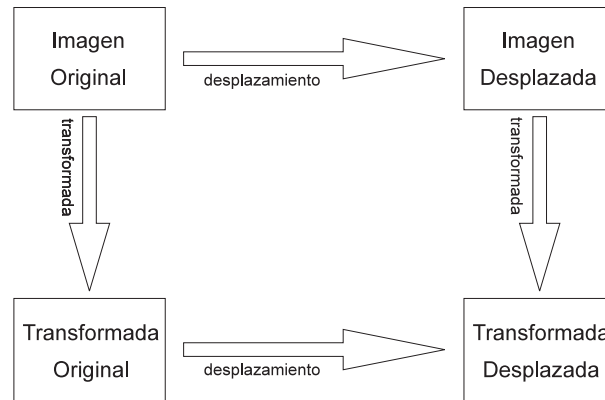


Figura 5.2: Propiedad de invarianza frente a los desplazamientos de la transformada wavelet “à trous”.

5.2.1. Elección de la transformada “à trous”

En el presente trabajo se utiliza la transformada “à trous” para obtener las diferentes escalas wavelets. Las otras opciones consideradas fueron la transformada wavelet continua y la transformada wavelet tradicional (ver definiciones 3.3 y 3.6 respectivamente).

La razón fundamental para elegir la transformada “à trous” sobre la continua es que el cálculo de esta última sobre imágenes es demasiado costoso. En el capítulo en donde se introdujo esta transformada se la definió sobre señales continuas, pero se puede extender a señales discretas e imágenes: el cálculo se realiza entonces mediante una gran cantidad de convoluciones [2]. En cambio, la transformada “à trous” al calcularse solamente sobre escalas diádicas puede obtenerse en forma completa con mucho menos procesamiento.

Las razones para descartar la transformada tradicional en la detección de bordes son dos:

- No es invariante frente a los desplazamientos enteros de la imagen sobre la que se la aplica.
- Como la obtención de sus escalas implica una decimación (o submuestreo), se pierden detalles que pueden ser críticos.

Que una transformada sea invariante frente a los desplazamientos implica que aplicar la transformada a una imagen desplazada es similar a desplazar la transformada de la imagen original. La transformada “à trous” posee esta propiedad. En la figura 5.2 se esquematiza este concepto, para llegar al desplazamiento de la transformada se puede elegir cualquiera de los dos caminos mencionados.

La pérdida de detalles en la transformada tradicional se puede apreciar en la figura 3.8, donde se superponen la transformada tradicional y la “à trous”. En la figura se observa que los valores de la transformada tradicional se distancian entre sí al bajar la escala (debido al submuestreo), lo que produce que varios máximos locales de la transformada “à trous” no se encuentren en la tradicional.

5.3. Métodos alternativos de detección de bordes

Los métodos más conocidos para la detección de bordes de una imagen son el de Canny, el de Roberts, el de Prewitt, el de Sobel, y el llamado LoG (Laplaciano de una Gaussiana). Todos ellos se basan en la obtención aproximada de una derivada de la imagen.

En el presente trabajo se pretende obtener métodos alternativos de detección de bordes en imágenes que combinen la información que brindan las distintas escalas de la transformada wavelet “à trous”.

Se trata de combinar las escalas para generar una selección alternativa porque la elección de una sola escala como detector de bordes presenta varios inconvenientes:

- Es difícil elegir cuál de las escalas es la mejor para una determinada imagen (aún pudiendo elegir una escala intermedia).

- Las escalas más finas, si bien presentan una localización muy buena, son muy sensibles al ruido y a las altas frecuencias, detectándose bordes que es preferible descartar.
- Las escalas más gruesas, si bien descartan los bordes poco significativos y realzan los que corresponden a estructuras importantes, presentan un desplazamiento en la localización de los mismos producido por el suavizado necesario para obtenerlas.

De esta forma se intenta obtener una imagen de bordes con la buena localización que presentan las escalas más finas, pero priorizando los bordes más significativos de la imagen.

En este sentido es fundamental el efecto de propagación y evolución a través de las escalas que presentan los módulos máximos. Observando la transformada “à trous” de varias señales e imágenes, se nota claramente que los MM se propagan hacia escalas más gruesas sólo en las posiciones en donde se encuentran las singularidades o bordes más significativos. Por el contrario, donde los bordes son generados por ruido o por altas frecuencias, los MM se entremezclan y por lo general desaparecen.

5.3.1. Método de la propagación entre escalas

En base al hecho de que las singularidades más importantes se propagan más profundamente en el espacio de MM nace el primer método propuesto, que consiste en identificar hasta qué escala se propaga una determinada singularidad, y luego seleccionar las que se propagaron hasta las escalas más gruesas.

En una dimensión, el esquema del método es básicamente el siguiente:

```
Entradas: N, d, P
Salida  : Ret

Ret = [0 0 ... 0];      /* Ret tiene N posiciones */
for i = 1 : N
    if P[1,i] == 1
        Ret[i] = obtenerMayorEscalaDePropagacion( d, P, i );
    end
end
```

Donde:

- N es el tamaño de la señal a analizar.
- d representa los detalles de la transformada, $\{d_j\}_{1 \leq j \leq \log_2(N)}$.
- P representa las posiciones de los módulos máximos de la transformada, $\{P_j\}_{1 \leq j \leq \log_2(N)}$.
- Ret contiene la escala más gruesa hasta la que se propaga el MM que se encuentra en la primera escala. Tiene N posiciones.
- obtenerMayorEscalaDePropagacion hace referencia a la rutina que revisa hasta qué escala se propaga cada MM.

Intuitivamente, para obtener la escala hasta la que se propaga un MM, se intentan revisar los niveles de la transformada buscando módulos máximos que sean *cercanos y parecidos* al MM del primer nivel.

Esta rutina parece tener una implementación sencilla, sin embargo hay algunos factores que la dificultan:

- *Variación del Módulo.* Al avanzar de una escala a la siguiente los valores de los coeficientes de la transformada en una determinada posición van variando. Esto se debe a que el proceso de filtrado de una escala a la otra, implica que el coeficiente que se encuentra en la posición i en una escala es un promedio ponderado de los coeficientes cercanos a i de la escala anterior. Por otra parte, el tipo de variación depende del tipo de singularidad que presente la señal en ese entorno [18].

- *Desplazamiento.* Dependiendo del tipo de singularidad de la señal, las posiciones de los MM en las distintas escalas pueden o no desplazarse de la posición original (es decir de la posición de la singularidad). Esto se produce debido a que en las escalas más gruesas el tamaño de los filtros es muy superior al de las escalas finas, y por lo tanto abarcan una porción mucho mayor de la señal. En las figuras 5.3, 5.4 y 5.5 se muestran algunos tipos posibles de singularidades junto con las posiciones de los MM que generan en las distintas escalas de la transformada. En la fila superior se encuentra la señal, y en las siguientes las posiciones de los MM en cada nivel de la transformada.

En la figura 5.3 se observa que a mayor cantidad de incrementos se forman más MM , y que su desplazamiento es mayor. En cambio, en la figura 5.4, puede observarse que siempre se genera un único MM en cada nivel y que no hay desplazamiento entre ellos. Finalmente, en la figura 5.5, se nota que los MM se desplazan hacia afuera al aumentar la escala, y que este desplazamiento es proporcional al tamaño de los filtros utilizados.

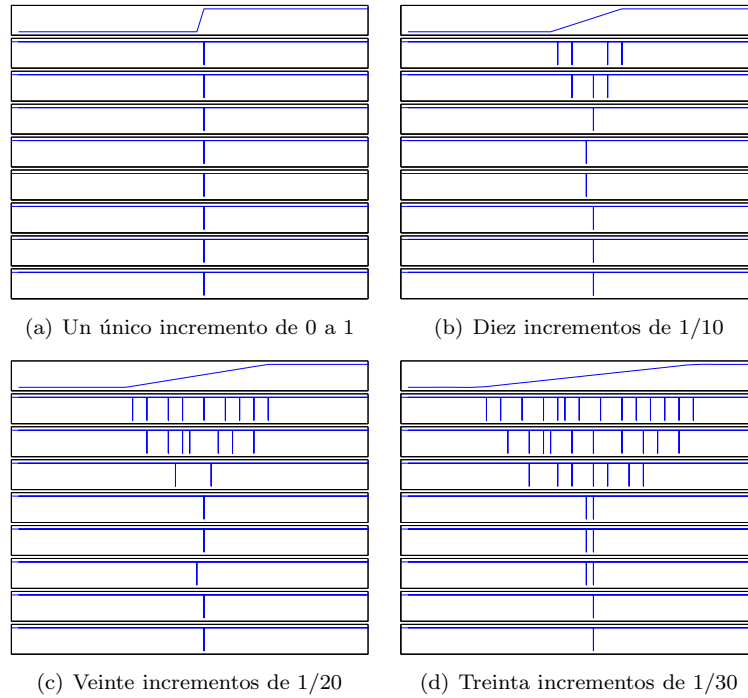


Figura 5.3: Desplazamientos de los MM sobre una señal que va de 0 a 1 en N incrementos iguales. En cada subfigura la primera fila es la señal original, la segunda fila es la escala más fina de la transformada, y la última la más gruesa.

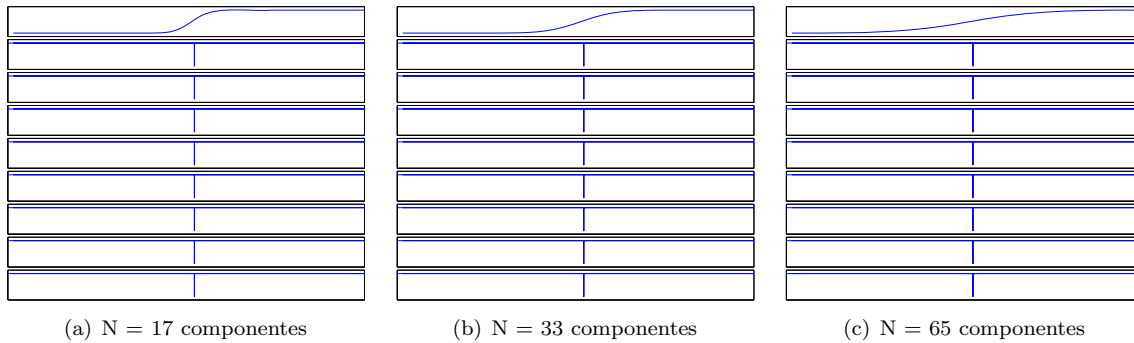


Figura 5.4: Desplazamientos de las posiciones de los MM sobre una señal generada a partir de un salto de 0 a 1 suavizado con un filtro gaussiano de N componentes. En cada subfigura la primera fila es la señal original, la segunda fila es la escala más fina de la transformada, y la última la más gruesa.

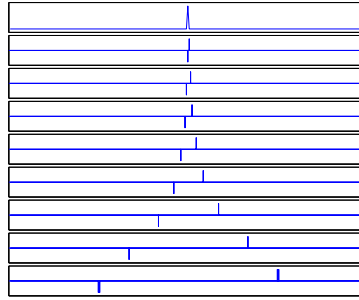


Figura 5.5: Desplazamientos de las posiciones de los MM sobre una señal con todos sus componentes en 0 salvo el central que está en 1. La primera fila es la señal original, la segunda fila es la escala más fina de la transformada, y la última la más gruesa.

- *Fusión de MM* . Si dos o más singularidades se encuentran lo suficientemente cerca, los MM correspondientes a ellas se fusionarán al propagarse hacia las escalas más gruesas. Esto se debe también al aumento del tamaño de los filtros, que a partir de una determinada escala dejan de diferenciarlas como singularidades independientes pasando a detectar una única singularidad. En la figura 5.6 se muestra este comportamiento para dos singularidades bruscas que se encuentran dentro de un entorno. Puede verse que cuanto más cercanas se encuentran entre sí, los MM que generan se fusionan en un nivel más fino.

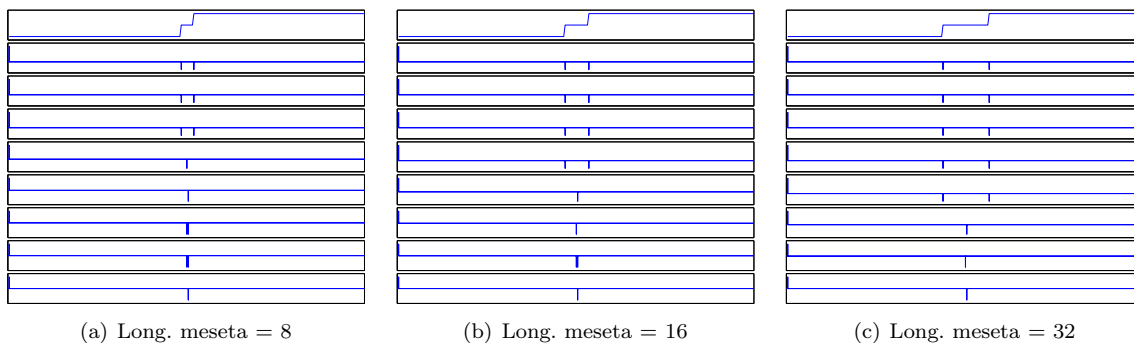


Figura 5.6: Fusión de los MM generados por dos singularidades cercanas. Las señales van de 0 a 0,5 en un primer salto, y luego a 1 en un segundo salto. La única diferencia entre las señales es la distancia entre estos dos saltos, o dicho de otra forma la longitud de la meseta central. En cada subfigura la primera fila es la señal original, la segunda fila es la escala más fina de la transformada, y la última la más gruesa.

La variación de los módulos nivel a nivel genera muchas dificultades al momento de comparar dos MM en niveles contiguos y decidir si fueron generados por la misma singularidad. Se trataron de implementar diferentes normalizaciones sobre los coeficientes de la transformada y de los filtros. Ninguna de estas normalizaciones dió resultados que permitieran independizarnos de esta variación. Finalmente se optó simplemente por verificar que el signo de los MM fuera el mismo en todos los niveles, sin importar el módulo de éstos. Esta simplificación no afecta el resultado del método en forma significativa dado que si el módulo es pequeño será descartado o absorbido por uno mayor que se encuentre cerca (en caso de que exista alguno).

Para atacar el problema del desplazamiento y la fusión de los MM de una escala a la siguiente, se revisan los valores de la transformada (que son MM) dentro de un entorno $2\Delta_j$ que varía dependiendo de la escala en la que nos encontramos. La razón para tomar un entorno dependiente de la escala es que en las escalas más finas (sobre todo en las dos primeras) los desplazamientos son muy pequeños o nulos, mientras que en las escalas gruesas suelen ser bastante grandes.

Resumiendo, dado un módulo máximo en la posición n de la escala 1, se considera que éste se propaga hasta la escala J si:

$$\begin{aligned} \forall j : 2 \leq j \leq J, \\ \exists n' : n - \Delta_j \leq n' \leq n + \Delta_j \text{ tal que :} \\ P_j[n'] = 1 \quad \wedge \quad \text{signo}(d_1[n]) = \text{signo}(d_j[n']) \end{aligned}$$

Extensión a imágenes

Para extender este método a imágenes, se trabaja con las matrices de módulos de la transformada (M_j), definidas en la ecuación 4.3.

De nivel a nivel se revisa un entorno bidimensional (en la implementación actual se trata de un cuadrado).

Dado que los valores de las matrices M_j son siempre positivos no se toma en cuenta la verificación de signo que se hace en una dimensión.

De esta forma, dado un módulo máximo en la posición $[n, m]$ de la escala 1, se considera que éste se propaga hasta la escala J si:

$$\begin{aligned} \forall j : 2 \leq j \leq J, \\ \exists n', m' : n - \Delta_j \leq n' \leq n + \Delta_j \quad , \quad m - \Delta_j \leq m' \leq m + \Delta_j \text{ tal que :} \\ P_j[n', m'] = 1 \end{aligned}$$

Además, para eliminar la mayor parte del ruido que pueda aparecer en la detección se realizaron dos mejoras importantes:

- La primera consiste en aplicar un pequeño umbral a los módulos de la transformada, del orden del 5% aproximadamente. Este umbral permite descartar los MM que se generan debido al ruido presente en la imagen.
- La segunda, consiste en eliminar los conjuntos conexos de menos de 4 píxeles en la imagen de selección. Esto permite descartar las imperfecciones en la detección de bordes en zonas que a simple vista son continuas, pero que poseen pequeñas irregularidades que son detectadas como MM .

Resultados

En esta sección se presentan algunos de los resultados obtenidos mediante este método.

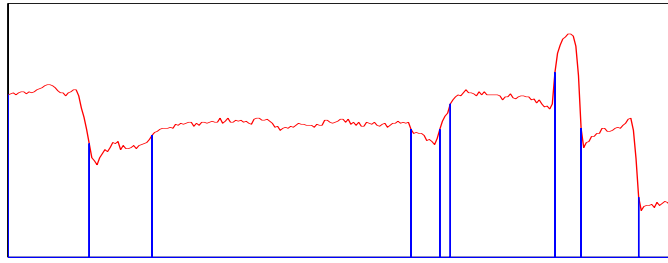
Experimentalmente se comprobó que es conveniente trazar la profundidad de los MM aproximadamente hasta la mitad de las escalas de la transformada, dado que en los niveles más gruesos el tamaño de los filtros es demasiado grande y la cantidad de MM se reduce notablemente. De esta forma, si la transformada tiene J niveles, se traza la profundidad hasta el nivel:

$$\lfloor J/2 \rfloor + 1$$

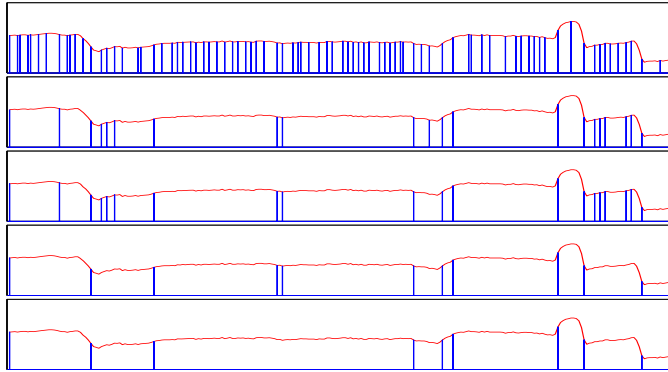
Las figuras 5.7, 5.8 y 5.9 se generaron en base a señales de entrada correspondientes a diferentes filas de la imagen de Lena (de 256x256 píxeles).

En la figura 5.10, se trabajó con una señal artificial para poder controlar el comportamiento del método frente a singularidades bien definidas. Puede verse que todas las singularidades son detectadas, y que la ubicación de las mismas coincide con la posición de máximo cambio, es decir, justo en el centro de cada transición.

Desde la figura 5.11 hasta la figura 5.14 se muestran los resultados obtenidos, adaptando este método para imágenes. Como puede verse en esas imágenes, los bordes más importantes son detectados.

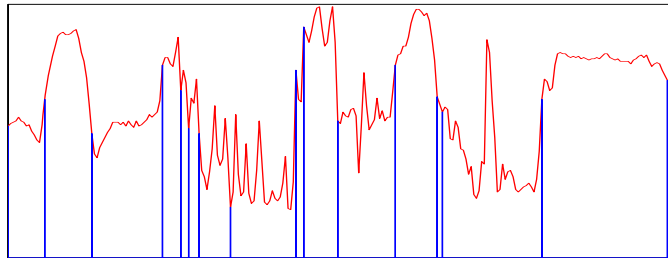


(a) Ubicación de las singularidades propagadas hasta el nivel 5

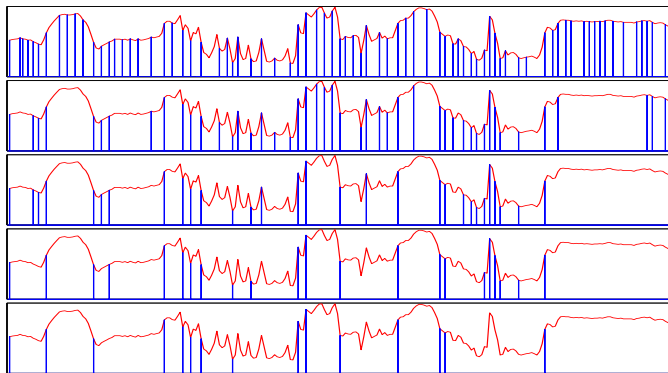


(b) Ubicación de las singularidades propagadas según el nivel. Arriba el primer nivel, el más fino, y abajo el más grueso.

Figura 5.7: Detección de singularidades sobre la fila 16 de la imagen de Lena.

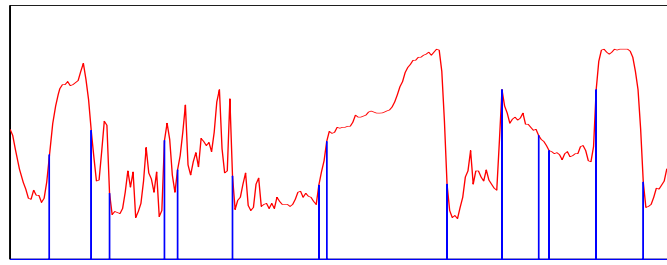


(a) Ubicación de las singularidades propagadas hasta el nivel 5

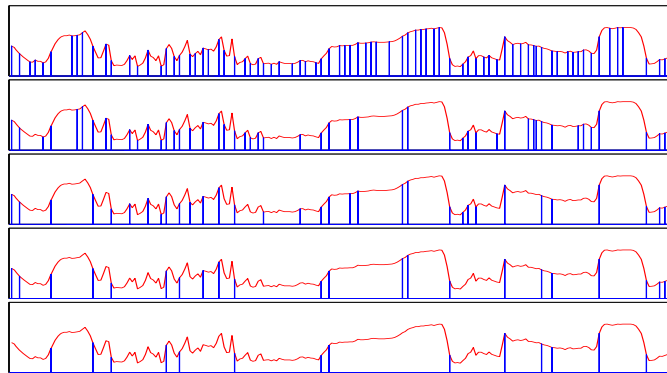


(b) Ubicación de las singularidades propagadas según el nivel. Arriba el primer nivel, el más fino, y abajo el más grueso.

Figura 5.8: Detección de singularidades sobre la fila 128 de la imagen de Lena.

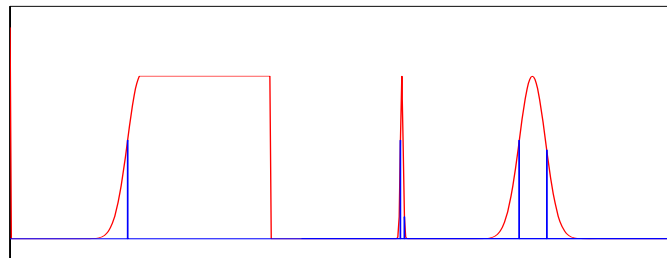


(a) Ubicación de las singularidades propagadas hasta el nivel 5

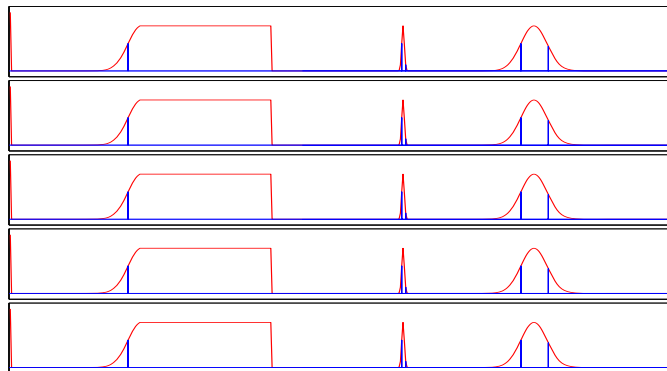


(b) Ubicación de las singularidades propagadas según el nivel. Arriba el primer nivel, el más fino, y abajo el más grueso.

Figura 5.9: Detección de singularidades sobre la fila 200 de la imagen de Lena.



(a) Ubicación de las singularidades propagadas hasta el nivel 5



(b) Ubicación de las singularidades propagadas según el nivel. Arriba el primer nivel, el más fino, y abajo el más grueso.

Figura 5.10: Detección de singularidades sobre una señal unidimensional artificial.



Figura 5.11: Bordes detectados sobre Lena (256x256 píxeles)



Figura 5.12: Bordes detectados sobre Peppers (256x256 píxeles)



Figura 5.13: Bordes detectados sobre Camera (128x128 píxeles)



Figura 5.14: Bordes detectados sobre Amber (128x128 píxeles)

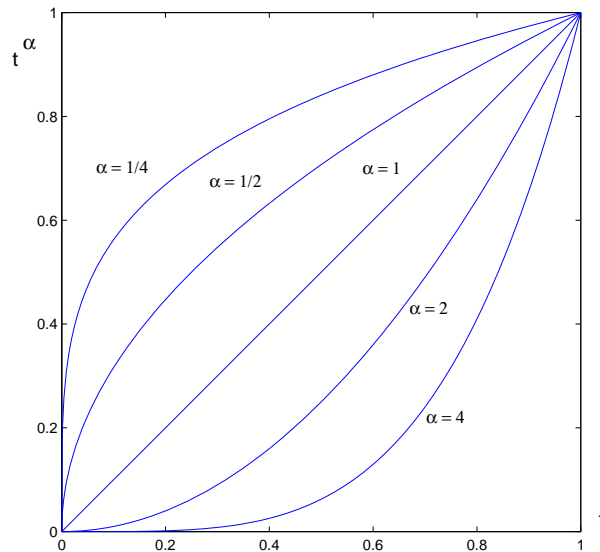


Figura 5.15: Las curvas t^α muestran el comportamiento de una función Lipschitz α en $x = 0$.

5.3.2. Método de la evolución de los coeficientes MM

El método anterior, aunque es intuitivo y produce buenos resultados, tiene como desventaja la necesidad de revisar las escalas hasta el nivel 5, lo que hace que sea bastante costoso su procesamiento. Debido a esto se trató de encontrar un método alternativo en donde no hiciera falta revisar las escalas más gruesas.

El método que se explicará en esta sección se basa en la evolución de los valores de la transformada en las posiciones de los MM a través de las primeras escalas. Esta evolución caracteriza el valor del exponente de *Lipschitz* de la señal en un entorno cercano a la ubicación del MM en el primer nivel.

Definición:

Se dice que una función f es puntualmente Lipschitz α (con $0 < \alpha$) en x cuando²:

a) $0 < \alpha \leq 1$ si $\exists K > 0$ tal que:

$$\forall t \in \mathbb{R} \quad , \quad |f(x) - f(t)| \leq K|x - t|^\alpha$$

b) $1 < \alpha$ si $\exists K > 0$, y un polinomio p_x de grado $m = \lfloor \alpha \rfloor$ tal que:

$$\forall t \in \mathbb{R} \quad , \quad |f(t) - p_x(t)| \leq K|x - t|^\alpha$$

- Si una función f satisface la definición anterior para todos los $x \in [a, b]$ con una constante K independiente de x , se dice que f es uniformemente Lipschitz α sobre $[a, b]$.
- La *regularidad Lipschitz* de f en x sobre $[a, b]$ es el supremo de estos α .

Tratamos de caracterizar a una señal mediante sus exponentes Lipschitz para determinados entornos. Este exponente da una idea de su suavidad o rugosidad. Cuanto menor sea el exponente de Lipschitz más discontinua será la señal.

Observando la definición de Lipschitz para $0 < \alpha \leq 1$ en $x = 0$ (ver figura 5.15), vemos que:

$$|f(t)| \leq K|t|^\alpha$$

Se puede demostrar [20], que una función $f(x)$ es uniformemente Lipschitz α (con $\alpha \geq 0$) sobre (a, b) , si y sólo si existe una constante $K > 0$ tal que para todo $x \in (a, b)$, la transformada wavelet satisface:

²También suele llamársela Holder α .

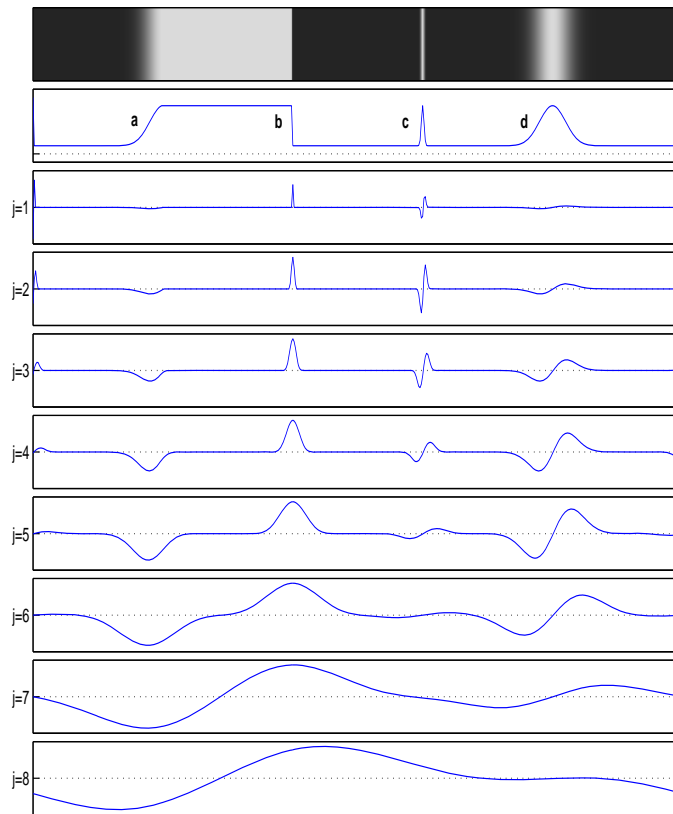


Figura 5.16: Evolución de distintos tipos de singularidades a través de los niveles de la transformada wavelet “à trous”.

$$|Wf(u, s)| \leq K s^\alpha \quad (5.7)$$

Notar que:

- Si $\alpha > 0$, s^α es creciente.
- Si $\alpha < 0$, s^α es decreciente.

Esta ecuación indica que, si α es positivo, la amplitud de los coeficientes que son *MM* debe incrementarse hacia las escalas más gruesas (un valor de s chico indica una escala fina y uno grande una más gruesa).

Inversamente, si α es negativo, la amplitud de los coeficientes que son *MM* debe reducirse hacia las escalas más gruesas.

De esta forma, se estudia la evolución de los *MM* a través de las primeras escalas, desde la primera, la más fina hacia las más gruesas. Si la amplitud de los coeficientes que son *MM* se incrementa, estamos en presencia de una singularidad suave, y si decrece, de una singularidad abrupta.

La figura 5.16 muestra esta situación para algunos tipos de singularidades diferentes. En esta figura se muestran cuatro singularidades:

- La más abrupta es la identificada con la letra **c**, esta singularidad es un salto abrupto entre un nivel de gris bajo a uno alto a ambos lados de la misma. Calculando se puede observar que el valor absoluto de su transformada (y por ende los *MM*) va decreciendo a medida que se avanza de nivel. En una imagen este tipo de singularidades se corresponde con las líneas muy finas y no con los cambios de contraste significativos. Por lo tanto trataremos de no detectarlas como un borde importante, porque el método está orientado a detectar bordes en imágenes reales y en estas imágenes este tipo de singularidades por lo general no delimitan estructuras.

- La singularidad **b** es la segunda más abrupta, en este caso se pasa de un valor alto a la izquierda a uno bajo a la derecha sin valores intermedios. En este caso, el valor más alto (en módulo) de la transformada se mantiene constante a través de los niveles. Este tipo de singularidades sí corresponden a la delimitación de estructuras, y nos interesa detectarlas. Por ejemplo, esta singularidad se puede presentar cuando un objeto negro se encuentra sobre un fondo blanco.
- Las singularidades **a** y **d** son muy similares. Ambas corresponden a transiciones relativamente suaves de valores de gris muy diferentes. La primera de ellas pasa de un valor bajo a uno alto, y la segunda se puede ver como dos singularidades juntas. El valor absoluto de la transformada en estas singularidades va aumentando nivel a nivel. Estas singularidades también deben ser detectadas, debido a que la mayoría de los bordes presentan estas características.

Como se explicó al comienzo de este capítulo, consideramos a los bordes como las fronteras divisorias de porciones de la imagen donde el nivel de gris es más o menos constante. Teniendo ese concepto en mente, y siguiendo el análisis anterior, se deduce que las únicas singularidades que no tomaremos en cuenta son aquellas en las que el módulo de su transformada decrece nivel a nivel. Por lo tanto, las singularidades que deseamos detectar son aquellos en donde el **coeficiente Lipschitz α no sea negativo**. Estas singularidades son las que corresponden a transiciones de un nivel de gris a otro, ya sea de forma muy abrupta (como en la singularidad **b**), o de forma más suave (como en las singularidades **a** y **d**).

Como se están buscando las singularidades con coeficientes Lipschitz no negativos, se pide que los valores se incrementen de nivel en nivel.

El proceso de detección utiliza las matrices de módulos de la transformada (M_j , definidas en la ecuación 4.3), y las matrices de posiciones de los módulos máximos (P_j , definidas en la ecuación 4.5). Ambas definiciones se transcriben a continuación para facilitar la lectura:

$$M_j[n, m] = \sqrt{(DH_j[n, m])^2 + (DV_j[n, m])^2}$$

$$P_j[n, m] = \begin{cases} 1 & \text{si } M_j[n, m] \text{ es máximo} \\ 0 & \text{si no} \end{cases}$$

En base a estas matrices definimos la matriz de módulos en las posiciones de los MM como³:

$$MP_j[n, m] = M_j[n, m] \cdot P_j[n, m] \quad (5.8)$$

Luego tomamos⁴:

$$v_1 = MP_1[n, m]$$

$$v_2 = MP_2[n, m]$$

$$v_3 = MP_3[n, m]$$

Entonces, se clasifica al píxel en la posición $[n, m]$ como “borde” (**correspondiente a $\alpha > 0$**) si:

$$v_1 < (v_2 \cdot \text{tolerancia}) \quad \wedge \quad v_2 < (v_3 \cdot \text{tolerancia}) \quad (5.9)$$

El valor de *tolerancia* se puede utilizar para variar el valor de los coeficientes Lipschitz detectados:

- Si hacemos *tolerancia* = 1, sólo se detectarán los que son no negativos.
- Si se fija *tolerancia* > 1, también se detectarán algunos coeficientes negativos (más negativos cuanto mayor sea el valor de *tolerancia*).
- Si en cambio hacemos *tolerancia* < 1, no se detectarán algunos coeficientes nulos o positivos (se detectarán menos cuanto menor sea el valor de *tolerancia*).

³El punto representa el producto escalar, de esta forma la matriz MP_j se obtiene multiplicando componente a componente las matrices M_j y P_j .

⁴En la práctica, como las posiciones de los MM se van desplazando en las escalas más gruesas, el valor de v_3 se calcula como el valor promedio dentro de una vecindad de $k \times k$ píxeles centrada en la posición $[n, m]$. En la implementación actual se toma $k = 3$.

Problemas de la selección y mejoras propuestas

El resultado del algoritmo propuesto es una imagen binaria de bordes en donde los bordes aparecen en 1 y el resto de la imagen en 0. Esta imagen resultado tiene tres inconvenientes, que se generan a raíz de la naturaleza de los *MM*. Estos son:

- En las zonas en donde la imagen original tiene tonos muy parejos y constantes a simple vista, se producen manchas en la imagen binaria de bordes. Esto se debe a que en esas zonas que parecen constantes para el ojo, en realidad hay, por lo general, una textura subyacente que genera máximos locales de módulo muy pequeño. Si éstos se propagan en las escalas sin llegar a anularse por completo, aparecerán en la imagen binaria de bordes. Para eliminar estas manchas indeseadas de la imagen binaria, se aplica un umbral muy pequeño (del orden del 5 %) sobre la imagen binaria ponderada por los módulos de los detalles en la primera escala (MP_1).

En la figura 5.17 se puede ver la selección entregada por el algoritmo, y el resultado obtenido luego de eliminar las manchas mediante un pequeño umbralado. En la imagen de la derecha se nota claramente la desaparición de las manchas en las zonas de bajas frecuencias de la imagen original.

- Si bien en la imagen de bordes luego de la mejora han desaparecido las manchas que ensuciaban las zonas constantes, todavía siguen apareciendo algunas pequeñas rayas o puntos sueltos que tal vez correspondan a defectos en la imagen o a valores que sobrevivieron por muy poco al umbral aplicado.

Si definimos como *isla conexa* a cualquier conjunto de unos que estén conectados vertical, horizontal o diagonalmente (directa o indirectamente), podemos recorrer la imagen binaria identificando estas islas. Luego eliminamos las que tienen menos de un número determinado de píxeles, y de esta forma limpiamos sensiblemente la imagen de selección. En nuestro caso, sólo conservamos las islas con 4 o más píxeles.

En la figura 5.18 se puede ver el resultado de esta operación aplicada al resultado de la primera mejora, es decir, a la imagen umbralada.

- Por último, si bien la imagen de bordes luego de las mejoras anteriores es mucho más nítida, las curvas de los bordes todavía pueden considerarse gruesas. Los bordes son claros, pero en la mayoría de los casos las curvas tienen un grosor de dos píxeles. En algunas aplicaciones ésto no es un problema, pero en otras es necesario tener curvas de un píxel de grosor. Para lograr este objetivo, se procesa la imagen binaria mediante una operación de morph binario. Esta operación se llama *thinning* [27, 14, 33], y al aplicarla sobre imágenes binarias con objetos blancos sobre fondo negro, reduce todos los objetos a líneas. En nuestro caso, los objetos son curvas de a lo sumo 2 píxeles de grosor, por lo tanto al aplicar esta operación logramos reducir su grosor a 1 sólo píxel. En cierta forma esta mejora es optativa, ya que como se mencionó anteriormente, hay aplicaciones donde no es necesario obtener bordes con un grosor tan fino.

En la figura 5.18 se puede ver el resultado de esta operación aplicada al resultado de la segunda mejora, es decir, a la imagen umbralada y sin píxeles aislados.

Cabe aclarar que el orden de estas dos últimas mejoras puede invertirse, es decir, primero puede aplicarse el *thinning*, y luego descartar las islas pequeñas. El resultado de invertirlas es que también se eliminan las islas que antes del *thinning* tienen 4 o más componentes, pero que luego de éste tienen 3 o menos.

Algunos resultados

Desde la figura 5.19 hasta la figura 5.23, se muestran algunos de los resultados obtenidos con este método (la matriz *Ret* en el algoritmo).



Figura 5.17: Bordes detectados sobre Lena. A la izquierda se muestra el resultado crudo de la selección mediante el método de evolución a través de las escalas. A la derecha se muestran los bordes mejorados mediante un pequeño umbralado.



Figura 5.18: Bordes detectados sobre Lena. Resultado luego de eliminar todas las islas conexas de menos de 4 píxeles sobre la imagen binaria de la figura 5.18 (izquierda). Resultado de aplicar una operación de morph binario que reduce el grosor de los bordes detectados (derecha).



Figura 5.19: Bordes detectados sobre Lena (256x256 píxeles)

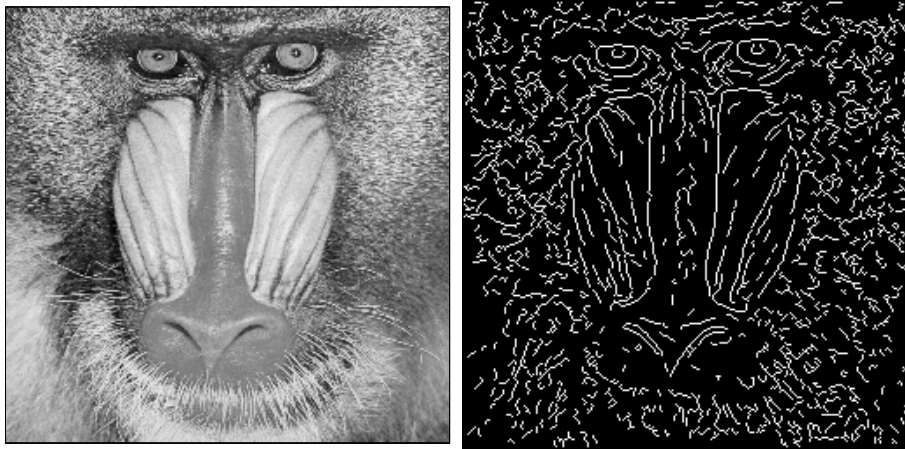


Figura 5.20: Bordes detectados sobre Baboon (256x256 píxeles)



Figura 5.21: Bordes detectados sobre Peppers (256x256 píxeles)

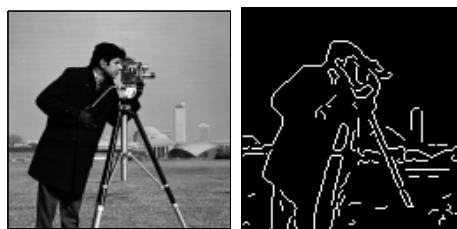


Figura 5.22: Bordes detectados sobre Camera (128x128 píxeles)



Figura 5.23: Bordes detectados sobre Amber (128x128 píxeles)

5.4. Comparación con los métodos clásicos

El objetivo de esta sección es comparar el método propuesto para detectar bordes en imágenes de la sección anterior contra los métodos clásicos de detección de bordes.

Los métodos contra los que se comparará son:

- Roberts
- Prewitt
- Sobel
- LoG
- Canny

Para cada uno de estos métodos se dará una breve descripción de su funcionamiento y luego se mencionarán las diferencias que presentan con el método propuesto, sus ventajas y desventajas.

Se optó por comparar contra métodos cuyo funcionamiento esté basado en el esquema de derivación, descrito en 5.1, dado que presentan un comportamiento similar, y de esta forma la comparación es más adecuada. Además de este tipo de métodos existen otros basados en técnicas totalmente distintas como caminos mínimos sobre grafos, o la aplicación de la transformada de Hough, por nombrar algunos [9].

Roberts, Prewitt y Sobel

El funcionamiento de estos tres métodos es muy parecido. Calculan una aproximación de la derivada de la imagen, y aplican un umbral al módulo de la misma. La diferencia entre los tres radica en que cada uno utiliza una aproximación distinta a la derivada de la imagen.

El esquema de derivación y posterior umbralización del módulo de la derivada es básico dentro de esta categoría de detectores de bordes, y es un punto en común de estos métodos con el propuesto en este trabajo. Sin embargo, estos tres métodos se basan fuertemente en la elección del umbral para obtener resultados satisfactorios, mientras que el método propuesto en este trabajo simplemente utiliza un umbralado para eliminar puntos con módulos muy pequeños que aparecen debido a la naturaleza misma de los módulos máximos de la transformada wavelet “à trous”. En las comparaciones se utilizará el mismo porcentaje de umbral para los tres métodos que el que se utiliza en el método propuesto.

Ninguno de estos tres métodos utiliza multirresolución para la selección de bordes.

Tienen como desventaja la necesidad de aplicar un umbral que es dependiente del método a utilizar, y de la imagen a procesar. Un umbral que funciona muy bien para una imagen A con el método de Roberts, puede funcionar muy mal si aplicamos el método de Prewitt sobre la misma imagen. Asimismo, si procesamos la imagen B con Roberts puede ser necesario volver a seleccionar un umbral, porque el anterior puede no entregar buenos resultados. Esto implica que es necesario ajustar a mano cada umbral para cada imagen si se desea obtener buenos resultados. También hay heurísticas que proponen el cálculo del umbral dependiendo de algunas propiedades de la imagen, como el nivel de ruido, algunas características del histograma, etc [9].

El método propuesto en este trabajo no necesita la fijación manual de un umbral, ya que selecciona los bordes según su evolución a través de las escalas de la transformada wavelet “à trous”.

Para calcular la aproximación de la derivada de la imagen, los tres métodos utilizan dos máscaras para filtrar la imagen y obtener dos componentes separadas de la misma (horizontal o X y vertical o Y para Prewitt y Sobel, y diagonales a 45 y 135 grados para Roberts). Luego, a partir de éstas se obtiene el módulo de la misma.

Las máscaras utilizadas en el método de Roberts son las más sencillas, y se muestran en la figura 5.24. Las máscaras de Prewitt también son muy sencillas (figura 5.25), y hasta podrían reemplazarse por un filtro unidimensional que se aplique por filas y columnas.

+1	0	0	0
0	-1	-1	0

M_{45° M_{135°

Figura 5.24: Máscaras de Roberts para calcular la aproximación de la derivada de una imagen.

+1	+1	+1	-1	0	+1
0	0	0	-1	0	+1
-1	-1	-1	-1	0	+1

M_x M_y

Figura 5.25: Máscaras de Prewitt para calcular la aproximación de la derivada de una imagen.

En el caso de Sobel, las máscaras que se utilizan realizan un suavizado y una derivación al mismo tiempo. Este pequeño suavizado le proporciona cierta resistencia al ruido. Las máscaras de Sobel se muestran en la figura 5.26.

En las figuras 5.27 y 5.28 se muestran los bordes detectados por estos tres métodos para las imágenes Lena y Baboon respectivamente. Los resultados pueden compararse con las figuras 5.19 y 5.20 en donde se encuentran los obtenidos con el método de evolución a través de las escalas. En todos los casos se utilizó el mismo método y porcentaje de umbralado.

Como se puede ver en las figuras, los resultados son bastante pobres si tomamos un nivel de umbralado tan bajo como el que se usa en el método propuesto. Si en cambio fuéramos eligiendo el umbral dependiendo del método y la imagen, se podrían obtener resultados mucho mejores. Como el objetivo es obtener bordes sin asistencia humana, no se realizaron estas pruebas. También se podrían haber incluido pruebas en donde el nivel de umbral se determine en forma automática dependiendo de alguna propiedad de la imagen. Cuando se estaban investigando estas posibilidades, se encontraron algunos resultados muy buenos y otros muy malos. De esto se dedujo que la determinación automática es ya un problema en sí mismo, y se decidió no abordarlo en este trabajo.

LoG

Las siglas *LoG* significan Laplacian of Gaussian. Este método es muy parecido a los anteriores, salvo que en lugar de obtener una aproximación a la derivada de la imagen, obtiene una aproximación a la segunda derivada. Para esto, filtra la imagen con el laplaciano de una función gaussiana. Al obtener la segunda derivada, los bordes se encuentran en los cruces por cero de la misma, por lo tanto, ahora el umbral se tiene que aplicar reteniendo sólo los coeficientes que se encuentran cercanos al cero.

En las figuras 5.30 y 5.31 se pueden ver los resultados de aplicar este método a las imágenes de Lena

+1	+2	+1	-1	0	+1
0	0	0	-2	0	+2
-1	-2	-1	-1	0	+1

M_x M_y

Figura 5.26: Máscaras de Sobel para calcular la aproximación de la derivada de una imagen.



Figura 5.27: Métodos clásicos sobre Lena. Umbral 5 %. Arriba original y Roberts. Abajo a la izquierda Prewitt y a la derecha Sobel.

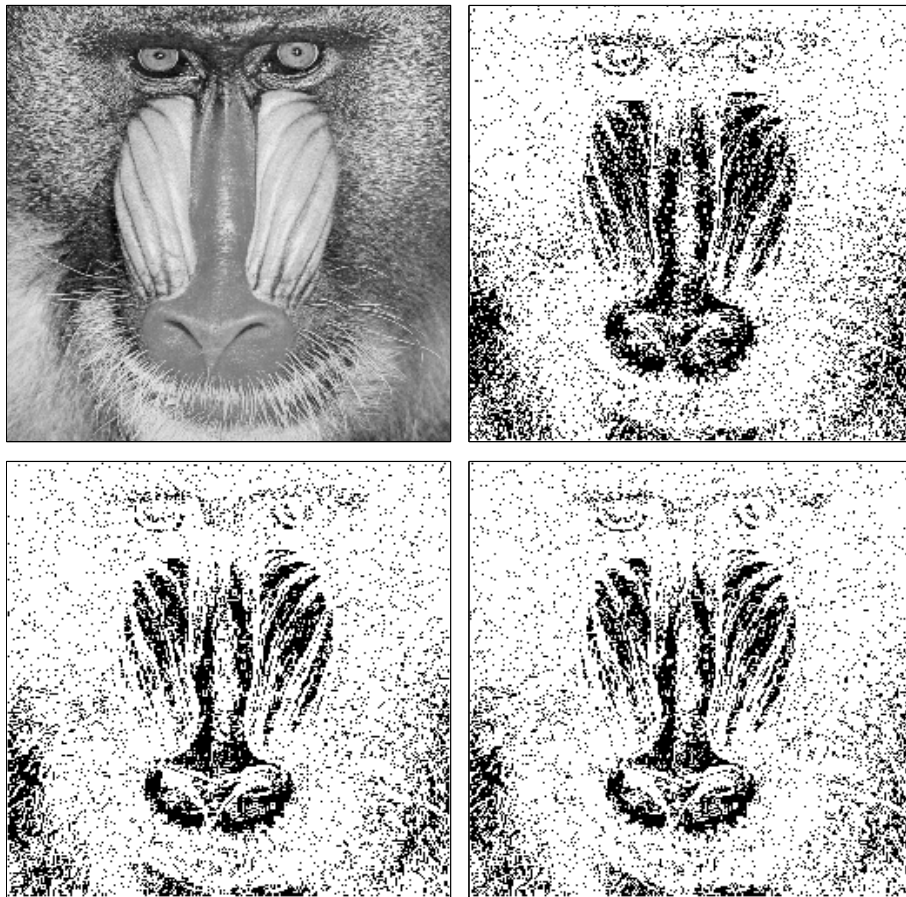


Figura 5.28: Métodos clásicos sobre Baboon. Umbral 5%. Arriba original y Roberts. Abajo a la izquierda Prewitt y a la derecha Sobel. En este caso los bordes son casi irreconocibles. En la figura 5.29 se aumentó el umbral para obtener un mejor resultado.

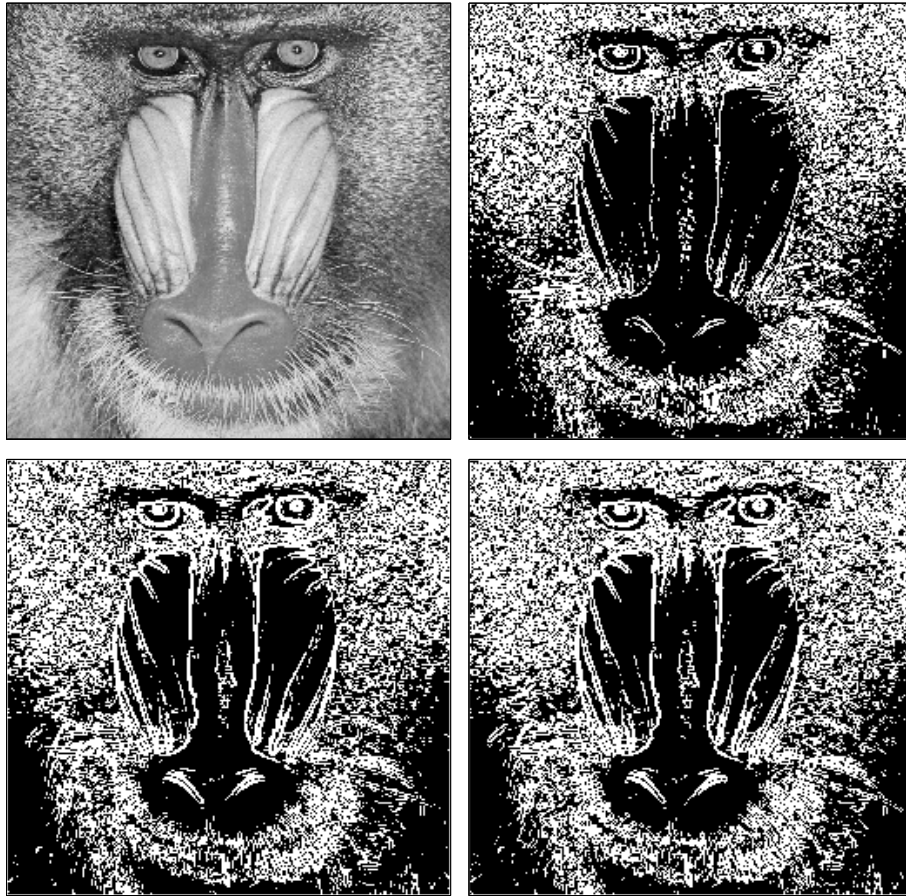


Figura 5.29: Métodos clásicos sobre Baboon. Umbral 15 %. Arriba original y Roberts. Abajo a la izquierda Prewitt y a la derecha Sobel.



Figura 5.30: Bordes de Lena utilizando el método LoG.

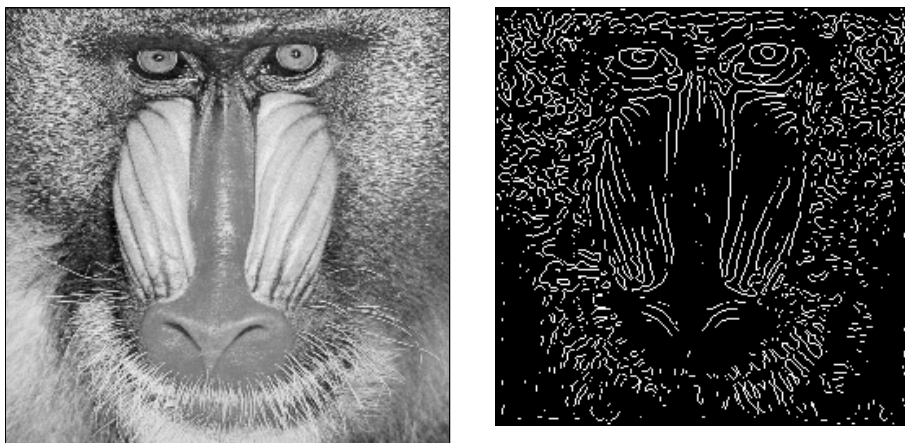


Figura 5.31: Bordes de Baboon utilizando el método LoG.

y Baboon.

Este método tiene como desventaja que al tratarse de una derivada doble es muy sensible al ruido. Por otro lado, en algunas ocasiones produce bordes dobles. Otro de los problemas es que no puede detectar la dirección de los bordes. El papel que cumple el laplaciano en la detección de bordes es secundario, por lo general se lo utiliza para establecer si un píxel está en la parte clara u oscura de un borde.

El problema de la sensibilidad al ruido se puede abordar haciendo un paso previo sobre la imagen para suavizarla. Sin embargo, esto ocasiona que los bordes se dilaten.

Canny

Este método es uno de los más aceptados para detección de bordes. Una explicación en detalle del mismo se puede encontrar en [4, 3].

En líneas generales consta de los siguientes pasos:

- a) Filtrar la imagen con un filtro pasa bajos para eliminar el ruido. Para esto se puede utilizar un filtro que sea una aproximación discreta a una función gaussiana.
- b) Obtener una aproximación al módulo y la dirección de la derivada de la imagen. Para esto se pueden utilizar las técnicas propuestas por los primeros tres métodos (Roberts, Prewitt o Sobel). También se pueden utilizar cualquier otro par de máscaras que generen aproximaciones de la



Figura 5.32: Bordes de Lena utilizando el método de Canny.

derivada en dos direcciones perpendiculares.

- c) Retener sólo los puntos que son máximos locales del módulo de la derivada. Para ésto se puede utilizar el mismo procedimiento que para detectar los módulos máximos de la transformada wavelet “à trous”, descrito en el capítulo 4.
- d) Aplicar finalmente un proceso de *histéresis* para obtener una imagen binaria y reducir las discontinuidades en los bordes. Este proceso consiste en aplicar dos umbrales a los valores cuyos módulos son máximos locales de la aproximación de la derivada. El primer umbral selecciona bordes fuertes (es un umbral alto), el segundo selecciona bordes débiles (es un umbral bajo). Los bordes definitivos se generan con todos los bordes fuertes más aquellos bordes débiles que estén conectados directamente con alguno fuerte.

Este método genera muy buenos resultados sobre una gran cantidad de imágenes, y se lo suele citar como el método óptimo de detección de bordes. Sus ventajas son varias. Por un lado, al realizar un filtrado pasabajos previo elimina el ruido de la imagen por lo que no se detectan bordes no deseados generados por altas frecuencias. Además, como selecciona sólo los puntos que son máximos locales del módulo de la derivada detecta los bordes en su posición de máximo cambio. Por último, la aplicación de dos umbrales permite obtener bordes claros y sin interrupciones.

Su principal desventaja es la necesidad de elegir los porcentajes de umbral que se aplicarán. Como se comentó antes, existen técnicas que realizan esta elección en forma automática, pero no siempre dan buenos resultados.

Este método tampoco utiliza información de multirresolución.

En las figuras 5.32 y 5.33 se pueden ver los resultados obtenidos sobre las imágenes de comparación. Los resultados sobre Lena son un poco menos localizados que los obtenidos con el método propuesto en este trabajo.

Los resultados sobre Baboon son un poco mejores que los obtenidos con el método propuesto. Sobre todo, no hay tantos bordes en los laterales y abajo. Sin embargo, los ojos y los pliegues del hocico están mejor detectados en nuestro método.

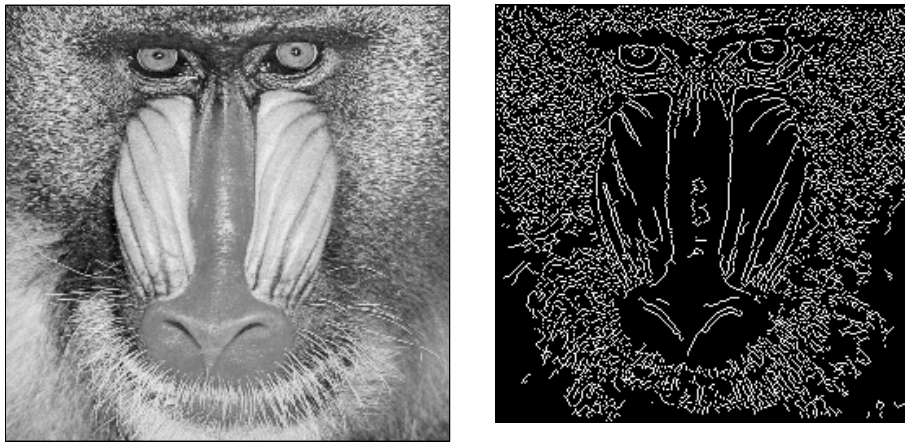


Figura 5.33: Bordes de Baboon utilizando el método de Canny.

Capítulo 6

Reconstrucción usando los módulos máximos

6.1. Introducción

La mayor parte de la información de una imagen se concentra en los bordes de la misma. Por lo tanto en teoría debería ser posible tomar una imagen, quedarnos sólo con los bordes de ésta, y luego reconstruirla a partir de ellos. Esto es justamente lo que se hace en este capítulo: se describe un método de selección y retención de bordes a diferentes escalas y un algoritmo iterativo que a partir de ellos reconstruye una aproximación bastante fiel de la imagen original. La imagen reconstruida no es exáctamente igual a la original, pero las diferencias son imperceptibles al ojo humano.

El método consiste en una implementación de la técnica de *aproximaciones sucesivas* propuesta por Mallat en [18, 16].

La idea detrás de este método es muestrear la imagen en forma irregular. Las posiciones de las muestras serán las posiciones donde se detectan los módulos máximos, y los valores de estas muestras son los vectores de detalle (ángulo, módulo) en cada una de estas posiciones. Luego para reconstruir la imagen se piensa a estas muestras como un *frame* (ver 6.2.1), es decir una representación redundante de la información y se utiliza un algoritmo iterativo (gradiente conjugado) para llegar a partir de esta representación hasta la imagen que la originó.

De esta forma, entre la imagen original y la reconstruida se generan dos tipos de errores. El primero es el error que existe entre la imagen original y la imagen a la que tiende el algoritmo iterativo, que no tienen por qué ser la misma. Y el segundo error se debe a que para fines prácticos, se realiza una cantidad finita de iteraciones, por lo que obtendremos una aproximación de la imagen a la que tiende el algoritmo iterativo.

El capítulo comienza introduciendo este concepto para señales de una dimensión y luego lo extiende para imágenes.

6.2. Reconstrucción en 1D

Si asumimos que nuestra señal de entrada es la discretización de cierta función continua f , entonces al tomar la transformada wavelet de ésta tendremos:

$$Wf(u, s) = \langle f, \psi_{u,s} \rangle \quad (6.1)$$

Siendo u la posición y s la escala. En los desarrollos discretos siempre usamos escalas diádicas (potencias de 2), por lo tanto, tomaremos $s = 2^j$.

Al detectar los *módulos máximos* de la transformada, en cada escala 2^j conocemos las posiciones de los mismos, a las que llamaremos $\{u_{p,j}\}_p$ y sus valores:

$$Wf(u_{p,j}, 2^j) = \langle f, \psi_{p,j} \rangle \quad (6.2)$$

siendo:

$$\psi_{p,j}(t) = \frac{1}{\sqrt{2^j}} \psi\left(\frac{t - u_{p,j}}{2^j}\right)$$

Si llamamos ψ' a la derivada de ψ , como $Wf(u, 2^j)$ tiene un extremo local en $u = u_{p,j}$ (por ser MM) nos queda:

$$\frac{\partial Wf(u_{p,j}, 2^j)}{\partial u} = -2^{-j} \langle f, \psi'_{p,j} \rangle = 0 \quad (6.3)$$

La idea es obtener un algoritmo que basándose en las posiciones $\{u_{p,j}\}_p$ y en los valores $Wf(u_{p,j}, 2^j)$, nos devuelva una función \tilde{f} que sea una aproximación a la original f y que cumpla:

$$W\tilde{f}(u_{p,j}, 2^j) = \langle \tilde{f}, \psi_{p,j} \rangle = \langle f, \psi_{p,j} \rangle \quad (6.4)$$

y

$$\langle \tilde{f}, \psi'_{p,j} \rangle = \langle f, \psi'_{p,j} \rangle = 0 \quad (6.5)$$

En la ecuación 6.4 se pide que los valores de la transformada de \tilde{f} en las posiciones $\{u_{p,j}\}_p$ (las posiciones de los MM de Wf) sean los mismos que los valores de la transformada de f .

En 6.5 se pide que la derivada de $W\tilde{f}(u, 2^j)$ se haga cero en las posiciones $\{u_{p,j}\}_p$. Esto sólo no alcanza para garantizar que en esas posiciones tengamos un MM en $W\tilde{f}$, lo cual se logra minimizando $\|\tilde{f}\|$.

La señal que satisface 6.4 y 6.5 y tiene norma mínima es la proyección ortogonal $P_v f$ de f sobre el espacio V que se genera con $\{\psi_{p,j}, \psi'_{p,j}\}_{p,j}$, y como en los cálculos discretos hay un número finito de MM, $\{\psi_{p,j}, \psi'_{p,j}\}_{p,j}$ es una familia finita, y por lo tanto, una base o un *frame* de V [16].

6.2.1. Frames

Un frame de un espacio X es un conjunto (redundante o no) de vectores de X que nos permiten expresar cualquier elemento de X como combinaciones de estos vectores. En nuestro caso, el frame en cuestión son las wavelets $\psi_{p,j}$ y sus derivadas $\psi'_{p,j}$.

Este conjunto o familia de vectores se nota $\{\nu_n\}_{n \in \Gamma}$, donde los ν_n son los vectores y el conjunto de índices Γ puede ser finito o infinito.

Dada una función $f \in X$, su representación o proyección en el frame $\{\nu_n\}_{n \in \Gamma}$ se obtiene mediante el operador U :

$$\forall n \in \Gamma \quad , \quad Uf[n] = \langle f, \nu_n \rangle \quad (6.6)$$

A los $Uf[n]$ se los llama coeficientes del frame.

Formalmente la familia de vectores $\{\nu_n\}_{n \in \Gamma}$ son un frame del espacio X si existen dos constantes $A > 0$ y $B > 0$ tales que para cualquier $f \in X$ tenemos:

$$A \|f\|^2 \leq \sum_{n \in \Gamma} |\langle f, \nu_n \rangle|^2 \leq B \|f\|^2 \quad (6.7)$$

En los casos en donde los vectores $\{\nu_n\}_{n \in \Gamma}$ dependen de la señal que se analiza, como en el caso de los MM, para volver a obtener f a partir de los coeficientes del frame, se define el operador L como :

$$Lf = U^t Uf = \sum_{n \in \Gamma} \langle f, \nu_n \rangle \nu_n \quad (6.8)$$

y luego f se puede volver a obtener haciendo:

$$f = L^{-1}Lf = L^{-1}g \quad (6.9)$$

Cálculo de $L^{-1}g$

El Teorema 5.4 de [16] presenta un algoritmo de Gradiente Conjugado que nos permite calcular $L^{-1}g$ en forma eficiente:

Entrada:

g : Vector al que se le quiere aplicar el operador L^{-1}

Salida:

f_{n+1} : Vector resultado en la iteración n

Inicialización:

$f_0 = 0$, /* Comenzamos con una aproximación inicial nula */
 $r_0 = g$, /* Variables auxiliares */
 $p_0 = g$,
 $p_{-1} = 0$

Para $n \geq 0$:

$$\begin{aligned} \lambda_n &= \frac{\langle r_n, p_n \rangle}{\langle p_n, Lp_n \rangle} \\ f_{n+1} &= f_n + \lambda_n p_n \\ r_{n+1} &= r_n - \lambda_n Lp_n \\ p_{n+1} &= Lp_n - \frac{\langle Lp_n, Lp_n \rangle}{\langle p_n, Lp_n \rangle} p_n - \frac{\langle Lp_n, Lp_{n-1} \rangle}{\langle p_{n-1}, Lp_{n-1} \rangle} p_{n-1} \end{aligned}$$

La convergencia del algoritmo depende de las constantes A y B de la ecuación 6.7:

$$\|f - f_n\|_L \leq \frac{2\sigma^n}{1 + \sigma^{2n}} \|f\|_L$$

con

$$\sigma = \frac{\sqrt{B} - \sqrt{A}}{\sqrt{B} + \sqrt{A}}$$

y entonces $\lim_{n \rightarrow +\infty} f_n = f$.

Ejemplo sencillo de reconstrucción

Supongamos que nuestro espacio X es el plano real (\mathbb{R}^2).

En este espacio la base canónica es $\{e_1 : (1, 0); e_2 : (0, 1)\}$. Tomemos como posible frame el conjunto de vectores:

$$\{\nu_n\}_{n \in \Gamma} = \{\nu_1 : (3, 0); \nu_2 : (0, 2); \nu_3 : (1, 3)\}$$

Como X es el plano real y $f \in X$, entonces nuestra función f tiene que ser un punto de \mathbb{R}^2 (o visto de otra forma, f es una función de dimensión 2). Tomemos por ejemplo $f = (5, 5)$. Utilizando la ecuación 6.6 obtenemos:

$$\begin{aligned} Uf[1] &= \langle (5, 5), (3, 0) \rangle = 15 + 0 = 15 \\ Uf[2] &= \langle (5, 5), (0, 2) \rangle = 0 + 10 = 10 \\ Uf[3] &= \langle (5, 5), (1, 3) \rangle = 5 + 15 = 20 \end{aligned}$$

Luego usando la ecuación 6.8 obtenemos Lf como:

$$\begin{aligned}
 L(5, 5) &= Uf[1](3, 0) + Uf[2](0, 2) + Uf[3](1, 3) \\
 &= 15(3, 0) + 10(0, 2) + 20(1, 3) \\
 &= (45, 0) + (0, 20) + (20, 60) \\
 &= (65, 80) = g
 \end{aligned}$$

Ahora usando el algoritmo de reconstrucción que describimos anteriormente tenemos:

Inicialización:

$$\begin{aligned}
 f_0 &= (0, 0) \\
 r_0 &= (65, 80) \\
 p_0 &= (65, 80) \\
 p_{-1} &= (0, 0)
 \end{aligned}$$

Para $n = 0$:

$$\begin{aligned}
 Lp_0 &= L(65, 80) = (890, 1235) \\
 \lambda_0 &= \frac{\langle (65, 80), (65, 80) \rangle}{\langle (65, 80), (890, 1235) \rangle} = 0,0678 \\
 f_1 &= (0, 0) + 0,0678(65, 80) = (4,4087, 5,4261) \\
 r_1 &= (65, 80) - 0,0678(890, 1235) = (4,6345, -3,7656) \\
 p_1 &= (890, 1235) - \frac{\langle (890, 1235), (890, 1235) \rangle}{\langle (65, 80), (890, 1235) \rangle}(65, 80) = (-71,5456, 51,5592)
 \end{aligned}$$

Para $n = 1$:

$$\begin{aligned}
 Lp_1 &= L(-71,5456, 51,5592) = (-560,7788, 455,6328) \\
 \lambda_1 &= \frac{\langle (4,6345, -3,7656), (-71,5456, 51,5592) \rangle}{\langle (-71,5456, 51,5592), (-560,7788, 455,6328) \rangle} = -0,0083 \\
 f_2 &= (4,4087, 5,4261) + -0,0083(-71,5456, 51,5592) = (5, 5)
 \end{aligned}$$

En la segunda iteración tenemos $f_2 = f$.

6.2.2. Definición del operador L en 1D

En esta sección se indica cómo calcular el operador L que nos permite utilizar el algoritmo de reconstrucción sobre el frame del espacio V compuesto por las wavelets $\psi_{p,j}$ y sus derivadas $\psi'_{p,j}$.

En un principio, se puede tomar:

$$\forall v \in V, \quad Lv = \sum_{j,p} (\langle v, \psi_{p,j} \rangle \psi_{p,j} + \langle v, \psi'_{p,j} \rangle \psi'_{p,j})$$

Sin embargo, se puede obtener una implementación más rápida si no se impone en forma explícita la restricción sobre las derivadas de las wavelets, y en cambio se utiliza un espacio V reducido, generado sólo por $\{\psi_{p,j}\}_{p,j}$, y definiendo :

$$\forall v \in V, \quad Lv = \sum_{j,p} \langle v, \psi_{p,j} \rangle \psi_{p,j} \tag{6.10}$$

Al eliminar la restricción sobre las derivadas de las wavelets se pierde la seguridad de obtener máximos locales en las posiciones $\{u_{p,j}\}_p$, pero como estamos minimizando la norma $\|\tilde{f}\|$, en esas posiciones se obtienen por lo general valores que están muy cerca de ser máximos.

Cálculo eficiente del operador L

La transformada “à trous” nos permite realizar el cálculo del operador L aplicado a un determinado vector v de forma eficiente [16].

Los elementos del espacio V en este caso son señales unidimensionales finitas (asumiremos que su largo es N), y por lo tanto, se puede aplicar la transformada “à trous” sobre ellos.

Los pasos a seguir son los siguientes:

- * Calcular la transformada “à trous” de v obteniendo $\{d_1, d_2, \dots, d_J, a_J\}$, donde $J = \log_2(N)$.
- * Detectar las posiciones donde se encuentran los módulos máximos de la transformada. De esta forma obtenemos $\{P_j\}_{1 \leq j \leq J}$.
- * Poner en cero todas las posiciones de la transformada que no sean módulos máximos, es decir, aquellas que tienen un cero en $\{P_j\}_{1 \leq j \leq J}$. Así obtenemos los valores $\{\tilde{d}_j\}_{1 \leq j \leq J}$.
- * Aplicar sobre $\{\tilde{d}_j\}_{1 \leq j \leq J}$ la transformada “à trous” inversa con las siguientes modificaciones:
 - Utilizar los filtros h y g , en lugar de utilizar los filtros duales \tilde{h} y \tilde{g} , dado que en la ecuación 6.2.2 las wavelets de descomposición y reconstrucción son las mismas.
 - No aplicar la constante de normalización $1/2$ al pasar de un paso al siguiente, debido a que las wavelets utilizadas en la reconstrucción no están atenuadas en 2^{-j} .

De esta forma nos queda¹:

$$\tilde{a}_j = \tilde{a}_{j+1} \otimes (h \uparrow 2^j) + \tilde{d}_{j+1} \otimes (g \uparrow 2^j)$$

- * Finalmente nos queda que $Lv = \tilde{a}_0$.

6.2.3. Obtención de una aproximación de la señal original

Basándonos en la ecuación 6.9, obtenemos:

$$\tilde{f} = L^{-1}L\tilde{f} = L^{-1}\tilde{g} \quad (6.11)$$

donde \tilde{f} es la aproximación a la señal original que estamos buscando, y \tilde{g} (por la ecuación 6.10) es:

$$\tilde{g} = L\tilde{f} = \sum_{j,p} \langle \tilde{f}, \psi_{p,j} \rangle \psi_{p,j}$$

Dado que en las posiciones $\{u_{p,j}\}_p$ la ecuación 6.4 nos garantiza que los valores de la transformada de \tilde{f} son iguales que los de la transformada de f tenemos que:

$$\tilde{g} = Lf = \sum_{j,p} \langle f, \psi_{p,j} \rangle \psi_{p,j} \quad (6.12)$$

y por lo tanto:

$$\tilde{f} = L^{-1}Lf \quad (6.13)$$

¹Para una demostración más detallada puede consultarse la sección 6.2.2 de [16], pág. 188.

Escala	Figura 6.1	Figura 6.2	Figura 6.3	Figura 6.4
1	31.25 %	34.38 %	17.97 %	2.73 %
2	24.61 %	26.56 %	16.41 %	3.13 %
3	10.94 %	16.41 %	12.50 %	3.13 %
4	6.64 %	6.64 %	6.25 %	3.13 %
5	3.91 %	3.13 %	3.13 %	3.13 %
6	1.95 %	2.34 %	2.34 %	1.56 %
7	1.17 %	0.78 %	0.78 %	1.56 %
8	0.78 %	0.78 %	0.78 %	0.78 %
Total	10.16 %	11.38 %	7.52 %	2.40 %

Tabla 6.1: Porcentajes de MM detectados en los ejemplos de reconstrucción en 1D.

6.2.4. Ejemplos de reconstrucción en 1D

En esta sección se presentan algunos ejemplos de reconstrucción en 1D utilizando la representación en base a bordes a diferentes escalas y el algoritmo presentado en este capítulo.

Se utilizaron cuatro señales diferentes (figuras 6.1, 6.2, 6.3 y 6.4). La primera y la segunda son las filas centrales de Lena y Baboon respectivamente. Se las tomó como ejemplo porque son una clara muestra de señales reales con altas y bajas frecuencias.

La tercera es nuevamente la fila central de Baboon, pero luego de aplicarle una serie de suavizados para descartar las altas frecuencias. De esta forma se intenta mostrar el desempeño del método en señales con cambios suaves.

Por último, la cuarta es una imagen artificial que se utilizó para destacar el comportamiento de la reconstrucción en las zonas de cambios abruptos.

Para cada señal se muestra, arriba el original, luego en las tres primeras filas la reconstrucción obtenida y el error entre ésta y el original luego de aplicar 3, 5 y 20 iteraciones del algoritmo de reconstrucción. Por último se muestra la evolución del PSNR entre el original y las reconstrucciones aplicando entre 1 y 30 iteraciones del algoritmo de reconstrucción.

Como puede verse en las figuras mencionadas, ya a partir de la tercera iteración se logran muy buenos resultados cuando las señales tienen un alto contenido de ruido. También se puede ver en los gráficos de evolución del PSNR que el algoritmo de reconstrucción tiene una gran mejora en las primeras 5 iteraciones aproximadamente, y luego la mejora se vuelve asintótica. Conocer ésto nos permite realizar pocas iteraciones del mismo (5 ó 7) dado que realizar muchas más no nos beneficiará demasiado.

En la figura 6.5 se puede observar en forma comparativa la evolución de los cuatro PSNR de las figuras anteriores. Se observa que los PSNR de Lena y Baboon tienen una evolución muy parecida, con valores cercanos a 40, que para esta medida se consideran muy buenos. La evolución correspondiente a la señal de Baboon suavizada es un poco mejor debido a la ausencia de frecuencias muy altas. Por otra parte, la evolución de la señal artificial presenta PSNR muy altos con respecto a los anteriores (del orden de 80). Esto se debe a que grandes porciones de la señal presentan valores constantes, y el PSNR no es una muy buena medida de comparación en estos casos.

En todas estas reconstrucciones se utilizó la representación completa de MM. Esto implica retener por cada escala de la transformada, todos los coeficientes de detalle que se ubican en las posiciones que son MM. En la tabla 6.1 se muestran las cantidades de MM detectados (en porcentaje) para cada figura, por escala y en total.

6.3. Reconstrucción en Imágenes

En esta sección se extienden a señales bidimensionales (imágenes) los resultados obtenidos para el caso unidimensional.

La idea es similar, partiendo de una imagen a la que llamaremos I , calculamos la transformada “à trous” de la misma, y sobre ésta calculamos los MM. Luego a partir de la información retenida

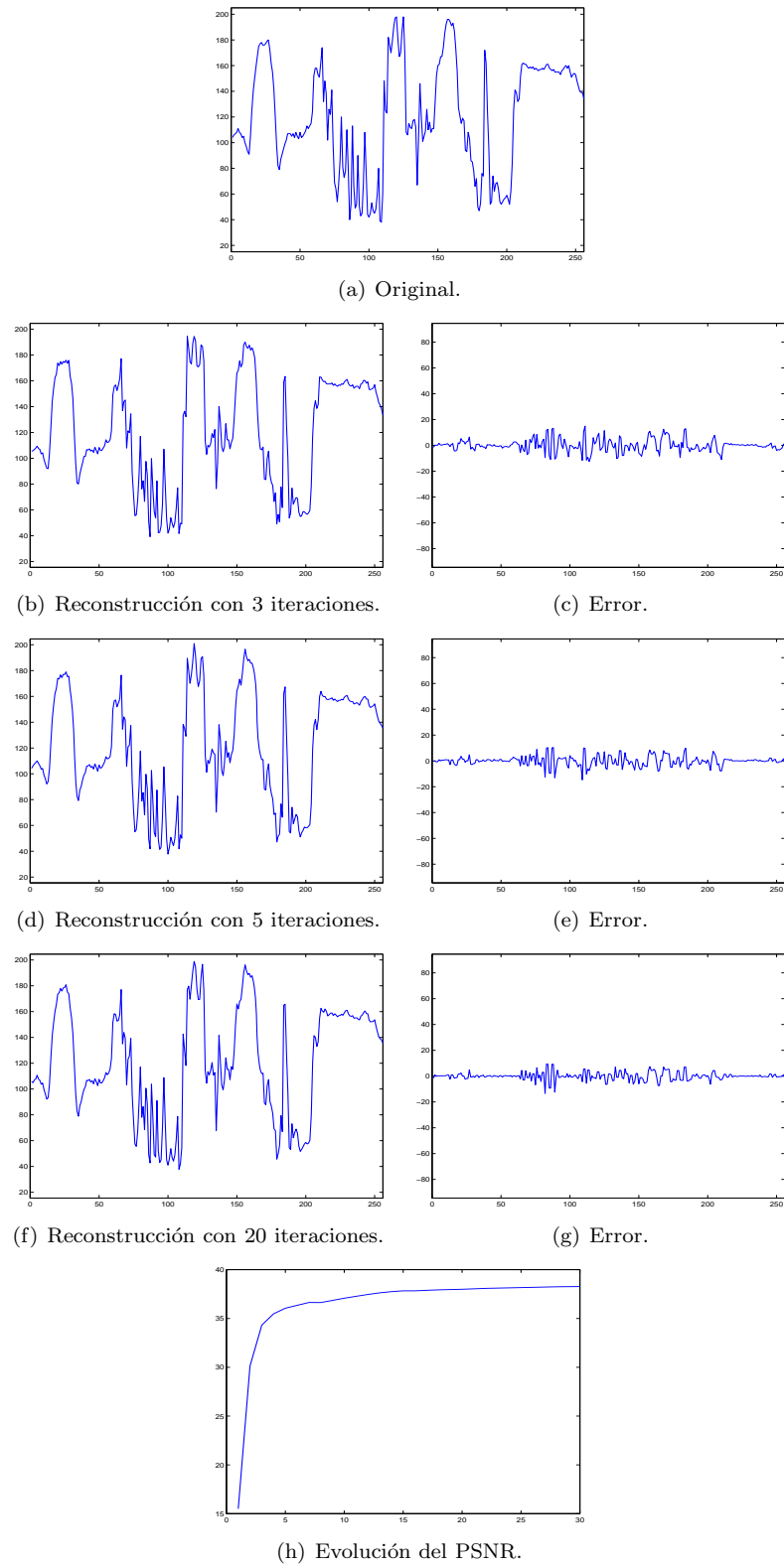


Figura 6.1: Reconstrucción sobre la fila central de Lena.

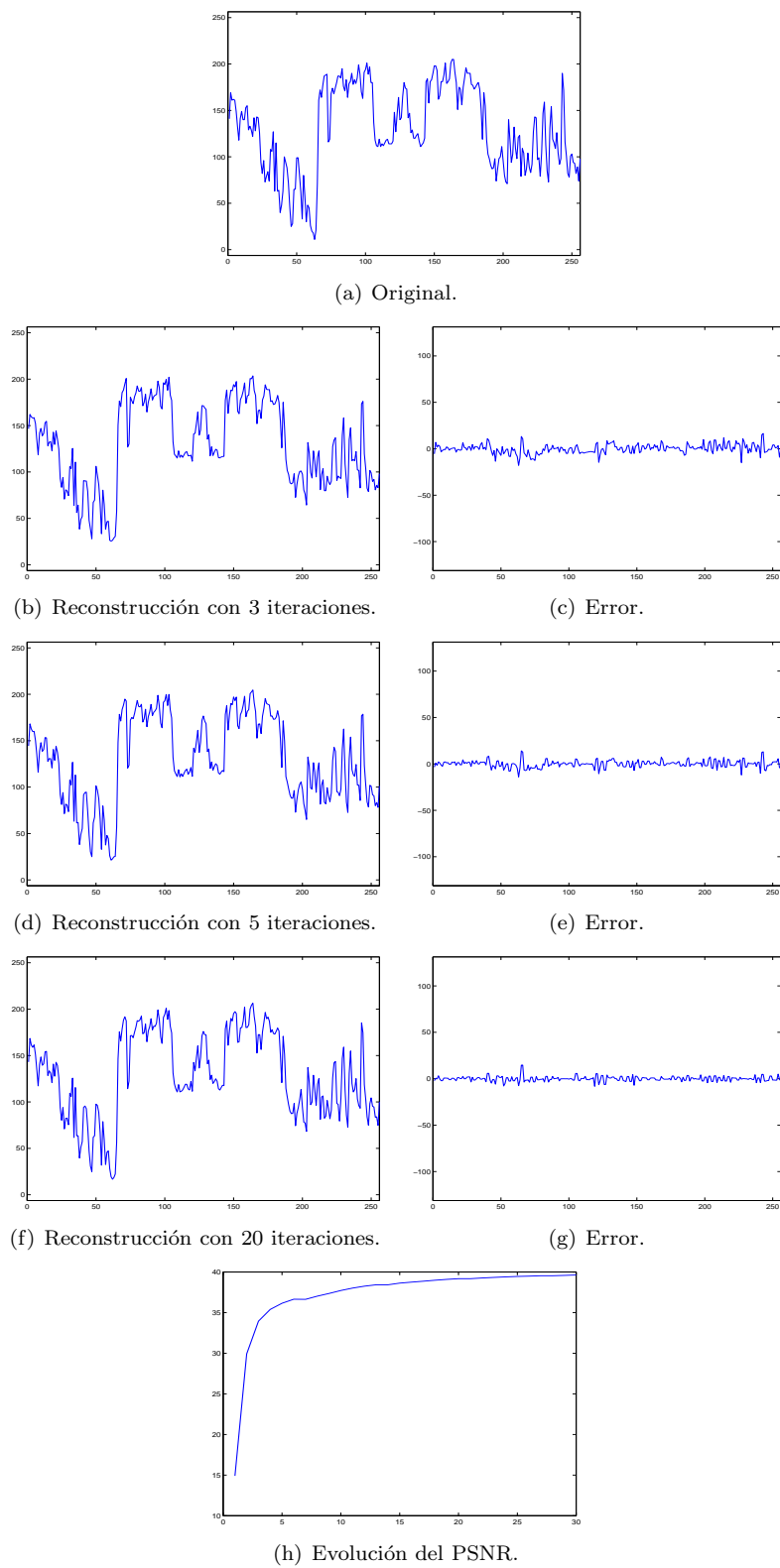


Figura 6.2: Reconstrucción sobre la fila central de Baboon.

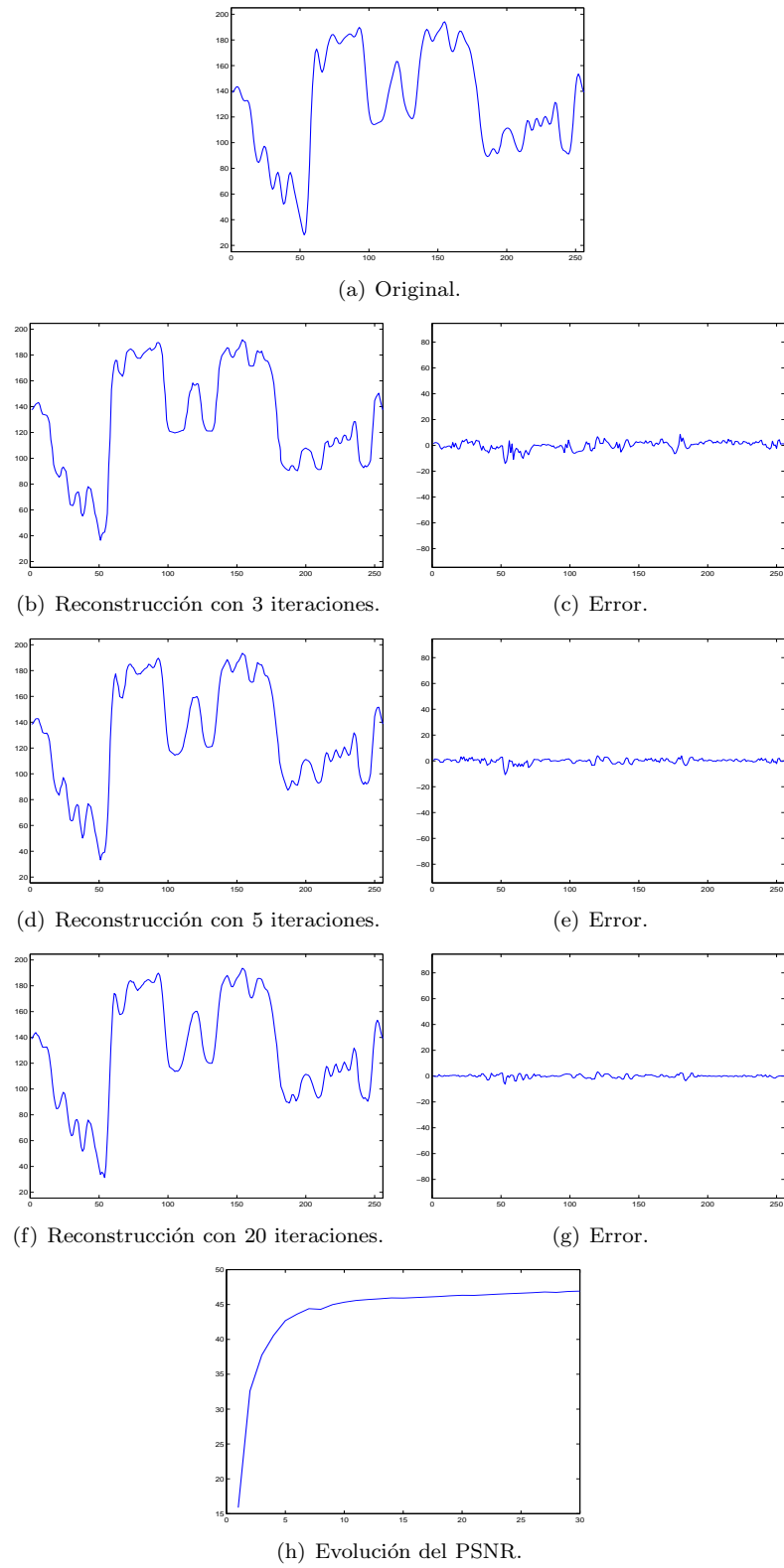
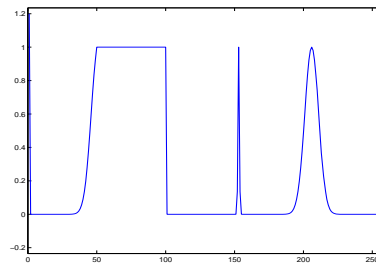
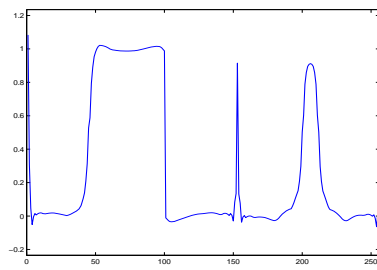


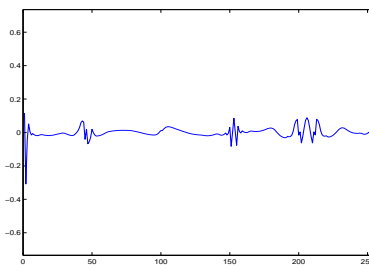
Figura 6.3: Reconstrucción sobre la fila central de Baboon suavizada.



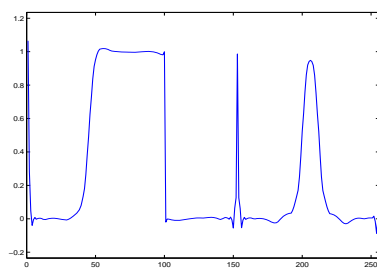
(a) Original.



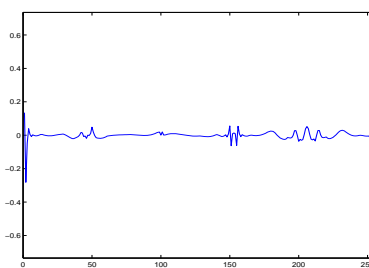
(b) Reconstrucción con 3 iteraciones.



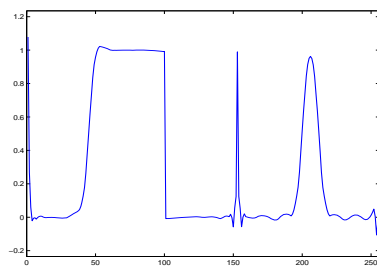
(c) Error.



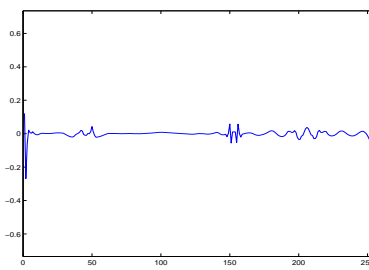
(d) Reconstrucción con 5 iteraciones.



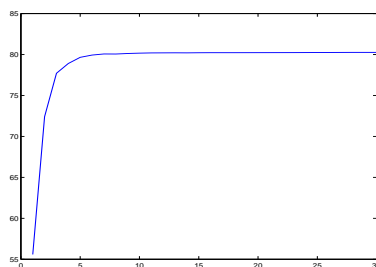
(e) Error.



(f) Reconstrucción con 20 iteraciones.



(g) Error.



(h) Evolución del PSNR.

Figura 6.4: Reconstrucción sobre una señal artificial.

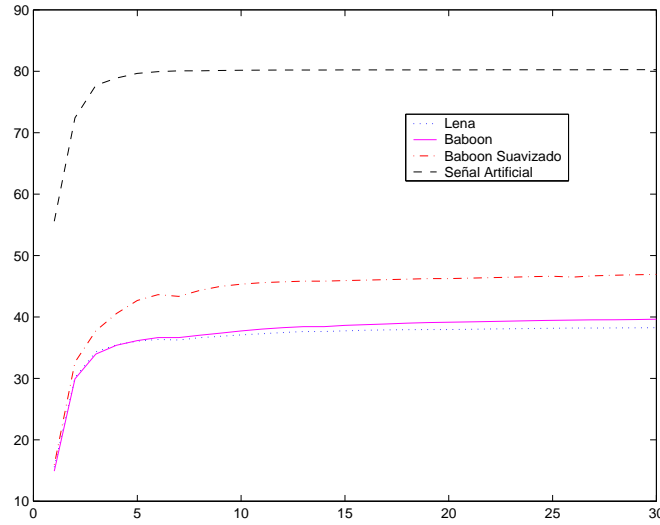


Figura 6.5: Comparación de las evoluciones de PSNR en los ejemplos de reconstrucción en 1D.

obtenemos una imagen aproximada a la original a la que llamaremos \tilde{I} .

De la misma forma que en la ecuación 6.13, en el caso bidimensional tendremos que:

$$\tilde{I} = L^{-1}LI \quad (6.14)$$

pero ahora los operadores de proyección L^{-1} y L se aplican sobre elementos bidimensionales.

Para calcular en forma eficiente el resultado de aplicar el operador L sobre una imagen I , también se utiliza un procedimiento similar al caso unidimensional:

- * Se calcula la transformada “à trous” de la imagen.

Sólo se calculan las matrices de detalles horizontales DH_j y verticales DV_j , ya que el algoritmo de reconstrucción no utiliza los detalles diagonales.

- * Se detectan las posiciones de los MM.

- * Se ponen en cero los valores de la transformada que no son MM.

- * Y por último se aplica la transformada inversa modificada.

En este caso como la transformada se compone de las matrices de detalles horizontales y verticales, nos queda:

$$\begin{aligned} \tilde{I}_j &= \tilde{I}_{j+1} \otimes_{filas} (h \uparrow 2^j) \otimes_{columnas} (h \uparrow 2^j) \\ &+ \tilde{D}H_{j+1} \otimes_{columnas} (g \uparrow 2^j) \\ &+ \tilde{D}V_{j+1} \otimes_{filas} (g \uparrow 2^j) \end{aligned}$$

- * Finalmente nos queda que $LI = \tilde{I}_0$.

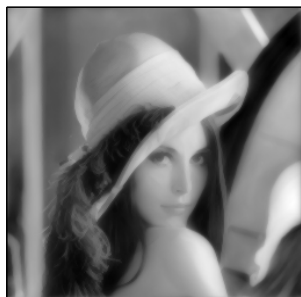
6.3.1. Ejemplos de reconstrucción en imágenes

De la misma forma que se presentaron algunos ejemplos de reconstrucción de señales utilizando la representación de MM, en esta sección se muestran algunos ejemplos sobre imágenes.

Se utilizaron cinco imágenes diferentes (figuras 6.6, 6.7, 6.8, 6.9 y 6.10). Para cada imagen se muestra, arriba el original, luego en las tres primeras filas la reconstrucción obtenida y el error entre ésta y el original luego de aplicar 5, 10 y 20 iteraciones del algoritmo de reconstrucción. Por último se muestra la evolución del PSNR entre el original y las reconstrucciones aplicando entre 1 y 80 iteraciones del algoritmo de reconstrucción.



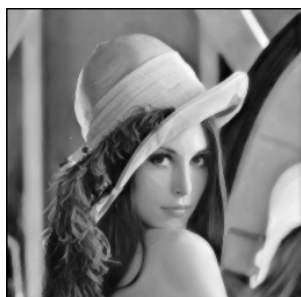
(a) Original.



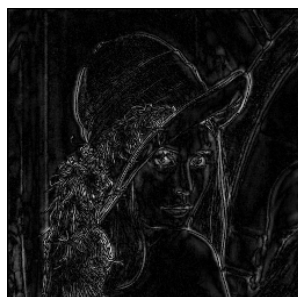
(b) Reconstrucción 5 it.



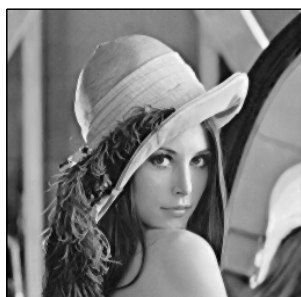
(c) Error.



(d) Reconstrucción 10 it.



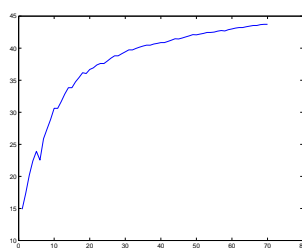
(e) Error.



(f) Reconstrucción 20 it.

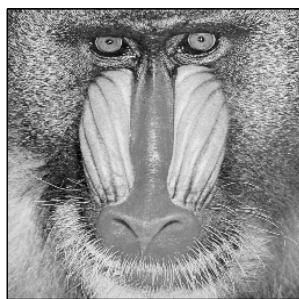


(g) Error.

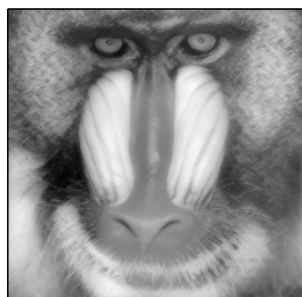


(h) Evolución del PSNR. Detección MM simple.

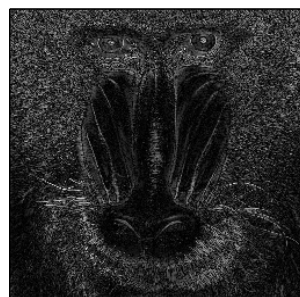
Figura 6.6: Reconstrucción sobre Lena.



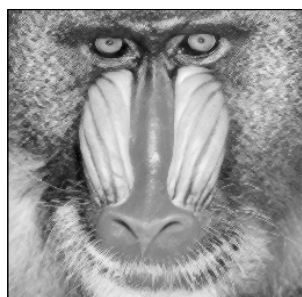
(a) Original.



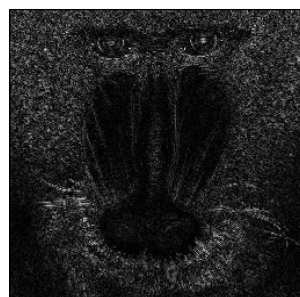
(b) Reconstrucción 5 it.



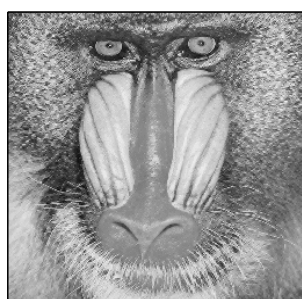
(c) Error.



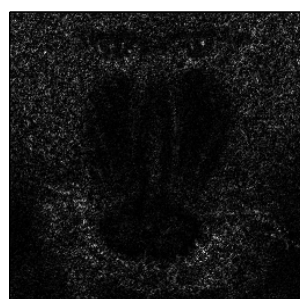
(d) Reconstrucción 10 it.



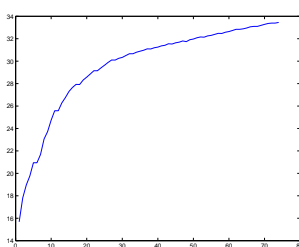
(e) Error.



(f) Reconstrucción 20 it.

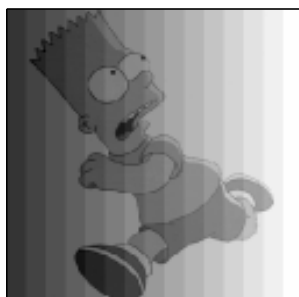


(g) Error.

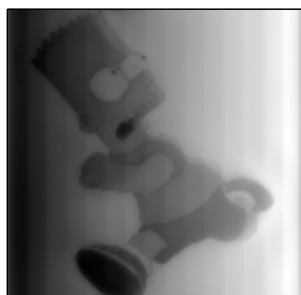


(h) Evolución del PSNR.

Figura 6.7: Reconstrucción sobre Baboon.



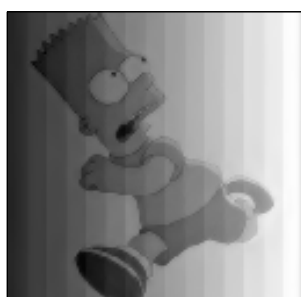
(a) Original.



(b) Reconstrucción 5 it.



(c) Error.



(d) Reconstrucción 10 it.



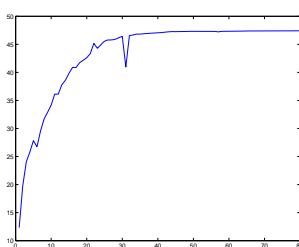
(e) Error.



(f) Reconstrucción 20 it.

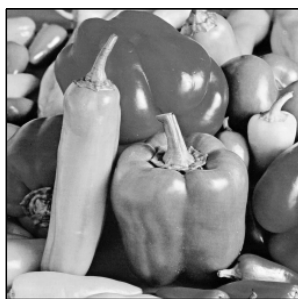


(g) Error.

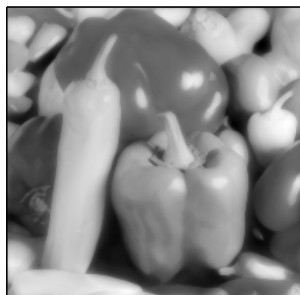


(h) Evolución del PSNR.

Figura 6.8: Reconstrucción sobre Bart.



(a) Original.



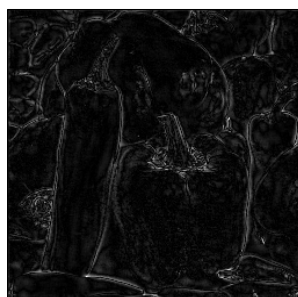
(b) Reconstrucción 5 it.



(c) Error.



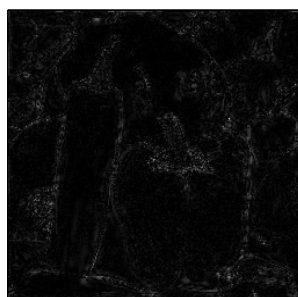
(d) Reconstrucción 10 it.



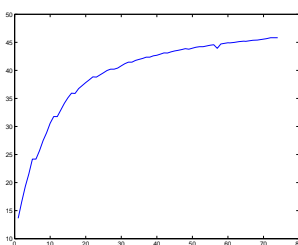
(e) Error.



(f) Reconstrucción 20 it.



(g) Error.



(h) Evolución del PSNR.

Figura 6.9: Reconstrucción sobre Peppers.



(a) Original.



(b) Reconstrucción 5 it.



(c) Error.



(d) Reconstrucción 10 it.



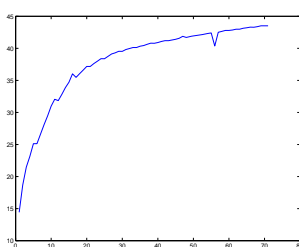
(e) Error.



(f) Reconstrucción 20 it.



(g) Error.



(h) Evolución del PSNR.

Figura 6.10: Reconstrucción sobre Village.

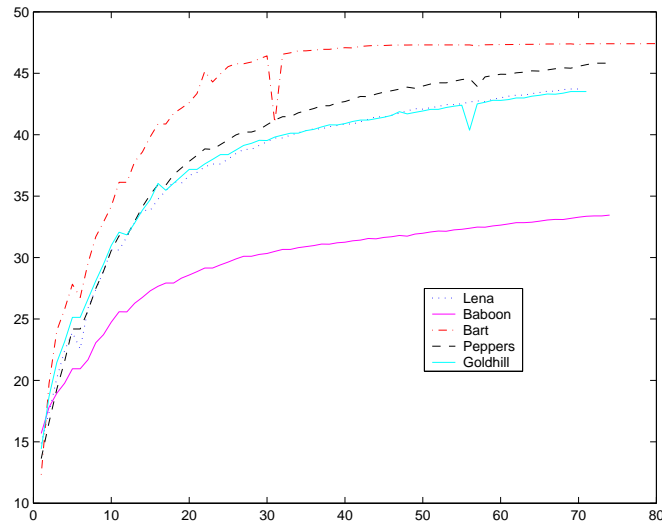


Figura 6.11: Comparación de las evoluciones de PSNR en los ejemplos de reconstrucción en 2D.

Escala	Figura 6.6	Figura 6.7	Figura 6.8	Figura 6.9	Figura 6.10
1	32.41 %	30.68 %	19.20 %	33.37 %	33.71 %
2	26.40 %	34.83 %	18.57 %	24.89 %	28.80 %
3	14.42 %	19.28 %	14.16 %	14.03 %	16.03 %
4	7.88 %	8.93 %	7.31 %	8.04 %	8.89 %
5	5.13 %	4.82 %	4.32 %	5.30 %	4.95 %
6	3.21 %	3.53 %	3.29 %	3.12 %	3.05 %
7	1.95 %	2.30 %	3.20 %	1.90 %	1.74 %
8	1.87 %	1.57 %		1.54 %	1.27 %
Total	11.66 %	13.24 %	10.01 %	11.52 %	12.31 %

Tabla 6.2: Porcentajes de MM detectados en los ejemplos de reconstrucción en 2D.

De la misma forma que en los ejemplos en una dimensión, se puede ver que la evolución de los PSNR es exponencial al principio, para luego tornarse asintótica aproximadamente a partir de la iteración número 25. Recordemos que en los ejemplos unidimensionales la evolución se hacía asintótica aproximadamente a partir de la iteración número 5, como en este caso estamos trabajando en dos dimensiones tiene sentido que se alcancen valores altos de PSNR cerca de la iteración $25 = 5^2$.

En la figura 6.11 se puede observar en forma comparativa la evolución de los cinco PSNR de las figuras anteriores. En este caso cabe aclarar que si bien los valores de PSNR obtenidos en la reconstrucción de Baboon no son muy altos, las diferencias con la imagen original se dan en los sectores en donde la imagen tiene frecuencias muy altas (ver imagen de error), y es justo en esos lugares en donde el ojo humano no distingue las diferencias.

De la misma forma que en los ejemplos unidimensionales, en todas estas reconstrucciones se utilizó la representación completa de MM. En el caso bidimensional, esto implica retener por cada escala de la transformada, todos los coeficientes de detalles horizontales y verticales que se ubican en las posiciones que son MM. En la tabla 6.2 se muestran las cantidades de MM detectados (en porcentaje) para cada figura, por escala y en total.

Capítulo 7

Conclusiones

En el presente trabajo se exploró la capacidad de la transformada wavelet como detector de singularidades a varias resoluciones.

Para alcanzar nuestros objetivos se investigaron varios temas, como la representación de señales unidimensionales, la obtención de distintas transformadas wavelet y sus propiedades, métodos de detección de módulos máximos y reconstrucción de imágenes a partir de sus singularidades entre otros.

Se utilizó exhaustivamente la transformada wavelet “à trous” y se estudiaron sus distintas propiedades: invarianza frente a los desplazamientos, similitudes y diferencias con la transformada tradicional y las ventajas que presenta en la extracción de singularidades.

Se estudiaron los problemas de centrado que presentan las distintas formas de obtención de esta transformada, y se propusieron métodos para corregirlos. De esta forma se obtuvo un método altamente eficiente para su obtención en forma completa, evitando el problema que genera el crecimiento exponencial de los filtros utilizados.

Se exploraron las capacidades que presenta esta transformada para detectar y caracterizar las singularidades de una imagen implementando algoritmos concretos. En este sentido, los resultados obtenidos por nuestros métodos son altamente satisfactorios, superando ampliamente nuestras expectativas. Logramos cumplir nuestro objetivo de detectar los bordes importantes en diferentes imágenes generando imágenes de bordes agradables a la vista y sin la utilización de umbrales manuales.

Se observó que esta transformada presenta varias ventajas para este fin, ya que se puede obtener en forma eficiente, con una muy buena localización y nos brinda un excelente punto intermedio entre la transformada tradicional y la continua.

Este compromiso (o punto intermedio) se nota principalmente en que la transformada tradicional se puede obtener en forma muy eficiente pero nos entrega poca información de la señal original (sobre todo en las escalas más gruesas), y no es invariante frente a los desplazamientos.

Por otro lado, la transformada continua nos brinda una gran cantidad de información acerca de la señal. Concretamente nos permite obtener información de la misma a cualquier escala intermedia y no sólo las diádicas. Como contraparte, la obtención de esta transformada es muy costosa.

Durante la realización del trabajo se extendió la obtención de la transformada wavelet “à trous” a imágenes, agregando la imagen de detalles diagonales. Esta modificación nos permitió desarrollar un método de reconstrucción sin pérdida de la imagen original a partir de los coeficientes de la transformada.

Se estudiaron también los métodos de detección de máximos locales en imágenes, generando en ese sentido un método original que permite obtener dichos máximos con mucha precisión teniendo en cuenta la dirección del vector de borde en cada píxel. Posteriormente se presentó un método que genera resultados muy parecidos al nuestro, pero cuya obtención es mucho más simple. Este último método es el que utilizan la mayoría de los métodos clásicos que realizan detección de máximos locales, como el de Canny. Lo importante del trabajo en este sentido es mostrar la relación entre ambos métodos, lo que nos permite afirmar que la utilización del método simplificado no compromete la buena localización de los módulos máximos.

En el capítulo dedicado a la reconstrucción de imágenes a partir de sus singularidades se estudió la

eficiencia del método propuesto por Mallat. Como resultado de este análisis se presentó un trabajo en el *5th Argentine Symposium on Computing Technology* (AST 2004), [23]. En él se describen posibles métodos de codificación de la información seleccionada por los detectores de bordes.

Los aportes originales del presente trabajo son:

- Comparación entre las distintas transformadas wavelet.
- Diferentes métodos de obtención de la transformada wavelet “à trous” con sus respectivos centrados.
- Un algoritmo de reconstrucción sin pérdida de la transformada wavelet “à trous” para imágenes.
- Un método de detección de módulos máximos en imágenes que brinda mejores resultados que el tradicional.
- Dos métodos concretos de detección de singularidades en señales e imágenes utilizando las diferentes resoluciones de la transformada wavelet “à trous”.
- La implementación y el estudio de la eficiencia del método de reconstrucción de señales e imágenes a partir de sus singularidades propuesto por Mallat [18].

Por otra parte, durante la investigación se realizaron varias implementaciones importantes, entre las que podemos nombrar la vectorización del algoritmo de detección de módulos máximos en imágenes y diferentes optimizaciones para la obtención completa de la transformada wavelet “à trous” en imágenes.

Quedan como trabajos futuros varios temas:

- Mejorar el desempeño de nuestros detectores de bordes para que se adapten a la presencia de distintos tipos de ruido.
- Extender nuestros métodos de detección de bordes a imágenes en color y a video.
- Generar un detector de contornos basado en los bordes obtenidos por nuestros métodos, utilizando además la información de dirección de bordes que entrega la transformada wavelet “à trous” sobre imágenes.
- Generar algoritmos de compresión eficientes basados en la codificación de las singularidades de una imagen.
- Desarrollar métodos de realce o atenuación de bordes en imágenes mediante la antitransformada “à trous” desarrollada.

Bibliografía

- [1] Giovanni Adorni, Stefano Cagnoni, Marcello Carletti, and Monica Mordonini. Omnidirectional vision algorithms in robotics, 2002.
- [2] A. Muñoz Barrutia, R. Ertlé, and M. Unser. Fast continuous wavelet transform based on B-splines. *Proceedings of the SPIE Conference on Mathematical Imaging: Wavelet Applications in Signal and Image Processing IX*, 4478:224–229, 2001.
- [3] John Canny. Finding edges and lines in images. Master’s thesis, MIT, 1983.
- [4] John Canny. A computational approach to edge detection. *PAMI*, Vol.8(6), pages 679–698, 1986.
- [5] William F. Fagan, Marie-Josée Fortin, and Candan Soykan. Integrating edge detection and dynamic modeling in quantitative analyses of ecological boundaries. *BioScience*, Vol.53 No.8, pages 730–738, 2003.
- [6] Guoliang Fan and Wai-Kuen Cham. Postprocessing of low bit-rate wavelet-based image coding using multiscale edge characterization. *IEEE Trans. Circuits Syst. Video Techn.*, 11(12):1263–1272, 2001.
- [7] Olivier Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, MA, 1993.
- [8] Marlin L. Gendron and Juliette W. Ioup. Wavelet multi-scale edge detection for extraction of geographic features to improve vector map databases. *Journal of Navigation*, January 2000.
- [9] Rafael C. González and Richard E. Woods. *Tratamiento digital de imágenes*. Addison-Wesley / Dias de Santos, 1996.
- [10] Amara Graps. An introduction to wavelets. *IEEE Computational Science and Engineering*, 2, 1995.
- [11] Forrest M. Hoffman. An introduction to Fourier theory. *Physics 641- Instrument Design and Signal Enhancement*, 1997.
- [12] M. Holschneider, R. Kronland-Martinet, J. Morlet, and A. Grossmann. A real-time algorithm for signal analysis with the help of the wavelet transform. In J. M. Combes, A. Grossmann, and Ph. Tchamitchian, editors, *Wavelets, Time-Frequency Methods and Phase Space*, pages 286–297. Springer-Verlag, 1989.
- [13] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. *Spoken Language Processing: A Guide to Theory Algorithm and System Development*. Prentice Hall, 2001.
- [14] L. Lam, S.-W. Lee, and C. Suen. Thinning methodologies: a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):869–885, 1992.
- [15] Stéphane Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693, 1989.
- [16] Stéphane Mallat. *A wavelet tour of signal processing (Second Edition)*. Academic Press, 1998.

- [17] Stéphane Mallat and Wen Liang Hwang. Singularity detection and processing with wavelets. *Courant Institute of Mathematical Sciences*, 1991.
- [18] Stéphane Mallat and Sifen Zhong. Characterization of signals from multiscale edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:710–732, 1992.
- [19] David Marr. *Vision*. W.H. Freeman and Co., 1982.
- [20] Y. Meyer. *Ondelettes et Operateurs*. Hermann Press, 1990.
- [21] D. Darian Muresan and Thomas W. Parks. Prediction of image detail. *The IEEE International Conference on Image Processing*, 2000.
- [22] Eric Neufeld, Haruna Popoola, David Callele, and David Mould. Mixed initiative interactive edge detection. *Graphics Interface*, 2003.
- [23] Joaquín Marcos J. Núñez Cortés and Ana María C. Ruedin. Compresión de imágenes usando los módulos máximos de la transformada wavelet “à trous”. *5th Argentine Symposium on Computing Technology (AST)*, 2004.
- [24] Jung-Me Park, Hui-Chuan Chen, and Shu T. Huang. A new gray level edge thinning method. Technical report, University of Alabama, Department of Computer Science, 2000.
- [25] Ana María C. Ruedin. *Multiwavelets no separables: construcción y aplicaciones para el procesamiento de imágenes*. PhD thesis, Universidad de Buenos Aires, 2002.
- [26] Mará Simon, Alvaro Pardo, and Federico Lecumberry. Codificación de video. *Escuela de Ciencias Informáticas (ECI)*, 2003.
- [27] P. Soille. *Morphological Image Analysis*. Springer-Verlag, Heidelberg, 2nd edition, 2003.
- [28] Gilbert Strang. Signal processing for everyone. *Computational Mathematics Driven by Industrial Problems*, Lecture Notes in Mathematics 1739, 2000.
- [29] Gilbert Strang and Truong Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1997.
- [30] Antonio Turiel and Angela del Pozo. Reconstructing images from edges. *Submitted to IEEE Transactions on Image Processing*, 2000.
- [31] Antonio Turiel and Nestor Parga. The multi-fractal structure of contrast changes in natural images: from sharp edges to textures. *Neural Computation*, 12:763, 2000.
- [32] Brani Vidaković and Peter Muller. *Wavelets for kids, a tutorial introduction*, 1991.
- [33] George Wolberg. Image morphing: a survey. *The Visual Computer*, 14:360–372, 1998.