



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

Reconstrucción de Señales a partir de una Representación Dispersa

T E S I S

QUE PARA OBTENER EL GRADO DE:
MAESTRO EN CIENCIAS
(COMPUTACIÓN)

PRESENTA:
MAZAY OSWALDO JIMÉNEZ SALINAS

DIRECTOR DE TESIS:
DR. FRANCISCO JAVIER GARCÍA UGALDE

MÉXICO, D.F.

2011



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Resumen

La mayoría de los esquemas de compresión de imágenes que existen hasta la fecha funcionan bajo el mismo principio: la imagen o señal se representa en una base apropiada y posteriormente se conservan y codifican los coeficientes de mayor peso o importancia energética. Tomando en cuenta el proceso de adquisición, se podría preguntar por qué se gastan tantos recursos de cómputo y de memoria para adquirir una información, que posteriormente, en una buena parte, será desechada para poder ser comprimida y almacenada. O ¿Existirá alguna forma mejor de adquirir y comprimir la información? La respuesta afirmativa es reciente y se llama muestreo compresivo. El muestreo compresivo es un área emergente en el procesamiento de señales, cuya principal motivación es adquirir y comprimir la información al mismo tiempo. Este esquema se divide en dos etapas, una es la adquisición de la información y otra es la reconstrucción de la misma.

El objetivo principal de este trabajo es implementar un esquema de muestreo compresivo en software y evaluar su desempeño en la compresión de imágenes. Para evaluar el desempeño, primero comparamos el esquema de muestreo compresivo con el algoritmo *embedded zerotree wavelet*, un algoritmo de codificación progresiva que permite agregar tantos bits como uno quiera para obtener la aproximación objetivo. Posteriormente, se hacen algunas simulaciones con el algoritmo de reconstrucción para constatar que el algoritmo sea robusto al ruido. Finalmente, la última parte de la tesis consiste en aplicar el muestreo compresivo para reconstruir imágenes formadas por elipses ó también denominadas como “fantasmas”. Los resultados obtenidos se comparan con dos métodos de reconstrucción, uno mediante retroproyecciones y el otro mediante retroproyecciones filtradas.

Contenido

Resumen	iii
Notación	xiii
1 Introducción	1
1.1 Motivación	1
1.2 Bases teóricas	4
1.3 Aplicaciones	7
1.3.1 Compresión de imágenes	7
1.3.2 Cifrado de la información	7
1.3.3 Señales de Radar	8
1.4 Organización de la tesis	8
2 Muestreo Compresivo	9
2.1 Adquisición de la información	9
2.2 Reconstrucción de la Información	13
2.2.1 Basis Pursuit	14
2.2.2 Orthogonal Matching Pursuit	16
2.2.3 Compressive Sampling Matching Pursuit	18
2.2.4 Principales Teoremas	19
2.3 Resumen	21
3 Otras Técnicas de Compresión y Reconstrucción de Imágenes	25
3.1 Wavelets	25
3.1.1 Transformada Discreta wavelet	25
3.1.2 Transformada Discreta wavelet en 2-D	26
3.2 EZW	29
3.3 Otras Técnicas de Reconstrucción de Imágenes	34
3.3.1 Transformada Radon	34
3.3.2 Reconstrucción utilizando Retroproyecciones Filtradas	36
3.4 Resumen	37

4 Simulaciones y resultados	39
4.1 Resultados de las simulaciones entre el CS y EZW	40
4.2 Resultados de las simulaciones entre el CS y retroproyecciones . .	46
4.3 Resumen	48
5 Conclusiones y trabajo futuro	51
5.1 Trabajo a Futuro	52
Bibliografía	53
Lista de Acrónimos	57
Índice alfabético	59

Índice de Figuras

1.1	Aproximación utilizando la wavelet Haar y cinco niveles de descomposición. Las señales aproximadas fueron reconstruidas a partir de $M = 80$ coeficientes wavelet que son los primeros coeficientes ó los coeficientes más significativos. Para el caso del muestreo compresivo la señal fue reconstruida con $3M = 240$ mediciones lineales. La señal es perfectamente reconstruida para el caso de la aproximación no lineal y la aproximación con muestreo compresivo. El error E es medido como la raíz de la suma de las diferencias de las muestras al cuadrado.	3
1.2	Ejemplos de señal dispersa (a), y compresible (b).	5
2.1	Comparación de las soluciones mínimas ℓ_1 y ℓ_2	16
3.1	Proceso de descomposición	26
3.2	Proceso de reconstrucción	26
3.3	Ejemplo codificación de los coeficiente <i>wavelet</i> por subbandas. . .	27
3.4	a) Descomposición <i>wavelet</i> en un nivel. b) Descomposición <i>wavelet</i> en tres niveles.	28
3.5	Ejemplo descomposición Wavelet en tres niveles.	29
3.6	a) Orden de búsqueda zigzag para la transformada <i>wavelet</i> , con dos <i>quadtrees</i> . b) Clasificación de un coeficiente.	31
3.7	Etapas de EZW. a) Transformada <i>wavelet</i> con tres niveles de descomposición. b) Etapa 1, umbral=32. c) Etapa 2, umbral=16. d) Cuantización al final de la etapa 2.	33
3.8	Geometría de un haz de rayos paralelos.	34
4.1	Imagen de prueba “man”.	40
4.2	Comparación de métodos de compresión con diferentes funciones <i>wavelet</i> . a) Gráfica del nivel de compresión .vs. PSNR, (muestreo compresivo). b) Gráfica del nivel de compresión vs PSNR, (método EZW).	41
4.3	Imágenes reconstruidas con un factor de compresión 4:1 utilizando el esquema de muestreo compresivo.	43

4.4	Imágenes reconstruidas con un factor de compresión 4:1 utilizando el algoritmo EZW.	44
4.5	Imágenes reconstruidas a partir de mediciones contaminadas con ruido Gaussiano.	45
4.6	Imágenes de prueba. a) Imagen de prueba conocido como fantasma Logan-Shepp. b) Imagen de prueba formada por 10 elipses. c) Dominio de muestreo en plano de frecuencia; coeficientes de Fourier son recolectados a lo largo de aproximadamente 22 líneas radiales	47
4.7	Imágenes de Reconstruidas. a) Imagen original fantasma Shepp-Logan. b) Reconstrucción con CS a partir de 22 líneas radiales, PSNR = 48.89[dB]. c) Reconstrucción con BP a partir de 22 proyecciones, PSNR = 4.78 [dB]. d) Reconstrucción con FBP a partir de 22 proyecciones, PSNR = 14.42 [dB]. e) Reconstrucción con BP a partir de 360 proyecciones, PSNR = 4.78 [dB]. f) Reconstrucción con FBP a partir de 360 proyecciones, PSNR = 20.89 [dB]. . . .	49
4.8	Imágenes de Reconstruidas. a) Imagen original formada por diez elipses. b) Reconstrucción con CS a partir de 22 líneas radiales, PSNR = 47.69[dB]. c) Reconstrucción con BP a partir de 22 proyecciones, PSNR = 7.06 [dB]. d) Reconstrucción con FBP a partir de 22 proyecciones, PSNR = 13.86 [dB]. e) Reconstrucción con BP a partir de 360 proyecciones, PSNR = 7.02 [dB]. f) Reconstrucción con FBP a partir de 360 proyecciones, PSNR = 21.26 [dB]. . . .	50

Índice de Tablas

2.1	Análisis de complejidad del algoritmo CoSaMP. Las dimensiones de la matriz de muestreo son $m \times n$, el nivel de dispersidad es d y el número \mathcal{M} acota el costo de la multiplicación matriz-vector con Φ o Φ^*	20
2.2	Comparación de algunos algoritmos de recuperación. La notación d se refiere al nivel de dispersidad; m se refiere al número de mediciones; n se refiere al tamaño de la señal.	22
4.1	PSNR de la imágenes reconstruida a partir de muestras contaminadas.	46

Índice de Algoritmos

2.2.1 CoSaMP Algoritmo de Recuperación	19
3.2.2 Codificación bit-plane	30

Notación

$\langle f, g \rangle$	Producto interno.
$\ f\ _2$	Norma Euclidiana.
$\ f\ _1$	Norma L_1 o ℓ_1 .
$\ f\ _\infty$	Norma L_∞ o ℓ_∞ .
$f[n] = O(g[n])$	Orden de: existe k tal que $f[n] \leq k \cdot g[n]$.
$A \ll B$	A es mucho menor que B .

Conjuntos

\mathbb{N}	Números naturales.
\mathbb{Z}	Números enteros.
\mathbb{R}	Números reales.
\mathbb{C}	Números complejos.
$ \Lambda $	Número de elementos en el conjunto Λ .

Señales

$f(t)$	Señal continua.
$f[n]$	Señal discreta.
$\delta(t)$	Distribución Dirac.
$\delta[n]$	Dirac discreta.

Transformadas

$\hat{f}(w)$	Transformada de Fourier.
--------------	--------------------------

Probabilidad

$\mathbb{E}[\cdot]$	Valor esperado.
---------------------	-----------------

Capítulo 1

Introducción

1.1 Motivación

El principio bajo el cual trabajan la mayoría de los algoritmos de compresión, consiste en representar una señal en una base apropiada y posteriormente codificar los coeficientes de mayor peso o importancia energética en la expansión. La eficiencia de estos algoritmos radica principalmente en encontrar una “buena transformación”, es decir, si la transformada provee una buena compactación de la energía de la señal el algoritmo de compresión mejora. Este problema ya ha sido estudiado ampliamente no sólo desde el punto de vista teórico [1], sino también desde el punto de vista práctico [2].

Durante la década de los noventa, los algoritmos de compresión de señales fueron revolucionados por la introducción de las *wavelets*, una representación que desplazó a la *transformada discreta coseno* [3] ya que proveía una mejor representación de fenómenos no estacionarios y al mismo tiempo un análisis multiresolución. Tal fue el impacto de las *wavelets* en el área de compresión que esta es la principal diferencia entre los estándares de compresión JPEG [4] y JPEG2000 [5].

El área del análisis armónico computacional ha permanecido muy activa, a finales de los noventa y principios del dos mil surgieron varias representaciones orientadas geométricamente, entre las más comunes se encuentran las *curvelets* [6], *contourlets* [7], *bandelets* [8], etc. Al conjunto de estas representaciones se les refiere genéricamente como las “X-lets”. Estas representaciones fueron diseñadas como una sustitución de las *wavelets* para representar objetos en dos dimensiones. En el trasfondo con el mismo objetivo de encontrar una “buena transformación”.

El esquema convencional de adquirir la información y después comprimirla fue cuestionado por David Donoho [9]. Tomando en cuenta el proceso de adquisición de las señales, uno se podría preguntar ¿por qué se gastan tantos recursos de computo y de memoria para adquirir información?, que posteriormente, en una buena parte, será desechada para poder ser comprimida y almacenada. O ¿Existirá alguna forma de adquirir y comprimir la información al mismo tiempo?

La respuesta afirmativa es reciente y se llama muestreo compresivo. El muestreo compresivo (CS), o también conocido como sensado compresivo es un paradigma de muestreo que en cierta medida va en contra de los criterios tradicionales de adquisición de datos.

Este paradigma fue desarrollado desde los años setenta, pero permaneció ignorado hasta el 2004, y a partir de entonces ha cobrado gran importancia entre la comunidad científica, gracias a los resultados obtenidos por Emmanuel Candés y David Donoho [10, 11, 12, 13, 14]. La teoría de muestreo compresivo afirma que se pueden recuperar ciertas señales e imágenes con un número mucho menor de muestras que las que utilizan los métodos tradicionales actualmente utilizados. La motivación detrás del muestreo compresivo es hacer el “muestreo” y “compresión” al mismo tiempo.

En muchas aplicaciones, a uno le gustaría aproximar una señal a partir de su base de expansión [15], es decir, una señal \mathbf{x} representada por una combinación lineal de señales elementales $\{\psi_n\}_{n \in \mathbb{N}}$. Una forma de hacer esto es mantener los *primeros* M componentes en la expansión. Esto es referido como una *aproximación lineal* (AL) ya que es equivalente a proyectar la señal de entrada sobre un subespacio lineal fijo generado con los primeros M elementos de la base. La aproximación lineal puede ser expresada como

$$\hat{\mathbf{x}}_M^{(AL)} = \sum_{n=0}^{M-1} c_n \psi_n \quad \text{donde} \quad c_n = \langle \mathbf{x}, \psi_n \rangle. \quad (1.1.1)$$

Sin embargo, la reconstrucción con los primeros M elementos por lo general no es muy buena ya que estos coeficientes no siempre capturan las principales características del objeto. Un mejor esquema de aproximación consiste en mantener los *mejores* M componentes en la expansión. Esto es referido como una *aproximación no lineal* (ANL) ya que la señal de entrada es proyectada sobre un subespacio lineal adaptado a esta. En general, la aproximación no lineal puede ser expresada como

$$\hat{\mathbf{x}}_M^{(ANL)} = \sum_{n \in I_M}^{M-1} c_n \psi_n. \quad (1.1.2)$$

donde I_M es el conjunto de los M índices correspondiente a los M coeficientes más significativos $|c_n|$.

A modo de ejemplo, consideremos la aproximación de una señal unidimensional *suave por tramos* con $N = 1024$ muestras usando tanto el esquema de aproximación lineal como el de aproximación no lineal [16] y el esquema de reconstrucción con muestreo compresivo.

En los casos de aproximación lineal y no lineal sólo $M = 80$ coeficientes de la base de expansión fueron retenidos. En el caso de la reconstrucción con muestreo compresivo se tomaron $3M = 240$ mediciones lineales para obtener una aproximación con el mismo número de coeficientes que las dos anteriores ¹. La

¹En el capítulo 2 se explicará a detalle como se llevan a cabo estas mediciones lineales y también el esquema de reconstrucción de la señal.

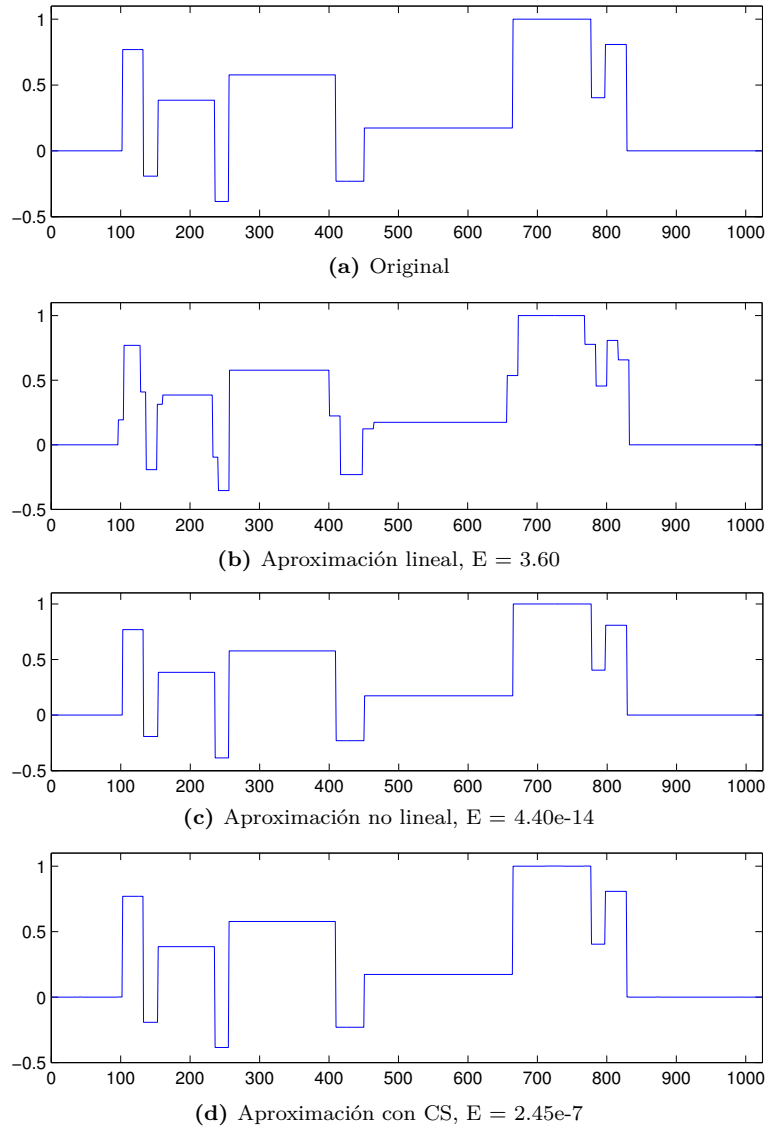


Figura 1.1: Aproximación utilizando la wavelet Haar y cinco niveles de descomposición. Las señales aproximadas fueron reconstruidas a partir de $M = 80$ coeficientes wavelet que son los primeros coeficientes ó los coeficientes más significativos. Para el caso del muestreo compresivo la señal fue reconstruida con $3M = 240$ mediciones lineales. La señal es perfectamente reconstruida para el caso de la aproximación no lineal y la aproximación con muestreo compresivo. El error E es medido como la raíz de la suma de las diferencias de las muestras al cuadrado.

figura 1.1 muestra los resultados obtenidos utilizando las bases wavelet de Haar. En este caso podemos observar que la reconstrucción lineal es pobre ya que los primeros elementos en la expansión contiene más elementos de baja frecuencia que de alta frecuencia (*cambios bruscos ó discontinuidades*), por lo tanto la aproximación resulta con ciertas discontinuidades producto de la descomposición. La aproximación no lineal resulta en una aproximación casi perfecta ya que los coeficientes más significativos incluyen coeficientes tanto de baja frecuencia como de alta frecuencia. La aproximación con el muestreo compresivo da un resultado muy similar a la aproximación con los coeficientes más significativos, utilizando alrededor de $3M$ mediciones lineales incoherentes [17, 18].

Los experimentos con las señales 1-D, indican que la eficiencia del muestreo compresivo es muy buena ya que a partir de una representación *dispersa*, el algoritmo de reconstrucción encuentra una aproximación con un error muy cercano al de la aproximación no lineal. Esto también indica que las ventajas del muestreo compresivo parecen ser muchas y actuales, respecto a los esquemas convencionales de adquisición de la información, por lo que se busca explorar sus ventajas en el procesamiento de imágenes.

1.2 Bases teóricas

Antes de continuar, es importante definir algunos conceptos que son de gran importancia en el muestreo compresivo. Se dice que una señal \mathbf{x} de dimensión n es d -dispersa si esta tiene d o menos componentes distintos de cero,

$$\mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\|_0 := |\text{supp}(\mathbf{x})| = |\{j : x_j \neq 0\}| \leq d \ll n$$

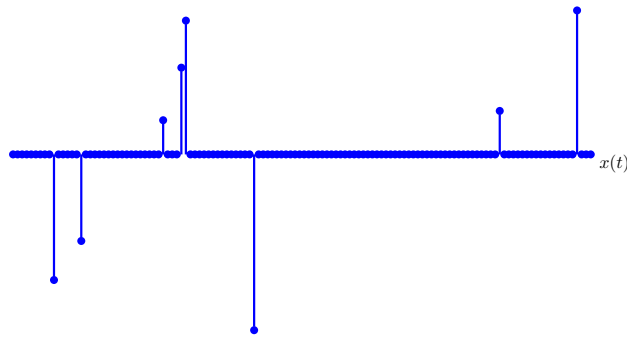
donde $\text{supp}(\mathbf{x})$ denota la cardinalidad del conjunto soporte de \mathbf{x} , y $\|\cdot\|_0$ es una cuasinorma. Para $1 \leq p < \infty$, la norma p denotada como $\|\cdot\|_p$ se define como

$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}$$

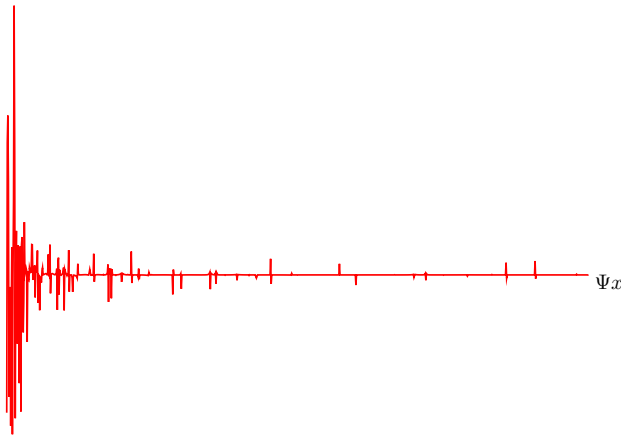
para el caso cuando $p = \infty$ tenemos que $\|\mathbf{x}\|_\infty = \max |x_i|$, donde x_i representa el i -ésimo componente del vector. Las señales dispersas son una idealización que no encontramos en la práctica, sin embargo es común encontrar señales que son *compresibles*, esto significa que sus elementos decaen rápidamente cuando son ordenados por magnitud. También se puede decir que una señal es compresible con respecto a otras bases ortogonales, por ejemplo Fourier ó wavelets. Sea q un número en el intervalo $(0,1)$. Se dice que \mathbf{x} es q compresible con magnitud C si los elementos ordenados de la señal decaen a una tasa

$$|x|_{(i)} \leq C \cdot i^{(-1/q)} \quad \text{para } i = 1, 2, 3, \dots \quad (1.2.1)$$

Las señales dispersas son un caso especial de las señales compresibles. La figura 1.2a muestra un ejemplo de una señal dispersa y la figura 1.2b muestra un ejemplo de un señal compresible en el dominio wavelet.



(a) Señal dispersa



(b) Señal Compresible

Figura 1.2: Ejemplos de señal dispersa (a), y compresible (b).

Otro concepto que se utilizará más adelante es el de la restricción de una señal. Supongamos que \mathbf{x} es una señal en \mathbb{R}^n y r es una constante positiva. Escribimos \mathbf{x}_r para indicar que \mathbf{x} está restringida a sus r componentes más grandes en magnitud. Los empates se rompen lexicográficamente. Además esta señal es la mejor aproximación r -dispersa de \mathbf{x} con respecto a cualquier norma ℓ_p . Ahora supongamos que T es un subconjunto de $\{1, 2, \dots, n\}$. Definimos la restricción de la señal al conjunto T como

$$\mathbf{x}|_T = \begin{cases} x_i, & i \in T \\ 0, & \text{otro caso.} \end{cases}$$

Escribimos Φ_T para indicar la submatriz de Φ cuyas columnas están listadas en

el conjunto T . Y la pseudoinversa de una matriz alta, de rango completo Φ la definimos por la siguiente formula $\Phi^\dagger = (\Phi^* \Phi)^{-1} \Phi^*$.

En capítulos posteriores utilizaremos algunos programas de optimización, por tanto será necesario describirlos. Un problema de optimización tiene la forma

$$\begin{aligned} \text{minimizar} \quad & f_0(\mathbf{x}) \\ \text{sujeto a} \quad & f_i \leq b_i, \quad \text{para } i = 1, \dots, m \end{aligned} \quad (1.2.2)$$

el vector \mathbf{x} es la *variable de optimización* del problema, la función $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$ es la *función objetivo*, las funciones $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ son las *funciones de restricción* y las constantes b_i son los límites o cotas, para las restricciones. Un vector \mathbf{x}^* es llamado la *solución* del problema (1.2.2), si tiene el menor valor objetivo entre todos los vectores que satisfacen las restricciones.

Una de las subclases de optimización ampliamente conocida es la de mínimos cuadrados. Un problema de *mínimos cuadrados* es un problema de optimización sin restricciones y un objetivo, el cual es la suma del cuadrado de los términos de la forma $a_i^T \mathbf{x} - b_i$:

$$\text{minimizar} \quad f_0(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|_2^2 = \sum_{i=1}^k (a_i^T \mathbf{x} - b_i)^2. \quad (1.2.3)$$

donde $\mathbf{A} \in \mathbb{R}^{k \times n}$ con ($k \geq n$), a_i^T son los renglones de \mathbf{A} , y el vector $\mathbf{x} \in \mathbb{R}^n$ es la variable de optimización.

La solución al problema de mínimos cuadrados (1.2.3) puede ser reducido a resolver el sistema de ecuaciones lineales,

$$(\mathbf{A}^T \mathbf{A})\mathbf{x} = \mathbf{A}^T \mathbf{b}$$

entonces se tiene la solución analítica $\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$. Este problema puede ser resuelto en un tiempo aproximado a $n^2 k$, con una constante conocida. Una computadora de escritorio puede resolver un problema de estos con cientos de variables, en pocos segundos [19].

Otro problema de optimización muy utilizado es de la minimización la norma ℓ_1 :

$$\text{minimizar} \quad \|\mathbf{x}\|_1 \quad \text{sujeto a} \quad \mathbf{Ax} = \mathbf{b} \quad (1.2.4)$$

este problema tiende a producir una solución \mathbf{x} con un gran número de componentes iguales a cero. En otras palabras, el problema de mínima norma ℓ_1 tiende a producir soluciones *dispersas* de $\mathbf{Ax} = \mathbf{b}$, usualmente con m componentes distintas de cero. El problema de encontrar la menor \mathbf{x} con m componentes distintas de cero en general requiere examinar y comparar todas las $n!/(m!(n-m)!)$ opciones de los m coeficientes distintos de cero de los n coeficientes en \mathbf{x} . Resolver el problema de la mínima norma ℓ_1 , por otro lado, da una buena heurística para encontrar una solución pequeña y dispersa de $\mathbf{Ax} = \mathbf{b}$.

1.3 Aplicaciones

El área del muestreo compresivo es muy reciente, sin embargo, ha traído muchos avances al estado del arte de muchas aplicaciones tales como el procesamiento de imágenes, imágenes de resonancia magnética (MRI), aplicaciones de radar, corrección de errores y comunicaciones. Es muy amplio el campo de aplicación que uno recomienda estudiar [20, 14] para más detalles.

A continuación, se listan algunos ejemplos:

1.3.1 Compresión de imágenes

Muchas imágenes de escenas naturales tienen una representación dispersa cuando son representadas en alguna base. Hoy en día las cámaras digitales capturan las imágenes con un sensor para cada píxel y cada píxel es adquirido antes de ser comprimido, en otras palabras, adquieren una gran cantidad de información que después en su mayor parte es deshechada. Gracias al avance de los circuitos integrados y del silicio, las cámaras de hoy en día pueden adquirir imágenes de varios megapíxeles utilizando este esquema. Como se mencionó anteriormente, la principal motivación del muestreo compresivo es muestrear y comprimir al mismo tiempo y de esta forma evitar etapas superfluas.

Una arquitectura que utilice muestreo compresivo puede sonar muy alentador porque se podrían reducir el número de etapas en el procesamiento pero también trae consigo grandes retos. Varios investigadores [21] están trabajando en un prototipo de una cámara de un solo píxel, ésta consiste a grosso modo de un microespejo digital (DMD), dos lentes, un detector de fotones y un convertidor analógico digital (A/D).

También en el área de imágenes médicas, especialmente las imágenes de resonancia magnética que tienen una representación dispersa en su representación espacial, por lo que el muestreo compresivo es adecuado para reconstruir una imagen de esta naturaleza a partir de muestras aleatorias de coeficientes de Fourier [10].

1.3.2 Cifrado de la información

El muestreo compresivo también tiene impacto en el área de cifrado. En la práctica cuando uno transmite información de un punto a otro esta sufre cierta distorsión debido al medio o canal de transmisión, estos errores suelen ocurrir en ráfagas corrompiendo sólo algunas partes de la información. El problema del cifrado de la información puede ser formulado como sigue. Supongamos que tenemos un vector $\mathbf{f} \in \mathbb{R}^n$, el mensaje plano, que deseamos transmitir de forma segura. Después se transmiten las mediciones $\mathbf{y} = \mathbf{A}\mathbf{f}$, del mensaje cifrado, donde \mathbf{A} es la *matriz de medición* o *matriz de codificación*. Posteriormente este mensaje cifrado es distorsionado por un vector disperso de error \mathbf{e} , $\mathbf{y}' = \mathbf{A}\mathbf{f} + \mathbf{e}$. En el caso de que la información no haya sido alterada, el mensaje puede ser reconstruido siempre y cuando \mathbf{A} sea de rango completo. En el caso que la información haya sido alterada por un vector de errores \mathbf{e} , uno puede considerar

una matriz B cuyo kernel sea el rango de A . Aplicando a ambos miembros de la expresión $\mathbf{y}' = A\mathbf{f} + \mathbf{e}$, obtenemos $B\mathbf{y}' = B\mathbf{e}$. Por lo tanto, el problema de codificación se reduce a recuperar el vector de error \mathbf{e} de las observaciones $B\mathbf{e}$. Una vez que es conocido \mathbf{e} , $A\mathbf{f}$ también, y finalmente \mathbf{f} también es conocido ya que A tiene rango completo. Para más detalles uno puede consultar [22, 23]. En la teoría de codificación uno asume que la información pertenece a un campo finito, en la práctica 0 o 1, en el caso del CS este asume que la codificación se lleva a cabo sobre los números reales. Varios investigadores ya se encuentran estudiando este problema [24].

1.3.3 Señales de Radar

Un sistema convencional de radar transmite un pulso, y posteriormente utiliza un filtro acoplado para correlacionar la señal recibida con el pulso. El receptor entonces utiliza un sistema de compresión de pulsos junto con un convertidor A/D de alta velocidad para procesar la señal. Este enfoque no sólo es complicado y costoso, sino que además la resolución de la señal esta limitada por el principio de incertidumbre de Hiesenberg asociado al radar. EL CS ataca estos problemas discretizando el plano tiempo frecuencia dentro de una rejilla y considerando cada posible objetivo como una matriz. Si el número de objetivos es suficientemente pequeño, entonces la rejilla será dispersa, y las técnicas de muestreo compresivo pueden ser usadas para recuperar la señal [25].

1.4 Organización de la tesis

La tesis esta organizada como sigue. En el siguiente capítulo, nos concentramos en establecer los principios del muestreo compresivo. Las formas de adquisición de la información utilizando este esquema y los algoritmos más utilizados para la reconstrucción de señales dispersas. En el capítulo 3 se presentan algunas técnicas usualmente utilizadas para la compresión y la reconstrucción de imágenes. Se explica a detalle el esquema EZW, un algoritmo utilizado para la compresión de imágenes en el dominio wavelet, su principal ventaja es que utiliza codificación progresiva por lo que uno puede agregar o quitar bits para alcanzar una aproximación deseada. El esquema de reconstrucción estudiado en este capítulo es el de retroproyección a partir de rayos X, se analiza sus ventajas y desventajas de la versión filtrada como la no filtrada. En el capítulo 4 se hace la comparación de la compresión de imágenes utilizando el esquema de muestreo compresivo vs EZW. También se realizan algunos experimentos para validar la robustez del algoritmo CoSaMP al ruido. En la segunda parte de este capítulo se hace una comparación de la reconstrucción de imágenes tipo “fantasmas” con muestreo compresivo y con la transformada Radon.

Finalmente, concluimos con el capítulo 5 con un resumen y algunas observaciones para trabajo futuro.

Capítulo 2

Muestreo Compresivo

2.1 Adquisición de la información

El muestreo compresivo se realiza en dos etapas. La primera es la adquisición de la información y la segunda es la reconstrucción de la misma. Aquí el *muestreo o sensado* se lleva a cabo mediante mediciones lineales, es decir, una muestra es una función lineal aplicada a la señal. El *operador de información* tiene la siguiente forma:

$$y_k = \langle \mathbf{x}, \varphi_k \rangle, \quad k \in M. \quad (2.1.1)$$

donde \mathbf{x} es la señal de interés de dimensión n , $M \subset \{1, \dots, n\}$ es un subconjunto de cardinalidad $m \ll n$ y las φ_k son las funciones de muestreo, estas funciones de muestreo no necesariamente se adaptan a la estructura de la señal, es decir, son fijas independientemente de \mathbf{x} . El proceso de obtener múltiples muestras puede ser visto como el proceso de aplicar la *matriz de muestreo* Φ a la señal de interés. Si uno toma m muestras, o mediciones, de una señal en \mathbb{R}^n , entonces la matriz Φ tiene dimensiones $m \times n$. La pregunta natural que surgiría sería, ¿Cuál es el número necesario de mediciones para adquirir señales d -dispersas? Esta pregunta puede ser respondida principalmente desde dos enfoques. El primero es mediante la *coherencia mutua* [9, 26] y el segundo mediante la *propiedad de isometría restringida* (RIP) [24].

Analicémoslo primero desde el punto de vista de la coherencia. La coherencia entre la base de sensado Φ y la base de representación Ψ se define como

$$\mu(\Phi, \Psi) = \sqrt{n} \max_{1 \leq k, j \leq n} |\langle \varphi_k, \psi_j \rangle|. \quad (2.1.2)$$

La coherencia mide la mayor similitud entre cualquiera de dos elementos de las bases Φ y Ψ . Si Φ y Ψ contienen elementos correlacionados, la coherencia es grande. En caso contrario es pequeña. La coherencia puede tener valores entre 1 y \sqrt{n} , véase [27]. Para ver como funciona esta medida tomemos los siguientes ejemplos:

1. Supongamos que para representar la señal tenemos los siguientes vectores $\psi_1 = (1,0,0)$, $\psi_2 = (0,1,0)$, $\psi_3 = (0,0,1)$ y para sensarlo tenemos el vector $\varphi = (0, \frac{1}{\sqrt{3}}, 0)$. Calculando la coherencia mutua entre estos elementos tenemos:

$$\begin{cases} |\langle \psi_1, \varphi \rangle| = 0 \\ |\langle \psi_2, \varphi \rangle| = \frac{1}{\sqrt{3}} \\ |\langle \psi_3, \varphi \rangle| = 0 \end{cases} \Rightarrow \mu = \frac{1}{\sqrt{3}} \cdot \sqrt{3} = 1$$

en este caso los elementos tienen coherencia mínima, o a veces también se le llama como *incoherencia máxima*.

2. Como segundo ejemplo tomemos los siguientes vectores $\psi_1 = (1,0,0)$, $\psi_2 = (0,1,0)$, $\psi_3 = (0,0,1)$ y $\varphi = (0,0,1)$. Calculando la coherencia tenemos:

$$\begin{cases} |\langle \psi_1, \varphi \rangle| = 0 \\ |\langle \psi_2, \varphi \rangle| = 0 \\ |\langle \psi_3, \varphi \rangle| = 1 \end{cases} \Rightarrow \mu = 1 \cdot \sqrt{3} = \sqrt{3}$$

en este caso los elementos resultaron ser máximamente coherentes.

Una vez establecida la coherencia mutua podemos introducir el siguiente teorema:

Teorema 2.1.1 (Principio de incertidumbre débil [27]). *Se fija $x \in \mathbb{R}^n$ y se supone que es d -dispersa en una base Ψ . Se seleccionan m mediciones aleatorias uniformemente distribuidas en el dominio Φ . Entonces si*

$$m \geq C \cdot \mu^2(\Phi, \Psi) \cdot d \cdot \log n$$

para alguna constante positiva C , la solución a (1.2.4) es exacta con muy alta probabilidad.

Este teorema nos establece una cota acerca del número de mediciones necesarias para poder reconstruir la señal. Lo primero que podemos notar es que entre menor sea la coherencia, menor será el número de mediciones necesarias para reconstruir la señal. También nos establece que no se sufre de pérdida de información si uno mide un conjunto cualesquiera de m coeficientes que es por mucho menor al tamaño de la señal. Si la coherencia ($\mu(\Phi, \Psi)$) es igual o cercana a uno, entonces en el orden de $d \log n$ muestras son suficientes en lugar de n . Que tan alta es la probabilidad de recuperar la señal a partir de las m muestras, en [27] se demuestra que la probabilidad de éxito excede $1 - \delta$ si $m \geq C \cdot \mu^2(\Phi, \Psi) \cdot d \cdot \log(n/\delta)$.

Como vimos arriba, si uno estuviera interesado en implementar un esquema de sensado basado en muestreo compresivo, estaría principalmente interesado en

buscar pares de bases (Φ, Ψ) con coherencia mínima, además que estas tuvieran algoritmos rápidos, es decir, con tiempos de procesamiento de orden $O(n)$ o $O(n \log n)$. La elección de estas bases también depende de la aplicación, para explicar más a detalle consideremos los siguientes ejemplos:

Como primer ejemplo, consideremos a Φ como la base de espigas de Dirac $\varphi_k(t) = \delta(t - k)$ y Ψ como la base de Fourier, $\psi_j(t) = n^{-\frac{1}{2}} e^{i2\pi jt/n}$. Ya que Φ es la matriz de sensado, esta corresponde al esquema clásico de muestreo en tiempo o espacio. Este par tiempo-frecuencia tiene coherencia $\mu(\Phi, \Psi) = 1$, es decir, son máximamente incoherentes. Además, las espigas y las sinusoidales son máximamente incoherentes no sólo en una dimensión sino en cualquier dimensión. La base de Fourier también puede aplicarse como matriz de sensado Φ para adquirir señales que son dispersas en el tiempo, por ejemplo imágenes de resonancia magnética que son dispersas en espacio [10]. La transformada de Fourier puede aplicarse en un tiempo de orden $O(n \log n)$, además no necesita ser almacenada para poder ser aplicada a una señal.

Como segundo ejemplo, consideremos a las bases wavelets como matriz de representación Ψ y las noiselets como matriz de sensado Φ . Como se explico antes, las wavelets son de las bases más utilizadas para representar señales ya que proveen una buena compactación de la energía y un análisis multiresolución de imágenes en 2-D. La transformada wavelet puede ser aplicada en un orden de tiempo $O(n)$, de ahí su gran éxito. Las noiselets son una familia de funciones que están relacionadas con las wavelets, justo como las bases de Fourier están relacionadas con una señal en el dominio del tiempo. Como su nombre lo refiere, este tipo de señales son como ruido, y en particular, las noiselets son incomprensibles por wavelets ortogonales. En [28], Coifman et al. presentan dos tipos de noiselets complejas:

La primera familia de funciones son definidas recursivamente en el intervalo $[0,1)$ como:

$$\begin{aligned} f_1(x) &= \chi_{[0,1)}(x) \\ f_{2n}(x) &= (1 - i)f_n(2x) + (1 + i)f_n(2x - 1) \\ f_{2n+1}(x) &= (1 + i)f_n(2x) + (1 - i)f_n(2x - 1) \end{aligned} \quad (2.1.3)$$

donde $\chi_{[0,1)}(x) = 1$ en la definición del intervalo $[0,1)$ y 0 en otro caso. La segunda familia de funciones, conocidas también como *Noiselets Dragón*, son simétricas bajo el cambio de coordenadas $x \rightarrow 1 - x$. Su recursividad esta definida como:

$$\begin{aligned} g_1(x) &= \chi_{[0,1)}(x) \\ g_{2n}(x) &= (1 - i)g_n(2x) + (1 + i)g_n(2 - 2x) \\ g_{2n+1}(x) &= (1 + i)g_n(2x) + (1 - i)g_n(2 - 2x) \end{aligned} \quad (2.1.4)$$

En [28] se demuestra que estas funciones son bases ortonormales para funciones constantes por tramos con resolución 2^N , es decir, para funciones

$$V_N = \{h(x) : h \text{ es constante sobre cada } [2^{-N}k, 2^{-N}(k+1)), 0 \leq k < 2^N\}.$$

Estas funciones pueden verse como un diagrama de mariposa, por lo que una implementación podría llevarse acabo con un algoritmo parecido al de

transformada rápida de Fourier (FFT) tomando un orden de tiempo $O(n \log n)$. En particular la coherencia entre las noiselets y las wavelets Haar es $\sqrt{2}$ y entre las noiselets Daubechies D4 y D8 es, respectivamente, cercana de 2.2 y 2.9 a lo largo de un amplio rango de muestras de tamaño n . Esto también se extiende a dimensiones mayores [14]. Este par de bases es adecuado para adquirir imágenes de escenas naturales ya que en primera instancia las wavelets proveen una buena representación dispersa para este tipo de señales y las noiselets son incoherentes con el dominio wavelet. Otra ventaja es que al igual que la matriz de Fourier, la matriz de noiselets tampoco necesita ser almacenada para poder ser aplicada a la señal de interés. Finalmente ambas representaciones existen con algoritmos rápidos, pudiendo aplicar la transformación wavelet y noiselet en $O(n)$ y $O(n \log n)$, respectivamente.

Finalmente, las matrices aleatorias tienen incoherencia grande con cualquier base fija Ψ . Se ha demostrado con alta probabilidad que la coherencia entre Φ y Ψ es cercana de $\sqrt{2 \log n}$ [14]. Téngase en cuenta la implicación que llama la atención aquí, en el sentido de que si el sensado con sistemas incoherentes es bueno, entonces los mecanismos eficientes deben adquirir correlaciones con formas de onda aleatorias, por ejemplo: ruido blanco. Una de las grandes desventajas de las matrices aleatorias es que necesitan ser almacenadas para poder ser aplicadas a la señal por lo que resulta prohibitivo utilizarlas en un aplicación real. Por ejemplo, si uno quiere reconstruir una imagen de un megapixel ($n = 1000000$) a partir de $m = 25,000$ mediciones, se necesitarían más de 3 gigabytes de memoria para almacenar la matriz de medición en doble precisión y del orden de gigaflops para aplicarla [27].

Ahora analicemos desde el punto de vista de RIP. Esta propiedad fue introducida por Candès y Tao en [10] y refinada en [24] y se define como sigue:

Definición 2.1.2 (RIP). *Para cada entero $r = 1, 2, \dots$, se define la constante de isometría δ_r de una matriz Φ como el valor más pequeño tal que*

$$(1 - \delta_r) \cdot \|\mathbf{x}\|_2^2 \leq \|\Phi \mathbf{x}\|_2^2 \leq (1 + \delta_r) \cdot \|\mathbf{x}\|_2^2 \quad (2.1.5)$$

se cumple para todo vector \mathbf{x} r -disperso.

Se dice que una matriz Φ cumple con la propiedad de isometría restringida de orden r si δ_r no es muy cercano a uno. Cuando $\delta_r < 1$, estas desigualdades implican que cada colección de r columnas de Φ es no singular, lo cual es el requerimiento mínimo para invertir señales $(r/2)$ -dispersas. Cuando $\delta_r \ll 1$, el operador de muestreo mantiene aproximadamente la distancia Euclidiana entre cada par de señales $(r/2)$ -dispersas, lo que implica que los vectores en cuestión no pueden estar en el espacio nulo de Φ . (Es decir, es posible invertir el proceso de muestro establemente). Si uno quiere utilizar en la práctica esta propiedad para adquirir señales d -dispersas, uno debe lograr obtener una constante de isometría δ_{2d} pequeña con un número de muestras tan pequeño como sea posible. Aunque es muy difícil computacionalmente verificar si una matriz satisface las desigualdades (2.1.5), se ha demostrado que muchas matrices satisfacen RIP con muy alta probabilidad y pocas mediciones. De hecho las matrices aleatorias

Gaussianas de Bernoulli y Fourier satisfacen RIP con un número de mediciones aproximadamente lineal sobre el nivel de dispersidad. Por ejemplo, utilizando matrices aleatorias uno puede comúnmente obtener $\delta_{2d} \leq 0.1$ con

$$m = O(d \log^\alpha n) \quad (2.1.6)$$

mediciones, donde α es un número entero pequeño (se conjetura que (2.1.6) se cumple con solo $\log n$ mediciones).

A continuación se presentan dos ejemplos de matrices que cumplen con RIP:

1. Matrices Gaussianas: Si los valores de $\sqrt{m} \Phi$ son independientes e idénticamente distribuidos de una distribución normal con media cero y varianza 1, entonces

$$m \geq \frac{Cr \log(n/r)}{\varepsilon^2} \Rightarrow \delta_r \leq \varepsilon \quad (2.1.7)$$

excepto con probabilidad e^{-cm} . Para detalles vea [11].

2. Matrices parciales de Fourier: Si $\sqrt{m} \Phi$ es un conjunto aleatorio uniformemente distribuido de m renglones obtenidos de la matriz unitaria de Fourier de dimensiones $n \times n$, entonces

$$m \geq \frac{Cr \log^5 n \cdot \log(\varepsilon^{-1})}{\varepsilon^2} \Rightarrow \delta_r \leq \varepsilon \quad (2.1.8)$$

excepto con probabilidad N^{-1} . Véase [29].

Las letras C y c representan constantes positivas. Al igual que en el caso de la coherencia las matrices aleatorias gaussianas requieren un número pequeño de muestras, por ejemplo con $\delta_r \leq 0.1$ uno requiere alrededor de $Cr \log(n/r)$ muestras. En el caso de las matrices de Fourier uno requiere un número un poco mayor de muestras, sin embargo como ya se mencionó anteriormente para aplicaciones prácticas las matrices de Fourier proveen mayores ventajas.

Se presentaron dos enfoques, los cuales nos proveen cotas bien definidas sobre el número m de muestras para reconstruir una señal. Como pudimos observar, ambas cotas son muy similares, sin embargo es más sencillo medir la coherencia mutua que comprobar las desigualdades de RIP.

Una vez establecido el proceso de muestreo, podemos abordar el proceso de reconstrucción de la información.

2.2 Reconstrucción de la Información

Como vimos en la sección anterior, el proceso de muestreo sólo recolecta un subconjunto de m mediciones de la señal de dimensión n . Por lo tanto, el proceso de reconstrucción consistiría en recuperar la señal d -dispersa $\mathbf{x} \in \mathbb{R}^n$ a partir del vector de mediciones $\mathbf{y} = \Phi \mathbf{x} \in \mathbb{R}^m$. Este proceso podría sonar un poco desalentador ya que Φ es de dimensiones $m \times n$, es decir, su solución implicaría resolver un sistema indeterminado de ecuaciones. Lo único que sabemos *a priori*

es que el vector es disperso por lo tanto podríamos utilizar un programa de optimización para encontrar la solución de los elementos distintos de la señal. Una forma de optimización sería utilizando el problema de minimización de la norma ℓ_0

$$\min_{\tilde{\mathbf{x}} \in \mathbb{R}^n} \|\tilde{\mathbf{x}}\|_0 \quad \text{sujeto a} \quad \Phi \tilde{\mathbf{x}} = \mathbf{y} (= \Phi \mathbf{x}) \quad (2.2.1)$$

Si \mathbf{x} es d -dispersa y el rango de Φ es mayor a $2d$, entonces la solución a (2.2.1) debe ser la señal \mathbf{x} . De hecho, si el minimizador es $\tilde{\mathbf{x}}$, entonces \mathbf{x} es la solución factible y $\tilde{\mathbf{x}}$ deber ser d -dispersa también. Ya que $\Phi \tilde{\mathbf{x}} = \mathbf{y}$, $\mathbf{y} - \mathbf{x}$ deberá estar en el espacio nulo de Φ . Pero como $\tilde{\mathbf{x}} - \mathbf{x}$ es $2d$ -disperso y como el rango de Φ es mayor a $2d$, se debe tener que $\tilde{\mathbf{x}} = \mathbf{x}$. De este modo el problema de optimización de la norma ℓ_0 , teóricamente funciona a la perfección. Sin embargo, en la práctica este es un problema computacionalmente intratable y un problema NP-duro [24] en general. Afortunadamente la teoría del muestreo compresivo ha desarrollado algoritmos cuya solución es equivalente a este problema NP-duro. Los métodos de reconstrucción se dividen principalmente en tres categorías [30]:

- a) Búsquedas *greedy*: Estos algoritmos forman una aproximación en cada paso haciendo elecciones locales óptimas en cada paso. Algunos ejemplos incluyen *orthogonal matching pursuit* (OMP) [31], OMP por etapas [32] y OMP regularizado [33, 34].
- b) Relajación Convexa: Estos métodos resuelven un programa de optimización convexo para aproximar la señal objetivo. Dentro de este grupo existen algunas variantes del programa de optimización conocido como *Basis Pursuit* [35].
- c) Algoritmos Combinatorios: Estos métodos utilizan matrices de medición muy estructuradas que soportan una rápida reconstrucción a través de un grupo de pruebas. Algunos algoritmos en esta categoría son muestreo de Fourier [36, 37], *chaining pursuit* [38] y *heavy hitters on steroids* (HHS) pursuit [39].

Dentro de los enfoques mencionados cada uno de los algoritmos tiene sus ventajas y desventajas. A continuación describiremos un algoritmo basado en búsqueda *greedy* y otro en relajación convexa para después describir con mayor detalle un algoritmo conocido como *compressive sampling matching pursuit* (CoSaMP) que utiliza ambos enfoques para adquirir mayor estabilidad y velocidad en la reconstrucción de la información. Nos concentraremos un poco más en CoSaMP ya que este algoritmo será utilizado en todas nuestras simulaciones.

2.2.1 Basis Pursuit

Considerando que el problema (2.2.1) es numéricamente inviable, es necesario buscar una solución numéricamente más manejable. Uno de los enfoques más utilizados para resolver sistemas indeterminados de ecuaciones es mediante la

solución de mínima energía o mínimos cuadrados:

$$\underset{\tilde{\mathbf{x}} \in \mathbb{R}^n}{\text{mín}} \|\tilde{\mathbf{x}}\|_2 \quad \text{sujeto a} \quad \Phi \tilde{\mathbf{x}} = \mathbf{y} \quad (2.2.2)$$

este problema es numéricamente tratable, sin embargo, en el caso de información incompleta, este enfoque pone en cero todas las componentes no observadas, resultando en un minimizador de mucha menor energía que el de la señal original y también introduce muchos artefactos.

La minimización de la norma ℓ_1 , o también conocida como *Basis Pursuit*, ha sido propuesta como una alternativa convexa a la norma combinatoria ℓ_0 [35].

$$\underset{\tilde{\mathbf{x}} \in \mathbb{R}^n}{\text{mín}} \|\tilde{\mathbf{x}}\|_1 \quad \text{sujeto a} \quad \Phi \tilde{\mathbf{x}} = \mathbf{y} \quad (2.2.3)$$

este problema intenta encontrar la solución con mínima norma ℓ_1 . Esta relajación usualmente recupera \mathbf{x} de forma exacta cuando \mathbf{x} es dispersa y de forma muy aproximada cuando la señal es cuasi-dispersa, o compresible. Candès y Tao demostraron que para ciertas matrices que cumplen alguna propiedad cuantitativa, los programas (2.2.1) y (2.2.3) son equivalentes [24]. En otras palabras, aunque la norma ℓ_1 es diferente de la cuasi-norma ℓ_0 , la solución ℓ_1 usualmente provee la solución más dispersa.

Desde el punto de vista práctico esta minimización convexa puede ser replanteada como un programa de optimización lineal [35]:

$$\underset{\substack{\mathbf{x} \in \mathbb{R}^n \\ \mathbf{t} \in \mathbb{R}^m}}{\text{mín}} \mathbf{1}^T \mathbf{t} \quad \text{sujeto a} \quad -\mathbf{t} \preceq \Phi \tilde{\mathbf{x}} - \mathbf{y} \preceq \mathbf{t} \quad (2.2.4)$$

esencialmente descompone $|\mathbf{x}|$ en sus partes positivas y negativas, $\mathbf{x} = \mathbf{x}_+ - \mathbf{x}_-$, haciendo disponible un gran número de estrategias computacionalmente efectivas [19], estableciendo también cotas polinomiales para su solución.

Desde el punto de vista geométrico Basis Pursuit también explica porque puede recuperar soluciones dispersas.

Sin pérdida de generalidad tomemos un ejemplo en \mathbb{R}^2 . Supongamos que tenemos un vector disperso $\tilde{\mathbf{x}}$ (sólo una de sus componentes es cero) del cual sólo tomamos una medición y el subespacio afín H el cual es el conjunto de todas las \mathbf{x} que comparten el mismo valor de medición, $H = \{\mathbf{x} : \Phi \mathbf{x} = \mathbf{y}\}$.

Si queremos encontrar en H el elemento $\tilde{\mathbf{x}}$, utilizando el enfoque de mínima norma ℓ_1 y mínima norma ℓ_2 . Esto puede hacerse expandiendo sus respectivas formas

$$B_{\tau_1} = \{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x}\|_1 \leq \tau_1\} \quad \text{y} \quad B_{\tau_2} = \{\mathbf{x} \in \mathbb{R}^2 : \|\mathbf{x}\|_2 \leq \tau_2\}$$

incrementando τ_1 y τ_2 hasta intersectar H . Esta interpretación se ilustra en la figura 2.1.

La forma ℓ_1 se mantiene más cerca del eje de coordenadas que la forma ℓ_2 . Podemos ver que el punto de intersección de H y la forma ℓ_2 no es dispersa del todo. En dimensiones mayores esta diferencia se vuelve más dramática, ya

que el volumen de la forma ℓ_1 se vuelve más pequeño que el volumen de la forma ℓ_2 . Por tanto, la solución óptima $\tilde{\mathbf{x}}$ es más probable que tenga más ceros o valores cercanos a cero cuando sea calculada utilizando la norma ℓ_1 que cuando se calcula utilizando la norma ℓ_2 .

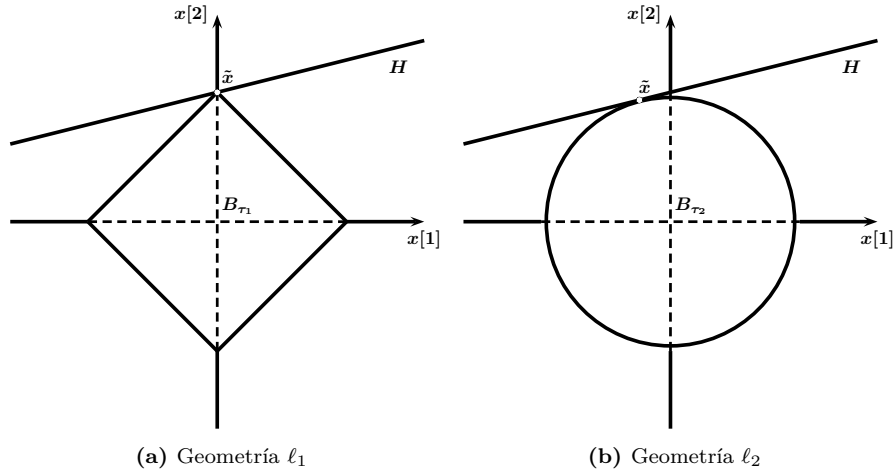


Figura 2.1: Comparación de las soluciones mínimas ℓ_1 y ℓ_2 .

2.2.2 Orthogonal Matching Pursuit

Como se mencionó antes, este algoritmo cae dentro del enfoque greedy. Este algoritmo primero fue introducido por Zhang y Mallat [40] con aplicaciones en compresión, reconocimiento de patrones, entre otras, y fue retomado por Gilbert y Tropp [31] como una alternativa para reconstruir señales dispersas a partir de mediciones gaussianas. OMP es relativamente rápido, desafortunadamente no es tan estable como Basis Pursuit.

Este algoritmo se describe a continuación [31], OMP para recuperación de señales:

ENTRADA:

- Matriz de Medición Φ de dimensión $m \times n$.
- Vector de mediciones $\mathbf{y} = \Phi \mathbf{x}$ de dimensión m .
- Nivel de dispersión d de la señal ideal.

SALIDA:

- Un estimado $\tilde{\mathbf{x}} \in \mathbb{R}^n$ de la señal ideal.
- Un conjunto de índices $I \subset \{1, \dots, n\}$.

PROCEDIMIENTO:

1. Se inicializa el residuo $\mathbf{r}_0 = \mathbf{y}$, el conjunto de índices $I_0 = 0$, y el contador de iteración $t = 1$.
2. Se encuentran los índices λ_t que resuelven el programa de optimización:

$$\lambda_t = \arg \max_{j=1, \dots, n} |\langle \mathbf{r}_{t-1}, \boldsymbol{\varphi}_j \rangle|$$

Si el máximo ocurre para múltiples índices, se rompe el empate de forma determinista.

3. Se aumenta el conjunto de índices y las matrices de las entradas elegidas:

$$I_t = I_{t-1} \cup \{\lambda_t\} \quad \text{y} \quad \boldsymbol{\Phi}_t = [\boldsymbol{\Phi}_{t-1}, \boldsymbol{\varphi}_{\lambda_t}]$$

Se usa $\boldsymbol{\Phi}_0$ para indicar la matriz vacía.

4. Se resuelve el problema de mínimos cuadrados para obtener una nueva señal estimada:

$$\mathbf{x}_t = \arg \min_{\mathbf{x}} \|\mathbf{y} - \boldsymbol{\Phi}_t \mathbf{x}\|_2$$

5. Se calcula la nueva aproximación de la información y el residuo

$$\begin{aligned} \mathbf{a}_t &= \boldsymbol{\Phi}_t \mathbf{x}_t \\ \mathbf{r}_t &= \mathbf{y} - \mathbf{a}_t \end{aligned}$$

6. Se incrementa t y se repite el paso 2 si $t < d$.
7. El estimado $\tilde{\mathbf{x}}$ de la señal ideal tiene los índices diferentes de cero listados en I . El valor de la estimación $\tilde{\mathbf{x}}$ en la componente λ_j igual a la j -ésima componente de \mathbf{x}_t .

El tiempo de ejecución del algoritmo OMP esta dominado por el paso 2, cuyo costo es $O(dmn)$. El algoritmo se itera d veces y en cada iteración se hace una selección a través de los n elementos, se multiplica por $\boldsymbol{\Phi}^*$, y se resuelve el problema por mínimos cuadrados. En la iteración t , el problema de mínimos cuadrados puede ser resuelto en tiempo $O(tn)$. Sin embargo, se debe mantener una factorización QR de $\boldsymbol{\Phi}_t$ y también usar el algoritmo de Gram-Schmidt modificado (MGS), ya que la matriz es no estructurada y densa. En el caso de que la matriz sea estructurada, una implementación más eficiente puede ser implementada.

Este algoritmo, como muchos otros de tipo greedy lo que hace es calcular el soporte de la señal iterativamente y una vez que el soporte es calculado, la señal es reconstruida a partir de sus mediciones. La mayoría de estos algoritmos recuperan un menor nivel de dispersidad comparado con la minimización ℓ_1 . Además, la mayoría a excepción de algunos pocos no proveen garantías comprobables y estabilidad.

A continuación presentaremos un algoritmo que combina tanto el enfoque greedy como el de optimización.

2.2.3 Compressive Sampling Matching Pursuit

El algoritmo CoSaMP es muy parecido a OMP, primero selecciona los elementos más grandes de la señal provisional y después agrega estos elementos al conjunto soporte. CoSaMP a diferencia de OMP realiza una estimación de la señal mediante el programa de mínimos cuadrados y después establece en cero a todos los elementos de esta estimación, excepto los más significativos, para hacer la señal dispersa. Este algoritmo puede dividirse en cinco pasos principales [30]:

1. **Identificación:** El algoritmo forma una señal provisional de las muestras actuales y localiza las componentes más grandes de esta.
2. **Unión de soportes:** El conjunto de los elementos recientemente identificados es unido con el conjunto de elementos que aparecen en la aproximación actual.
3. **Estimación:** El algoritmo resuelve un problema de mínimos cuadrados para aproximar la señal objetivo en las componentes del conjunto unión.
4. **Aislamiento:** El algoritmo produce una nueva aproximación manteniendo sólo los elementos más grandes de la aproximación de mínimos cuadrados.
5. **Actualización de muestras:** Por último, las muestras son actualizadas de forma que se refleje en el residuo la parte de la señal que no ha sido aproximada.

Estos pasos se repiten hasta que se activa el criterio de paro. El pseudocódigo para CoSaMP se muestra en el algoritmo 2.2.1.

Analicemos ahora la complejidad del algoritmo. La complejidad está principalmente regida por el problema de mínimos cuadrados, parecería que este paso es muy costoso, sin embargo ya que la matriz Φ_I nunca tendrá más de $3d$ columnas ésta puede ser aplicada de forma rápida utilizando un método iterativo, tal como la iteración de Richardson o el gradiente conjugado, véase [41] para más detalles sobre estos métodos iterativos. Estos métodos tienen la ventaja de que sólo interactúan con la matriz Φ_I a través de operaciones matriz-vector. Por tanto, si inicializamos el método de mínimos cuadrados con la aproximación actual \mathbf{a}^{k-1} , entonces el costo de resolver este problema es $O(\mathcal{M})$, donde \mathcal{M} acota el costo de multiplicar una matriz con un vector con Φ_T o Φ_T^* .

El análisis de los pasos restantes es mucho más sencillo. Formar la señal provisional está acotado por la multiplicación $\Phi^* \mathbf{u}$, es decir $O(\mathcal{M})$. La identificación de las $2d$ componentes más grandes de la señal puede ser realizada en tiempo lineal $O(n)$ con algún algoritmo de selección aleatoria [42]. Otra forma es haciendo un ordenamiento decreciente y quedarnos con los primeros $2d$ elementos, esto tiene una complejidad de $O(d \cdot \log d)$. El paso de aislamiento es muy parecido al de selección ya que requiere ordenar los elementos por magnitud y después seleccionar los primeros d con un costo de $O(d \cdot \log d)$. Finalmente la actualización de las muestras está definida por el costo de la multiplicación de Φ por el vector d -disperso \mathbf{a}^k . En resumen la complejidad del algoritmo CoSaMP

Algoritmo 2.2.1: CoSaMP Algoritmo de Recuperación

CoSaMP(Φ, \mathbf{y}, d)
Entrada: Matriz de Muestreo Φ , vector de mediciones \mathbf{y} , nivel de dispersidad d .
Salida: Una aproximación \mathbf{a} d -dispersa de la señal objetivo.

$\mathbf{a}^0 \leftarrow \mathbf{0}$ { aproximación inicial }
 $\mathbf{u} \leftarrow \mathbf{y}$ { muestras actuales = muestras de entrada }
 $k \leftarrow 0$

repeat
 $k \leftarrow k + 1$

$\mathbf{v} \leftarrow \Phi^* \mathbf{u}$ { Formación de señal de mínima energía }
 $\Omega \leftarrow \text{supp}(\mathbf{v}_{2d})$ { Identificar componentes mayores }
 $I \leftarrow \Omega \cup \text{supp}(\mathbf{a}^{k-1})$ { Unir soportes }

$\mathbf{b}_{|I} \leftarrow \Phi_I^\dagger \mathbf{y}$ { Estimación de señal con mínimos cuadrados }
 $\mathbf{b}_{|I^c} \leftarrow \mathbf{0}$

$\mathbf{a}^k \leftarrow \mathbf{b}_d$ { Actualiza siguiente aproximación }
 $\mathbf{u} \leftarrow \mathbf{y} - \Phi \mathbf{a}^k$ { Actualizar muestras actuales }

until criterio de paro *verdadero*

esta acotada por el costo de multiplicación matriz-vector, es decir, $O(\mathcal{M})$. La tabla 2.1 muestra un resumen de lo antes explicado.

En cuanto al espacio de almacenamiento, CoSaMP (sin contar el espacio de la matriz de muestreo) utiliza un vector de dimensión n para el vector de la señal provisional y otros dos vectores de dimensión m para almacenar los vectores de muestras \mathbf{u}, \mathbf{y} . Por lo tanto en el peor de los casos se utiliza un espacio de almacenamiento de $O(n)$.

2.2.4 Principales Teoremas

El principal teorema de CoSaMP garantiza la reconstrucción exacta de señales dispersas y una reconstrucción aproximada de señales compresibles o cuasi-dispersas.

Teorema 2.2.1. (CoSaMP [30]). *Supóngase que Φ es una matriz de muestreo de dimensiones $m \times n$ con una constante de isometría restringida $\delta_{2d} \leq c$. Sea $\mathbf{u} = \Phi \mathbf{x} + \mathbf{e}$ un vector de muestras de una señal arbitraria contaminada con un ruido arbitrario. Para un parámetro de precisión η dado, el algoritmo CoSaMP*

Paso	Costo
Formar señal provisional	$O(\mathcal{M})$
Identificación	$O(n \log n)$
Unión de soportes	$O(d \log d)$
Estimación mínimos cuadrados	$O(\mathcal{M})$
Aislamiento	$O(d \log d)$
Actualización de muestras	$O(\mathcal{M})$
Costo total por iteración	$O(\mathcal{M})$

Tabla 2.1: Análisis de complejidad del algoritmo CoSaMP. Las dimensiones de la matriz de muestreo son $m \times n$, el nivel de dispersidad es d y el número \mathcal{M} acota el costo de la multiplicación matriz-vector con Φ o Φ^* .

produce una aproximación \mathbf{a} d -dispersa que satisface

$$\|\mathbf{x} - \mathbf{a}\|_2 \leq C \cdot \max \left\{ \eta, \frac{1}{\sqrt{d}} \|\mathbf{x} - \mathbf{x}_{d/2}\|_1 + \|\mathbf{e}\|_2 \right\}$$

donde $\mathbf{x}_{d/2}$ es la mejor aproximación $(d/2)$ -dispersa de \mathbf{x} . El tiempo de ejecución es $O(\mathcal{M} \cdot \log(\|\mathbf{x}\|_2 / \eta))$, donde \mathcal{M} acota el costo de una multiplicación matriz-vector con Φ o Φ^* . Ocupando un espacio $O(n)$.

Este teorema nos dice que el algoritmo CoSaMP da una aproximación d -dispersa, cuyo error ℓ_2 es comparable con el error ℓ_1 de la mejor aproximación $(d/2)$ -dispersa de la señal. El error garantiza que el algoritmo puede recuperar una señal d -dispersa con alta precisión sin presencia de ruido. Su desempeño se va degradando conforme la energía del ruido aumenta o también debido a que la señal es compresible. La cota del tiempo de ejecución indica que cada multiplicación matriz-vector reduce el error por un factor constante.

Otro resultado importante de CoSaMP establece que al final de cada iteración se hace un avance significativo ya que el error de aproximación es relativamente grande respecto a la *energía no recuperable*. La energía no recuperable ν de una señal mide el error base en la aproximación de la señal, que ocurre debido al ruido en las muestras, o porque la señal es no dispersa.

$$\nu = \|\mathbf{x} - \mathbf{x}_d\|_2 + \frac{1}{\sqrt{d}} \|\mathbf{x} - \mathbf{x}_d\|_1 + \|\mathbf{e}\|_2 \quad (2.2.5)$$

Este resultado se enuncia en el siguiente teorema:

Teorema 2.2.2. (*Invarianza a la iteración [30]*). Para cada iteración $k \geq 0$, la señal de aproximación \mathbf{a}^k es d -dispersa y

$$\|\mathbf{x} - \mathbf{a}^{k+1}\|_2 \leq 0.5 \|\mathbf{x} - \mathbf{a}^k\|_2 + 10\nu$$

En particular,

$$\|\mathbf{x} - \mathbf{a}^k\|_2 \leq 2^{-k} \|\mathbf{x}\|_2 + 20\nu$$

donde ν es la energía no recuperable.

Por último, otro punto que no se ha abordado fue el de criterio de paro. Hay tres principales enfoques para el paro del algoritmo. El primero es detenerlo después de un número fijo de iteraciones. El segundo es usando la norma $\|\mathbf{u}\|_2$ de las muestras actuales como evidencia acerca de la norma $\|\mathbf{r}\|_2$ del residuo. El último criterio es utilizar la norma $\|\mathbf{v}\|_\infty$ de la señal provisional para acotar la magnitud $\|\mathbf{r}\|_\infty$ de los elementos del residuo. El residuo \mathbf{r} se define como $\mathbf{r} = \mathbf{x} - \mathbf{a}$.

Los siguientes teoremas nos establecen las garantías para el paro del algoritmo:

Teorema 2.2.3. (*Paro I [30]*). *El criterio de paro $\|\mathbf{v}\|_2 \leq \varepsilon$ asegura que*

$$\|\mathbf{x} - \mathbf{a}\|_2 \leq 1.06 \cdot (\varepsilon + \|\mathbf{e}\|_2)$$

El criterio de paro $\|\mathbf{y}\|_\infty \leq \eta/\sqrt{2d}$ asegura que

$$\|\mathbf{x} - \mathbf{a}\|_\infty \leq 1.12\eta + 1.17\|\mathbf{e}\|_2$$

Teorema 2.2.4. (*Paro II [30]*). *El criterio de paro $\|\mathbf{v}\|_2 \leq \varepsilon$ se activa tan pronto como*

$$\|\mathbf{x} - \mathbf{a}\|_2 \leq 0.95 \cdot (\varepsilon - \|\mathbf{e}\|_2)$$

El criterio de paro $\|\mathbf{y}\|_\infty \leq \eta/\sqrt{2d}$ es activado tan pronto como

$$\|\mathbf{x} - \mathbf{a}\|_\infty \leq \frac{0.45\eta}{d} - \frac{0.68\|\mathbf{e}\|_2}{\sqrt{d}}$$

Finalmente la tabla 2.2 muestra el análisis de algunos algoritmos de acuerdo a varios criterios [30]. Las muestras generales se refieren a la variedad de esquemas de muestreo que el algoritmo soporta. El número óptimo de muestras se refiere a que si el algoritmo puede recuperar una señal d -dispersa con un número de muestras $O(d \cdot \log n)$ o si requiere una cota superior a esta. La uniformidad se refiere a que si el algoritmo puede recuperar todo tipo de señales dada una matriz de muestreo fija, y la estabilidad a si el algoritmo resulta exitoso cuando la señal es compresible, pero no dispersa, y/o cuando las muestras están contaminadas con ruido. La complejidad se refiere al costo de recuperar una señal d -dispersa en el peor de los casos.

2.3 Resumen

En este capítulo introducimos varios conceptos acerca del muestreo compresivo, el cual consiste en reconstruir una señal dispersa, o compresible, a partir de un número mucho menor de muestras que el de la señal original. Las muestras son obtenidas mediante el producto interno de la señal original con unas funciones bases que son fijadas independientemente de la señal original. El proceso de muestreo puede verse como la aplicación de la matriz de muestreo a la señal

	CoSaMP	OMP	ROMP
Muestras generales	RIP	Subgauss.	RIP
Opt. # muestras	sí	sí	no
Uniformidad	sí	no	sí
Estabilidad	sí	No conocida	sí
Complejidad	$O(mn)$	$O(dmn)$	$O(dmn)$
	Opt. Convexa	Muestreo de Fourier	HHS Pursuit
Muestras generales	RIP	no	no
Opt. # muestras	sí	no	no
Uniformidad	sí	no	sí
Estabilidad	sí	sí	sí
Complejidad	$LP(n, m)$	$d \text{ polylog}(n)$	$\text{poly}(d \log n)$

Tabla 2.2: Comparación de algunos algoritmos de recuperación. La notación d se refiere al nivel de dispersidad; m se refiere al número de mediciones; n se refiere al tamaño de la señal.

original. Las matrices que son “buenas” candidatas para ser utilizadas en el proceso de adquisición de la información son aquellas que obedecen el principio de RIP, o aquellas que tengan coherencia mínima con la base de representación de la señal. Dependiendo del enfoque utilizado, RIP o coherencia mutua, el número de muestras puede estar acotado por $O(d \log n)$, o por $m \geq C\mu^2(\Phi, \Psi)d \log n$. Las matrices aleatorias obedecen el principio de RIP y además son incoherentes con casi cualquier base de representación, sin embargo, tienen la desventaja de que necesitan ser almacenadas para ser aplicadas a la señal original, lo que resulta prohibitivo para una aplicación real. Las matrices que resultan convenientes para una aplicación real son las matrices noiselets y las matrices de coeficientes de Fourier, ya que estas pueden ser aplicadas en tiempo $O(n \log n)$, además de que no necesitan de almacenamiento extra. En la reconstrucción de la información se analizaron tres métodos, el de minimización de la norma ℓ_1 , OMP y CoSaMP. El método Basis Pursuit es muy bueno para aproximar las señales dispersas debido a su geometría, y ha sido demostrado que en muchos casos es equivalente a la cuasi-norma ℓ_0 . Además como puede ser replanteado con programación lineal, este método se convierte en muy estable. Sin embargo, la desventaja de este método es que es muy lento debido a que realiza una optimización global, si por ejemplo uno implementa este programa con un método de punto interior, la complejidad del algoritmo sería $O(m^2 n^{1.5})$. Debido a que OMP tiene un enfoque greedy es mucho más rápido, su complejidad es $O(dmn)$, sin embargo no provee una solución tan dispersa como Basis Pursuit y suele introducir artefactos. CoSaMP es un algoritmo que combina las ventajas tanto del enfoque greedy, como del de optimización lo que resulta en un algoritmo estable y uniforme. Este algoritmo identifica muchas de las componentes en cada iteración, lo que permite que el algoritmo se ejecute de manera más rápida. El tiempo de ejecución de CoSaMP es de orden $O(mn)$. Finalmente, una de las novedades de CoSaMP es que

explota las propiedades de la isometría restringida para asegurar la identificación exitosa de la señal en cada paso, desde el punto de vista greedy. Candès y Tao habían explotado esta característica pero en métodos de relajación convexa para muestreo compresivo [24].

Capítulo 3

Otras Técnicas de Compresión y Reconstrucción de Imágenes

En general en la literatura se suelen encontrar dos tipos de compresión de imágenes: con pérdidas, o sin pérdidas. En el caso de la compresión sin pérdidas, la imagen original es recuperada de forma exacta después de la descompresión. Sin embargo, con imágenes de escenas naturales es muy difícil obtener una compresión libre de errores a una tasa de compresión mayor a 2:1 [43]. Para el caso de compresión con pérdidas, mayores proporciones de compresión pueden ser obtenidas con un error difícil de percibir entre la imagen descomprimida y la imagen original. En este capítulo presentaremos un algoritmo que sirve como referencia de compresión de imágenes conocido como embedded zerotree wavelet (EZW), este método de compresión fue introducido por Shapiro en [44]. El algoritmo EZW utiliza la transformada discreta *wavelet* (DWT) en 2-D por lo que antes de explicar el algoritmo será conveniente que presentemos algunos conceptos breves acerca de esta transformación ¹.

3.1 Wavelets

3.1.1 Transformada Discreta wavelet

El origen de la transformada discreta *wavelet* data de 1976 cuando Croiser, Esteban y Galand desarrollaron una técnica para descomponer señales discretas. Crochiere, Weber y Flanagan hicieron un trabajo similar sobre codificación de señales de voz en ese mismo año. Ellos nombraron su esquema de análisis como codificación por subbandas. En 1983, Burt definió una técnica muy similar a la codificación por subbandas que posteriormente la nombró como codificación piramidal, la cual es también conocida como análisis multiresolución. Después en 1989, Vetterli y Le Gall hicieron algunas mejoras a estos esquemas, removiendo la redundancia remanente en el esquema de codificación piramidal.

¹El lector puede referirse a [2, 15] para una explicación más detallada.

La DWT es obtenida usando técnicas de filtrado. Al proceso de descomponer la señal se le conoce como *análisis* y al proceso de reconstrucción como *síntesis*. Es decir, filtros de diferentes frecuencias de corte son utilizados para analizar la señal en diferentes escalas. Las señales son pasadas a través de una serie de filtros paso-altas para analizar componentes de altas frecuencias, y a su vez filtros paso-bajas son utilizados para analizar componentes de bajas frecuencias.

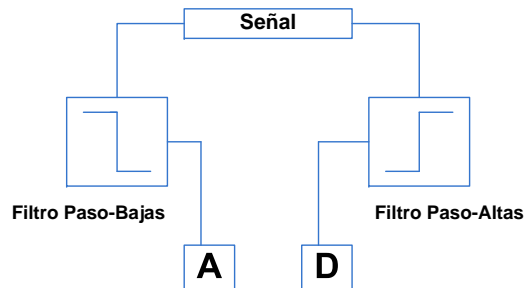


Figura 3.1: Proceso de descomposición

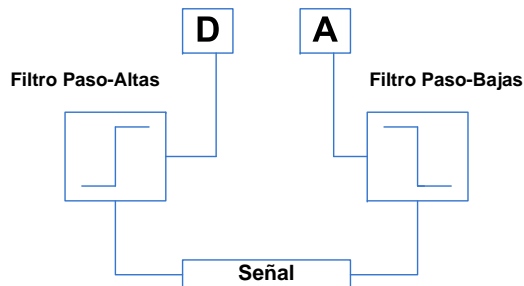


Figura 3.2: Proceso de reconstrucción

La DWT analiza la señal en diferentes bandas de frecuencia con diferentes resoluciones descomponiendo la señal en una aproximación reducida, e información detallada, como se muestra en la figura 3.1. Y su proceso inverso consiste en reconstruir la señal a partir de estas aproximaciones y detalles, como se muestra en la figura 3.2. Es de notar que el desempeño de este proceso dependerá mucho de la selección de los filtros y por ende de las funciones *wavelet*².

3.1.2 Transformada Discreta wavelet en 2-D

La DWT en dos dimensiones (DWT2) es ampliamente utilizada en procesamiento de imágenes e inclusive fue adoptada por el estándar JPEG2000 debido a que

²El número de funciones *wavelet* es muy variado, en la actualidad existe un gran número de filtros ortogonales y biortogonales

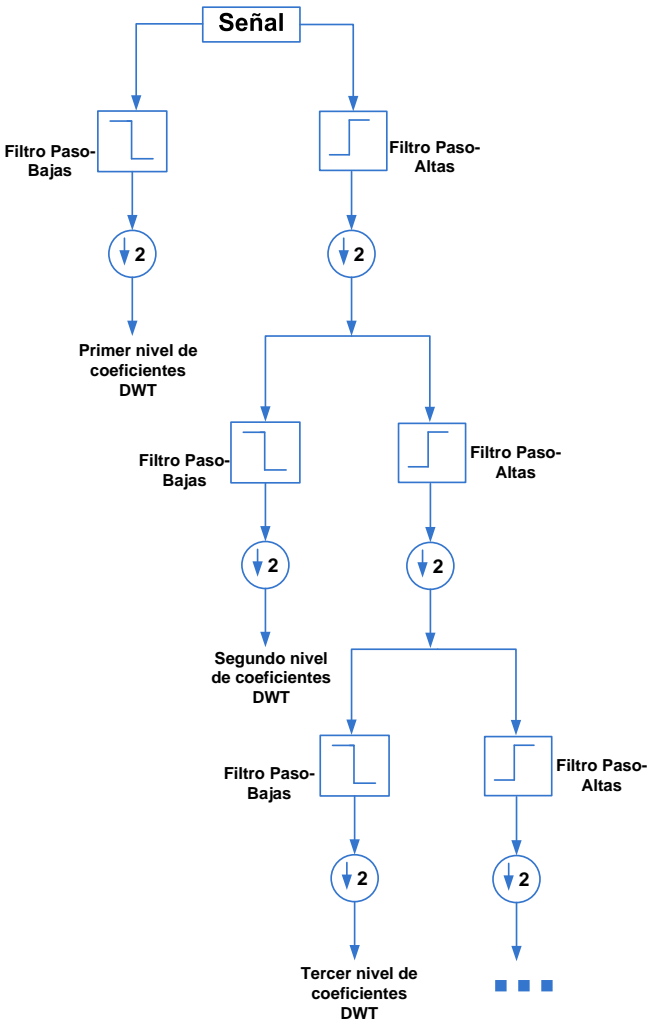


Figura 3.3: Ejemplo codificación de los coeficiente *wavelet* por subbandas.

proporciona una buena representación de imágenes con pocos coeficientes. La DWT2 descompone una imagen en uno o más niveles, para cada nivel la DWT2 genera una matriz de coeficientes; estos coeficientes a su vez están clasificados en cuatro categorías, o bandas, a saber: aproximaciones, detalles horizontales, detalles verticales y detalles diagonales. Como la transformada *wavelet* es separable ésta puede ser obtenida con un filtrado horizontal seguido de un filtrado vertical. Las bandas también suelen denominarse como LL para la banda de aproximaciones, HL para la banda de detalles verticales, LH para la banda de detalles horizontales y HH para la banda de detalles diagonales. La letra L se refiere a un filtrado paso-bajas y la letra H a un filtrado paso-altas. Por tanto la banda LL significa que tuvo un filtrado paso-bajas en la dirección horizontal y también un filtrado paso-bajas en la dirección vertical. En el caso de varios niveles de descomposición se suele agregar un subíndice a las letras para indicar el nivel de la banda de descomposición. Por ejemplo LL_3 significaría que es la banda de aproximaciones del nivel 3 de descomposición. Esto se ilustra en la figura 3.4.

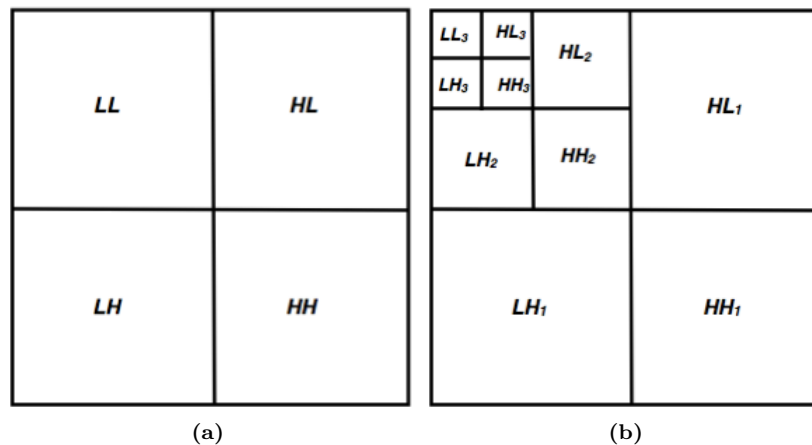


Figura 3.4: a) Descomposición *wavelet* en un nivel. b) Descomposición *wavelet* en tres niveles.

La figura 3.5 muestra la imagen de “lena” con tres niveles de descomposición. Podemos observar en la figura que la aproximación, es la matriz de coeficientes que posee la mayor energía, mientras que los detalles son coeficientes con muy poca energía, ya que éstos sólo dan detalles acerca de la orientación debido al tipo de filtrado que se hace de la imagen original.

Resumiendo, el éxito de la transformada Wavelet ha sido muy grande principalmente debido a que provee una buena aproximación no lineal para funciones suaves por tramos en una dimensión (1-D) y también porque su tiempo de ejecución es de orden $O(n)$. También debido a su distribución de coeficientes ha dado lugar al surgimiento de varios algoritmos para compresión de imágenes, por mencionar algunos: EZW, *set partitioning in hierarchical trees* (SPIHT), *wavelet difference reduction* (WDR), *adaptively scanned wavelet difference reduction* (ASWDR)

[43].

Ahora que hemos presentado de un forma muy breve y general la transformada *wavelet* podemos abordar uno de los primeros algoritmos de compresión de imágenes que utilizan esta transformada, el EZW, y que ha sido la base de otros.

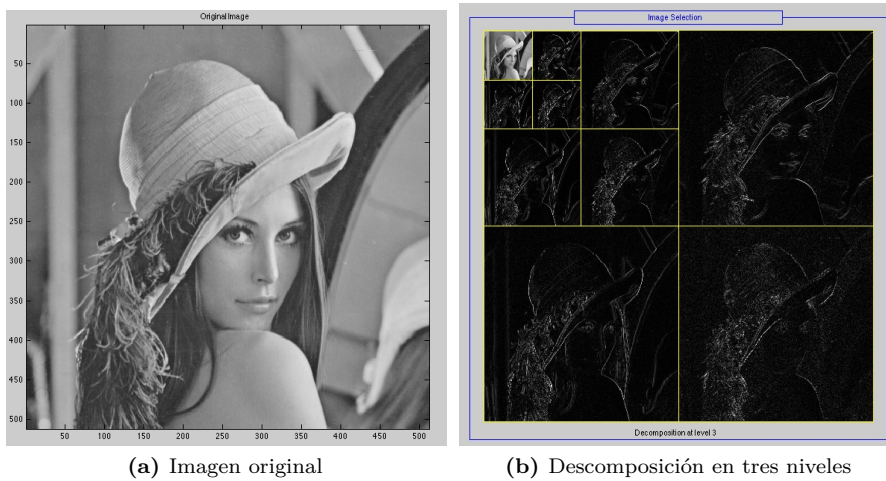


Figura 3.5: Ejemplo descomposición Wavelet en tres niveles.

3.2 EZW

El algoritmo EZW fue uno de los primeros algoritmos en mostrar el poder de las wavelets en la compresión de imágenes. Este algoritmo se dice que es embebido (o embedded del inglés) porque utiliza codificación progresiva para comprimir una imagen en una cadena de bits con una precisión cada vez mayor. Es decir, entre más bits agreguemos a la cadena de bits mayor será el contenido de detalles de la imagen decodificada. El esquema *zerotree* permite codificar de una forma concisa las posiciones de los valores más significativos que resultan del proceso de codificación embebido. El codificador EZW explota algunas propiedades de la transformada *wavelet* tales como que las imágenes de escenas naturales, en general, tienen un espectro paso-bajas. Es decir, cuando una imagen es transformada al dominio *wavelet* la energía de los coeficientes se compacta conforme el nivel de descomposición aumenta (o de manera equivalente, la energía es menor en las subbandas altas que en las subbandas bajas). De aquí también surge la estrategia de codificar los coeficientes de forma progresiva. El proceso de embebido utilizado por EZW es llamado codificación bit-plane y consiste de cinco pasos, los cuales se explican en el algoritmo 3.2.2.

Como podemos observar el algoritmo de codificación bit-plane [43] consiste en calcular la expansión binaria para los valores transformados y guardar la magnitud de los bits significativos en la expansión. Este procedimiento se puede

realizar tantas veces como sea necesario para obtener la magnitud de los coeficientes cuantizados $w_Q(m)$, los cuales son una aproximación de los valores de la magnitud de los coeficientes $|w(m)|$. En el proceso de decodificación los signos y los bits de salida pueden ser utilizados para construir una aproximación para cualquier grado de precisión deseado.

Algoritmo 3.2.2: Codificación bit-plane

PROCEDIMIENTO:

1. Inicialización. Se elige un umbral inicial, $T = T_0$, tal que todos los valores transformados satisfagan $|w(m)| < T_0$ y al menos un valor transformado satisfaga $|w(m)| \geq T_0/2$.
2. Actualización de umbral. $T_k = T_{k-1}/2$.
3. Pase significativo. Se hace una búsqueda a través de los valores no significativos usando el algoritmo base de orden de búsqueda. Se prueba cada valor como sigue:
 - if** $|w(m)| \geq T_k$
 - return** signo de $w(m)$
 - $w_Q(m) = T_k$
 - else if** $|w(m)| < T_k$
 - $w_Q(m)$ retiene su valor inicial de 0.
4. Pase de refinamiento. Búsqueda a través de los valores significativos encontrados con valores de umbral mayores T_j , para $j < k$ (si $k = 1$ este paso se salta). Para cada valor significativo $w(m)$, se hace lo siguiente:
 - if** $|w(m)| \in [w_Q(m), w_Q(m) + T_k]$
 - return** bit 0
 - $w_Q(m) = T_k$
 - else if** $|w(m)| \in [w_Q(m) + T_k, w_Q(m) + 2T_k]$
 - return** bit 1
 - $w_Q(m) = w_Q(m) + T_k$
5. Ciclo. Repetir pasos del 2 al 4.

La transformada *wavelet* se adapta muy bien a este tipo de codificación ya que como se mencionó este tipo de transformada provee representaciones dispersas de imágenes de escenas naturales, muchos de los coeficientes son cercanos a cero, y la mayor parte de la energía se concentra en las bajas frecuencias. Una vez descrito el proceso de codificación embebido de los coeficientes *wavelet*, podemos describir el método *zerotree* mediante el cual EZW transmite las posiciones de los coeficientes significativos. El método *zerotree* da un descripción implícita y compacta de la ubicación de los valores significativos. Antes de definir un *zerotree* primero definiremos un *quadtrees*, un *quadtrees* es un árbol

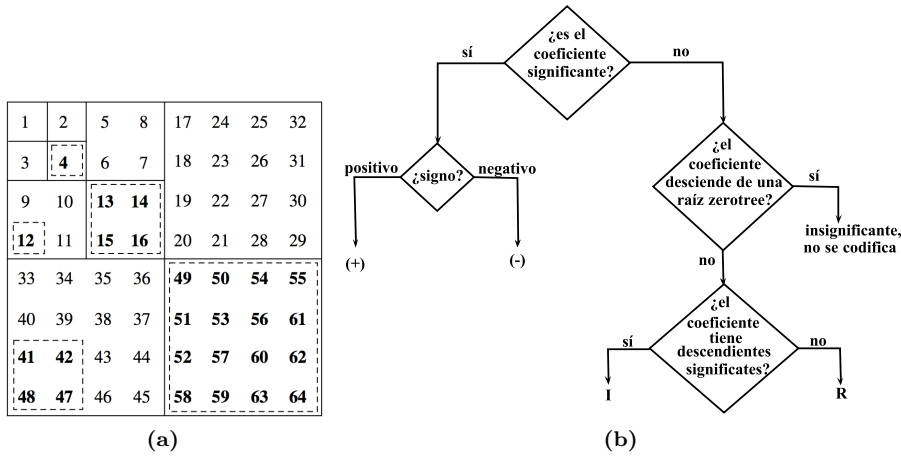


Figura 3.6: a) Orden de búsqueda zigzag para la transformada *wavelet*, con dos *quadtrees*. b) Clasificación de un coeficiente.

de ubicaciones en el dominio *wavelet* con raíz $[i, j]$, luego sus hijos establecidos en $[2i, 2j], [2i + 1, 2j], [2i, 2j + 1], [2i + 1, 2j + 1]$, y cada uno de sus hijos forma una raíz, por tanto también tiene asociados sus hijos, y así sucesivamente. La figura 3.6a muestra un ejemplo de dos *quadtrees*, encerrados en bloques con líneas discontinuas. Uno de ellos tiene raíz en el índice 12 y sus hijos en los índices $\{41, 42, 47, 48\}$. Este *quadtree* tiene dos niveles y se denota por $\{12 \mid 41, 42, 47, 48\}$. El otro *quadtree* tiene tres niveles, su raíz la tiene en el índice $\{4\}$, los hijos de su raíz en los índices $\{13, 14, 15, 16\}$, sucesivamente sus hijos en los índices $\{49, 50, \dots, 64\}$. Y es denotado por $\{4 \mid 13, \dots, 15 \mid 49, \dots, 64\}$. Una vez conocida la definición de un *quadtree* podemos definir un *zerotree*, un *zerotree* es un *quadtree* que, para un umbral T dado, tiene coeficientes *wavelet* insignificantes en cada una de sus ubicaciones.

Los *zerotrees* pueden proveer descripciones muy compactas de las ubicaciones de los valores insignificantes porque solo es necesario codificar un símbolo, tal como R , para indicar la ubicación de la raíz. El decodificador puede inferir que todas las ubicaciones en el *zerotree* tienen valores insignificantes, por tanto sus ubicaciones no son codificadas. Este esquema es muy útil en la compresión de wavelets porque la estructura multiresolución de la transformada *wavelet* de imágenes produce muchos zeros (especialmente con umbrales altos).

Ahora que tenemos la bases de la codificación *zerotree*, podemos completar el esquema de codificación EZW. El algoritmo EZW consiste en el simple reemplazo del *pase significativo* en la codificación bit-plane con el siguiente paso:

EZW paso 3: *Pase significativo*. Búsqueda a través de los valores insignificantes usando el algoritmo base de orden de búsqueda. Prueba cada valor como sigue:

$$\text{if } |w(m)| \geq T_k$$

```

return signo de  $w(m)$ 
 $w_Q(m) = T_k$ 
else if  $|w(m)| < T_k$ 
  return  $w_Q(m) = 0$ 
if  $m$  esta en el primer nivel
  return  $I$ 
else
  Busca a través del quadtrees con raíz  $m$ 
  if este quadtrees es zerotree
    return  $R$ 
  else
    return  $I$ 

```

Durante la búsqueda a través de un *quadtrees*, los valores que son encontrados como significativos en umbrales altos son tratados como ceros. Todos los descendientes de una raíz de ceros son saltados en el resto de la búsqueda en ese umbral.

El proceso antes descrito también puede ser consultado en el diagrama de la figura 3.6b. Como un ejemplo de este método consideremos la matriz de coeficientes *wavelet* de 8×8 con tres niveles de descomposición mostrada en la figura 3.7a. Ya que el coeficiente más grande es 63, el umbral inicial es $T_1 = 32$.

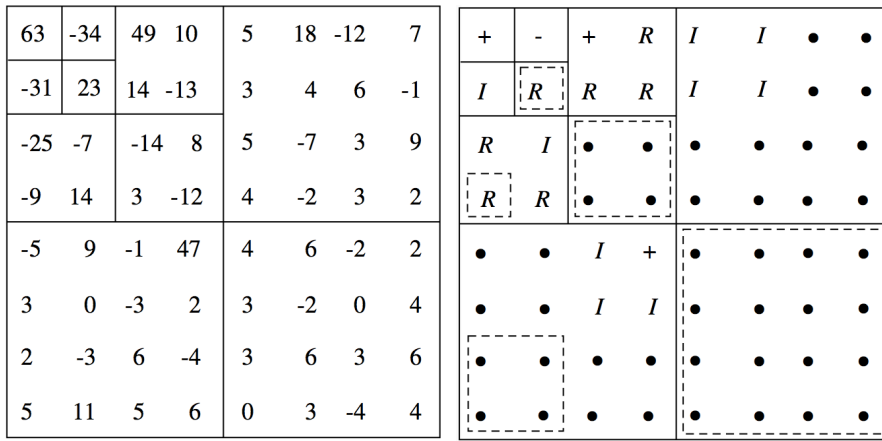
Empezamos con el orden de búsqueda establecido en la figura 3.6a. 63 es mayor que 32 por tanto obtiene signo +. -34 es mayor que 32 en valor absoluto y obtiene -. Continuamos con -31 que resulta menor en valor absoluto que 32. Sin embargo, recorriendo su *quadtrees*, observamos que no es una raíz de un *zerotree* porque existe un valor mayor de 47. Por lo tanto el símbolo asignado es I . Seguimos con 23 y vemos que es un *zerotree* por tanto se le asigna el símbolo R . Continuamos con el proceso siguiendo el orden de búsqueda y nos saltamos todos aquellos coeficientes para los cuales hemos establecido previamente que pertenecen a un *zerotree*. Los resultados de este primer paso significativo se muestran en la figura 3.7b. La salida del codificador después de este primer ciclo sería:

$$+ - IR + RRRRIRRIIII + II$$

correspondiente a una transformación cuantizada teniendo sólo valores entre ± 32 y $+32$ en cada ubicación marcada con un signo de + en la figura 3.7b, -32 en cada ubicación marcada por el signo - y cero en el resto de las ubicaciones. En el segundo ciclo, con umbral $T_2 = 16$, el resultado del pase significativo es mostrado en la figura 3.7c. Notemos, en particular, que el símbolo R esta en la décima posición del orden de búsqueda porque el signo + cae en la ubicación de un hijo en el ciclo anterior, entonces es tratado como cero. De allí, que la posición diez está en la raíz de un *zerotree*. También hay un paso de refinamiento hecho en el segundo ciclo. La salida del codificador en este segundo ciclo es entonces:

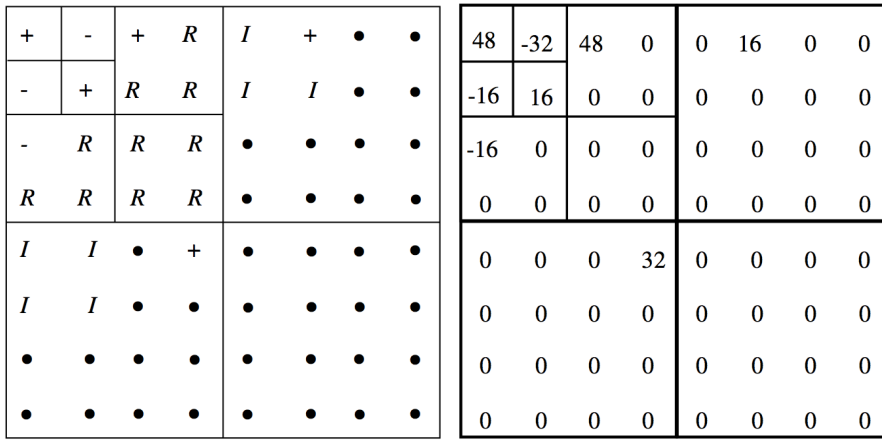
$$- + RRR - RRRRRRIII + IIII1010$$

con correspondientes coeficientes *wavelet* cuantizados mostrados en la figura 3.7d.



(a) coeficientes wavelet

(b) umbral, $T_1 = 32$



(c) umbral, $T_2 = 16$

(d) Cuantización al final de la segunda etapa

Figura 3.7: Etapas de EZW. a) Transformada *wavelet* con tres niveles de descomposición. b) Etapa 1, umbral=32. c) Etapa 2, umbral=16. d) Cuantización al final de la etapa 2.

Dado que en un mundo digital, es necesario transmitir solo bits. Una simple codificación de los símbolos del algoritmo EZW en bits podría utilizar un código como $+$ = 01, $-$ = 00, R = 10 y I = 11. En Shapiro [44], un codificador aritmético sin pérdidas es recomendado para obtener una mayor compresión de la cadena de bits del codificador.

3.3 Otras Técnicas de Reconstrucción de Imágenes

Una de las primeras técnicas utilizadas para la reconstrucción de imágenes médicas fue la tomografía computarizada (CT), la cual es una técnica en la que se utiliza una serie de mediciones con rayos X tomados en diferentes ángulos alrededor de un objeto. Ésta técnica tuvo un impacto revolucionario para el diagnóstico en medicina y también se ha utilizado exitosamente en aplicaciones para la industria. En 1972 Hounsfield patentó el primer escáner de CT y fue galardonado con el premio Nobel junto con Cormack por su invención en 1979. Desde entonces, nuevos desarrollos han producido escaneos más rápidos y mejoras tanto en el uso como en la calidad de las imágenes.

Una parte importante en esta historia ha sido el desarrollo de nuevos algoritmos de reconstrucción de imágenes. Aunque el problema de reconstrucción de la imagen en su más pura forma matemática fue resuelto por Johann Radon en 1917, el área da pie a la aparición de nuevos algoritmos.

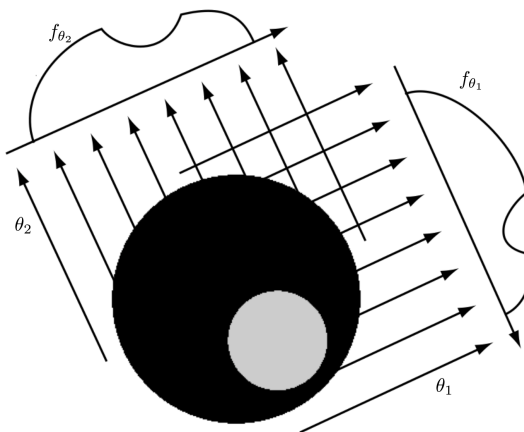


Figura 3.8: Geometría de un haz de rayos paralelos.

3.3.1 Transformada Radon

La proyección de un haz de rayos paralelos puede ser modelada como un conjunto de líneas, como se muestra en la figura 3.8. Un punto arbitrario en la proyección de una señal está dado por la suma a lo largo de la línea $x \cos \theta_k + y \sin \theta_k = \rho_j$. En el espacio continuo, la suma de rayos está dada por la integral de línea siguiente,

$$g(\rho_j, \theta_k) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta_k + y \sin \theta_k - \rho_j) dx dy. \quad (3.3.1)$$

debido a la propiedad del impulso: el valor de la ecuación es cero, a menos que el argumento del impulso δ sea cero, indicando que la integral es calculada solo

a lo largo de la línea $x \cos \theta_k + y \sin \theta_k = \rho_j$. Para el caso general de todos los valores de ρ y θ , la ecuación anterior se generaliza a

$$g(\rho, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) dx dy. \quad (3.3.2)$$

esta ecuación, la cual da la proyección de $f(x, y)$ a lo largo de una línea arbitraria en el plano xy , es conocida como la *transformada Radon*. La transformada Radon es la piedra angular en la reconstrucción de imágenes a partir de las proyecciones.

En el caso discreto, (3.3.2) se convierte en

$$g(\rho, \theta) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f[x, y] \delta[x \cos \theta + y \sin \theta - \rho] \quad (3.3.3)$$

donde x, y, ρ y θ son variables discretas. Si uno fija el valor de θ y varía el valor de ρ . La ecuación 3.3.3 simplemente suma los píxeles de $f(x, y)$ a lo largo de la línea de los valores especificados por estos dos parámetros. Incrementando a través de todos los valores de ρ necesarios para generar la imagen se obtiene una proyección. Cambiando θ y repitiendo el procedimiento anterior se obtiene otra proyección, y así sucesivamente. La transformada Radon, $g(\rho, \theta)$, también es conocida como el *sinograma* de la imagen y es mostrada en un sistema de coordenadas ρ, θ . Al igual que la transformada de Fourier, la transformada Radon contiene información necesaria para reconstruir $f(x, y)$. Un sinograma puede ser fácilmente interpretado para regiones simples, pero también puede complicarse cuando la geometría del objeto es compleja.

El principal objetivo de la CT es obtener una representación 3-D de un volumen a través de sus proyecciones. Esto se lleva a cabo mediante la retroproyección (BP) de cada proyección y después se suman todas las retroproyecciones para generar una imagen (o rebanada). El apilamiento de todas estas imágenes produce el volumen 3-D. La imagen formada por una sola retroproyección con el ángulo θ de la transformada Radon es calculada como sigue

$$f_{\theta}(x, y) = g(x \cos \theta + y \sin \theta, \theta) \quad (3.3.4)$$

La imagen final es formada mediante la integración a través de todas las imágenes retroproyectadas:

$$f(x, y) = \int_0^{\pi} f_{\theta}(x, y) d\theta \quad (3.3.5)$$

Para el caso discreto, la integral se transforma en la suma de todas las imágenes retroproyectadas:

$$f(x, y) = \sum_{\theta=0}^{\pi} f_{\theta}(x, y) \quad (3.3.6)$$

Una imagen retroproyectada formada en la manera descrita, algunas veces se le refiere como *laminograma*.

Una de las desventajas del método de retroproyección es que las imágenes resultan emborronadas, por tanto dificulta su interpretación, especialmente en el caso de imágenes médicas.

3.3.2 Reconstrucción utilizando Retroproyecciones Filtradas

Como se mencionó anteriormente, si las retroproyecciones son obtenidas directamente las imágenes sufren un emborronamiento. Para eliminar este emborronamiento lo que comúnmente se hace es filtrar las proyecciones antes de ser retroproyectadas. Antes de introducir la técnica de retroproyecciones filtrada, enunciaremos el siguiente teorema:

Teorema 3.3.1 (Rebanada de Fourier). *La transformada de Fourier de proyecciones satisface*

$$\forall \theta \in [0, \pi], \forall \omega \in \mathbb{R} \quad \hat{g}(\omega, \theta) = \hat{f}(\omega \cos \theta, \omega \sin \theta)$$

donde, $\hat{f}(u, v)$ denota la transformada 2-D de $f(x, y)$.

Este teorema establece que la transformada de Fourier de una proyección es una rebanada de la región de la transformada de Fourier en 2-D de la cual la proyección fue obtenida.

La técnica de retroproyección filtrada se obtiene a través de la transformada inversa de Fourier. La transformada inversa de Fourier en 2-D de $\hat{f}(u, v)$ se define como

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \hat{f}(u, v) e^{j2\pi(ux+vy)} du dv \quad (3.3.7)$$

realizando el cambio de variables con $u = \omega \cos \theta$ y $v = \omega \sin \theta$, las diferenciales se convierten en $du dv = \omega d\omega d\theta$, entonces se puede expresar (3.3.7) en coordenadas polares como:

$$f(x, y) = \int_0^{2\pi} \int_0^{\infty} \hat{f}(\omega \cos \theta, \omega \sin \theta) e^{j2\pi\omega(x \cos \theta + y \sin \theta)} \omega d\omega d\theta \quad (3.3.8)$$

utilizando el teorema 3.3.1 en la ecuación anterior obtenemos,

$$f(x, y) = \int_0^{2\pi} \int_0^{\infty} \hat{g}(\omega, \theta) e^{j2\pi\omega(x \cos \theta + y \sin \theta)} \omega d\omega d\theta \quad (3.3.9)$$

si dividimos la integral en dos expresiones una para θ en el rango de 0 a π y la otra en el rango de π a 2π , y usamos el hecho que $\hat{g}(\omega, \theta + \pi) = \hat{g}(-\omega, \theta)$, podemos reescribir (3.3.9) como:

$$f(x, y) = |\omega| \int_0^{\pi} \int_{-\infty}^{\infty} \hat{g}(\omega, \theta) e^{j2\pi\omega(x \cos \theta + y \sin \theta)} d\omega d\theta \quad (3.3.10)$$

si integramos respecto a ω , el término $x \cos \theta + y \sin \theta$ es una constante, la cual es una recta de la forma $x \cos \theta + y \sin \theta = \rho$. Sustituyendo en (3.3.10) tenemos

$$f(x,y) = \int_0^\pi \left[\int_{-\infty}^{\infty} |\omega| \hat{g}(\omega, \theta) e^{j2\pi\omega\rho} \right]_{\rho=x \cos \theta + y \sin \theta} d\theta \quad (3.3.11)$$

La expresión interna en (3.3.11) tiene forma de una transformada inversa de Fourier en 1-D a excepción del término $|\omega|$, el cual puede ser identificado como un filtro *rampa* en una dimensión. Esta función no es integrable porque su amplitud se extiende hasta ∞ en ambas direcciones, por tanto la transformada inversa esta indefinida. En la práctica, lo que comúnmente se hace es limitar la banda del filtro con una ventana. Algunas de las ventanas más utilizadas son las de Hamming y Hann, las cuales se definen para una ventana discreta de M puntos como:

$$h(\omega) = \begin{cases} c + (c - 1) \cos \frac{2\pi\omega}{M-1} & 0 \leq \omega \leq (M - 1) \\ 0 & \text{otro caso} \end{cases} \quad (3.3.12)$$

cuando $c = 0,54$, la función corresponde a la ventana Hamming y, cuando $c = 0,5$ corresponde a la ventana de Hann. La diferencia entre ambas en aplicaciones de procesamiento de imágenes, es prácticamente imperceptible. En resumen la imagen retroproyectada es obtenida como sigue:

1. Calcular la transformada de Fourier en 1-D de cada proyección.
2. Multiplicar cada transformada de Fourier por el filtro $|\omega|$, limitado con una ventana adecuada.
3. Obtener la transformada inversa en 1-D de cada transformada filtrada.
4. Sumar todas las transformadas inversas del paso 3.

Debido a que un filtro es utilizado en la reconstrucción, este método se llama retroproyección filtrada (FBP). El resultado de la imagen reconstruida es mucho más nítido ya que el filtro elimina el emborronamiento.

3.4 Resumen

En ese capítulo explicamos un método de compresión de imágenes en 2-D y un método de reconstrucción de imágenes a partir de proyecciones. El algoritmo EZW explota las características de la transformada *wavelet* para codificar los coeficientes más significativos del objeto, este algoritmo es muy similar a la codificación bit-plane, excepto en un paso. La gran ventaja de este algoritmo es que usa codificación progresiva por lo que uno puede agregar tantos bits como quiera hasta obtener la aproximación deseada. La transformada Radon es muy utilizada para reconstruir imágenes a partir de rayos X, sin embargo, no es la única forma en la cual se pueden reconstruir imágenes médicas, uno puede revisar [45] para conocer más acerca de este tema. La retroproyección filtrada es muy popular en implementaciones reales de reconstrucción de imágenes.

Sin embargo, tiene sus inconvenientes, uno es la complejidad computacional que alenta el proceso de reconstrucción. Esto es en parte rectificado por un método de retroproyección rápida. Por otro lado, la calidad de la imagen es también un problema. Objetos metálicos pequeños pueden inducir artefactos en la imagen reconstruida. Comparada con técnicas de reconstrucción algebraicas, la retroproyección filtrada no es adaptable para ausencia de datos y efectos de oclusión parcial. Sin embargo, las técnicas algebraicas son típicamente más costosas computacionalmente.

Capítulo 4

Simulaciones y resultados

En este capítulo haremos la evaluación del muestreo compresivo respecto a otros métodos comúnmente utilizados en el procesamiento de señales. La primera parte consiste en la comparación del muestreo compresivo con el algoritmo de compresión EZW. Después se harán algunas pruebas para evaluar la robustez del algoritmo de reconstrucción CoSaMP ante el ruido. Y por último compararemos el muestreo compresivo con los métodos de retroproyecciones vistos en el capítulo 3, para reconstruir imágenes tipo médicas. Todos los algoritmos fueron programados en el lenguaje de programación C y el lenguaje Cython [46], excepto el de retroproyecciones que fue simulado en MATLAB.

Para evaluar la calidad de las imágenes reconstruidas utilizaremos una medida objetiva promedio conocida como la relación ó razón señal pico a ruido (PSNR). La razón señal pico a ruido, es la proporción de la potencia máxima de una señal y la potencia del ruido que afecta la fidelidad de su representación. Para obtener la PSNR, primero consideremos el error cuadrático medio (MSE), ó, distorsión D , como:

$$D = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N \mathbb{E}[|I(i,j) - \tilde{I}(i,j)|^2], \quad (4.0.1)$$

donde I es la imagen original, \tilde{I} es la imagen reconstruida y M, N son las dimensiones del tamaño de la imagen. La PSNR en decibeles se define como:

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAX}_I^2}{D} \right) [\text{dB}]. \quad (4.0.2)$$

donde MAX_I es el máximo valor pico-a-pico en la imagen (típicamente 255 para imágenes de 8 bits).

4.1 Resultados de las simulaciones entre el CS y EZW

Para evaluar el desempeño del muestreo compresivo, reconstruiremos una imagen de 512×512 píxeles a partir de varios niveles de compresión y la compararemos con el algoritmo de compresión en el dominio *wavelet* explicado en el capítulo 3 EZW. Las imágenes de escenas naturales no tienen una representación dispersa en espacio por lo que es necesario buscar una base de expansión en la que sí lo sean. Como vimos anteriormente la transformada *wavelet* provee una representación dispersa para este tipo de señales, además su tiempo de procesamiento tanto para la transformada directa como inversa es de $O(n)$.

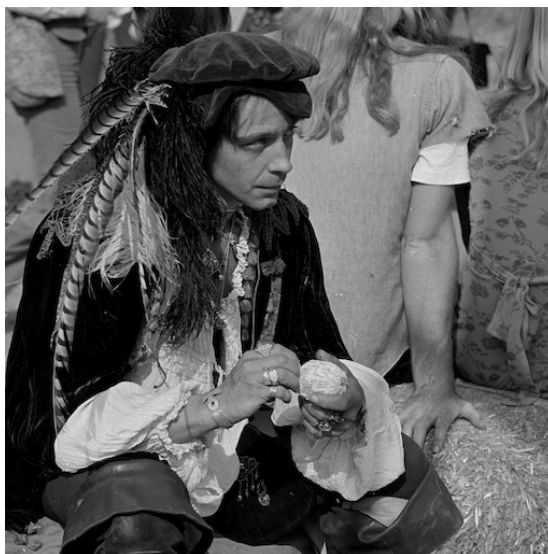


Figura 4.1: Imagen de prueba “man”.

La imagen utilizada para las simulaciones se muestra en la figura 4.1, nos referiremos a esta imagen como “man”, en las simulaciones con la imagen “man” la base de representación Ψ será la transformada *wavelet*. Una vez que obtenemos la representación dispersa de la imagen procedemos a realizar la adquisición de la información con el *operador de información* Φ , en este caso el operador de información está conformado por las funciones *noiselet* definidas en el capítulo 2. La elección de este par de bases (Ψ, Φ) se debe a que son incoherentes, por lo tanto el número de mediciones de acuerdo al principio de coherencia mutua está acotado por $O(d \log n)$, donde d es el nivel de dispersidad de la imagen y n es el tamaño de la imagen en este caso $M \times N$. El proceso de reconstrucción se hará con el algoritmo greedy CoSaMP estudiado en 2.2.3.

La figura 4.2 muestra dos gráficas del nivel de compresión vs la PSNR. Los radios de compresión utilizados fueron 128:1, 64:1, 32:1, 16:1, 8:1, 4:1 y 2:1

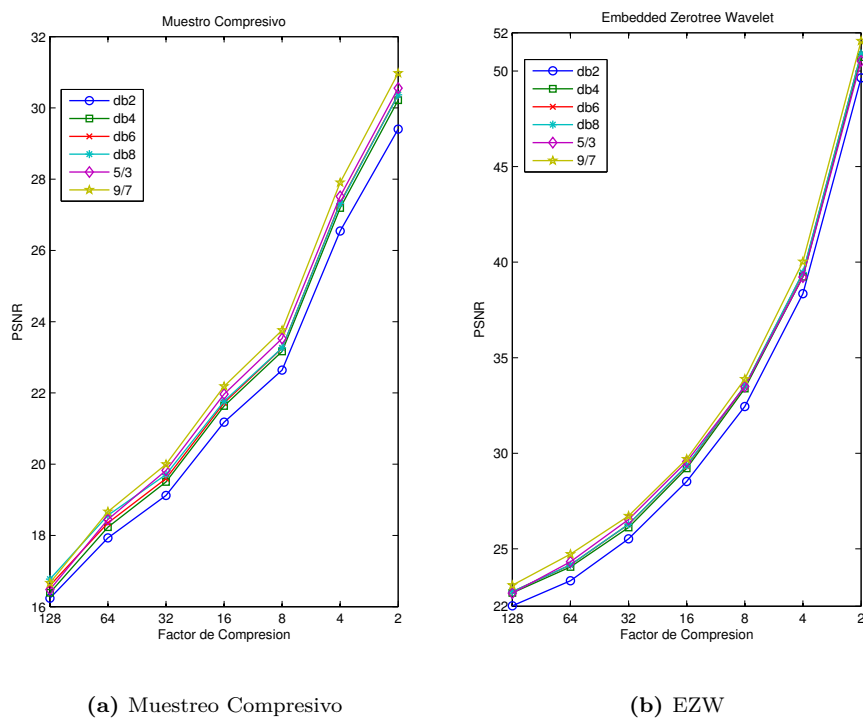


Figura 4.2: Comparación de métodos de compresión con diferentes funciones *wavelet*. a) Gráfica del nivel de compresión .vs. PSNR, (muestreo compresivo). b) Gráfica del nivel de compresión vs PSNR, (método EZW).

tanto para el muestreo compresivo como para el algoritmo EZW. Y también se utilizaron diferentes funciones *wavelet* entre ellas las *wavelets* Daubechies 2, 4, 6, y 8 y las famosas *wavelets* biortogonales “5/3” y “9/7” utilizadas en el estándar JPEG2000. En primer lugar podemos observar que entre mejor se adapte la función *wavelet* a la naturaleza de la señal, ésta provee no sólo una representación dispersa, sino sus coeficientes capturan mejor la información de la señal. De allí que las *wavelets* biortogonales presenten una PSNR mayor en la reconstrucción de la imagen que las *wavelets* Daubechies. En particular la *wavelet* “9/7” es la que mejor se adapta a la señal, proveyendo la mayor PSNR tanto para el esquema de muestreo compresivo, como para el esquema EZW. Por otro lado, podemos ver que el algoritmo EZW alcanza valores de la PSNR mucho más altos para todos los niveles de compresión que el esquema CS. La pregunta natural sería ¿porqué existe tanta diferencia entre los resultado si ambos algoritmos reconstruyen la imagen con los coeficientes más importantes? Por un lado el muestreo compresivo da una reconstrucción semejante al de la aproximación no lineal, es decir, da una reconstrucción con los mejores coeficientes. La compresión en el CS simplemente se fija de acuerdo al número de mediciones, por ejemplo si la señal es de longitud $n = 1024$ y se toma un número de mediciones $m = 512$ entonces la señal es comprimida a un radio 2:1. De estas 512 muestras el algoritmo CoSaMP reconstruye una señal d -dispersa, d se obtiene de la expresión de coherencia mutua $m \approx d \log n \Rightarrow d \approx m / \log n$, sustituyendo los valores correspondientes resulta que el algoritmo CoSaMP nos da una aproximación $d \approx 170$ y esta aproximación tiene un error equivalente al de la aproximación no lineal. En esencia el algoritmo EZW hace lo mismo, escanea los coeficientes en un orden predefinido y codifica la ubicación de los elementos más importantes, pero adicionalmente hace uso de un codificador aritmético. En teoría de la información, el codificador aritmético es conocido por proveer mayor capacidad de compresión sobre otros codificadores estadísticos, tales como el de Huffman y sus variantes. Es gracias a esta técnica que el algoritmo EZW puede comprimir mucho más la información y guardar más detalles de la imagen.

Las figuras 4.3 y 4.4 muestran las imágenes obtenidas con un radio de compresión 4:1 tanto para el esquema EZW como para el esquema CS. Las imágenes reconstruidas con el esquema EZW tienen una mayor calidad que aquellas reconstruidas con el muestreo compresivo. También cabe destacar que la calidad de la imagen mejora entre filtro y filtro, aunque tengan el mismo nivel de compresión, esta diferencia aunque es más notable en las imágenes del muestreo compresivo también esta presente en el esquema EZW. Por ejemplo si uno observa las imágenes 4.3a y 4.3f, se puede ver que la imagen reconstruida a partir de coeficientes Daubechies 2, esta más “píxeleada” porque la forma del filtro es cuadrada, en el caso de la *wavelet* “9/7” como el filtro es más suave, la imagen resultante mejora su aspecto.

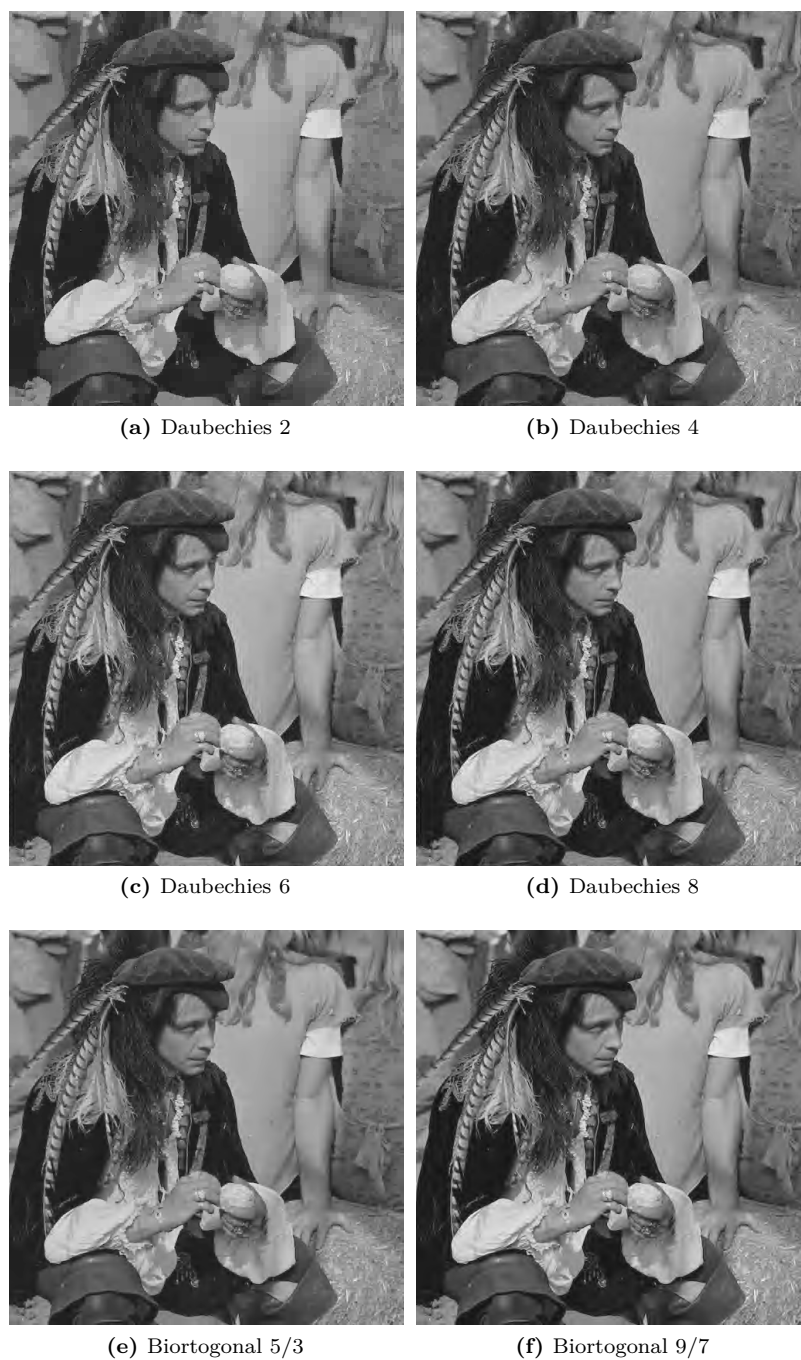


Figura 4.3: Imágenes reconstruidas con un factor de compresión 4:1 utilizando el esquema de muestreo compresivo.



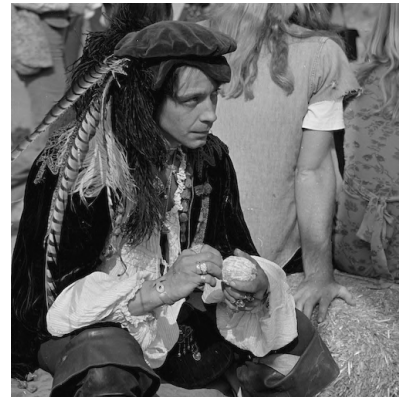
(a) Daubechies 2



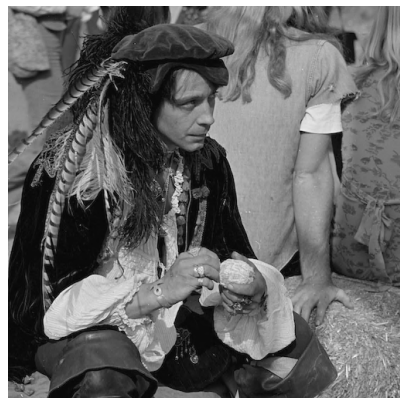
(b) Daubechies 4



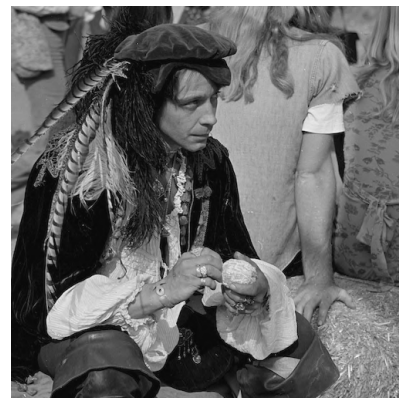
(c) Daubechies 6



(d) Daubechies 8



(e) Biortogonal 5/3



(f) Biortogonal 9/7

Figura 4.4: Imágenes reconstruidas con un factor de compresión 4:1 utilizando el algoritmo EZW.

Para evaluar el desempeño del algoritmo de reconstrucción CoSaMP contaminamos las mediciones $\mathbf{y} = \Phi\mathbf{x}$ con un ruido Gaussiano \mathbf{e} , con diferentes valores de desviación estándar σ . Es decir el algoritmo reconstruye la información a partir de $\mathbf{z} = \Phi\mathbf{x} + \mathbf{e} = \mathbf{y} + \mathbf{e}$. En este experimento se seleccionó la *wavelet* biortogonal “9/7” porque experimentalmente fue la que proveía una mejor calidad en la reconstrucción de la imagen.



(a) Reconstrucción con muestras contaminadas con $\mu = 0$, $\sigma = 0.01$. (b) Reconstrucción con muestras contaminadas con $\mu = 0$, $\sigma = 0.015$.



(c) Reconstrucción con muestras contaminadas con $\mu = 0$, $\sigma = 0.1$. (d) Reconstrucción con muestras contaminadas con $\mu = 0$, $\sigma = 0.15$.

Figura 4.5: Imágenes reconstruidas a partir de mediciones contaminadas con ruido Gaussiano.

La figura 4.5 muestra las imágenes reconstruidas a partir de 9072 coeficientes significativos, cabe hacer notar que el valor de estas mediciones está normalizado, por tanto sus valores son muy pequeños, casi todos son cercanos a cero. La imagen 4.5a no sufre tanta distorsión porque el nivel de ruido es pequeño comparado con el valor de las mediciones, sin embargo cuando la desviación estándar del ruido es cercana a, o mayor a, $\sigma = 0.1$ la imagen recuperada resulta muy distorsionada. Las imágenes reconstruidas se deterioran mucho debido a que la energía no recuperable de la señal se ve incrementada por dos factores: uno es porque la señal es compresible, y otro porque el ruido agregado a las

muestras disminuye la razón señal a ruido.

Desviación estándar σ del ruido e	PSNR [dB]
0.01	26.83
0.015	24.18
0.1	17.31
0.15	16.28

Tabla 4.1: PSNR de la imágenes reconstruida a partir de muestras contaminadas.

La tabla 4.1 muestra los valores de la PSNR obtenida de las imágenes reconstruidas a partir de las muestras contaminadas con ruido Gaussiano. Como podemos observar si el nivel del ruido no es muy grande las imágenes resultantes poseen buena calidad considerando que sólo son reconstruidas con 9072 coeficientes del total de 262144.

4.2 Resultados de las simulaciones entre el CS y retroproyecciones

Como último ejemplo realizaremos la reconstrucción de imágenes formadas por elipses, o también conocidas como fantasmas. Este tipo de imágenes son dispersas en espacio por lo que no tienen que ser transformadas a otro dominio. Las imágenes prueba que utilizaremos se muestran en las figuras 4.6a y 4.6b. El operador de muestreo Ω estará formado por un patrón en forma de estrella en el plano de frecuencia; este se muestra en la figura 4.6c. Se escogió este patrón para poderlo comparar con los métodos de retroproyección y retroproyección filtrada. Para la reconstrucción de la señal con este esquema de muestreo compresivo se utilizará un programa de optimización en lugar de CoSaMP. Este programa de optimización consiste en minimizar la norma de la variación total (TV) del objeto g en dos dimensiones. La norma de la variación total se define para el caso discreto como,

$$\|g\|_{TV} = \sum_{x,y} \sqrt{|D_1g[x,y]|^2 + |D_2g[x,y]|^2}, \quad (4.2.1)$$

donde $g[x,y]$, $0 \leq x \leq M-1$, $0 \leq y \leq N-1$, D_1 es la diferencia finita $D_1g = g[x,y] - g[x-1,y]$ y $D_2g = g[x,y] - g[x,y-1]$. Entonces para recuperar f a partir de muestras parciales de Fourier, se encuentra la solución \tilde{f} al problema de optimización [10],

$$\min \|g\|_{TV} \quad \text{sujeto a} \quad \hat{g}(\omega) = \hat{f}(\omega) \quad \text{para todo } \omega \in \Omega \quad (4.2.2)$$

lo que hace este programa de optimización es buscar la solución \tilde{f} con mínima variación total y también busca aquellos coeficientes que se acoplen a los coeficientes desconocidos del objeto g . De esta forma el programa borra parcialmente

los artificios producidos por los coeficientes desconocidos y al mismo tiempo mantiene la fidelidad de los coeficientes conocidos, bajo las restricciones de los coeficientes de Fourier de la reconstrucción.

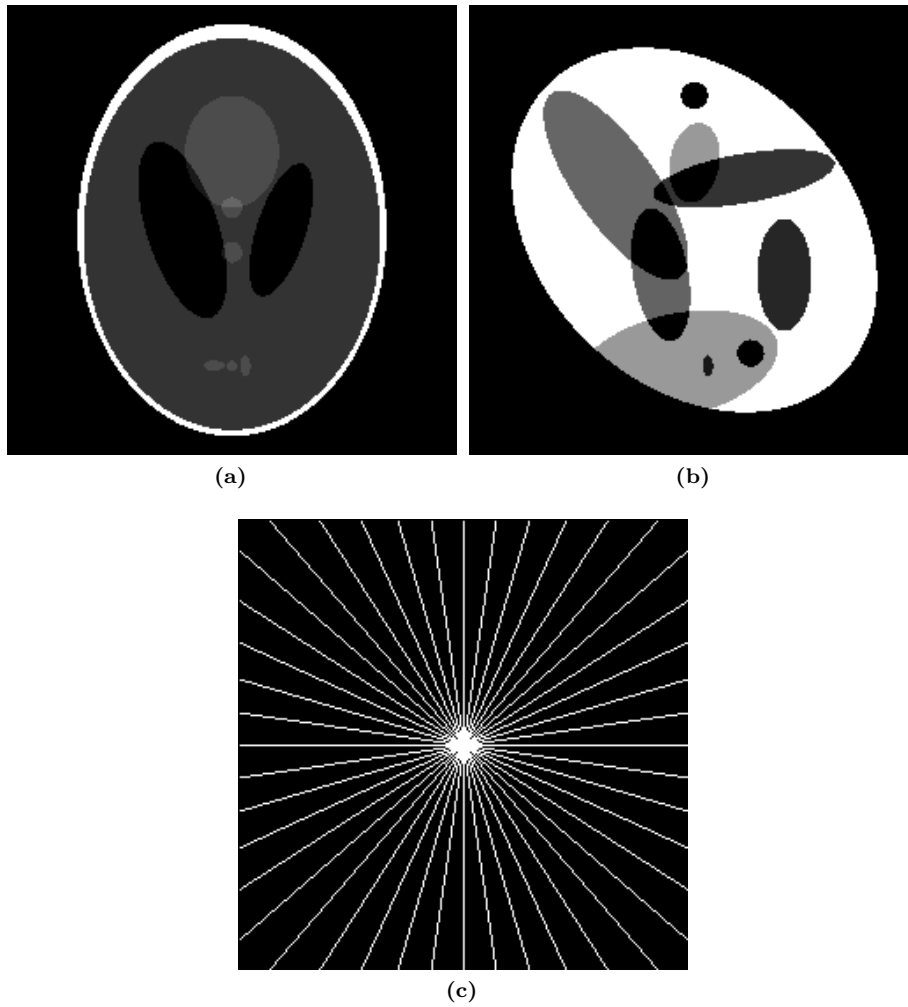


Figura 4.6: Imágenes de prueba. a) Imagen de prueba conocido como fantasma Logan-Shepp. b) Imagen de prueba formada por 10 elipses. c) Dominio de muestreo en plano de frecuencia; coeficientes de Fourier son recolectados a lo largo de aproximadamente 22 líneas radiales

La figura 4.7b muestra el resultado de las imágenes reconstruidas del fantasma Shepp-Logan con 22 líneas radiales utilizando como sensado coeficientes parciales de Fourier y como reconstrucción el programa de optimización (4.2.2). La diferencia entre la imagen original y la reconstruida es imperceptible, además la PSNR es cercana a 50 [dB]. Las figuras 4.7c y 4.7d muestran las imágenes reconstruidas con retroproyección y retroproyección filtrada, respectivamente. Ambas figuras fueron reconstruidas a partir de 22 proyecciones, la calidad de ambas imágenes es mala ya que tienen muchos artefactos debido al submuestreo. Las figuras 4.7e y 4.7f muestran las mismas imágenes reconstruidas con el método de retroproyección y retroproyección filtrada, pero ahora a partir de 360 proyecciones; el aspecto de ambas imágenes mejora bastante sin embargo la versión no filtrada resulta muy emborronada dificultado su interpretación. La figura 4.8 muestra otro ejemplo de una imagen formada por diez elipses, que es reconstruida bajo las mismas condiciones que en el ejemplo anterior utilizando muestreo compresivo, retroproyecciones y retroproyecciones filtradas. Los resultados de reconstrucción con CS son superiores que el de BP y FBP. Esto nos indica que aunque el muestreo compresivo no es tan competitivo en comprimir imágenes de escenas naturales como otros métodos de compresión, sí lo es para reconstruir imágenes tipo médicas a partir de un número mucho menor de muestras.

4.3 Resumen

En este capítulo hicimos algunos experimentos para evaluar el desempeño del muestreo compresivo. En primer lugar lo comparamos con el método EZW y concluimos que debido a que el algoritmo EZW hace uso de técnicas de la teoría de la información puede alcanzar un tasa de compresión mucho mayor, por tanto puede guardar mayor nivel de detalle de esta. Ambos algoritmos reconstruyen la información con los *mejores* elementos de la base de expansión, sin embargo, debido a que el algoritmo EZW utiliza un codificador aritmético este puede alcanzar mayores órdenes de compresión con mayores niveles de detalle. Por otro lado el algoritmo CoSaMP resultó ser robusto respecto al ruido, sin embargo se observó que el nivel de calidad de la imagen disminuye debido a que la distorsión no sólo es provocada por el ruido, sino también por el hecho de que la señal no es completamente dispersa. Por último, en la parte final realizamos algunos experimentos con imágenes dispersas en espacio, por lo tanto no necesitaban ser transformadas a otro dominio. Las mediciones estaban formadas por coeficientes parciales de Fourier y se reconstruyeron a partir de un programa de optimización convexa. Este método de CS fue comparado con los métodos de reconstrucción de imágenes con retroproyecciones, y retroproyecciones filtradas. El método de muestreo CS resultó superior por una parte porque el programa de optimización minimiza las oscilaciones de la imagen provocadas por el submuestreo, y por otra parte porque los algoritmos de retroproyección simplemente consideran que los elementos que no son vistos por proyecciones son nulos, de allí que surjan varios artefactos en la imagen reconstruida.

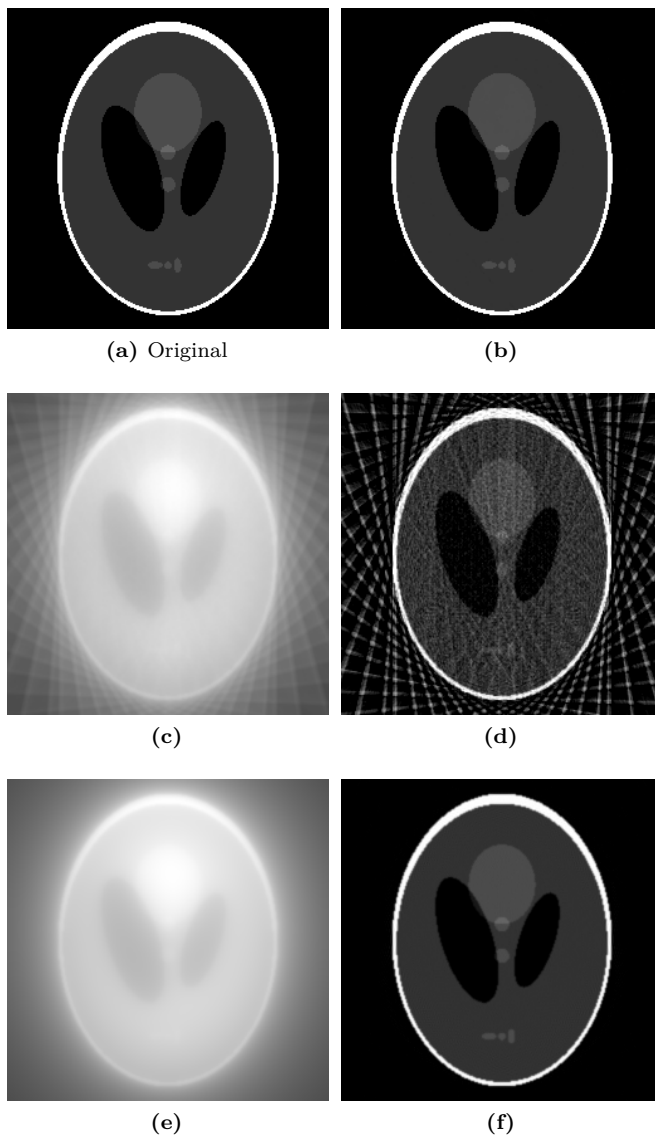


Figura 4.7: Imágenes de Reconstruidas. a) Imagen original fantasma Shepp-Logan. b) Reconstrucción con CS a partir de 22 líneas radiales, PSNR = 48.89[dB]. c) Reconstrucción con BP a partir de 22 proyecciones, PSNR = 4.78 [dB]. d) Reconstrucción con FBP a partir de 22 proyecciones, PSNR = 14.42 [dB]. e) Reconstrucción con BP a partir de 360 proyecciones, PSNR = 4.78 [dB]. f) Reconstrucción con FBP a partir de 360 proyecciones, PSNR = 20.89 [dB].

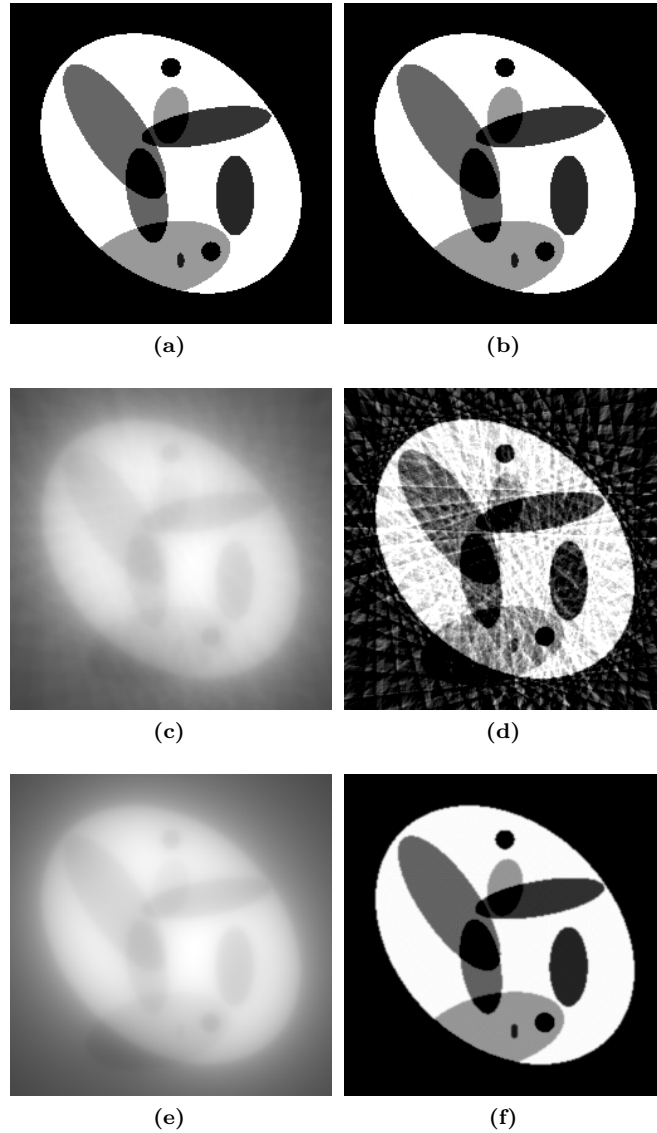


Figura 4.8: Imágenes de Reconstruidas. a) Imagen original formada por diez elipses. b) Reconstrucción con CS a partir de 22 líneas radiales, PSNR = 47.69[dB]. c) Reconstrucción con BP a partir de 22 proyecciones, PSNR = 7.06 [dB]. d) Reconstrucción con FBP a partir de 22 proyecciones, PSNR = 13.86 [dB]. e) Reconstrucción con BP a partir de 360 proyecciones, PSNR = 7.02 [dB]. f) Reconstrucción con FBP a partir de 360 proyecciones, PSNR = 21.26 [dB].

Capítulo 5

Conclusiones y trabajo futuro

En el desarrollo de este trabajo se logró implementar un esquema de muestreo compresivo en *software* bajo la plataforma de programación Python. El esquema implementado tiene una complejidad de $O(n \cdot \log n)$, la etapa que tiene mayor complejidad es la correspondiente a la reconstrucción de la información, es decir, el algoritmo CoSaMP. En el capítulo 2 se realizó un análisis del tiempo de ejecución de este algoritmo y se explicaron sus características sobre otros algoritmos de tipo *greedy*. En base a las simulaciones se pudo concluir que este algoritmo es robusto ya que permite reconstruir las señales a partir de mediciones ruidosas $\mathbf{y} = \Phi \mathbf{x} + \mathbf{e}$, en a lo más $6d$ iteraciones, con un error:

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_2 \leq C \left(\frac{1}{\sqrt{d}} \cdot \|\mathbf{x} - \mathbf{x}_s\|_1 + \|\mathbf{e}\|_2 \right)$$

es decir el nivel de calidad de la imagen disminuye debido a que la distorsión no sólo es provocada por el ruido, sino también por el hecho de que la señal no es completamente dispersa.

Para la compresión de imágenes naturales de tipo televisión, el esquema de muestreo compresivo resultó no ser tan competitivo como lo es el algoritmo de compresión EZW. Esto se debe principalmente a que el algoritmo EZW utiliza otras herramientas de la teoría de la información para codificar la información, una de ellas es el uso de un codificador aritmético, el cual permite alcanzar proporciones de compresión mayores, manteniendo un mayor nivel de detalle en la imagen comprimida. La compresión con el muestreo compresivo simplemente es fijada por el número de muestras y a partir de las muestras el algoritmo de reconstrucción provee una señal con los mejores d coeficientes de la base de expansión, es decir, equivalente a la aproximación no lineal. En esencia el algoritmo EZW hace lo mismo, escanea los coeficientes en un orden predefinido y codifica la ubicación de los elementos más importantes, pero con la adición del codificador aritmético. Por otro lado la adquisición de la información en el muestreo compresivo es bastante sencilla ya que lo único que se hace es correlacionar la señal de interés con formas de onda que se fijan independientemente de la estructura de la señal.

Finalmente, aunque el muestreo compresivo no es tan competitivo para imágenes de escenas naturales, para otro tipo de imágenes sí lo es, por ejemplo, algunas imágenes médicas que son dispersa en espacio. El esquema de muestreo CS utilizado en la reconstrucción de “fantasmas” resultó superior que los métodos basados en retroproyecciones, en parte porque el programa de optimización minimiza las oscilaciones de la imagen provocadas por el submuestreo y los algoritmos de retroproyección simplemente consideran que los elementos que no son vistos por proyecciones son nulos de allí que surjan varios artefactos en la imagen reconstruida. Aunque, la retroproyección filtrada es muy popular en implementaciones reales de reconstrucción de imágenes, tiene sus inconvenientes, uno de ellos es la complejidad computacional que alenta el proceso de reconstrucción. Esto es en parte corregido por un método de retroproyección rápida. Otro inconveniente es la calidad de la imagen, ya que objetos metálicos pequeños pueden inducir artefactos en la imagen reconstruida.

5.1 Trabajo a Futuro

Dentro de las simulaciones de la reconstrucción de imágenes con el muestreo compresivo se observó que entre mejor sea la base de expansión de la imagen, mejor será el nivel de detalle de la imagen reconstruida a partir de las mediciones. Por esta razón sería conveniente explorar otro tipo de representaciones orientadas geoméricamente tales como las “X-lets”, u otras transformadas redundantes que provean una representación dispersa de la imagen. Por otro lado se utilizó un algoritmo de tipo *greedy* en las simulaciones porque lo que se buscaba era obtener mayor rapidez en la reconstrucción, sin embargo los algoritmos de optimización en general proveen una mayor calidad de la señal reconstruida. Por tanto uno podría explorar con más detalle estos métodos, de tal manera que para alcanzar mayor rapidez en la reconstrucción uno podría implementar estos métodos con programación en paralelo. En la actualidad existen algunas plataformas de desarrollo en paralelo como CUDA y OpenCL, que explotan la potencia de las tarjetas gráficas ya que el número de núcleos en estas tarjetas es del orden de cientos (o miles en las tarjetas más sofisticadas), en cambio en los procesadores de computadora de tipo estación de trabajo es del orden de la decena.

En la actualidad el área del muestreo compresivo es muy activa y grandes avances se hacen día a día, en un reporte técnico publicado por Emmanuel Candès a finales del 2010 se propone otro enfoque probabilístico para la adquisición y reconstrucción de la información, por lo que el lector interesado podría estudiar este nuevo enfoque [47].

Bibliografía

- [1] D. Donoho, M. Vetterli, R. DeVore, and I. Daubechies, “Data compression and harmonic analysis,” *Information Theory, IEEE Transactions on*, vol. 44, no. 6, pp. 2435–2476, 1998.
- [2] M. Vetterli and J. Kovacevic, *Wavelets and subband coding*. Englewood Cliffs, NJ, USA: Prentice Hall, 1995.
- [3] N. Ahmed, T. Natarajan, and K. Rao, “Discrete cosine transform,” *Computers, IEEE Transactions on*, vol. C-23, no. 1, pp. 90 – 93, 1974.
- [4] W. Pennebaker and J. Mitchell, *JPEG still image data compression standard*. Norwell, MA, USA: Kluwer academic publishers, 1993.
- [5] A. Skodras, C. Christopoulos, and T. Ebrahimi, “Jpeg2000: The upcoming still image compression standard,” in *Proceedings of the 11th Portuguese Conference on Pattern Recognition, Porto, Portugal*, pp. 359–366, Citeseer, 2000.
- [6] E. Candes and D. Donoho, “Curvelets: A surprisingly effective nonadaptive representation of objects with edges,” tech. rep., Stanford University, 1999.
- [7] M. N. Do and M. Vetterli, “The contourlet transform: an efficient directional multiresolution image representation,” *Image Processing, IEEE Transactions on*, vol. 14, no. 12, pp. 2091–2106, 2005.
- [8] E. Le Pennec and S. Mallat, “Sparse Geometric Image Representations with bandelets,” *Image Processing, IEEE Transactions on*, vol. 14, no. 4, pp. 423–438, 2005.
- [9] D. L. Donoho and M. Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via l1-minimization,” *Proceedings of the National Academy of Sciences (PNAS)*, vol. 100, no. 5, pp. 2197–2202, 2003.
- [10] E. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 489–509, 2006.

- [11] E. Candes and T. Tao, “Near-optimal signal recovery from random projections: Universal encoding strategies?,” *Information Theory, IEEE Transactions on*, vol. 52, no. 12, pp. 5406–5425, 2006.
- [12] E. J. Candès and J. K. Romberg, “Quantitative robust uncertainty principles and optimally sparse decompositions,” *Foundations of Computational Mathematics*, vol. 7, pp. 529–531, 2007.
- [13] D. Donoho, “Compressed sensing,” *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [14] E. Candes and M. Wakin, “An introduction to compressive sampling,” *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 21–30, 2008.
- [15] S. Mallat, *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 3rd ed., 2008.
- [16] M. N. Do, *Directional Multiresolution Image Representations*. PhD thesis, Department of Communication Systems, Swiss Federal Institute of Technology Lausanne, 2001.
- [17] E. Candes and J. Romberg, “Practical signal recovery from random projections,” in *Proc. SPIE Computational Imaging*, vol. 5674, pp. 76–86, 2005.
- [18] A. Cohen, W. Dahmen, and R. DeVore, “Compressed sensing and best k-term approximation,” *J. Amer. Math. Soc.*, pp. 211–231, 2009.
- [19] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.
- [20] Y. Tsaig and D. L. Donoho, “Extensions of compressed sensing,” *Signal Process.*, vol. 86, no. 3, pp. 549–571, 2006.
- [21] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk, “Single-pixel imaging via compressive sampling,” *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 83–91, 2008.
- [22] E. Candes, M. Rudelson, T. Tao, and R. Vershynin, “Error correction via linear programming,” in *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, pp. 668–681, 2005.
- [23] E. Candes and T. Tao, “The dantzig selector: statistical estimation when p is much larger than n,” *The Annals of Statistics*, vol. 35, no. 6, pp. 2313–2351, 2007.
- [24] E. Candes and T. Tao, “Decoding by linear programming,” *Information Theory, IEEE Transactions on*, vol. 51, no. 12, pp. 4203–4215, 2005.
- [25] M. Herman and T. Strohmer, “High-resolution radar via compressed sensing,” *Signal Processing, IEEE Transactions on*, vol. 57, no. 6, pp. 2275–2284, 2009.

- [26] D. L. Donoho and X. Huo, “Uncertainty principles and ideal atomic decomposition,” *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2845–2862, 2001.
- [27] E. Candès and J. Romberg, “Sparsity and incoherence in compressive sampling,” *Inverse problems*, vol. 23, p. 969, 2007.
- [28] R. Coifman, F. Geshwind, and Y. Meyer, “Noiselets* 1,” *Applied and Computational Harmonic Analysis*, vol. 10, no. 1, pp. 27–44, 2001.
- [29] M. Rudelson and R. Vershynin, “Sparse reconstruction by convex relaxation: Fourier and gaussian measurements,” in *Information Sciences and Systems, 2006 40th Annual Conference on*, (Princeton, NJ), pp. 207–212, 2006.
- [30] D. Needell and J. Tropp, “Cosamp: Iterative signal recovery from incomplete and inaccurate samples,” *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.
- [31] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *Information Theory, IEEE Transactions on*, vol. 53, no. 12, pp. 4655–4666, 2007.
- [32] D. L. Donoho, Y. Tsaig, I. Drori, and J. L. Starck, “Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit,” tech. rep., Department of Statistics, Stanford University, 2007.
- [33] D. Needell and R. Vershynin, “Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit,” *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 2, pp. 310–316, 2010.
- [34] D. Needell and R. Vershynin, “Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit,” *Foundations of Computational Mathematics*, vol. 9, no. 3, pp. 317–334, 2008.
- [35] J. Chen and D. L. Donoho, “Atomic decomposition by basis pursuit,” *SPIE International Conference on Wavelets*, vol. 20, no. 1, pp. 33–61, 1995.
- [36] A. C. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss, “Near-optimal sparse fourier representations via sampling,” in *Proceedings of the 34th annual ACM symposium on Theory of computing*, STOC ’02, (New York, NY, USA), pp. 152–161, ACM, 2002.
- [37] A. C. Gilbert, S. Muthukrishnan, and M. Strauss, “Improved time bounds for near-optimal sparse fourier representations,” in *Proceedings of SPIE Wavelets XI*, (San Diego, CA, USA), 2003.
- [38] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin, “Algorithmic linear dimension reduction in the l_1 norm for sparse vectors,” in *Allerton 2006 (44th Annual Allerton Conference on Communication, Control, and Computing)*, 2006.

- [39] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin, “One sketch for all: fast algorithms for compressed sensing,” in *Proceedings of the 39th annual ACM symposium on Theory of computing*, STOC '07, (New York, NY, USA), pp. 237–246, ACM, 2007.
- [40] Z. Zhang, *Matching Pursuit*. PhD thesis, Courant Institute, New York University, 1993.
- [41] Å. Björck, *Numerical Methods for Least Squares Problems*. Philadelphia: SIAM, 1996.
- [42] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, 3rd ed., 2009.
- [43] K. R. RAO and P. C. YIP, *The Transform and Data Compression Handbook*. Boca Raton, FL, USA: CRC Press, 2000.
- [44] J. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3445–3462, 1993.
- [45] J. L. Prince and J. M. Links, *Medical Imaging Signal and Systems*. Upper Saddle River, NJ: Prentice Hallt, 2006.
- [46] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. Seljebotn, and K. Smith, “Cython: The best of both worlds,” *Computing in Science Engineering*, vol. 13, no. 2, pp. 31–39, 2011.
- [47] E. J. Candès and Y. Plan., “A probabilistic and riplless theory of compressed sensing,” tech. rep., Stanford University, 2010.

Lista de Acrónimos

A/D *Analog to Digital Converter*, Convertidor Analógico Digital.

AL Aproximación Lineal.

ANL Aproximación No Lineal.

ASWDR *Adaptively Scanned Wavelet Difference Reduction*.

BP *Backprojection*, Retroproyección [Pleaseinsertintopreamble]n.

CS *Compressive sampling*, Muestreo Compresivo.

CT *Computed Tomography*, Tomografía Computarizada.

DMD *Digital Micromirror Device*, Dispositivo Microespejo Digital.

DWT *Discrete Wavelet Transform*, Transformada Wavelet Discreta.

EZW *Embedded Zerotree Wavelet*.

FBP *Filtered Backprojection*, Retroproyección Filtrada.

FFT *Fast Fourier Transform*, Transformada Rápida de Fourier.

HHS *Heavy Hitters on Steroids*.

JPEG *Joint Photographic Experts Group*, Grupo Conjunto de Expertos en Fotografía.

JPEG2000 *Joint Photographic Experts Group 2000*, Grupo Conjunto de Expertos en Fotografía 2000.

MGS *Modified Gram-Schmidt*, Gram-Schmidt modificado.

MRI *Magnetic Resonance Imaging*, Imágenes de Resonancia Magnética.

MSE *Mean Square Error*, Error Cuadrático Medio.

OMP *Orthogonal Matching Pursuit.*

PSNR *Peak Signal to Noise Ratio, Relación Señal a Ruido Pico.*

RIP *Restricted Isometry Property, Propiedad de Isometría Restringida.*

SPIHT *Set Partitioning In Hierarchical Trees.*

TV *Total Variation, Variación Total.*

WDR *Wavelet Difference Reduction.*

Índice alfabético

- Adquisición de la información, 9
- Aplicaciones CS, 7
- Aproximación lineal, 2
- Aproximación no lineal, 2

- Bases teóricas, 4
- Basis Pursuit, 14

- Cifrado de la información, 7
- Coherencia Mutua, 9
- Compresión de imágenes, 7
- Compressive Sampling Matching Pursuit, 18

- Esbozo de la tesis, 8
- EZW, 29

- Motivación, 1

- Noiselets, 11

- Operador de información, 9
- Orthogonal Matching Pursuit, 16

- Propiedad de isometría restringida, 9

- Reconstrucción de la información, 13
- Reconstrucción utilizando Retroproyecciones Filtradas, 36
- Relación señal a ruido pico, 39
- Resultados de las simulaciones entre el CS y EZW, 40
- Resultados de las simulaciones entre el CS y retroproyecciones, 46

- Señales de Radar, 8

- Tomografía Computarizada, 34

- Transformada Radon, 35
- Transformada Discreta wavelet, 25
- Transformada Discreta Wavelet en 2-D, 26

- Wavelets, 25