

RECONSTRUCCIÓN DE SEÑALES MUESTREADAS A TASAS INFERIORES AL
CRITERIO DE NYQUIST

CINDY DAYANA SOLANO MEZA

LEIDY YOJANA GARCÍA ORTIZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICOMECAÑICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA

2011

RECONSTRUCCIÓN DE SEÑALES MUESTREADAS A TASAS INFERIORES AL
CRITERIO DE NYQUIST

CINDY DAYANA SOLANO MEZA

LEIDY YOJANA GARCÍA ORTIZ

Trabajo de grado para optar al título de
Ingeniería de Sistemas

Director:

MPE. HENRY ARGUELLO FUENTES

Codirector

CESAR AUGUSTO VARGAS

Ingeniero de Sistemas

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA

2011

A toda mi familia, Mi Mamá Teresa , Mi papa Carlos, mi hermana Johanita y mi cuñado Manuel, mi hermano Carlos y mi cuñada carolina, a todos por su gran amor, por guiarme todo este tiempo ,por creer siempre en mí y por apoyarme para lograr todos mis sueños.

A mis amigos Angel, Yuranis por ser mis grandes cómplices y amigos, por compartir conmigo sus experiencias, alegrías y momentos difíciles, por ayudarme ante cualquier situación, y por quererme mucho.

A Leidy por realizar este proyecto conmigo, y por brindarme su confianza y amistad, por ser tan paciente, por sus consejos y por todos los momentos felices e importantes compartidos.

*A todos mis amigos de carrera, Gina, Laura, Héctor, Juan David, Johan, Diana y todos los que compartieron conmigo momentos de alegría. Gracias por consentirme tanto.
¡Los quiero mucho a todos!*

Cindy Dayana Solano

Por su valiosa compañía dedico este triunfo a todas aquellas personas que compartieron a lo largo de mi carrera, las alegrías, las tristezas, las angustias, las noches de desvelo y que me ofrecieron su voz de aliento en los momentos difíciles cuando sentí que mis fuerzas se agotaban.

Especialmente:

A Mis padres Jesús y Nancy y mis hermanos Jorge y Lisney por apoyarme y respaldar mi carrera, por estar siempre a mi lado y creer en mí. Por el inmenso amor del que siempre me rodearon y me dio fuerzas para surgir.

A mi tía Doris, quien me apoyó para iniciar mis estudios.

A Rudvel, el amor de mi vida, el hombre que compartió conmigo su tiempo y su espacio para que yo saliera adelante.

A mi hija María José, quien es ahora el motor de mi vida y me dio fuerzas estos semestres para luchar y no rendirme.

A Cindy, mi compañera, confidente y cómplice, quien compartió conmigo momentos inolvidables.

A mis grandes amigos, Angel, Laurita, Dianis, Yura, Lina, Betty y todos aquellos que han compartido conmigo.

Con todos ellos quiero compartir la satisfacción por alcanzar la meta propuesta.

A ti Diosito... por no dejarme desvanecer.

Leidy Yojana García Ortiz

AGRADECIMIENTOS

Por habernos apoyado en el transcurso de nuestra formación y lograr el desarrollo del presente proyecto queremos agradecer:

Al profesor HENRY ARGUELLO FUENTES, Director de nuestro proyecto, por brindarnos sus conocimientos, por confiar en nosotras, por dedicarnos parte de su tiempo y la idea de desarrollar este proyecto.

Al Ingeniero CESAR AUGUSTO VARGAS GARCÍA, codirector de nuestro proyecto, por ser un gran amigo, por brindarnos muy buenos consejos, por apoyarnos en todo el transcurso del desarrollo de este proyecto, y por creer en nosotras.

A nuestros familiares por brindarnos su apoyo en todo momento, por ser las personas que más queremos y las más importantes en nuestras vidas.

A todos nuestros amigos y compañeros de carrera por compartir con nosotras en todos estos años de estudio.

A nuestros compañeros integrantes del grupo de Investigación en Ingeniería Biomédica, por compartir estos últimos semestres con nosotras todos sus conocimientos.

Gracias a Betty, Víctor, Hugo, Fabián, Lina y Andrés, por ser muy buenos amigos, y por compartir tantos momentos felices con nosotras.

Gracias a la Universidad Industrial de Santander por hacer de nosotras unas profesionales.

A Dios, por iluminar nuestro camino y ser el responsable de este logro .

RESUMEN

TÍTULO

RECONSTRUCCIÓN DE SEÑALES MUESTREADAS A TASAS INFERIORES AL CRITERIO DE NYQUIST*.

AUTORES

CINDY DAYANA SOLANO MEZA**

LEIDY YOJANA GARCÍA ORTÍZ**

PALABRAS CLAVE

ALGORITMOS GREEDY, ALGORITMO ORTHOGONAL MATCHING PURSUIT (OMP), ALGORITMO PUNTO INTERIOR, BASIS PURSUIT (BP), COMPRESSED SENSING (CS), SISTEMAS DE ECUACIONES SUBDETERMINADOS, TEOREMA DE NYQUIST, TRANSFORMADA WAVELET.

DESCRIPCIÓN

El presente proyecto se basa en la teoría denominada Compressive Sensing, en donde se establece que es posible la reconstrucción de señales que han sido muestreadas a tasas inferiores al criterio de Nyquist.

Esta teoría se centra en la reconstrucción de una señal $x \in \mathbf{R}^n$ mediante la solución del problema de optimización l_1 sujeto a $y = \Phi x$, donde $\Phi \in \mathbf{R}^{m \times n}$ con $m \ll n$, es decir, en el sistema el número de ecuaciones es menor al número de incógnitas. Para abordar esta problemática es posible utilizar una serie de algoritmos entre los cuales se encuentran los planteados en la teoría de algoritmos Greedy y aquellos que utilizan técnicas de optimización.

En este proyecto se utilizó el algoritmo Punto Interior de la Teoría de Basis Pursuit con algunas modificaciones, el cual busca una solución por medio de programación matemática y se realizó la implementación de un algoritmo Greedy denominado Orthogonal Matching Pursuit que se basa en un proceso iterativo. Estos algoritmos emplean una nueva técnica de compresión que trae beneficios en las áreas de almacenamiento y transmisión

Se presenta un análisis comparativo entre los dos algoritmos estudiados con respecto al tiempo de cómputo y la calidad de reconstrucción. Para dicho análisis se realizaron pruebas a imágenes dispersas en el dominio de la transformada Wavelet, en escala de grises haciendo variaciones en el tamaño de las señales y el porcentaje de compresión.

* TRABAJO DE GRADO

** FACULTAD DE INGENIERIAS FISICOMECAICAS. ESCUELA DE INGENIERIA DE SISTEMAS. DIRECTOR: MPE HENRY ARGUELLO. CODIRECTOR: ING DE SISTEMAS CESAR AUGUSTO VARGAS.

SUMMARY

TITLE

RECONSTRUCTION OF SIGNALS SAMPLED TO RATES LOWER THAN NYQUIST'S CRITERION. *

AUTHORS

CINDY DAYANA SOLANO MEZA**
LEIDY YOJANA GARCÍA ORTÍZ**

KEYWORDS

ALGORITHMS GREEDY, ALGORITHM ORTHOGONAL MATCHING PURSUIT (OMP), ALGORITHM INTERIOR POINT, BASIS PURSUIT (BP), COMPRESSIVE SENSING (CS), SYSTEMS OF EQUATIONS UNDERDETERMINED, NYQUIST'S THEOREM, WAVELET TRANSFORM.

DESCRIPTION

The present project is based on the theory known as Compressive Sensing, which states that it is possible the reconstruction of signals that have been sampled at rates lower than the criterion of Nyquist.

This theory focuses on the reconstruction of a signal $x \in \mathbf{R}^n$ by solving the optimization problem l_1 subject to $y = \Phi x$, with $\Phi \in \mathbf{R}^{m \times n}$ with $m \ll n$, i.e. in the system the number of equations is less than the number of unknowns. To approach this problematic is possible to use a series of algorithms between which the proposed ones are in the theory of algorithms Greedy and those that use techniques of optimization.

This project used the Interior Point Algorithm from the Theory of Basis Pursuit with some modifications, which looks for a solution by mathematical programming and realized the implementation of an algorithm Greedy named Orthogonal Matching Pursuit that is based on an iterative process.

It presents a comparative analysis between both algorithms studied with regard to the time of calculation and the quality of reconstruction. For this analysis were realized tests to sparse images in the domain of the transformed Wavelet, in grayscale making variations in the size of the signals and the percentage of compression.

* WORD DEGREE

** FACULTY OF ENGINEERINGS PHYSICOMECHANICAL. SCHOOL OF SYSTEMS ENGINEERING. DIRECTOR: MPE HENRY ARGUELLO. CODIRECTOR: SYSTEM ENGINEER CESAR AUGUSTO VARGAS.

CONTENIDO

INTRODUCCIÓN.....	18
1. DESCRIPCIÓN DEL PROYECTO	20
1.1 OBJETIVO GENERAL.....	20
1.2 OBJETIVOS ESPECÍFICOS.....	20
1.3 JUSTIFICACIÓN	20
2. ESTADO DEL ARTE	21
2.1 TÉCNICAS DE OPTIMIZACIÓN	21
2.2 ALGORITMOS GREEDY	22
3. MARCO TEÓRICO.....	25
3.1 TEOREMA DE NYQUIST	25
3.2 COMPRESSIVE SENSING	25
3.2.1 Representación dispersa y nivel de dispersión de una señal.....	27
3.2.2 Bases de representación	28
3.2.3 Bases de muestreo.....	33
3.2.4 Propiedad de incoherencia	35
4. ALGORITMOS DE RECONSTRUCCIÓN.....	36
4.1 BASIS PURSUIT (BP).....	36
4.1.1 Método de Punto Interior:.....	36
4.2 ALGORITMO GREEDY ORTHOGONAL MATCHING PURSUIT (OMP)	39
5. METODOLOGÍA	43
5.1 Señales Unidimensionales.	43
5.2 Señales Bidimensionales.....	43
6. RESULTADOS	45
CONCLUSIONES.....	91
RECOMENDACIONES.....	93
BIBLIOGRAFIA.....	96

LISTA DE FIGURAS

	Pág.
FIGURA 1. PRINCIPALES TÉCNICAS DE RECUPERACIÓN DE IMÁGENES	21
FIGURA 2. SISTEMA CONFORMADO POR LA BASE DE MUESTREO Φ , SEÑAL ORIGINAL x , SEÑAL COMPRIMIDA y	26
FIGURA 3. SEÑAL ORIGINAL EXPRESA EN EL DOMINIO DE UNA BASE DE REPRESENTACIÓN Ψ Y SU RESPECTIVO VECTOR DE COEFICIENTES.....	26
FIGURA 4. REPRESENTACIÓN DEL SISTEMA EN FUNCIÓN DE LA BASE DE MUESTREO Φ , LA BASE DE REPRESENTACIÓN Ψ Y EL VECTOR DE COEFICIENTES s	27
FIGURA 5. (A) IMAGEN ORIGINAL. (B) COEFICIENTES OBTENIDOS A PARTIR DE LA BASE DE REPRESENTACIÓN WAVELET (C) IMAGEN RECUPERADA A PARTIR DE LOS COEFICIENTES WAVELET.	28
FIGURA 6. REPRESENTACIÓN DE LA BASE DE FOURIER. CADA UNA DE LAS FUNCIONES SINUSOIDALES REPRESENTA UNA COLUMNA DE LA BASE.	28
FIGURA 7. DESCOMPOSICIÓN WAVELET EN DOS DIMENSIONES, EMPLEANDO EL FILTRO SYMLET 8 A UN NIVEL.	32
FIGURA 8. FAMILIA DE FILTROS SYM(N), DONDE N REPRESENTA EL ORDEN.	33
FIGURA 9. FORMA RECURSIVA DE CREAR UNA MATRIZ HADAMARD.	34
FIGURA 10. PORCENTAJE DE RECUPERACIÓN DE UNA SEÑAL K-DISPERSA $s \in R^{256}$ EMPLEANDO EL ALGORITMO ORTHOGONAL MATCHING PURSUIT CON $\Phi = \textit{Hadamard}$	45
FIGURA 11 . PORCENTAJE DE RECUPERACIÓN DE UNA SEÑAL K-DISPERSA $s \in R^{256}$ EMPLEANDO EL ALGORITMO ORTHOGONAL MATCHING PURSUIT CON $\Phi = \textit{Gaussiana}$	45
FIGURA 12. PORCENTAJE DE RECUPERACIÓN DE UNA SEÑAL K-DISPERSA $s \in R^{256}$ EMPLEANDO EL ALGORITMO PUNTO INTERIOR CON $\Phi = \textit{Hadamard}$	49
FIGURA 13. PORCENTAJE DE RECUPERACIÓN DE UNA SEÑAL K-DISPERSA $s \in R^{256}$ EMPLEANDO EL ALGORITMO PUNTO INTERIOR CON $\Phi = \textit{Gaussiana}$	49
FIGURA 14. PSNR PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DISPERSA DE 32X32 PÍXELES EMPLEANDO EL ALGORITMO DE PUNTO INTERIOR CON $\Psi = \textit{FWT2}$	52
FIGURA 15. TIEMPO DE CÓMPUTO PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DISPERSA DE 32X32 PÍXELES EMPLEANDO EL ALGORITMO DE PUNTO INTERIOR CON $\Psi = \textit{FWT2}$	52

FIGURA 16. PSNR PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DISPERSA DE 32X32 PÍXELES EMPLEANDO EL ALGORITMO DE PUNTO INTERIOR CON $\Psi = WTeórica$.	53
FIGURA 17. TIEMPO DE CÓMPUTO PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DISPERSA DE 32X32 PÍXELES EMPLEANDO EL ALGORITMO DE PUNTO INTERIOR CON $\Psi = WTeórica$.	54
FIGURA 18. PSNR PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DISPERSA DE 32X32 PÍXELES EMPLEANDO EL ALGORITMO DE ORTHOGONAL MATCHING PURSUIT CON $\Psi = FWT2$.	55
FIGURA 19. TIEMPO DE CÓMPUTO PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DISPERSA DE 32X32 PÍXELES EMPLEANDO EL ALGORITMO DE ORTHOGONAL MATCHING PURSUIT CON $\Psi = FWT2$.	55
FIGURA 20. PSNR PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DISPERSA DE 32X32 PÍXELES EMPLEANDO EL ALGORITMO DE ORTHOGONAL MATCHING PURSUIT CON $\Psi = Wmxt2$.	56
FIGURA 21. TIEMPO DE CÓMPUTO PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DISPERSA DE 32X32 PÍXELES EMPLEANDO EL ALGORITMO DE ORTHOGONAL MATCHING PURSUIT CON $\Psi = Wmxt2$.	57
FIGURA 22. PSNR PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DISPERSA DE 64X64 PÍXELES EMPLEANDO EL ALGORITMO DE PUNTO INTERIOR CON $\Psi = FWT2$.	58
FIGURA 23. TIEMPO DE CÓMPUTO PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DISPERSA DE 64X64 PÍXELES EMPLEANDO EL ALGORITMO DE PUNTO INTERIOR CON $\Psi = FWT2$.	58
FIGURA 24. PSNR PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DISPERSA DE 64X64 PÍXELES EMPLEANDO EL ALGORITMO DE PUNTO INTERIOR CON $\Psi = Wmxt2$.	59
FIGURA 25. TIEMPO DE CÓMPUTO PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DISPERSA DE 64X64 PÍXELES EMPLEANDO EL ALGORITMO DE PUNTO INTERIOR CON $\Psi = Wmxt2$.	60
FIGURA 26. PSNR PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DISPERSA DE 64X64 PÍXELES EMPLEANDO EL ALGORITMO DE ORTHOGONAL MATCHING PURSUIT CON $\Psi = FWT2$.	61
FIGURA 27. TIEMPO DE CÓMPUTO PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DISPERSA DE 64X64 PÍXELES EMPLEANDO EL ALGORITMO DE ORTHOGONAL MATCHING PURSUIT CON $\Psi = FWT2$.	61
FIGURA 28. PSNR PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DISPERSA DE 64X64 PÍXELES EMPLEANDO EL ALGORITMO DE ORTHOGONAL MATCHING PURSUIT CON $\Psi = Wmxt2$.	62
FIGURA 29. TIEMPO DE CÓMPUTO PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DISPERSA DE 64X64 PÍXELES EMPLEANDO EL ALGORITMO DE ORTHOGONAL MATCHING PURSUIT CON $\Psi = Wmxt2$.	63
FIGURA 30. PSNR PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DISPERSA DE 128X128 PÍXELES EMPLEANDO EL ALGORITMO DE PUNTO INTERIOR CON $\Psi = FWT2$.	64
FIGURA 31. PSNR PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DISPERSA DE 128X128 PÍXELES EMPLEANDO EL ALGORITMO DE PUNTO INTERIOR CON $\Psi = FWT2$.	64

LISTA DE TABLAS

	Pág.
TABLA 1. ALGORITMO PUNTO INTERIOR.....	39
TABLA 2. ALGORITMO ORTHOGONAL MATCHING PURSUIT	42
TABLA 3. RESULTADOS OBTENIDOS EMPLEANDO EL ALGORITMO ORTHOGONAL MATCHING PURSUIT CON $\Phi = \textit{Hadamard}$ PARA LA RECONSTRUCCIÓN DE UNA SEÑAL SPARSE $s \in \mathbf{R}^{256}$	47
TABLA 4. RESULTADOS OBTENIDOS EMPLEANDO EL ALGORITMO ORTHOGONAL MATCHING PURSUIT CON $\Phi = \textit{Gaussiana}$ PARA LA RECONSTRUCCIÓN DE UNA SEÑAL SPARSE $s \in \mathbf{R}^{256}$	48
TABLA 5. RESULTADOS OBTENIDOS EMPLEANDO EL ALGORITMO PUNTO INTERIOR CON $\Phi = \textit{Hadamard}$ PARA LA RECONSTRUCCIÓN DE UNA SEÑAL SPARSE $s \in \mathbf{R}^{256}$	50
TABLA 6. RESULTADOS OBTENIDOS EMPLEANDO EL ALGORITMO PUNTO INTERIOR CON $\Phi = \textit{Gaussiana}$ PARA LA RECONSTRUCCIÓN DE UNA SEÑAL SPARSE $s \in \mathbf{R}^{256}$	51
TABLA 7. RESULTADOS OBTENIDOS EMPLEANDO EL ALGORITMO PUNTO INTERIOR PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DE 32X32 PÍXELES, CON $\Psi = \textit{FWT2}$	53
TABLA 8. RESULTADOS OBTENIDOS EMPLEANDO EL ALGORITMO PUNTO INTERIOR PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DE 32X32 PÍXELES, CON $\Psi = \textit{WTeórica}$	54
TABLA 9. RESULTADOS OBTENIDOS EMPLEANDO EL ALGORITMO ORTHOGONAL MATCHING PURSUIT PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DE 32X32 PÍXELES, CON $\Psi = \textit{FWT2}$	56
TABLA 10. RESULTADOS OBTENIDOS EMPLEANDO EL ALGORITMO ORTHOGONAL MATCHING PURSUIT PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DE 32X32 PÍXELES, CON $\Psi = \textit{WTeórica}$	57
TABLA 11. RESULTADOS OBTENIDOS EMPLEANDO EL ALGORITMO PUNTO INTERIOR PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DE 64X64 PÍXELES, CON $\Psi = \textit{FWT2}$	59
TABLA 12. RESULTADOS OBTENIDOS EMPLEANDO EL ALGORITMO PUNTO INTERIOR PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DE 64X64 PÍXELES, CON $\Psi = \textit{WTeórica}$	60
TABLA 13. RESULTADOS OBTENIDOS EMPLEANDO EL ALGORITMO ORTHOGONAL MATCHING PURSUIT PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DE 64X64 PÍXELES, CON $\Psi = \textit{FWT2}$	62
TABLA 14. RESULTADOS OBTENIDOS EMPLEANDO EL ALGORITMO ORTHOGONAL MATCHING PURSUIT PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DE 64X64 PÍXELES, CON $\Psi = \textit{WTeórica}$	63

TABLA 15. RESULTADOS OBTENIDOS EMPLEANDO EL ALGORITMO PUNTO INTERIOR PARA LA RECONSTRUCCIÓN DE UNA IMAGEN DE 128X128 PÍXELES, CON $\Psi = \mathbf{FWT2}$	65
TABLA 16. IMAGEN DE 32X32 PÍXELES RECUPERADA EMPLEANDO EL ALGORITMO DE PUNTO INTERIOR CON $\Psi = \mathbf{FWT2}$, $\Phi = \mathbf{SBHE}$ Y NIVEL DE DISPERSIÓN DE 5%.	66
TABLA 17. IMAGEN DE 32X32 PÍXELES RECUPERADA EMPLEANDO EL ALGORITMO DE PUNTO INTERIOR CON $\Psi = \mathbf{FWT2}$, $\Phi = \mathbf{GAUSSIANA}$ Y NIVEL DE DISPERSIÓN DE 5%.	67
TABLA 18. IMAGEN DE 64X64 PÍXELES RECUPERADA EMPLEANDO EL ALGORITMO DE PUNTO INTERIOR CON $\Psi = \mathbf{FWT2}$, $\Phi = \mathbf{GAUSSIANA}$ Y NIVEL DE DISPERSIÓN DE 5%.	68
TABLA 19. IMAGEN DE 64X64 PÍXELES RECUPERADA EMPLEANDO EL ALGORITMO DE PUNTO INTERIOR CON $\Psi = \mathbf{WTeórica}$, $\Phi = \mathbf{GAUSSIANA}$ Y NIVEL DE DISPERSIÓN DE 5%.	69
TABLA 20. IMAGEN DE 32X32 PÍXELES RECUPERADA EMPLEANDO EL ALGORITMO PUNTO INTERIOR CON $\Psi = \mathbf{WTeórica}$, $\Phi = \mathbf{SBHE}$ Y NIVEL DE DISPERSIÓN DE 5%......	70
TABLA 21. IMAGEN DE 32X32 PÍXELES RECUPERADA EMPLEANDO EL ALGORITMO PUNTO INTERIOR CON $\Psi = \mathbf{WTeórica}$, $\Phi = \mathbf{GAUSSIANA}$ Y NIVEL DE DISPERSIÓN DE 5%......	71
TABLA 22. IMAGEN DE 64X64 PÍXELES RECUPERADA EMPLEANDO EL ALGORITMO PUNTO INTERIOR CON $\Psi = \mathbf{FWT2}$, $\Phi = \mathbf{SBHE}$ Y NIVEL DE DISPERSIÓN DE 5%.	72
TABLA 23. IMAGEN DE 64X64 PÍXELES RECUPERADA EMPLEANDO EL ALGORITMO PUNTO INTERIOR CON $\Psi = \mathbf{WTeórica}$, $\Phi = \mathbf{SBHE}$ Y NIVEL DE DISPERSIÓN DE 5%.	73
TABLA 24. IMAGEN DE 32X32 PÍXELES RECUPERADA EMPLEANDO EL ALGORITMO ORTHOGONAL MATCHING PURSUIT CON $\Psi = \mathbf{FWT2}$, $\Phi = \mathbf{HADAMARD}$ Y NIVEL DE DISPERSIÓN DE 5%.	74
TABLA 25. IMAGEN DE 32X32 PÍXELES RECUPERADA EMPLEANDO EL ALGORITMO ORTHOGONAL MATCHING PURSUIT CON $\Psi = \mathbf{FWT2}$, $\Phi = \mathbf{GAUSSIANA}$ Y NIVEL DE DISPERSIÓN DE 5%.	75
TABLA 26. IMAGEN DE 32X32 PÍXELES RECUPERADA EMPLEANDO EL ALGORITMO ORTHOGONAL MATCHING PURSUIT CON $\Psi = \mathbf{WTeórica}$, $\Phi = \mathbf{HADAMARD}$ Y NIVEL DE DISPERSIÓN DE 5%.	76
TABLA 27. IMAGEN DE 32X32 PÍXELES RECUPERADA EMPLEANDO EL ALGORITMO ORTHOGONAL MATCHING PURSUIT CON $\Psi = \mathbf{WTeórica}$, $\Phi = \mathbf{GAUSSIANA}$ Y NIVEL DE DISPERSIÓN DE 5%.	77
TABLA 28. IMAGEN DE 64X64 PÍXELES RECUPERADA EMPLEANDO EL ALGORITMO ORTHOGONAL MATCHING PURSUIT CON $\Psi = \mathbf{FWT2}$, $\Phi = \mathbf{HADAMARD}$ Y NIVEL DE DISPERSIÓN DE 5%.	78
TABLA 29. IMAGEN DE 64X64 PÍXELES RECUPERADA EMPLEANDO EL ALGORITMO ORTHOGONAL MATCHING PURSUIT CON $\Psi = \mathbf{FWT2}$, $\Phi = \mathbf{GAUSSIANA}$ Y NIVEL DE DISPERSIÓN DE 5%.	79

TABLA 30. IMAGEN DE 64x64 PÍXELES RECUPERADA EMPLEANDO EL ALGORITMO ORTHOGONAL MATCHING PURSUIT CON $\Psi = \mathbf{WTeórica}$, $\Phi = \mathbf{HADAMARD}$ Y NIVEL DE DISPERSIÓN DE 5%.....	80
TABLA 31. IMAGEN DE 64x64 PÍXELES RECUPERADA EMPLEANDO EL ALGORITMO ORTHOGONAL MATCHING PURSUIT CON $\Psi = \mathbf{WTeórica}$, $\Phi = \mathbf{GAUSSIANA}$ Y NIVEL DE DISPERSIÓN DE 5%.....	81
TABLA 32. IMAGEN DE 128x128 PÍXELES RECUPERADA EMPLEANDO EL ALGORITMO DE PUNTO INTERIOR CON $\Psi = \mathbf{FWT2}$, $\Phi = \mathbf{SBHE}$ Y NIVEL DE DISPERSIÓN DE 5%.....	82
TABLA 33. ANÁLISIS COMPARATIVO.....	88
TABLA 34. RESULTADOS OBTENIDOS EMPLEANDO LA BASE DE MUESTREO GAUSSIANA Y PORCENTAJE DE COMPRESIÓN DE 4%.....	89
TABLA 35. RESULTADOS OBTENIDOS EMPLEANDO EL ALGORITMO ORTHOGONAL MATCHING PURSUIT BUSCANDO EL NÚMERO DE MEDICIONES NECESARIAS PARA ALCANZAR UN PORCENTAJE DE RECUPERACIÓN DEL 75% DE UNA SEÑAL DISPERSA $s \in \mathbf{R256}$	90

INTRODUCCIÓN

Las señales, los signos y los símbolos han sido los medios de comunicación que los seres humanos han empleado para expresar sus ideas desde sus inicios. A través del tiempo, la evolución del pensamiento y los avances tecnológicos permitieron la creación de nuevas formas de intercambio de información para expresar sentimientos, opiniones, entre otros, mediante habla, escritura u otro tipo de señales.

En la vida real las señales que se pueden percibir tales como la luz y el sonido, son llamadas señales analógicas y se caracterizan porque tienen una variación continua en el tiempo. Durante la década de los 50 y los primeros años de los 60, diseñadores de sistemas analógicos desarrollaron una nueva técnica que procesaba dichas señales y las convertía en datos digitales para ser analizados.

Estas señales digitales se caracterizan por presentar una variación discontinua en el tiempo y sólo pueden tomar ciertos valores discretos. Su forma característica es ampliamente conocida; la señal básica es una onda cuadrada (pulsos) y las representaciones se realizan en el dominio del tiempo.

Esta nueva forma de transmisión es superior a la analógica, desde varios puntos de vista importantes.

Algunas ventajas son la multiplexación conjunta de voz, imágenes, música y el bajo costo de sus dispositivos de procesamiento. Igualmente permite la transmisión de grandes volúmenes de información a velocidades más altas, utilizando diversas técnicas de compresión, que reducen la cantidad de datos a transmitir para evitar problemas de envío en los dispositivos de transmisión y brindar mayores capacidades de almacenamiento.

Una muestra de ello, es el envío de imágenes comprimidas, que permite el aumento de la velocidad de transmisión, y le permite a la máquina receptora ocupar menos recursos. Sin embargo, se hace indispensable la utilización de métodos de descompresión que permitan visualizar la imagen original.

Este proyecto proporciona conceptos y métodos empleados para la reconstrucción de imágenes, mediante la implementación y el análisis de técnicas matemáticas existentes. En el capítulo 2 de este libro se hace la presentación de algunas técnicas de optimización y algunos algoritmos de la teoría Greedy para la reconstrucción de imágenes dispersas; el capítulo 3 se describe las características y principios fundamentales de la teoría de Compressive Sensing (CS); en el capítulo 4 se hará la explicación detallada de los algoritmos Orthogonal Matching Pursuit (OMP) propuesto por la teoría de algoritmos Greedy y Punto Interior propuesto en la teoría de Basis Pursuit (BP) y en los capítulos 5 y 6 se presenta la metodología y los resultados obtenidos en las simulaciones que se realizaron a señales haciendo variaciones en sus tamaños y el porcentaje de compresión.

1. DESCRIPCIÓN DEL PROYECTO

1.1 OBJETIVO GENERAL

Analizar e implementar los algoritmos Orthogonal Matching Pursuit (OMP) y Punto interior (PI) utilizados para la reconstrucción de imágenes digitales.

1.2 OBJETIVOS ESPECÍFICOS

- Implementar los algoritmos Ortogonal Matching Pursuit y punto interior para la reconstrucción de señales comprimidas.
- Realizar pruebas de velocidad y calidad de recuperación de los algoritmos implementados en una arquitectura de procesamiento determinada.
- Realizar un análisis comparativo entre los algoritmos implementados, y cada uno de estos con estudios ya realizados.

1.3 JUSTIFICACIÓN

En la actualidad manejar la información de forma rápida y segura es de vital importancia para cualquier tipo de organización. El tráfico en la red involucra grandes cantidades de datos, que en determinados espacios de tiempo provocan congestiones, lo cual podría provocar que la información pierda interés al llegar al destino. Para solucionar esta problemática, la información puede ser comprimida, y al llegar a su destino recuperarla mediante algoritmos que permitan este proceso.

El almacenamiento de cantidades significativas de datos, es otra de las preocupaciones que se presentan actualmente, lo cual ha hecho que los fabricantes de hardware se centren en la producción de equipos con mayor capacidad de almacenamiento. Si se comprime la información, se podría disponer de más espacio en un mismo equipo y a su vez reducir los tiempos de cómputo durante el procesamiento de señales e imágenes.

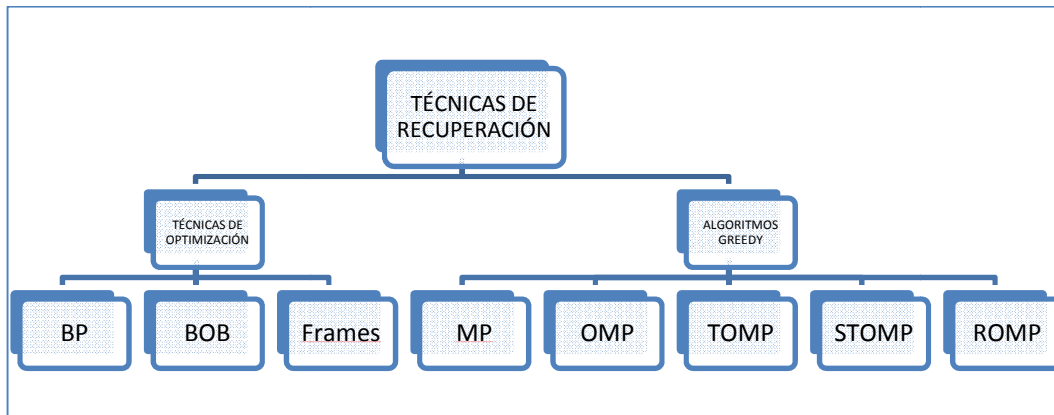
Según la problemática mencionada anteriormente, se han diseñado diversos algoritmos que permiten la compresión y descompresión de información; con lo cual, se logra reducir los costos en cambios de infraestructura y aumentar la eficiencia de los dispositivos de almacenamiento actuales.

2. ESTADO DEL ARTE

Actualmente existen algoritmos y técnicas computacionales que reconstruyen aproximaciones de una señal deseada que ha sido comprimida. Este problema de recuperación de señales (compresión) se puede convertir en un problema de recuperación de señales dispersas [1] [2] [3]. Dentro de estas técnicas de recuperación encontramos los algoritmos Greedy y las técnicas de optimización (ver Figura 1)

Los algoritmos Greedy son aquellos que, para resolver un determinado problema, siguen una metaheurística consistente en elegir la opción óptima en cada paso con el fin de llegar a una solución general óptima [4] y las técnicas de optimización también eligen la mejor solución para resolver un problema, pero empleando programación matemática [5].

Figura 1. Principales técnicas de recuperación de imágenes



2.1 TÉCNICAS DE OPTIMIZACIÓN

- **Basis Pursuit (BP):** El principio de BP es encontrar una representación de la señal cuyos coeficientes satisfacen la norma mínima L_1 ; expresada como:

$$\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix}, \|\mathbf{x}\|_1 = \sum_{i=0}^{n-1} |x_i| \text{ sujeto a } \mathbf{Ax} = \mathbf{b} \quad (1)$$

la cual puede ser resuelta empleando programación lineal (PL) [5] [6] [7] [8] [9], que en los últimos 40 años, ha servido de apoyo para la solución de una gran cantidad de problemas de optimización. En Basis Pursuit dichos problemas pueden ser tratados mediante el algoritmo simplex o el algoritmo de punto interior [6]. Este tema será tratado con más detalle en el capítulo 4.

- **Best Orthogonal Basis (BOB):** BP es mejor que BOB resolviendo estructuras no ortogonales debido a que Basis Pursuit trabaja sobre todas las bases del diccionario mientras que BOB solo lo hace con bases ortogonales[6][9]. Cuando la señal está compuesta por un número moderado de componentes no ortogonales, BOB no puede dar una representación dispersa[6].

- **Frames (MOF):** Contrario a Basis Pursuit, MOF emplea la norma de minimización L_2 ; expresada como[6][9]:

$$x = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix}, \|x\|_2 = \left(\sum_{i=0}^{n-1} (|x_i|^2) \right)^{1/2} \quad (2)$$

Este método presenta dos problemas claves. En primer lugar, no conserva el nivel de dispersión. Si los elementos fundamentales tienen una representación dispersa, entonces los coeficientes encontrados por MOF son probablemente menos dispersos. En segundo lugar, MOF tiene una resolución limitada. Frame tiene una relación interesante con el método BP-interior, este último puede ser inicializado con la solución Frame.

2.2 ALGORITMOS GREEDY

- **Matching Pursuit (MP):** MP [10], [11], [12] calcula aproximaciones de la señal de un diccionario redundante, seleccionando iterativamente un vector en el tiempo. Estas iteraciones requieren $O(N \log_2 N)$ operaciones para dar solución a un problema. N es el conjunto de vectores linealmente independientes que definen una base del espacio de la señal.

Una característica de este algoritmo es que se detiene después de pocos pasos. Cuando el diccionario es ortogonal, el método trabaja perfectamente. Por el contrario si no es ortogonal, la situación no es clara[3] [9].

- **Orthogonal Matching Pursuit (OMP):** Este algoritmo selecciona un óptimo local dentro de un grupo de posibles soluciones en cada una de las iteraciones que realiza, para encontrar un óptimo general [13]. Mientras que MP requiere $O(N \log_2 N)$ iteraciones, OMP requiere [1] [14] $O(n \log_2 N)$, donde n es el nivel de dispersión y N el conjunto de vectores que definen una base. Desde este punto de vista, OMP requiere menos iteraciones y por lo tanto sería más conveniente su utilización. Más adelante se presentará una descripción detallada del algoritmo OMP.

- **Tree-based Orthogonal Matching Pursuit (TOMP):** Algoritmos actuales como BP y OMP, emplean la estructura dispersa de señales desconocidas para dar solución a problemas inversos; mientras que TOMP, emplea además la estructura de árbol de las señales. Este último es mejor que BP y OMP en cuanto a reconstrucción de alta calidad. El número de iteraciones en TOMP es menor que en OMP ya que TOMP no selecciona un solo elemento sino todo el árbol a la vez. TOMP proporciona una buena reconstrucción a pesar de que los datos se vean afectados por el ruido[15].

- **Stagewise Orthogonal Matching Pursuit (STOMP):** STOMP es una extensión de OMP que selecciona en cada iteración varios vectores base. En cada iteración STOMP empieza la solución de un nuevo gradiente conjugado, mientras que OMP aproxima la proyección ortogonal usando un solo paso de optimización direccional que es más eficiente[17].

- **Regularized Orthogonal Matching Pursuit (ROMP):** Combina la velocidad y facilidad de la implementación de los algoritmos Greedy y las garantías de los métodos de programación convexa, con solo añadir un factor teórico logarítmico. ROMP es una variante de OMP [17]., que emplea $2n$ iteraciones, donde n es el nivel de dispersión, con un tiempo de ejecución de $O(dNn)$ al igual que el

encontrado para OMP. Este último selecciona un solo elemento en cada iteración, mientras que ROMP y STOMP seleccionan más de un elemento[12] [18].

3. MARCO TEÓRICO

3.1 TEOREMA DE NYQUIST

El Teorema del Muestreo, o Teorema de Nyquist-Shannon, establece que la frecuencia mínima de muestreo necesaria para evitar el Aliasing¹ debe ser menor que la frecuencia de muestreo:

$$f_m > 2 * \Omega_N \quad (3)$$

donde f_m representa la frecuencia mínima de muestreo, y Ω_N el ancho de banda de la señal a muestrear ($\Omega_N = f_{max} - f_{min}$).

Para señales con $f_{min} = 0$, se puede expresar como:

$$f_m > 2. f_{max} \quad (4)$$

A medida que aumenta el ancho de banda de la señal, aumenta la frecuencia de muestreo provocando que los convertidores analógico-digitales presenten dificultades para muestrear las señales debido a limitaciones de velocidad, problemas de almacenamiento, restricciones del ancho de banda para la transmisión, etc.

Actualmente la teoría de Compressive Sensing, permite solucionar estas problemáticas empleando una frecuencia mínima de muestreo inferior a la expresada en el criterio de Nyquist. Para tal fin esta teoría emplea señales cuyo nivel de dispersión es muy bajo, es decir, señales cuya información relevante está concentrada en muy pocos puntos. En los sistemas reales esta dispersión es expresada, por ejemplo, en dominios del tiempo o frecuencia.

3.2 COMPRESSIVE SENSING

Compressive Sensing² llamada también Compressed Sampling es una nueva teoría de muestreo de señales en la cual se establece que una señal que tiene

¹ Efecto que causa que señales continuas distintas se tornen indistinguibles cuando se les muestrea digitalmente.

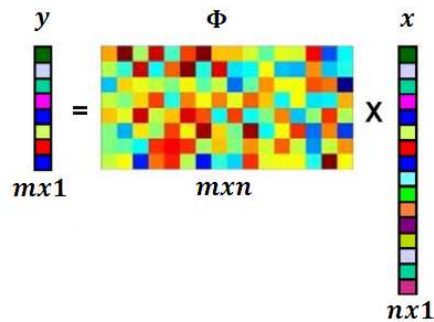
² Teoría actualmente estudiada por David Donoho, Emmanuel Candès, Justin Romberg and Terence Tao.

representación dispersa en una base determinada (Wavelet, Fourier, Coseno) puede ser reconstruida a partir de pocas muestras.

Matemáticamente la teoría de Compressive Sensing se representa como:

$$y = \Phi x \text{ con } \Phi \in R^{m \times n}, x \in R^n, y \in R^m \quad (5)$$

Figura 2. Sistema conformado por la base de muestreo Φ , señal original x , señal comprimida y .

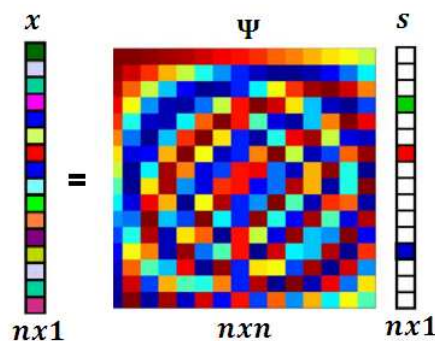


Fuente: Dave Mesecher, Larry Carin, Ivan Kadar, Ron Pirich, "Exploiting Signal Sparseness for Reduced- Rate Sampling", IEEE, 2009

Donde Φ es denominada base de muestreo y x es una señal k-dispersa (k-valores distintos de cero), en el dominio de una base Ψ , ver figura 3.

$$x = \Psi s \quad (6)$$

Figura 3. Señal original expresa en el dominio de una base de representación Ψ y su respectivo vector de coeficientes.



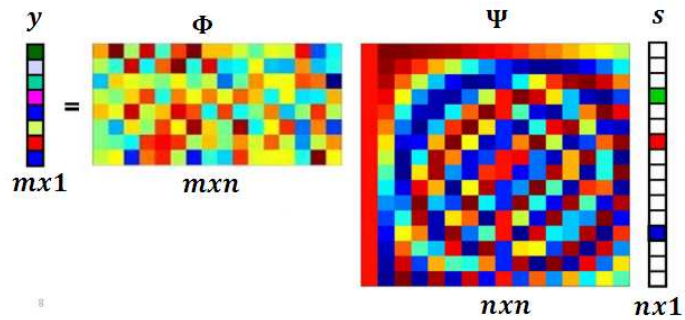
Fuente: Dave Mesecher, Larry Carin, Ivan Kadar, Ron Pirich, "Exploiting Signal Sparseness for Reduced- Rate Sampling", IEEE, 2009

Reemplazando (6) en (5) se obtiene:

$$y = \Phi \Psi s = \Theta s \quad (7)$$

Como se observa en la figura 4.

Figura 4. Representación del sistema en función de la base de muestreo Φ , la base de representación Ψ y el vector de coeficientes s .



Fuente: Dave Mesecher, Larry Carin, Ivan Kadar, Ron Pirich, "Exploiting Signal Sparseness for Reduced- Rate Sampling", IEEE, 2009

Para garantizar la reconstrucción de una señal x , CS requiere dos principios fundamentales:

- El nivel de dispersión de la señal de interés.
- La incoherencia entre las bases Φ y Ψ .

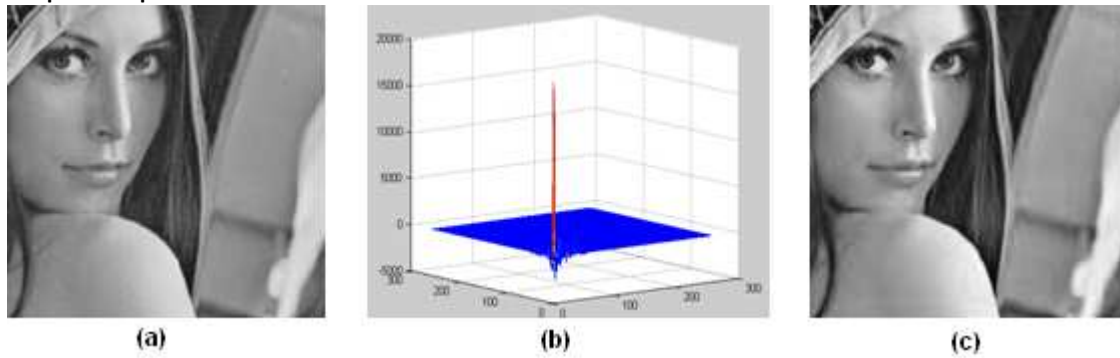
3.2.1 Representación dispersa y nivel de dispersión de una señal: Una señal x puede ser representada en términos de una base $\Psi \in \mathbb{R}^{n \times n}$ conformada por vectores columna $\{\Psi_i\}_{i=1}^n$, denominados átomos.

Esta señal x es k -dispersa en algún dominio cuando se puede representar por una combinación lineal de k -átomos de Ψ , y solo k coeficientes de s_i en (6) son distintos de cero.

El caso de interés es cuando se logra un nivel de dispersión alto, es decir $k \ll n$, ya que se puede establecer con alta probabilidad que la señal x puede ser reconstruida a partir de un reducido número de proyecciones, utilizando un proceso de optimización.

A continuación se expone un ejemplo de una imagen de tamaño 256x256, dispersa en el dominio de la transformada wavelet.

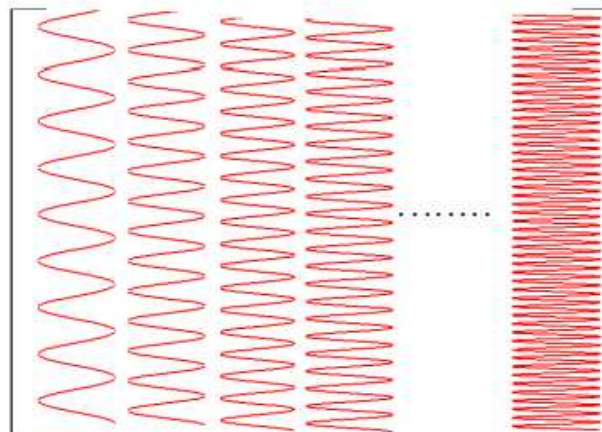
Figura 5. (a) Imagen Original. (b) Coeficientes obtenidos a partir de la base de representación Wavelet (c) Imagen recuperada a partir de los coeficientes Wavelet.



3.2.2 Bases de representación: Las diferentes bases de representación como su nombre lo indica permiten una representación compacta de una amplia clase de señales. Dicha representación se logra mediante una combinación lineal de varias funciones de la base. Obteniendo entonces (6). Entre estas bases encontramos Fourier, Coseno, Wavelet.

- **Transformada de Fourier:** Una base de representación en el dominio de Fourier, consta de una colección de ondas sinusoidales $\Lambda_{(w,v)}$, ver figura 6, donde $w \in [0, 2\pi]$ es la frecuencia angular, y v indica el tipo de fase (coseno o seno).

Figura 6. Representación de la base de Fourier. Cada una de las funciones sinusoidales representa una columna de la base.



- **Transformada Coseno:** Esta transformada es semejante a la transformada rápida de Fourier, con la diferencia de que es real, debido a que los vectores base se componen exclusivamente de funciones coseno muestreadas.

Matemáticamente en el plano unidimensional se define como:

$$C(n) = \alpha(n) \sum_{i=0}^{N-1} f(i) \cos \left[(2i+1)n \frac{\pi}{2N} \right]$$

$$\alpha(n) = \begin{cases} \frac{1}{\sqrt{2}} & n = 0 \\ 1 & n \neq 0 \end{cases} \quad (8)$$

Y (8) sirve de base para el análisis bidimensional, obteniéndose:

$$C(u, v) = \alpha(u)\alpha(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i, j) \cos \left[\frac{\pi(2i+1)u}{2N} \right] \cos \left[\frac{\pi(2j+1)v}{2N} \right] \quad (9)$$

(9) permite crear la matriz de representación coseno del sistema (6).

La transformada del coseno es muy utilizada para comprimir imágenes y videos, sin embargo actualmente la transformada Wavelet está logrando mejores resultados.

- **Transformada Wavelet:** La transformada wavelet es una herramienta matemática empleada para el análisis y la descomposición de señales. Especialmente es utilizada en la compresión de imágenes por sus ventajas al representar las señales en términos de elementos atómicos localizados en tiempo-frecuencia. La transformada de wavelet para una señal de interés x , con un filtro pasa bajo h y el correspondiente filtro pasa alto de h , se expresan en (10) y (11) respectivamente:

$$s_h^{(i)}(j) = \sum_{q=1}^{2^{n-i+1}} h(q-2j) s_h^{(i-1)}(q)$$

$$j = 1, 2, \dots, 2^{n-i} \quad (10)$$

$$s_g^{(i)}(j) = \sum_{q=1}^{2^{n-i+1}} g(q-2j) s_h^{(i-1)}(q) \quad (11)$$

$$j = 1, 2, \dots, 2^{n-i}$$

donde g representa el filtro pasa alto correspondiente a h , y q es el número de puntos diferentes de cero en h y g .

La forma matricial propuesta para la transformada discreta de Wavelet se puede representar de la siguiente manera:

$$\begin{bmatrix} s_h^{(i)} \\ s_g^{(i)} \\ s_g^{(i-1)} \\ \vdots \\ s_g^{(2)} \\ s_g^{(1)} \end{bmatrix} = \begin{bmatrix} H^{(i)}H^{(i-1)} & \dots & H^{(1)} \\ G^{(i)}H^{(i-1)} & \dots & H^{(1)} \\ G^{(i)}H^{(i-2)} & \dots & H^{(1)} \\ \vdots & & \\ G^{(2)}H^{(1)} & & \\ G^{(1)} & & \end{bmatrix} \mathbf{x} = \mathbf{W}\mathbf{x} \quad (12)$$

La transpuesta de W , es la matriz de transformación inversa, donde W es la transformada directa de wavelet.

De (10) y (11) se puede deducir la transformada wavelet en dos dimensiones, realizando la implementación unidimensional en dirección vertical (y) y luego horizontal (x), obteniéndose:

$$s_{hh}^{(i)}(j_1, j_2) = \sum_{l=1}^{2^{n-i+1}} \left[\sum_{q=1}^{2^{d-i+1}} h(q-2j_1) s_{hh}^{(i-1)}(q, l) \right] h(l-2j_2) \quad (13)$$

$$j_1 = 1, 2, \dots, 2^{d-i}; j_2 = 1, 2, \dots, 2^{n-i}$$

$$s_{hg}^{(i)}(j_1, j_2) = \sum_{l=1}^{2^{n-i+1}} \left[\sum_{q=1}^{2^{d-i+1}} h(q-2j_1) s_{hh}^{(i-1)}(q, l) \right] g(l-2j_2) \quad (14)$$

$$j_1 = 1, 2, \dots, 2^{d-i}; j_2 = 1, 2, \dots, 2^{n-i}$$

$$s_{gh}^{(i)}(j_1, j_2) = \sum_{l=1}^{2^{n-i+1}} \left[\sum_{q=1}^{2^{d-i+1}} g(q-2j_1) s_{hh}^{(i-1)}(q, l) \right] h(l-2j_2) \quad (15)$$

$$j_1 = 1, 2, \dots, 2^{d-i}; j_2 = 1, 2, \dots, 2^{n-i}$$

y

$$s_{gg}^{(i)}(j_1, j_2) = \sum_{l=1}^{2^{n-i+1}} \left[\sum_{q=1}^{2^{d-i+1}} g(q - 2j_1) s_{hh}^{(i-1)}(q, l) \right] g(l - 2j_2) \quad (16)$$

$$j_1 = 1, 2, \dots, 2^{d-i}; j_2 = 1, 2, \dots, 2^{n-i}$$

Reordenando las sub-matrices generadas por las ecuaciones (13 a 16) se tiene:

$$\begin{bmatrix} s_{hh} \\ s_{hg} \\ s_{gh} \\ s_{gg} \end{bmatrix} = \begin{bmatrix} H_x \mathbf{X} H_y^t \\ H_x \mathbf{X} G_y^t \\ H_y \mathbf{X} H_x^t \\ H_y \mathbf{X} G_x^t \end{bmatrix} \quad (17)$$

Para expresar en forma de vectores a \mathbf{s} y \mathbf{X} , (17) se convierte en:

$$\begin{bmatrix} s_{hh} \\ s_{hg} \\ s_{gh} \\ s_{gg} \end{bmatrix} = \begin{bmatrix} H_x \otimes H_y \\ H_x \otimes G_y \\ G_x \otimes H_y \\ G_x \otimes G_y \end{bmatrix} \mathbf{x} \quad (18)$$

dónde \otimes representa el producto Kronecker³.

La descomposición en la escala i , a partir de (13-18) expresada en forma matricial (Huiyuan Wang and José Vieira, 2010) está dada por:

³ Operación entre dos matrices de tamaño arbitrario dando como resultado una matriz de bloque (matriz conformada por matrices de menor tamaño).

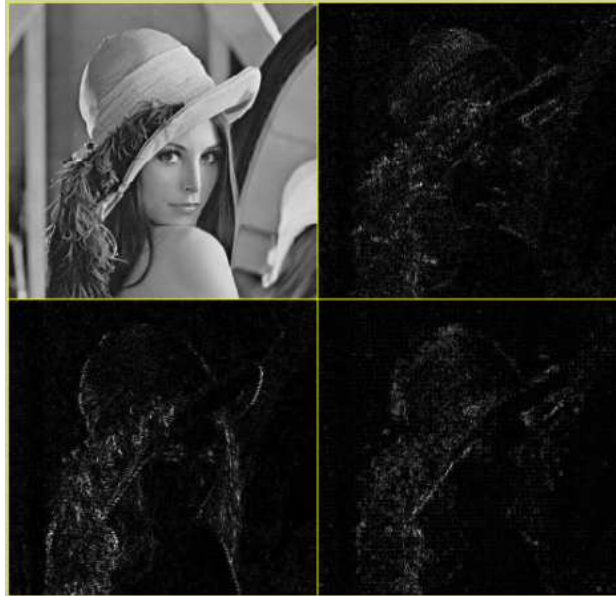
$$\begin{bmatrix} S_{hh}^{(i)} \\ S_{hg}^{(i)} \\ S_{gh}^{(i)} \\ S_{gg}^{(i)} \\ S_{hg}^{(i-1)} \\ S_{gh}^{(i-1)} \\ \vdots \\ S_{gh}^{(1)} \\ S_{gg}^{(1)} \end{bmatrix} = \begin{bmatrix} \left(H_x^{(i)} H_x^{(i-1)} H_x^{(i-2)} \dots H_x^{(1)} \right) \otimes \left(H_y^{(i)} H_y^{(i-1)} H_y^{(i-2)} \dots H_y^{(1)} \right) \\ \left(H_x^{(i)} H_x^{(i-1)} H_x^{(i-2)} \dots H_x^{(1)} \right) \otimes \left(G_y^{(i)} H_y^{(i-1)} H_y^{(i-2)} \dots H_y^{(1)} \right) \\ \left(G_x^{(i)} H_x^{(i-1)} H_x^{(i-2)} \dots H_x^{(1)} \right) \otimes \left(H_y^{(i)} H_y^{(i-1)} H_y^{(i-2)} \dots H_y^{(1)} \right) \\ \left(G_x^{(i)} H_x^{(i-1)} H_x^{(i-2)} \dots H_x^{(1)} \right) \otimes \left(G_y^{(i)} H_y^{(i-1)} H_y^{(i-2)} \dots H_y^{(1)} \right) \\ \left(H_x^{(i-1)} H_x^{(i-2)} \dots H_x^{(1)} \right) \otimes \left(G_y^{(i-1)} H_y^{(i-2)} \dots H_y^{(1)} \right) \\ \left(G_x^{(i-1)} H_x^{(i-2)} \dots H_x^{(1)} \right) \otimes \left(H_y^{(i-1)} H_y^{(i-2)} \dots H_y^{(1)} \right) \\ \vdots \\ G_x^{(1)} \otimes H_y^{(1)} \\ G_x^{(1)} \otimes G_y^{(1)} \end{bmatrix} \mathbf{x} \quad (19)$$

La forma matricial expuesta en (19) aplicada en la teoría de CS se expresa de la forma:

$$\mathbf{s} = \Psi^{-1} \mathbf{x} \quad (20)$$

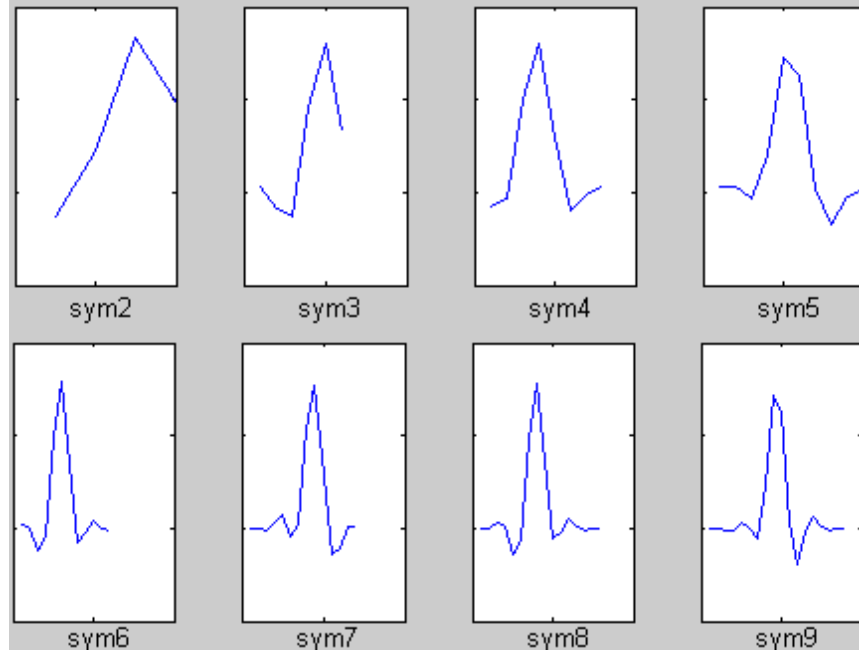
Por medio de la base de representación de Wavelet en dos dimensiones que se genera a partir de (19) utilizando determinados filtros pertenecientes a la familia Wavelet, se descompone una imagen (ver figura 7), generando su respectivo vector de coeficientes.

Figura 7. Descomposición Wavelet en dos dimensiones, empleando el filtro Symlet 8 a un nivel.



En este proyecto se emplean los filtros Symlet8, que son wavelets casi simétricas propuestas por Daubechies⁴ como modificaciones a la familia de filtros Daubechies (db), Ver figura 8.

Figura 8. Familia de filtros Sym(N), donde N representa el orden.



3.2.3 Bases de muestreo: La base de muestreo Φ permite la reconstrucción de la señal x a partir de las m muestras que pertenecen al vector de proyecciones y . Para lograr una buena reconstrucción de los coeficientes s las columnas de Φ deben ser linealmente independientes. Algunas de las matrices que cumplen esta propiedad y que fueron utilizadas en este proyecto son la base de muestreo Hadamard , Scramble block Hadamard ensemble y la base randómica con distribución normal (Gaussiana).

- **Matriz Hadamard:** Una matriz Hadamard de orden $N=2^n$, con $n \geq 0$, es una matriz cuadrada de $2^n \times 2^n$ compuesta por elementos $\{\pm 1\}$ en donde sus columnas y filas son mutuamente ortogonales y linealmente independientes.

⁴ Daubechies Ingrid. Matemática y física, creadora de la familia de filtros Wavelet Daubechies.

Para calcular una matriz T de orden 2^n , se obtiene de forma recursiva, como se muestra en la siguiente figura:

Figura 9. Forma recursiva de crear una matriz Hadamard.

$[1]$	$\left[\begin{array}{c c} 1 & 1 \\ \hline 1 & -1 \end{array} \right]$	$\left[\begin{array}{cc cc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \hline 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{array} \right]$	$\left[\begin{array}{c c} T_{n-1} & T_{n-1} \\ \hline T_{n-1} & -T_{n-1} \end{array} \right]$
$N=2^0$	$N=2^1$	$N=2^2$	$N=2^n$

- **Matriz randómica con distribución normal (Gaussiana):** La matriz Gaussiana de dimensión $m \times n$ se genera a partir de n vectores de medición con $n > 0$, estos se muestrean independientemente con distribución normal, media cero y varianza $1/m$, permitiendo crear una base con sus vectores y filas linealmente independientes.

- **Scramble block Hadamard ensemble:** Debido a los grandes volúmenes de datos que pueden llegar a ser manejados, se puede producir un desbordamiento de la memoria en la mayoría de los computadores. Una alternativa de solución a esta problemática es el procesamiento por bloques que presenta ventajas sobre la velocidad para realizar los cálculos ya que las bases de muestreo tienen tamaños pequeños.

Se ha creado entonces un operador de muestreo rápido denominado Scramble block Hadamard ensemble (SBHE), que toma bloques de la Hadamard parciales y los permuta randómicamente. Matemáticamente se tiene:

$$\Phi = Q_m U P_n \tag{21}$$

donde $U \in R^{n \times n}$ es una matriz de bloques diagonal expresada como:

$$U = \begin{bmatrix} U_B & & & \\ & U_B & & \\ & & \ddots & \\ & & & U_B \end{bmatrix} \tag{22}$$

en la cual U_B representa la matriz Hadamard BXB , P_n es un operador que reordena randómicamente la n columnas de U y Q_m es un operador que toma m filas de UP_n con distribución aleatoria uniforme.

3.2.4 Propiedad de incoherencia: La propiedad principal que se debe tener presente en la selección de la base de muestreo Φ y la base de representación Ψ se denomina incoherencia.

La incoherencia entre la matriz Φ y Ψ se entiende como la relación entre los elementos de cada una de ellas, en donde los elementos de Φ no se pueden obtener mediante alguna combinación lineal de los elementos de la matriz Ψ .

Para medir la coherencia entre dos matrices se puede utilizar la siguiente ecuación:

$$\mu(\Phi, \Psi) = \sqrt{n} \cdot \max_{1 \leq k, j \leq n} |\langle \phi_k, \psi_j \rangle| \text{ en donde } \mu \in [1, \sqrt{n}] \quad (23)$$

De (23) se puede establecer que si Φ y Ψ tienen elementos correlacionados, la incoherencia entre ellas será alta de lo contrario será baja.

Para la teoría de compresión de imágenes CS, la propiedad de incoherencia es muy importante para garantizar reconstrucción.

4. ALGORITMOS DE RECONSTRUCCIÓN

4.1 BASIS PURSUIT (BP)

El principio de BP es encontrar una representación de la señal cuyos coeficientes satisfacen la norma mínima L_1 ; expresada como:

$$x = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix}, \|x\|_1 = \sum_{i=0}^{n-1} (|x_i|), \text{ sujeto a } \Phi x = y \quad (24)$$

La cual puede ser resuelta empleando programación lineal (PL), que en los últimos 40 años, ha servido de apoyo para la solución de una gran cantidad de problemas de optimización. En Basis Pursuit dichos problemas pueden ser tratados mediante el algoritmo de punto interior o el algoritmo simplex.

4.1.1 Método de Punto Interior: Los métodos de punto interior presentan resultados impactantes en la práctica, lo cual los convierte en un método competitivo con el método simplex.

Dado el conjunto de puntos $\{x: \Phi x = y, x \geq 0\}$, donde $\Phi \in \mathbf{R}^{m \times n}$, $y \in \mathbf{R}^m$ y $x \in \mathbf{R}^n$; un punto posible está definido con un punto x bajo las restricciones $\Phi x = y$ y $x \geq 0$. Desde esta perspectiva y desde el punto de vista geométrico el algoritmo simplex trabaja a lo largo de la frontera de la región posible, mientras que los algoritmos de punto interior comienzan en un punto $x^{(0)}$ y se mueven en el interior de la restricción de desigualdad.

Para emplear este algoritmo de programación lineal en la técnica de Basis Pursuit, se inicia de una solución $x^{(0)} \geq 0$ para el problema de representación $\Phi x = y$. De forma iterativa el algoritmo modifica los coeficientes, conservando $\Phi x^{(0)} = y$ y haciendo disperso el vector $x^{(0)}$.

Los algoritmos de punto interior pueden ser descritos con los siguientes problemas:

- **Problema primal:**

$$\min \mathbf{z} = \mathbf{c}^T \mathbf{x}, \quad \text{sujeto a } \Phi \mathbf{x} = \mathbf{y} \text{ con } \mathbf{x} \geq 0 \quad (25)$$

donde \mathbf{c}^T es la transpuesta de los coeficientes. En cada iteración x_t del algoritmo se deben satisfacer las restricciones a las que el problema de minimización está sujeto.

- **Problema dual:**

$$\max \mathbf{w} = \mathbf{y}^T \boldsymbol{\beta}, \quad \text{sujeto a } \Phi^T \boldsymbol{\beta} + \boldsymbol{\delta} = \mathbf{c} \text{ con } \boldsymbol{\delta} \geq 0 \quad (26)$$

donde \mathbf{y}^T representa la transpuesta del vector \mathbf{y} , Φ^T es la transpuesta de la matriz Φ , $\boldsymbol{\beta}$ está definida como las variables de sombra y $\boldsymbol{\delta}$ son las variables de holgura.

- **Problema primal-dual:** tiene como objetivo resolver los problemas primal y dual simultáneamente, convergiendo cuando la brecha de dualidad tiende a cero.

Está definido como problema Primal (25) y problema dual (26) en donde cualquier solución $\mathbf{x}, (\mathbf{y}, \boldsymbol{\delta})$ a los problemas primal y dual respectivamente, será factible si satisfacen la condición de holgura complementaria:

$$\mu = \mathbf{x}_j \boldsymbol{\delta}_j = 0 \quad \text{con } j = 1 \dots \dots \dots n. \quad (27)$$

El objetivo en cada iteración es encontrar vectores $\mathbf{x}(\mu), \boldsymbol{\beta}(\mu)$ y $\boldsymbol{\delta}(\mu)$ donde la brecha de dualidad μ cercana a cero define la mejor solución. Por lo tanto el algoritmo está diseñado para encontrar una secuencia de soluciones primales y duales con la brecha de dualidad decreciente.

Específicamente en cada iteración n se tienen en cuenta:

$$\Phi x = y \quad (28)$$

$$\Phi^T \beta + \delta = c \quad (29)$$

$$x_j \delta_j = \mu \quad j = 1 \dots \dots n \quad (30)$$

$$x, \delta \geq 0 \quad (31)$$

hasta que el límite cuando n tiende a infinito es igual a cero.

En este proyecto se satisfacen las ecuaciones (28-31) teniendo en cuenta la base de muestreo Φ y la base de representación Ψ .

Si se define $x = \Psi s$, donde s representa los coeficientes de la transformada wavelet, (28) queda expresada como:

$$\Phi \Psi s = y \quad (32)$$

siendo s los coeficientes de Wavelet, de tal forma que Ψs es igual a la inversa de la transformada, entonces:

$$\Phi * iWT(x) = y \quad (33)$$

El problema queda expresado en cada iteración como:

$$(\Phi \Psi)^T \beta + \delta = c \quad (34)$$

$$\Psi^T \Phi^T \beta + \delta = c \quad (35)$$

$$dwt(\Phi^T \beta) + \delta = c \quad (36)$$

Tabla 1. Algoritmo Punto Interior

TAREA
<i>Resolver $\theta s = y$, $\theta \in R^{m \times n}$, $y \in R^m$, $m \ll n$</i>
INICIALIZACIÓN
<ul style="list-style-type: none"> • $\theta = \Phi\Psi$ • % <i>compresión</i> • % <i>dispersión</i> • $y =$ <i>señal comprimida</i>
FASE ITERATIVA
<ul style="list-style-type: none"> • <i>Punto inicial $s^{(0)}$</i> • <i>Aplicar Punto Interior</i> <i>Para $t = 0$</i> $\Phi s = y$ $(\Phi\Psi)^T \beta + \delta = c$ <p style="text-align: center;">ó</p> $\Psi^T \Phi^T \beta + \delta = c$ <p style="text-align: center;">ó</p> $dwt(\Phi^T \beta) + \delta = c$ $s_j \delta_j = \mu \quad j = 1 \dots n$ $s, \delta \geq 0$ <p style="text-align: center;">hasta $\lim_{t \rightarrow \infty} \mu = 0$</p> <ul style="list-style-type: none"> • <i>Almacenar los resultados</i> \hat{s} • <i>Reconstruir la señal</i> $s = \Psi \hat{s}$
VALIDAR
$Psnr = 20 \log_{10}(b/rms)$

4.2 ALGORITMO GREEDY ORTHOGONAL MATCHING PURSUIT (OMP)

Una de las técnicas utilizadas en la búsqueda de la solución óptima para los sistemas de ecuaciones lineales subdeterminados es el algoritmo Greedy OMP. Este algoritmo realiza un proceso iterativo, en donde hace una selección de una

posible solución dentro del grupo de soluciones que se obtienen en cada iteración, para encontrar así una solución óptima general.

El algoritmo OMP presenta tres pasos principales:

1. Definición de las entradas del algoritmo.
2. Fase Iterativa
3. Validación

La función principal de OMP es encontrar una aproximación a \mathbf{y} por medio de una combinación lineal de los vectores columna de la matriz \mathbf{A} :

$$\mathbf{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \dots \boldsymbol{\theta}_n] \quad (37)$$

los cuales minimizan la función:

$$\min_{\hat{s}} \|\mathbf{y} - \sum \hat{s}_i \boldsymbol{\theta}_i\|^2 \quad (38)$$

donde i corresponde al i -ésimo valor de \hat{s} .

La aproximación al valor de \mathbf{y} se realiza mediante un proceso iterativo, en donde el algoritmo selecciona los vectores columnas de $\mathbf{\Theta}$ mejor correlacionados con el vector residuo (Mallat y Zhang), el cual se define como la diferencia entre \mathbf{y} y su aproximación parcial, y se denota con la letra \mathbf{r} .

El valor inicial para \mathbf{r}_0 es \mathbf{y} , en seguida se busca la mayor proyección de \mathbf{r}_0 sobre el conjunto de los vectores columna de la matriz $\mathbf{\Theta}$.

Para encontrar estos valores se realiza el siguiente proceso:

$$j_0 = \operatorname{argmax}_{j=1,2,\dots,N} |\langle \mathbf{r}_0, \boldsymbol{\theta}_{j_0} \rangle| \quad (39)$$

$$\lambda_0 = \langle \mathbf{r}_0, \boldsymbol{\theta}_{j_0} \rangle \quad (40)$$

En (39) se almacena la posición j_0 del vector columna de $\mathbf{\Theta}$ con el que se obtuvo el mayor producto interno. El valor máximo que se obtiene entre $\boldsymbol{\theta}_{j_0}$ y \mathbf{r}_0 se almacena en otra variable (40).

Después de este proceso se calcula el nuevo residuo \mathbf{r}_1 de la siguiente forma:

$$\mathbf{r}_1 = \mathbf{r}_0 - |\langle \mathbf{r}_0, \boldsymbol{\theta}_{j_0} \rangle| \boldsymbol{\theta}_{j_0} \quad (41)$$

Este proceso se realiza de forma iterativa k veces, siendo k el número de valores distintos de cero en el vector disperso s . Para evitar que se haga una selección repetida de los vectores columna de Θ , es necesario aplicar un método de Ortogonalización, para esto se puede utilizar el procedimiento de Gram-Schmidt⁵.

$$\boldsymbol{\theta}'_{jt} = \boldsymbol{\theta}_{jt} - \sum_{l=0}^{t-1} \frac{\langle \boldsymbol{\theta}_{jt}, \boldsymbol{\theta}'_{jl} \rangle}{\|\boldsymbol{\theta}'_{jl}\|^2} \boldsymbol{\theta}'_{jl} \quad (42)$$

donde $\boldsymbol{\theta}'_{jt}$ es la versión ortogonalizada del vector columna seleccionado de Θ , con este se calcula el vector residuo de la iteración t .

$$\mathbf{r}_t = \mathbf{r}_{t-1} - \frac{\langle \mathbf{r}_{t-1}, \boldsymbol{\theta}'_{jt} \rangle}{\|\boldsymbol{\theta}'_{jt}\|^2} \boldsymbol{\theta}'_{jt}, \quad t = 2, \dots, m-1 \quad (43)$$

En el vector \mathbf{j} se almacenan los valores de las posiciones de los vectores columna seleccionados y en Λ los valores de los máximos productos internos calculados en cada iteración respectivamente:

$$\mathbf{J} = \{j_1, j_2, j_3, \dots, j_{m-1}\} \quad (44)$$

$$\Lambda = \{\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{m-1}\} \quad (45)$$

Al terminar las iteraciones, se reconstruye el vector:

$$\hat{\mathbf{s}}_i(\mathbf{J}) = \Lambda \quad (46)$$

⁵. Proceso utilizado en matemática y análisis numérico, para ortogonalizar un conjunto de vectores en un espacio Euclidiano R^n .

Tabla 2. Algoritmo Orthogonal Matching Pursuit

TAREA:
<i>Resolver</i> : $\Theta s = y, \quad \Theta \in R^{m \times n}, \quad y \in R^m, m \ll n$
INICIALIZACIÓN
<ul style="list-style-type: none"> • $\Theta = \Phi\Psi$ • $k =$ número de valores diferentes de cero del vector s • $n =$ número de filas de Θ • $y =$ señal comprimida
FASE ITERATIVA
<ul style="list-style-type: none"> • <i>Residuo inicial</i> $r_0 = y$ • <i>Encontrar la mayor proyección de r_0 sobre los vectores columna de Θ</i> $j_0 = \operatorname{argmax}_{j=1,2,\dots,N} \langle r_0, \theta_{j_0} \rangle$ $\lambda_0 = \langle r_0, \theta_{j_0} \rangle$ • <i>Calcular el nuevo residuo</i> $r_1 = r_0 - \langle r_0, \theta_{j_0} \rangle \theta_{j_0}$ • <i>Aplicar el método de Gram – Schmidt</i> $\theta'_{jt} = \theta_{jt} - \sum_{l=0}^{t-1} \frac{\langle \theta_{jt}, \theta'_{jl} \rangle}{\ \theta'_{jl}\ ^2} \theta'_{jl}$ • <i>Calcular el residuo en la iteración t</i> $r_t = r_{t-1} - \frac{\langle r_{t-1}, \theta'_{jt} \rangle}{\ \theta'_{jt}\ ^2} \theta'_{jt}, \quad t = 2, \dots, m - 1$ • <i>Almacenar los resultados</i> $J = \{j_1, j_2, j_3, \dots, j_{m-1}\}$ $\Lambda = \{\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_{m-1}\}$ • <i>Reconstruir el vector</i> $\hat{s}_i(J) = \Lambda$
VALIDAR:
$Psnr = 20 \log_{10}(b/rms)$

5. METODOLOGÍA

Esta sección muestra resultados experimentales que ilustran el rendimiento de los algoritmos Orthogonal Matching Pursuit y Punto Interior para señales en espacios unidimensional y bidimensional.

5.1 Señales Unidimensionales.

Para la realización de pruebas en señales unidimensionales, se tomo una señal x de n muestras, en donde se realizo una variación en el número de componentes k distintos de cero.

Se implementaron las respectivas bases de muestreo Φ (Hadamard y Gaussina), de estas se hicieron variaciones en el número de filas m , siendo $m \ll n$, de acuerdo al porcentaje de compresión deseado.

Para la reconstrucción de la imagen se utilizaron los algoritmos Orthogonal Matching Pursuit y Punto Interior. Se realizando 1000 iteraciones para cada algoritmo y se calculo el porcentaje de recuperación de cada prueba, estos resultados se utilizaron de base para calcular un promedio general.

Para el cálculo del porcentaje de recuperación se utilizó la siguiente ecuación:

$$PR(e) = \begin{cases} 100 - \frac{100e}{\|x\|} \%, & \text{si } 0 \leq e \leq \|x\| \\ 0\% & \text{si } e \geq \|x\| \end{cases} \quad (47)$$

También se observo el tiempo de cómputo que requirió cada algoritmo en el proceso de reconstrucción de la imagen establecida.

5.2 Señales Bidimensionales

Para la realización de pruebas en señales Bidimensionales, se tomaron 3 imágenes de dimensiones 32x32, 64x64 y 128x128 pixeles, en escala de grises y

dispersa en el dominio de Wavelet. Cada una de estas imágenes se toma como un vector x de longitud 1024, 4096 y 16384 respectivamente.

Se implementaron las respectivas bases de muestreo Φ (Hadamard, Scramble Block Hadamard Ensemble (SBHE) y Gaussina), haciendo variaciones en el número de filas m , con $m \ll n$, de acuerdo al porcentaje de compresión deseado (10%,20%,30%,40%,50%).

Para la base de representación Ψ se tienen en cuenta la transformada rápida de wavelet bidimensional (FWT2) y la transformada de wavelet generada teóricamente (WTeórica) dando como resultado una matriz $\Psi \in R^{n \times n}$.

Para la reconstrucción de la imagen se utilizaron los algoritmos Orthogonal Matching Pursuit y Punto Interior. Principalmente se ilustra bajo que porcentajes de compresión en el dominio de la transformada Wavelet es posible recuperar una imagen con un nivel de dispersión del 5%.

Para el cálculo del porcentaje de recuperación de cada algoritmo se utilizó la ecuación del PSNR (Pick Signal to Noise Ratio), este se define matemáticamente:

$$PSNR = 20 * \log_{10}(b/rms) \quad (48)$$

Donde b es el valor máximo que puede tomar la señal (1 a 255) y rms es la diferencia de mínimos cuadrados entre la imagen recuperada y la imagen original. Para cada algoritmo se observó también el tiempo de cómputo que tomo en reconstruir la imagen establecida.

A continuación se presentan las figuras que ilustran los resultados de cada una de las simulaciones aplicadas a señales dispersas unidimensionales y bidimensionales.

El equipo empleado para obtener los resultados consta de 2 GB de memoria RAM, un procesador Intel Core 2 Duo, sistema operativo Windows XP con la versión de matlab 7.9.0.

6. RESULTADOS

• Señales Unidimensionales

A continuación se presentan los resultados obtenidos para las simulaciones realizadas en señales unidimensionales con la implementación del algoritmo OMP.

Figura 10. Porcentaje de recuperación de una señal k -dispersa $s \in R^{256}$ empleando el algoritmo Orthogonal Matching Pursuit con $\Phi = Hadamard$

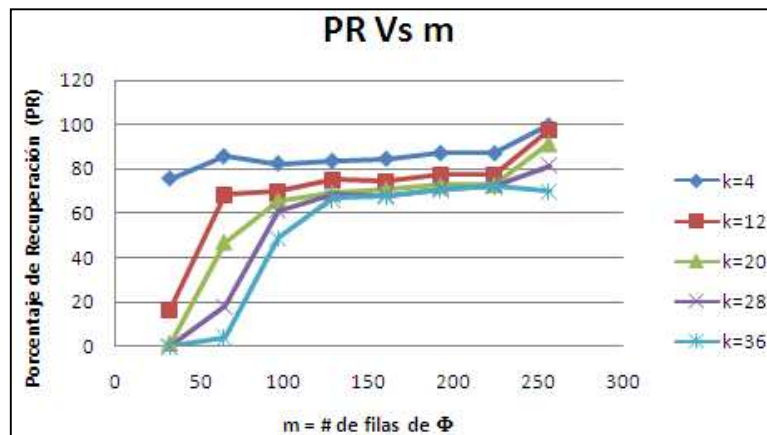


Figura 11 . Porcentaje de recuperación de una señal k -dispersa $s \in R^{256}$ empleando el algoritmo Orthogonal Matching Pursuit con $\Phi = Gaussiana$.

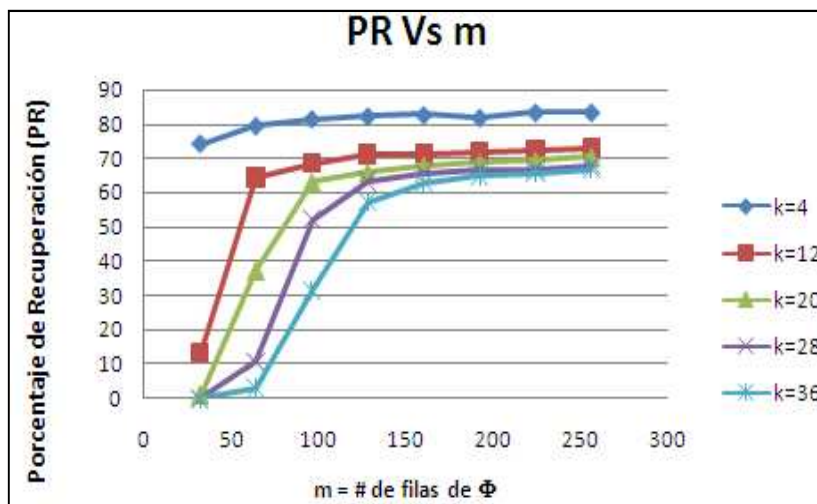


Tabla 3. Resultados obtenidos empleando el algoritmo Orthogonal Matching Pursuit con $\Phi = \text{Hadamard}$ para la reconstrucción de una señal sparse $s \in R^{256}$.

m	32	64	96	128	160	192	224	256
k = 4								
promPR	75,5521	85,7666	82,4739	83,6398	84,5774	87,1754	87,1075	99,9642
Tiempo Seg.	0,00050397	0,00051003	0,00055753	0,000647	0,000779	0,0007074	0,0007407	0,000785
k = 12								
promPR	16,4437	68,4219	69,585	75,3807	74,3922	77,073	77,2056	97,7114
Tiempo Seg.	0,0013	0,0015	0,0018	0,0021	0,0023	0,0027	0,0028	0,0028
k = 20								
promPR	1,0377	46,9742	65,8568	69,395	71,0147	72,6496	73,0401	91,7481
Tiempo Seg.	0,0026	0,0033	0,0037	0,0043	0,005	0,0055	0,006	0,0063
k = 28								
promPR	0,111	17,9098	61,5939	68,4431	68,2074	71,1794	72,7024	81,6106
Tiempo Seg.	0,0045	0,0054	0,0065	0,0076	0,0084	0,0093	0,0101	0,0105
k = 36								
promPR	0,0103	3,9953	49,0286	66,6486	67,6159	70,8872	72,1073	69,9802
Tiempo Seg.	0,0066	0,0081	0,0098	0,0112	0,0128	0,0142	0,0158	0,016

Tabla 4. Resultados obtenidos empleando el algoritmo Orthogonal Matching Pursuit con $\Phi = \text{Gaussiana}$ para la reconstrucción de una señal sparse $s \in R^{256}$.

m	32	64	96	128	160	192	224	256
k = 4								
promPR	74,1461	79,4491	81,509	82,2999	82,6774	81,8498	83,3806	83,4509
Tiempo Seg.	0,000362	0,000413	0,00045	0,000496	0,000566	0,00062	0,000673	0,000733
k = 12								
promPR	13,3027	64,1132	68,3608	70,8865	71,1105	71,4897	72,4145	72,772
Tiempo Seg.	0,0013	0,0015	0,0018	0,0021	0,0022	0,0025	0,0027	0,003
k = 20								
promPR	0,816	37,1159	62,9818	65,8562	67,8297	68,4969	69,2007	70,3566
Tiempo Seg.	0,0026	0,0032	0,0038	0,0042	0,0049	0,0054	0,006	0,0065
k = 28								
promPR	0.0637	10,6572	52,0135	62,9948	65,4745	66,6715	66,6655	67,7624
Tiempo Seg.	0,0049	0,0056	0,0063	0,0072	0,0083	0,0093	0,0103	0,0114
k = 36								
promPR	0,0166	2,9263	31,4061	57,2923	62,767	64,8471	65,8702	66,7325
Tiempo Seg.	0,0067	0,0082	0,0099	0,0112	0,0125	0,0142	0,0157	0,0175

A continuación se presentan los resultados obtenidos para las simulaciones realizadas en señales unidimensionales con la implementación del algoritmo Punto Interior.

Figura 12. Porcentaje de recuperación de una señal k -dispersa $s \in R^{256}$ empleando el algoritmo Punto Interior con $\Phi = \text{Hadamard}$.

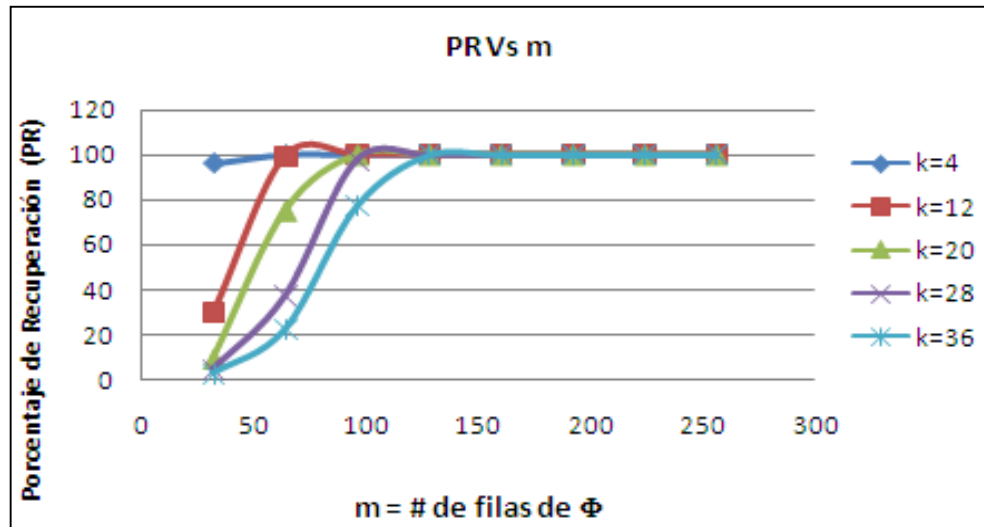


Figura 13. Porcentaje de recuperación de una señal k -dispersa $s \in R^{256}$ empleando el algoritmo Punto Interior con $\Phi = \text{Gaussiana}$.

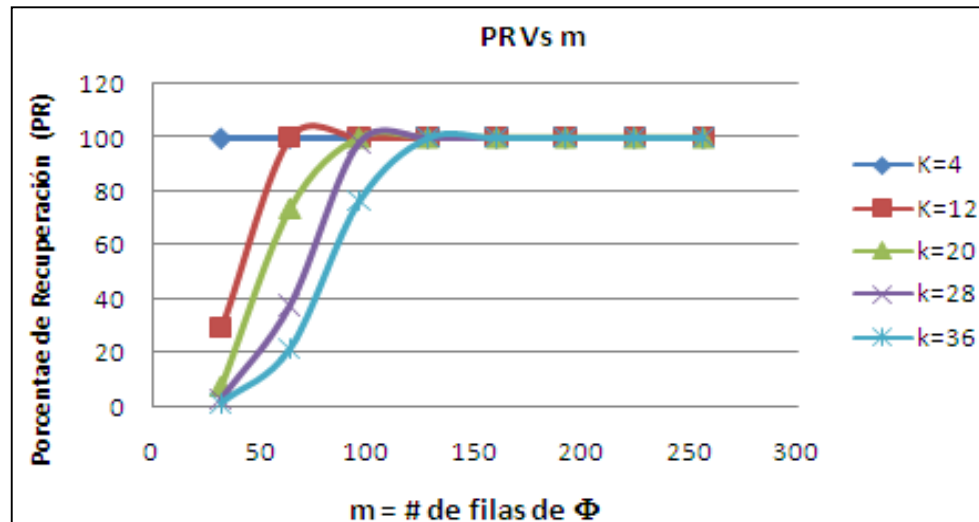


Tabla 5. Resultados obtenidos empleando el algoritmo Punto Interior con $\Phi = \text{Hadamard}$ para la reconstrucción de una señal sparse $s \in R^{256}$.

m	32	64	96	128	160	192	224	256
k = 4								
promPR	96,1596	99,9999	99,9999	100	100	100	100	100
Tiempo Seg.	0,04	0,0454	0,0572	0,0726	0,0851	0,1034	0,1254	0,1325
k = 12								
promPR	29,7963	99,0893	99,9999	100	100	100	100	100
Tiempo Seg.	0,0485	0,0592	0,0613	0,072	0,0942	0,1169	0,1382	0,1495
k = 20								
promPR	9,5043	75,0741	99,9735	99,9999	100	100	100	100
Tiempo Seg.	0,037	0,0631	0,0597	0,0687	0,0837	0,105	0,1185	0,1264
k = 28								
promPR	4,4679	37,0345	97,7875	99,9999	100	100	100	100
Tiempo Seg.	0,0437	0,06	0,0727	0,0753	0,0855	0,1057	0,1239	0,1387
k = 36								
promPR	2,5336	22,1703	77,1391	99,9079	100	100	100	100
Tiempo Seg.	0,0362	0,0617	0,0894	0,0814	0,1012	0,1137	0,1359	0,1493

Tabla 6. Resultados obtenidos empleando el algoritmo Punto Interior con $\Phi = \text{Gaussiana}$ para la reconstrucción de una señal sparse $s \in R^{256}$.

m	32	64	96	128	160	192	224	256
k = 4								
promPR	99,7818	99,9999	99,9999	99,9999	99,9999	100	99,9999	99,7818
Tiempo Seg.	0,0365	0,0413	0,0534	0,0659	0,0797	0,0968	0,1229	0,1326
k = 12								
promPR	29,3254	99,9867	99,9999	99,9999	99,9999	99,9999	99,9998	100
Tiempo Seg.	0,0546	0,0552	0,0611	0,0712	0,0853	0,1061	0,1219	0,1354
k = 20								
promPR	7,9641	73,7859	99,9998	99,9999	99,9999	99,9999	99,9998	100
Tiempo Seg.	0,0541	0,0758	0,0676	0,0774	0,0985	0,1181	0,1356	0,1493
k = 28								
promPR	3,0143	37,8669	98,1294	99,9999	99,9999	99,9999	99,9998	100
Tiempo Seg.	0,0548	0,0824	0,091	0,0931	0,1054	0,1193	0,1442	0,1671
k = 36								
promPR	1,4395	21,7673	76,3761	99,9998	99,9998	99,9999	99,9999	100
Tiempo Seg.	0,0528	0,0716	0,1132	0,1021	0,1099	0,1247	0,1516	0,1703

- **Señales Bidimensionales**

A continuación se presentan los resultados obtenidos para las simulaciones realizadas en señales unidimensionales con la implementación del algoritmo PI.

- Reconstrucción para imagen de tamaño 32x32 pixeles, Nivel de dispersión del 5%.

Figura 14. PSNR para la reconstrucción de una imagen dispersa de 32x32 pixeles empleando el algoritmo de Punto Interior con $\Psi = FWT2$.

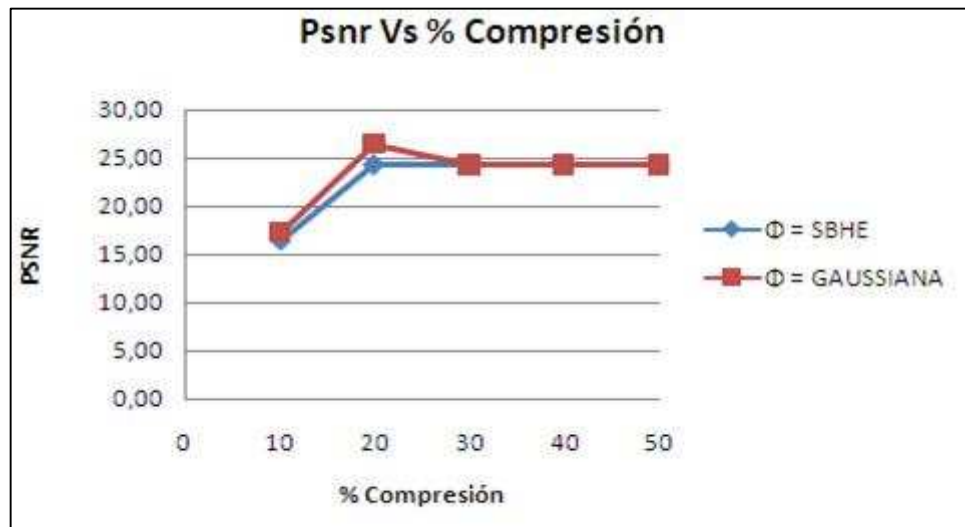


Figura 15. Tiempo de cómputo para la reconstrucción de una imagen dispersa de 32x32 pixeles empleando el algoritmo de Punto Interior con $\Psi = FWT2$.

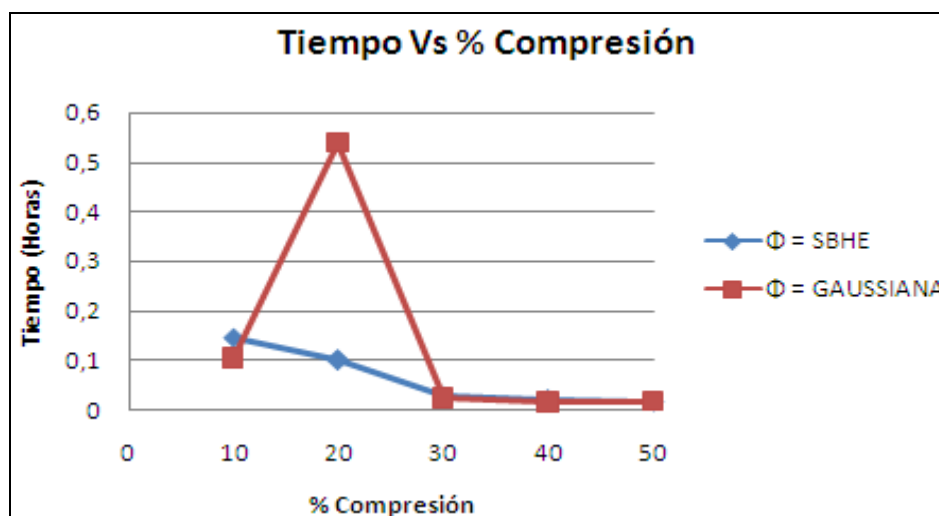


Tabla 7. Resultados obtenidos empleando el algoritmo Punto Interior para la reconstrucción de una imagen de 32x32 pixeles, con $\Psi = FWT2$.

Base de Muestreo Φ	% Compresión	PSNR	Tiempo (Horas)
SBHE	10	16,4427	0,14742017
	20	24,2924	0,09991319
	30	24,2927	0,02869822
	40	24,2926	0,01919978
	50	24,2933	0,0168855
GAUSSIANA	10	17,2422	0,10385167
	20	26,5348	0,53961111
	30	24,2930	0,02226006
	40	24,2936	0,01515036
	50	24,2928	0,01558731

Figura 16. PSNR para la reconstrucción de una imagen dispersa de 32x32 pixeles empleando el algoritmo de Punto Interior con $\Psi = W$ Teórica.

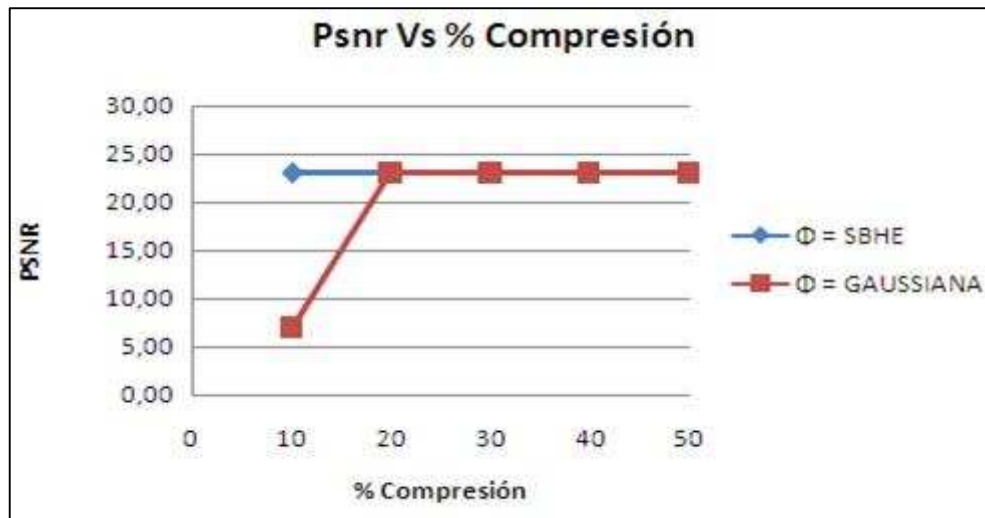


Figura 17. Tiempo de cómputo para la reconstrucción de una imagen dispersa de 32x32 pixeles empleando el algoritmo de Punto Interior con $\Psi = WTeórica$

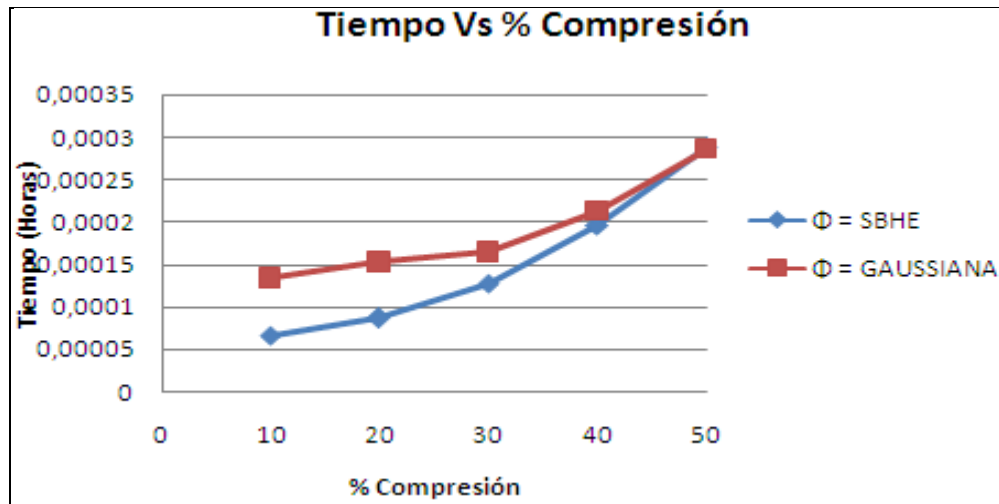


Tabla 8. Resultados obtenidos empleando el algoritmo Punto Interior para la reconstrucción de una imagen de 32x32 pixeles, con $\Psi = WTeórica$.

Base de Muestreo Φ	% Compresión	PSNR	Tiempo (Horas)
SBHE	10	23,0287	6,5778E-05
	20	23,0287	0,00008625
	30	23,0288	0,00012819
	40	23,0288	0,00019606
	50	23,0288	0,00028706
GAUSSIANA	10	6,9704	0,00013389
	20	23,0279	0,00015242
	30	23,0288	0,00016464
	40	23,0288	0,00021319
	50	23,0288	0,00028578

Figura 18. PSNR para la reconstrucción de una imagen dispersa de 32x32 pixeles empleando el algoritmo de Orthogonal Matching Pursuit con $\Psi = FWT2$.

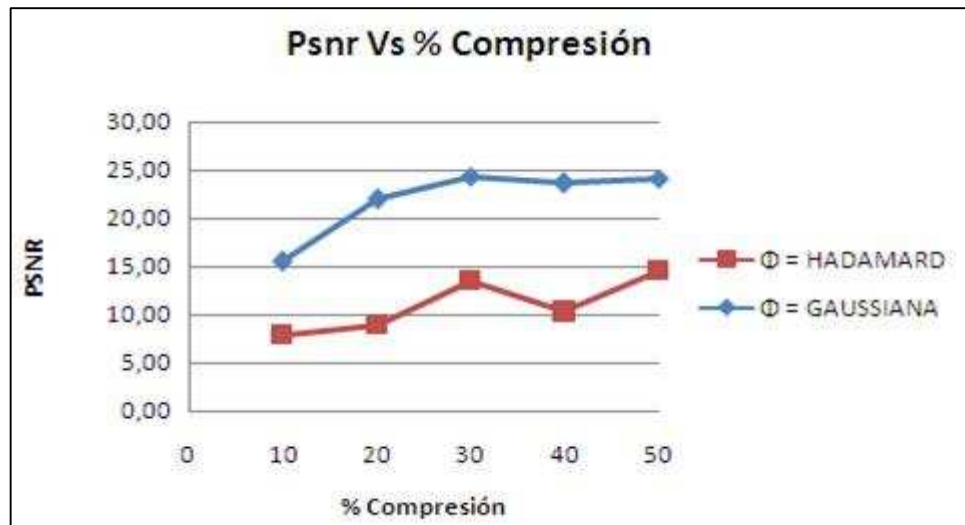


Figura 19. Tiempo de cómputo para la reconstrucción de una imagen dispersa de 32x32 pixeles empleando el algoritmo de Orthogonal Matching Pursuit con $\Psi = FWT2$.

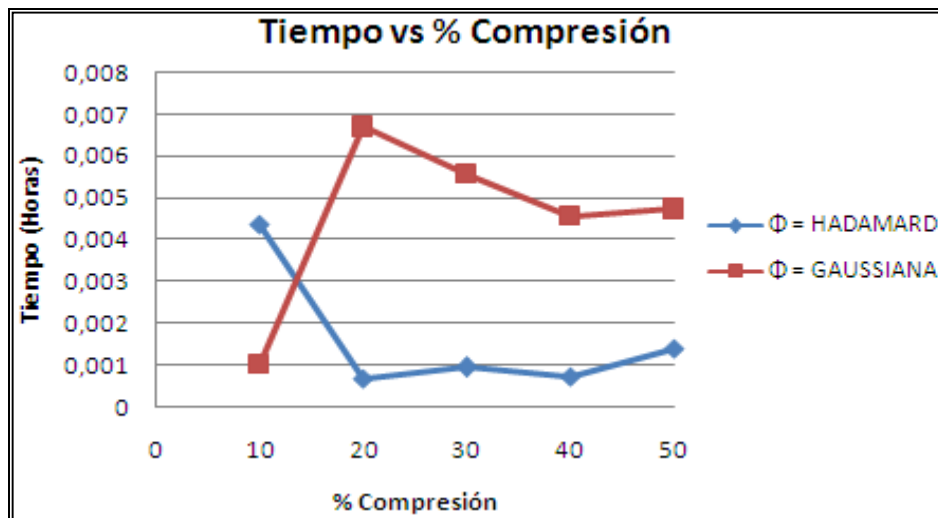


Tabla 9. Resultados obtenidos empleando el algoritmo Orthogonal Matching Pursuit para la reconstrucción de una imagen de 32x32 pixeles, con $\Psi = FWT2$.

Base de Muestreo Φ	% Compresión	PSNR	Tiempo (Horas)	Columnas Recuperadas
HADAMARD	10	7,9110	0,00437986	98 %
	20	8,9059	0,00067461	95 %
	30	13,5505	0,00093786	95 %
	40	10,3319	0,00073453	95 %
	50	14,6260	0,00138025	95 %
GAUSSIANA	10	15,5229	0,00101317	97%
	20	22,0031	0,00668617	98%
	30	24,4407	0,00556283	99%
	40	23,6569	0,00457419	99%
	50	24,1870	0,00473344	99%

Figura 20. PSNR para la reconstrucción de una imagen dispersa de 32x32 pixeles empleando el algoritmo de Orthogonal Matching Pursuit con $\Psi = WTeórica$.

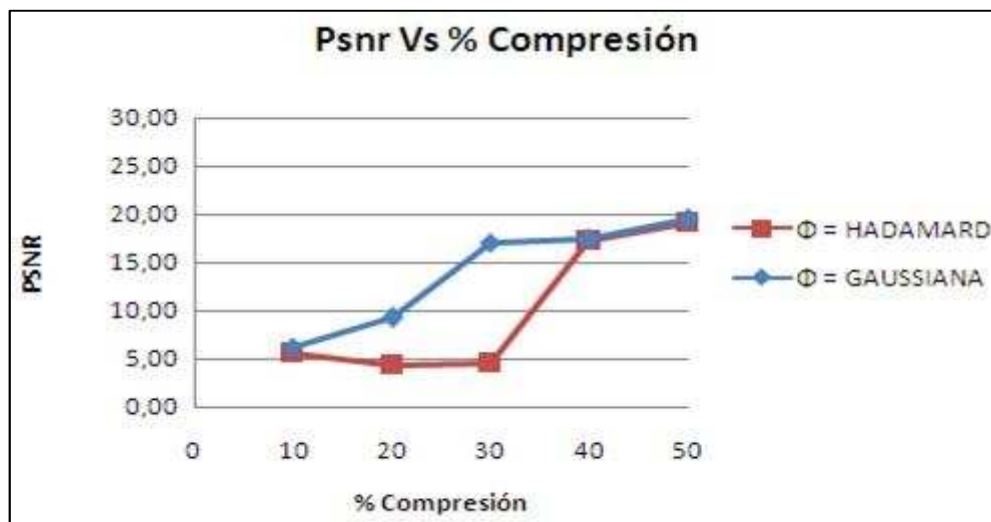


Figura 21. Tiempo de cómputo para la reconstrucción de una imagen dispersa de 32x32 pixeles empleando el algoritmo de Orthogonal Matching Pursuit con $\Psi = W_{Teórica}$.

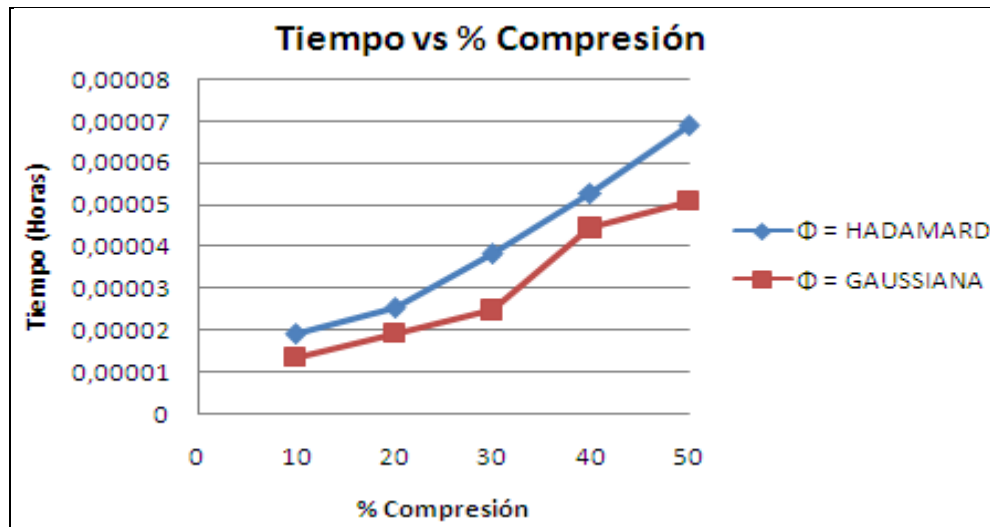


Tabla 10. Resultados obtenidos empleando el algoritmo Orthogonal Matching Pursuit para la reconstrucción de una imagen de 32x32 pixeles, con $\Psi = W_{Teórica}$

Base de Muestreo Φ	% Compresión	PSNR	Tiempo (Horas)	Columnas Recuperadas
HADAMARD	10	5,7034	1,8944E-05	91 %
	20	4,4328	0,00002525	96 %
	30	4,5494	3,8194E-05	97 %
	40	17,3291	5,2528E-05	99 %
	50	19,1138	6,8889E-05	99 %
GAUSSIANA	10	6,2055	1,3472E-05	91%
	20	9,2692	1,9194E-05	97%
	30	17,0328	2,4722E-05	99%
	40	17,4361	4,4444E-05	99%
	50	19,5179	5,0861E-05	99%

- Reconstrucción para imagen de tamaño 64x64 pixeles, Nivel de dispersión del 5%.

Figura 22. PSNR para la reconstrucción de una imagen dispersa de 64x64 pixeles empleando el algoritmo de PuntoInterior con $\Psi = FWT2$.

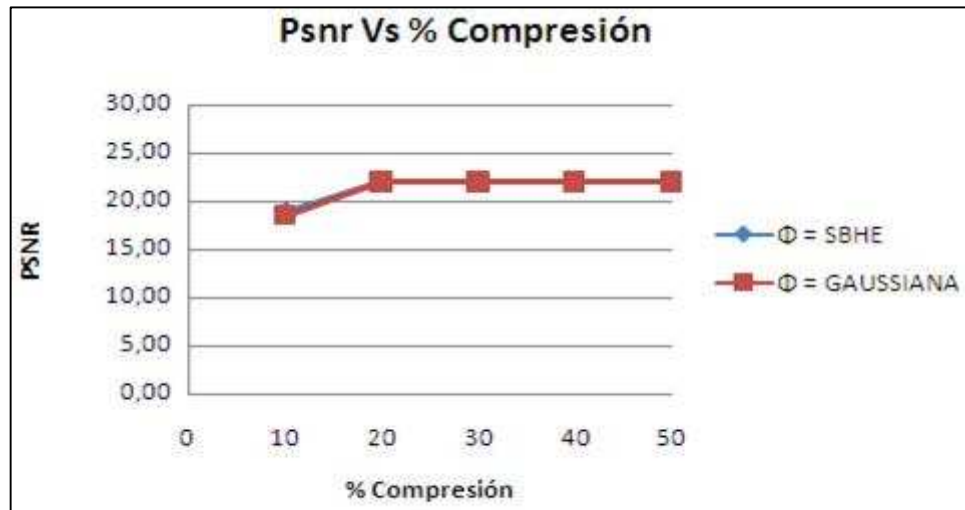


Figura 23. Tiempo de cómputo para la reconstrucción de una imagen dispersa de 64x64 pixeles empleando el algoritmo de Punto Interior con $\Psi = FWT2$.

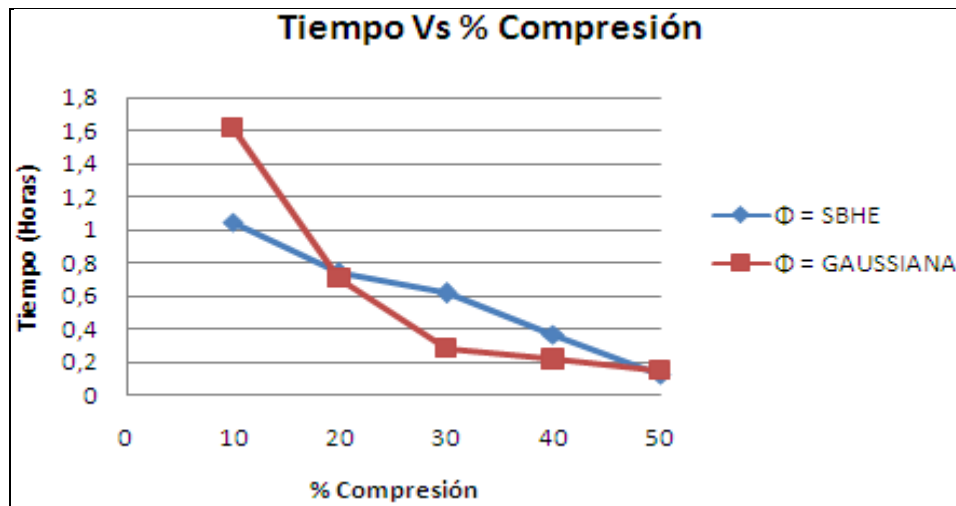


Tabla 11. Resultados obtenidos empleando el algoritmo Punto Interior para la reconstrucción de una imagen de 64x64 pixeles, con $\Psi = FWT2$.

Base de Muestreo Φ	% Compresión	PSNR	Tiempo (Horas)
SBHE	10	18,849	1,04388889
	20	21,981	0,735
	30	21,9762	0,61408333
	40	21,9771	0,36852778
	50	21,9762	0,12091383
GAUSSIANA	10	18,6105	1,62080556
	20	21,9747	0,70961111
	30	21,9739	0,27650136
	40	21,9667	0,21468322
	50	21,9784	0,14450481

Figura 24. PSNR para la reconstrucción de una imagen dispersa de 64x64 pixeles empleando el algoritmo de Punto Interior con $\Psi = WTeórica$.

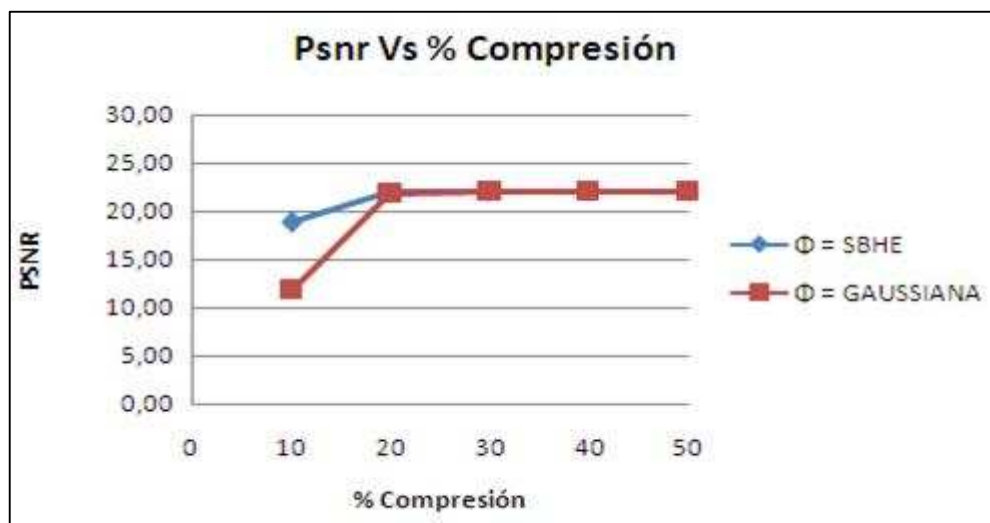


Figura 25. Tiempo de cómputo para la reconstrucción de una imagen dispersa de 64x64 pixeles empleando el algoritmo de Punto Interior con $\Psi = WTeórica$.

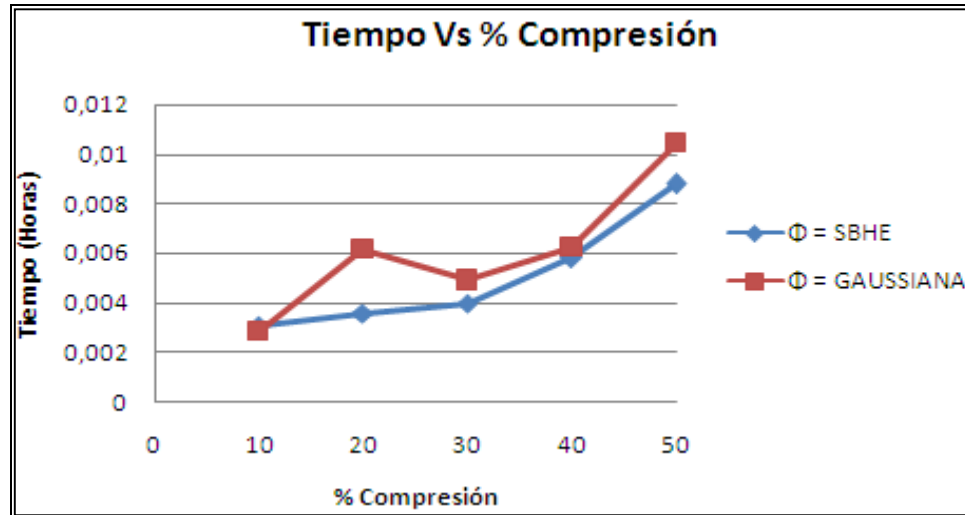


Tabla 12. Resultados obtenidos empleando el algoritmo Punto Interior para la reconstrucción de una imagen de 64x64 pixeles, con $\Psi = WTeórica$.

Base de Muestreo Φ	% Compresión	PSNR	Tiempo (Horas)
SBHE	10	18,8606	0,00307156
	20	22,1358	0,00357094
	30	22,1361	0,00397697
	40	22,1359	0,00579244
	50	22,1362	0,00878969
GAUSSIANA	10	11,9295	0,00282208
	20	21,8603	0,00615567
	30	22,1360	0,00489428
	40	22,1359	0,00623331
	50	22,1360	0,01044619

Figura 26. PSNR para la reconstrucción de una imagen dispersa de 64x64 pixeles empleando el algoritmo de Orthogonal Matching Pursuit con $\Psi = FWT2$.

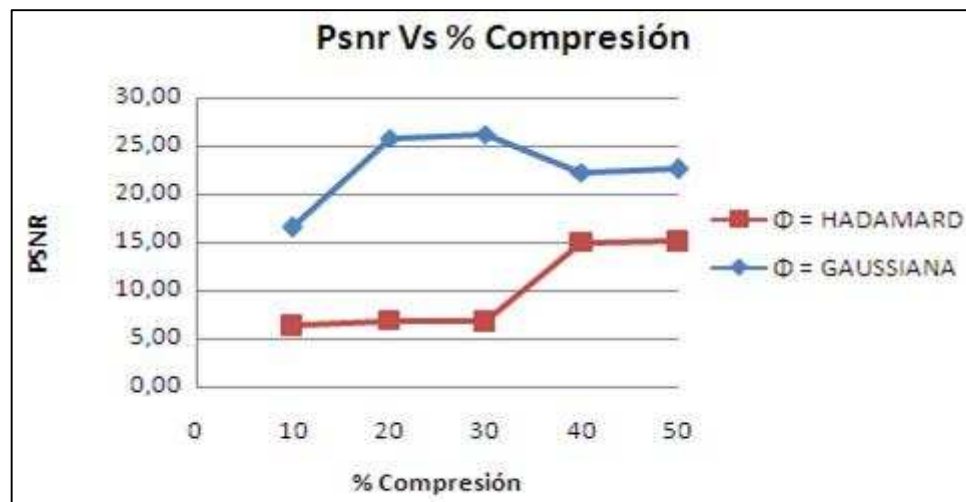


Figura 27. Tiempo de cómputo para la reconstrucción de una imagen dispersa de 64x64 pixeles empleando el algoritmo de Orthogonal Matching Pursuit con $\Psi = FWT2$.

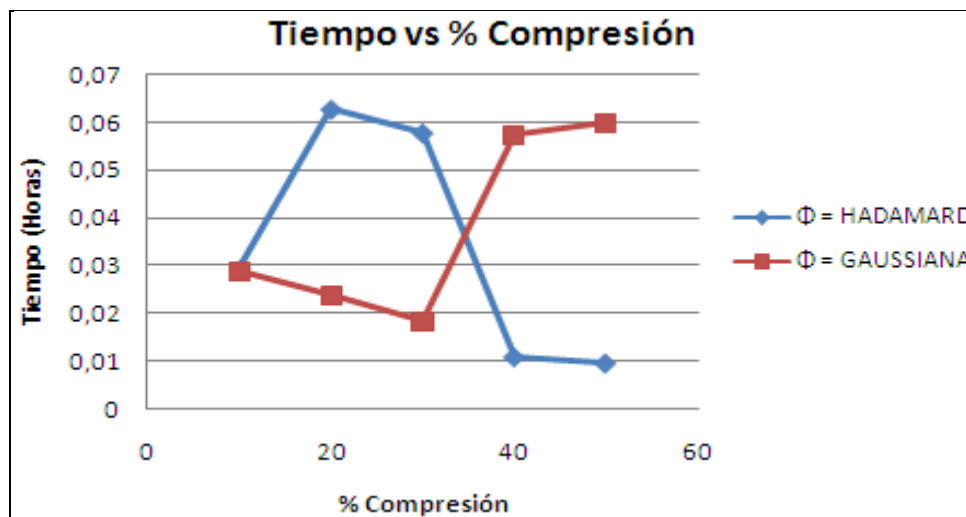


Tabla 13. Resultados obtenidos empleando el algoritmo Orthogonal Matching Pursuit para la reconstrucción de una imagen de 64x64 pixeles, con $\Psi = FWT2$.

Base de Muestreo Φ	% Compresión	PSNR	Tiempo (Horas)	Columnas Recuperadas
HADAMARD	10	6,3845	0,02920792	97 %
	20	6,8267	0,06271442	99 %
	30	6,7725	0,05776	99 %
	40	14,9821	0,010852	94 %
	50	15,0845	0,00955011	95 %
GAUSSIANA	10	16,5584	0,02877981	95%
	20	25,8731	0,02359697	98%
	30	26,2491	0,01827356	99%
	40	22,3020	0,05756053	97%
	50	22,6369	0,05987886	97%

Figura 28. PSNR para la reconstrucción de una imagen dispersa de 64x64 pixeles empleando el algoritmo de Orthogonal Matching Pursuit con $\Psi = WTeórica$.

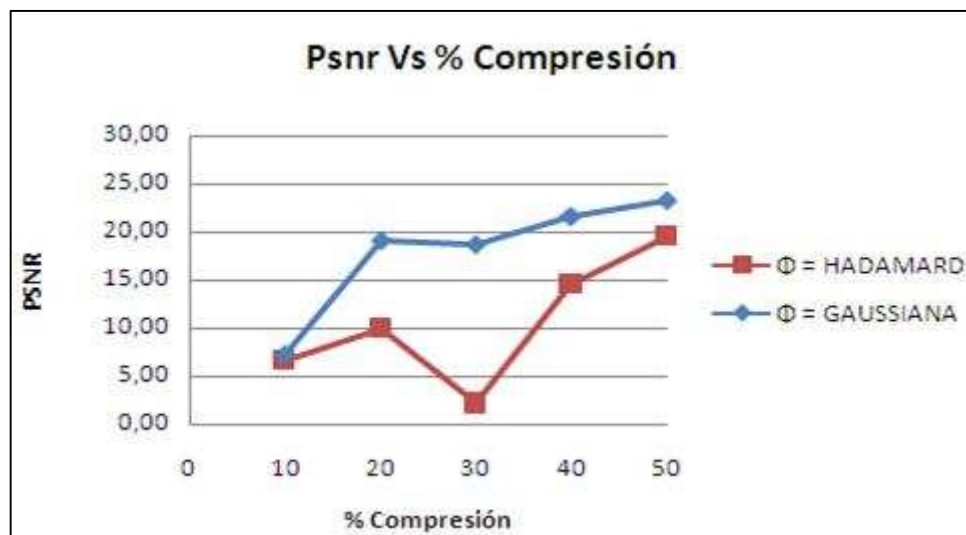


Figura 29. Tiempo de cómputo para la reconstrucción de una imagen dispersa de 64x64 pixeles empleando el algoritmo de Orthogonal Matching Pursuit con $\Psi = W_{Teórica}$.

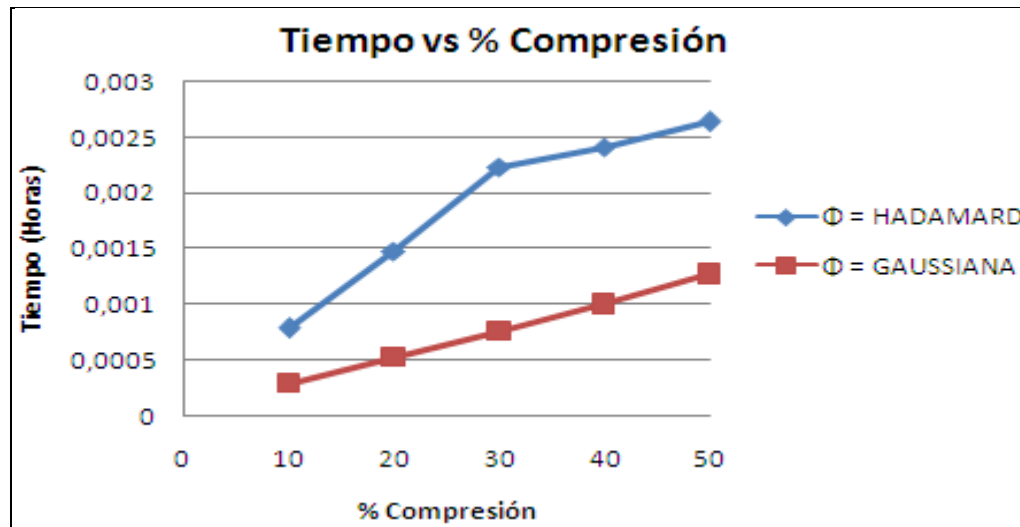


Tabla 14. Resultados obtenidos empleando el algoritmo Orthogonal Matching Pursuit para la reconstrucción de una imagen de 64x64 pixeles, con $\Psi = W_{Teórica}$

Base de Muestreo Φ	% Compresión	PSNR	Tiempo (Horas)	Columnas Recuperadas
HADAMARD	10	6,7314	0,000782611	95 %
	20	9,9568	0,001480389	96 %
	30	2,2104	0,002227833	99 %
	40	14,5345	0,002416111	96 %
	50	19,5876	0,002638222	99 %
GAUSSIANA	10	7,3003	0,000294778	96%
	20	19,2415	0,000519	98%
	30	18,7034	0,000760833	98%
	40	21,7057	0,000997583	99%
	50	23,2997	0,001270639	99%

- Reconstrucción para imagen de tamaño 128x128 pixeles, Nivel de dispersión del 5%.

Figura 30. PSNR para la reconstrucción de una imagen dispersa de 128x128 pixeles empleando el algoritmo de Punto Interior con $\Psi = FWT2$

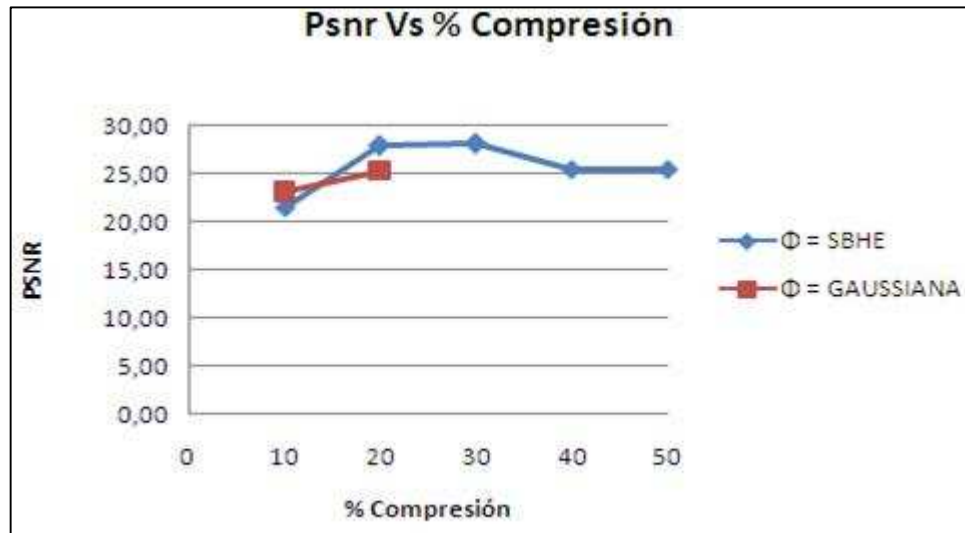


Figura 31. PSNR para la reconstrucción de una imagen dispersa de 128x128 pixeles empleando el algoritmo de Punto Interior con $\Psi = FWT2$

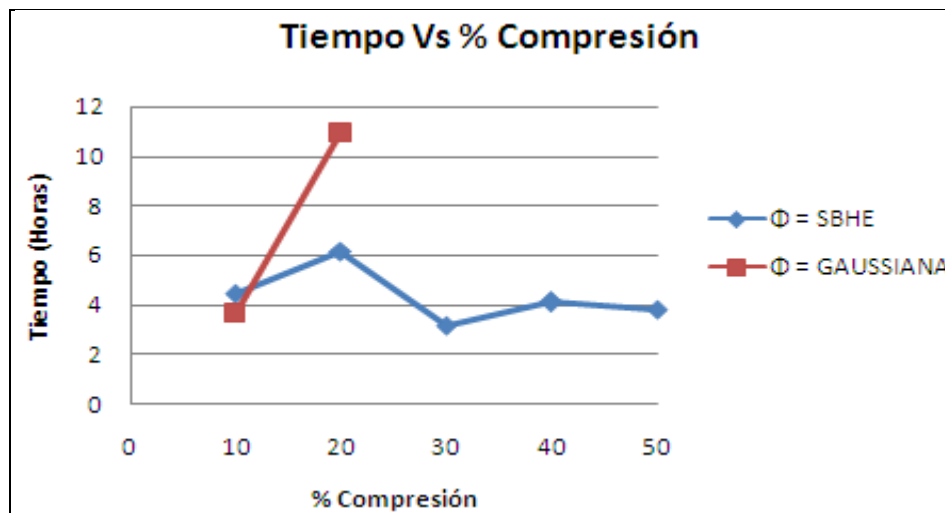


Tabla 15. Resultados obtenidos empleando el algoritmo Punto Interior para la reconstrucción de una imagen de 128x128 pixeles, con $\Psi = FWT2$.

Base de Muestreo Φ	% Compresión	PSNR	Tiempo (Horas)
SBHE	10	21,5064	4,46722222
	20	27,7932	6,14416667
	30	28,0022	3,1425
	40	25,4102	4,09555556
	50	25,4183	3,81138889
GAUSSIANA	10	23,0564	3,66361111
	20	25, 2310	10,96444444
	30	Limite de Memoria	Limite de Memoria
	40	Limite de Memoria	Limite de Memoria
	50	Limite de Memoria	Limite de Memoria

Para los porcentajes de compresión superiores al 20% empleando la base de muestreo Gaussiana no se obtuvieron resultados debido a que se presentaron limitaciones de Hardware, el tamaño de la base excedía los límites de memoria.

- **IMÁGENES**

A continuación se presentan las imágenes reconstruidas.

Tabla 16. Imagen de 32x32 pixeles recuperada empleando el algoritmo de Punto Interior con $\Psi = \text{FWT2}$, $\Phi = \text{SBHE}$ y nivel de dispersión de 5%.

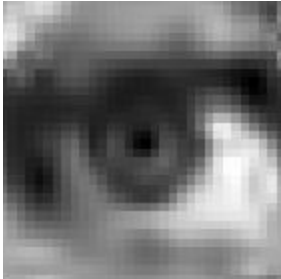
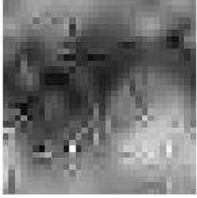

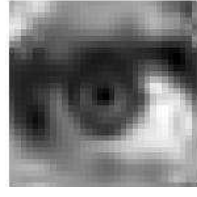

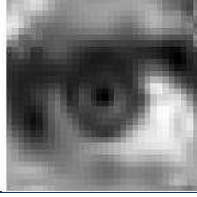
% Compresión	Imagen Original	Imagen Recuperada	PSNR
10			16,4427
20			24,2924
30			24,2927
40			24,2926
50			24,2933

Tabla 17. Imagen de 32x32 pixeles recuperada empleando el algoritmo de Punto Interior con $\Psi = \text{FWT2}$, $\Phi = \text{GAUSSIANA}$ y nivel de dispersión de 5%.

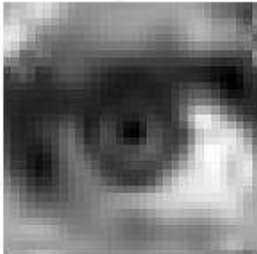
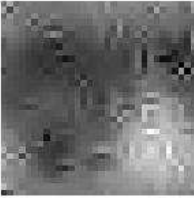
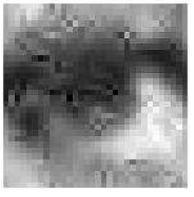



% Compresión	Imagen Original	Imagen Recuperada	PSNR
10			17,2422
20			26,5348
30			24,2930
40			24,2936
50			24,2928

Tabla 18. Imagen de 64x64 pixeles recuperada empleando el algoritmo de Punto Interior con $\Psi = \text{FWT2}$, $\Phi = \text{GAUSSIANA}$ y nivel de dispersión de 5%.

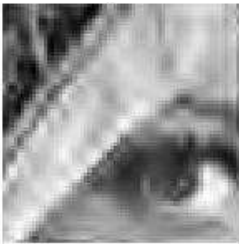

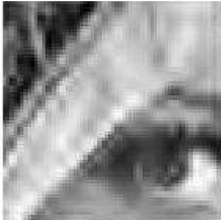

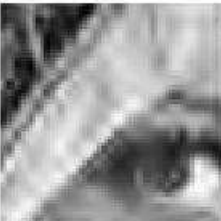

% Compresión	Imagen Original	Imagen Recuperada	PSNR
10			18,6105
20			21,9747
30			21,9739
40			21,9667
50			21,9784

Tabla 19. Imagen de 64x64 pixeles recuperada empleando el algoritmo de Punto Interior con $\Psi = W_{Teórica}$, $\Phi = GAUSSIANA$ y nivel de dispersión de 5%.

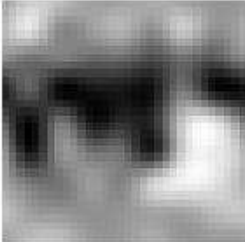
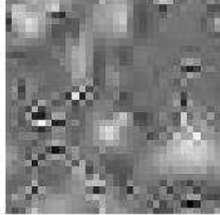
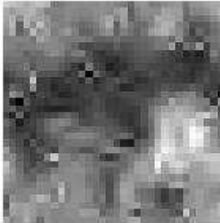

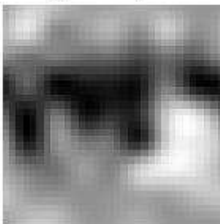

% Compresión	Imagen Original	Imagen Recuperada	PSNR
10			11,9295
20			21,8603
30			22,1360
40			22,1359
50			22,1360

Tabla 20. Imagen de 32x32 pixeles recuperada empleando el algoritmo Punto Interior con $\Psi = W_{\text{teórica}}$, $\Phi = \text{SBHE}$ y nivel de dispersión de 5%.





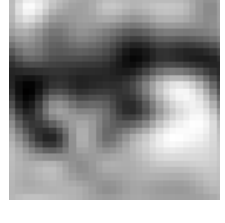

% Compresión	Imagen Original	Imagen Recuperada	PSNR
10			23,0287
20			23,0287
30			23,0288
40			23,0288
50			23,0288

Tabla 21. Imagen de 32x32 pixeles recuperada empleando el algoritmo Punto Interior con $\Psi = W$ Teórica, $\Phi = \text{GAUSSIANA}$ y nivel de dispersión de 5%.


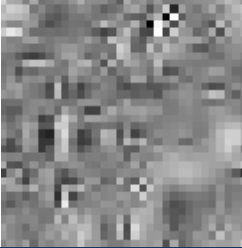


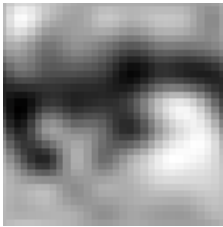
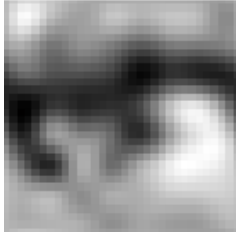
% Compresión	Imagen Original	Imagen Recuperada	PSNR
10			6,9704
20			23,0279
30			23,0288
40			23,0288
50			23,0288

Tabla 22. Imagen de 64x64 pixeles recuperada empleando el algoritmo Punto Interior con $\Psi = \text{FWT2}$, $\Phi = \text{SBHE}$ y nivel de dispersión de 5%.






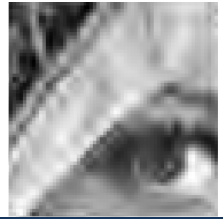
% Compresión	Imagen Original	Imagen Recuperada	PSNR
10			18,849
20			21,981
30			21,9762
40			21,9771
50			21,9762

Tabla 23. Imagen de 64x64 pixeles recuperada empleando el algoritmo Punto Interior con $\Psi = W$ Teórica, $\Phi = SBHE$ y nivel de dispersión de 5%.







% Compresión	Imagen Original	Imagen Recuperada	PSNR
10			18,8606
20			22,1358
30			22,1361
40			22,1359
50			22,1362

Tabla 24. Imagen de 32x32 pixeles recuperada empleando el algoritmo Orthogonal Matching Pursuit con $\Psi = \text{FWT2}$, $\Phi = \text{HADAMARD}$ y nivel de dispersión de 5%.


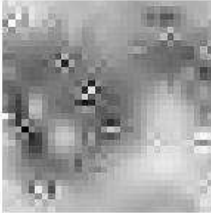
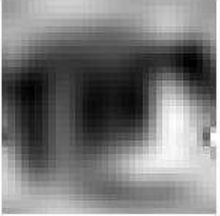
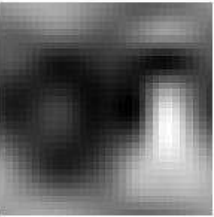
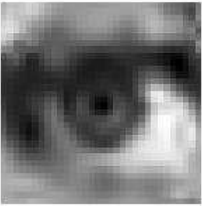

% Compresión	Imagen Original	Imagen Recuperada	PSNR
10			7,9110
20			8,9059
30			13,5505
40			10,3319
50			14,6260

Tabla 25. Imagen de 32x32 pixeles recuperada empleando el algoritmo Orthogonal Matching Pursuit con $\Psi = \text{FWT2}$, $\Phi = \text{GAUSSIANA}$ y nivel de dispersión de 5%.

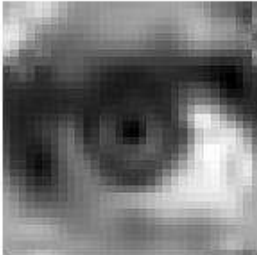
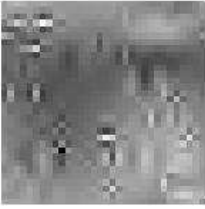


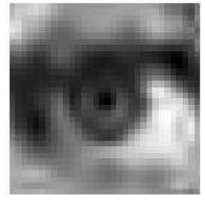
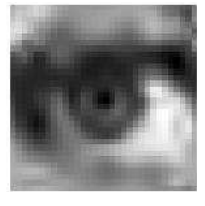
% Compresión	Imagen Original	Imagen Recuperada	PSNR
10			15,5229
20			22,0031
30			24,4407
40			23,6569
50			24,1870

Tabla 26. Imagen de 32x32 pixeles recuperada empleando el algoritmo Orthogonal Matching Pursuit con $\Psi = W_{\text{Teórica}}$, $\Phi = \text{HADAMARD}$ y nivel de dispersión de 5%.


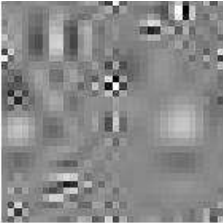
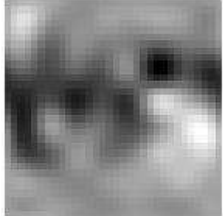



% Compresión	Imagen Original	Imagen Recuperada	PSNR
10			5,7034
20			4,4328
30			4,5494
40			17,3291
50			19,1138

Tabla 27. Imagen de 32x32 pixeles recuperada empleando el algoritmo Orthogonal Matching Pursuit con $\Psi = W_{\text{teórica}}$, $\Phi = \text{GAUSSIANA}$ y nivel de dispersión de 5%.


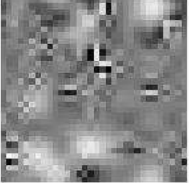




% Compresión	Imagen Original	Imagen Recuperada	PSNR
10			6,2055
20			9,2692
30			17,0328
40			17,4361
50			19,5179

Tabla 28. Imagen de 64x64 pixeles recuperada empleando el algoritmo Orthogonal Matching Pursuit con $\Psi = \text{FWT2}$, $\Phi = \text{HADAMARD}$ y nivel de dispersión de 5%.

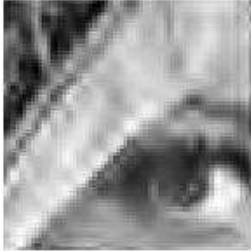
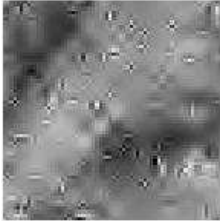




% Compresión	Imagen Original	Imagen Recuperada	PSNR
10			6,3845
20			6,8267
30			6,7725
40			14,9821
50			15,0845

Tabla 29. Imagen de 64x64 pixeles recuperada empleando el algoritmo Orthogonal Matching Pursuit con $\Psi = \text{FWT2}$, $\Phi = \text{GAUSSIANA}$ y nivel de dispersión de 5%.





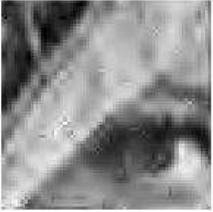

% Compresión	Imagen Original	Imagen Recuperada	PSNR
10			16,5584
20			25,8731
30			26,2491
40			22,3020
50			22,6369

Tabla 30. Imagen de 64x64 pixeles recuperada empleando el algoritmo Orthogonal Matching Pursuit con $\Psi = W$ Teórica, $\Phi = \text{HADAMARD}$ y nivel de dispersión de 5%.







% Compresión	Imagen Original	Imagen Recuperada	PSNR
10			6,7314
20			9,9568
30			2,2104
40			14,5345
50			19,5876

Tabla 31. Imagen de 64x64 pixeles recuperada empleando el algoritmo Orthogonal Matching Pursuit con $\Psi = W_{\text{Teórica}}$, $\Phi = \text{GAUSSIANA}$ y nivel de dispersión de 5%.


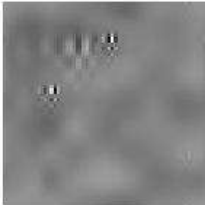
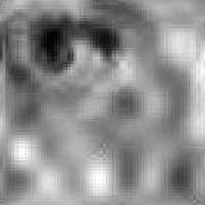
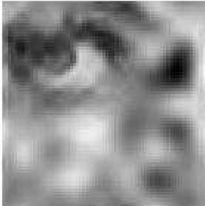







% Compresión	Imagen Original	Imagen Recuperada	PSNR
10			7,3003
20			19,2415
30			18,7034
40			21,7057
50			23,2997

Tabla 32. Imagen de 128x128 pixeles recuperada empleando el algoritmo de Punto Interior con $\Psi = \text{FWT2}$, $\Phi = \text{SBHE}$ y nivel de dispersión de 5%.

% Compresión	Imagen Original	Imagen Recuperada	PSNR
10			21,5064
20			27,7932
30			28,0022
40			25,4102

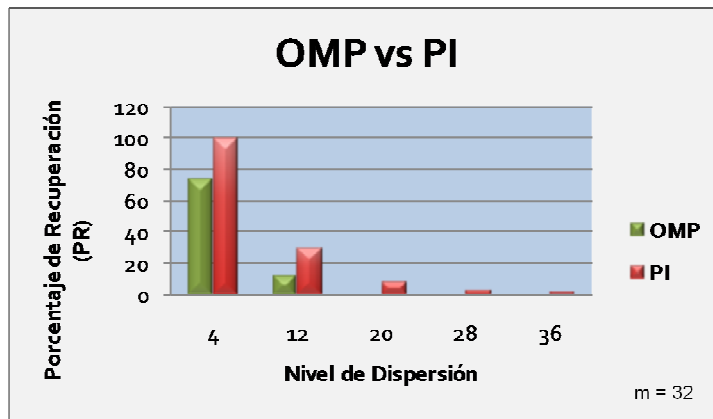
50			25,4183
----	--	--	---------

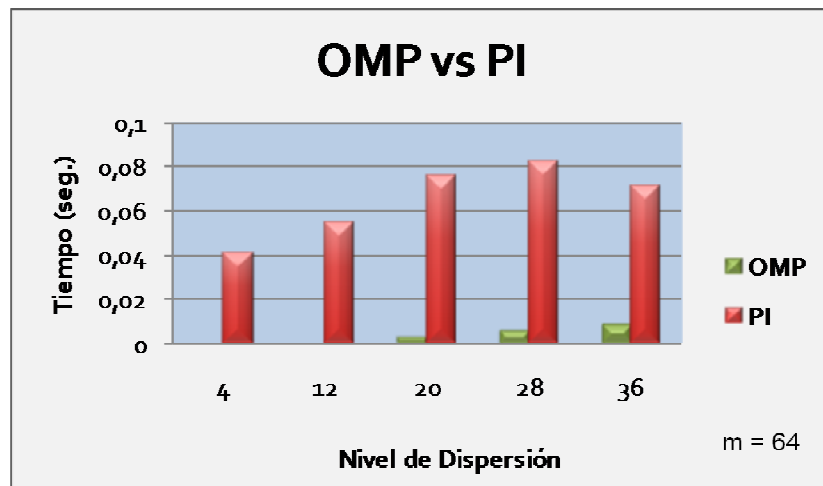
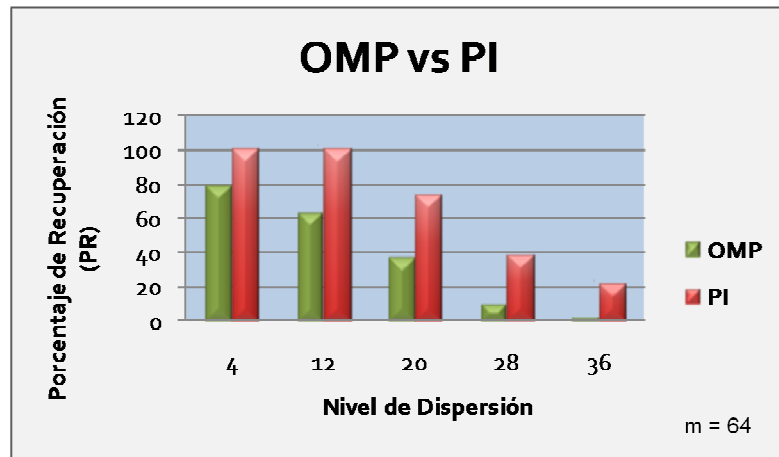
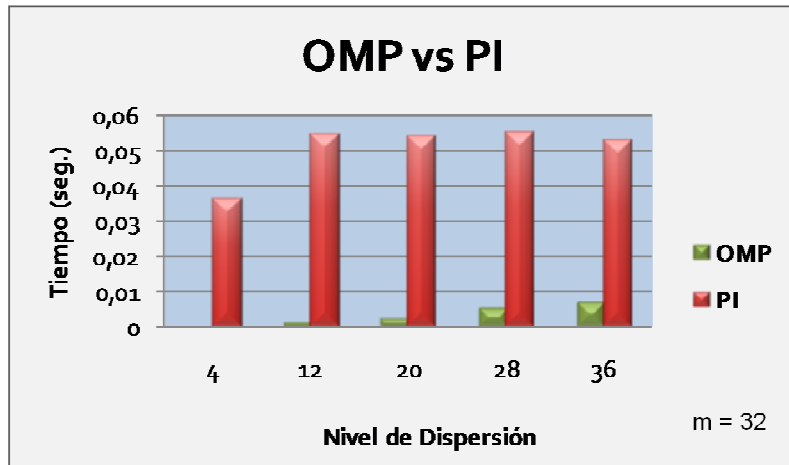
- **Análisis Comparativo**

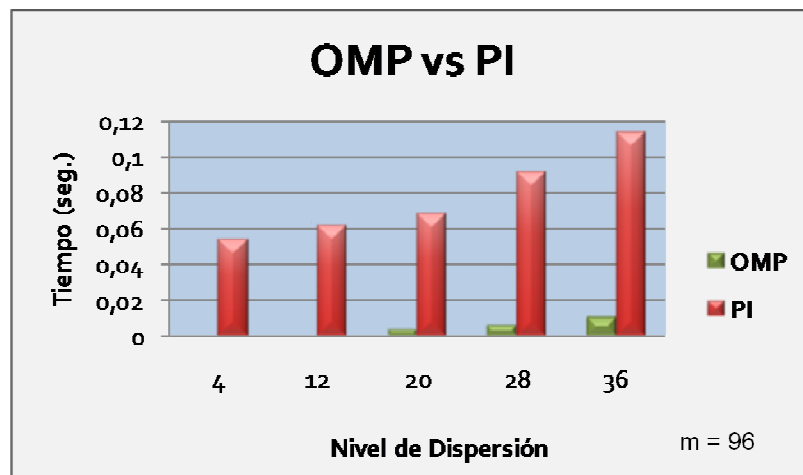
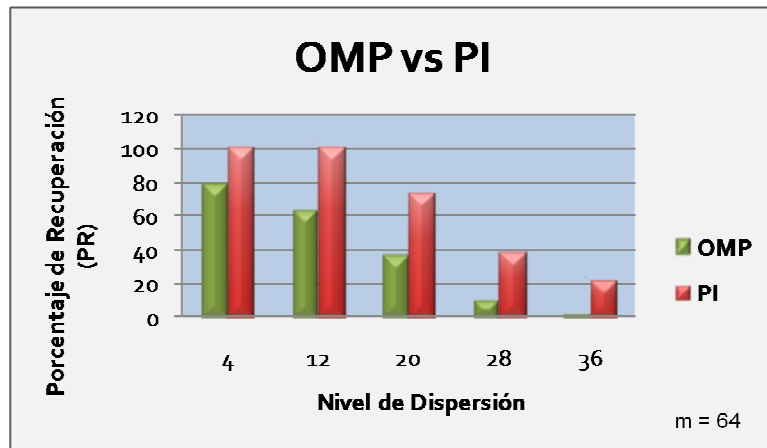
Las gráficas que se muestran a continuación son los resultados comparativos de las reconstrucciones obtenidas para señales unidimensionales y bidimensionales.

Análisis Comparativo para señales unidimensionales

Para las señales unidimensionales se obtuvo que la matriz de Hadarmard presenta mejores resultados de reconstrucción tanto para el algoritmo OMP como para el algoritmo Punto Interior.



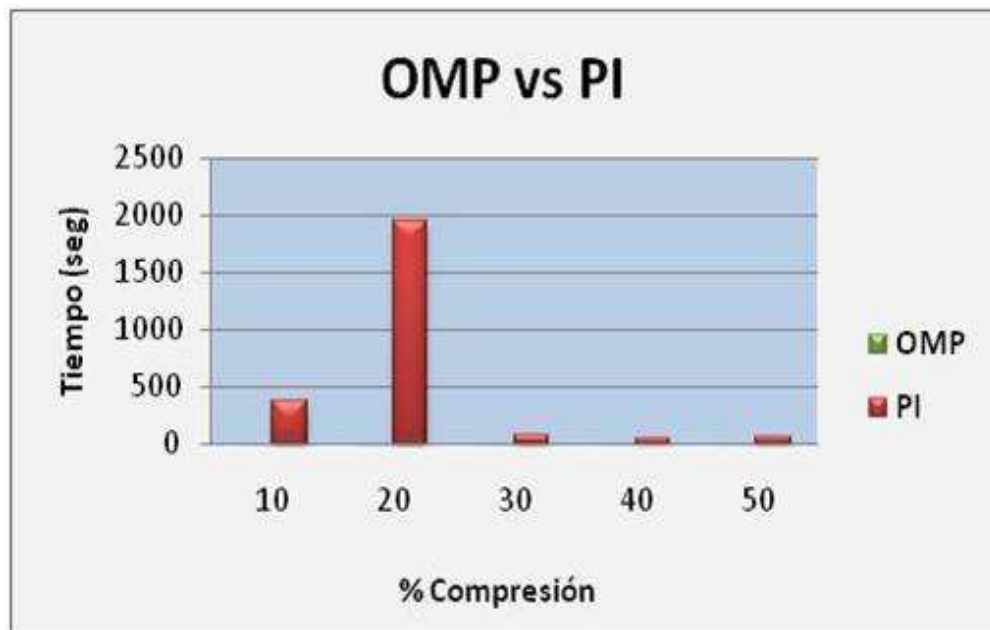
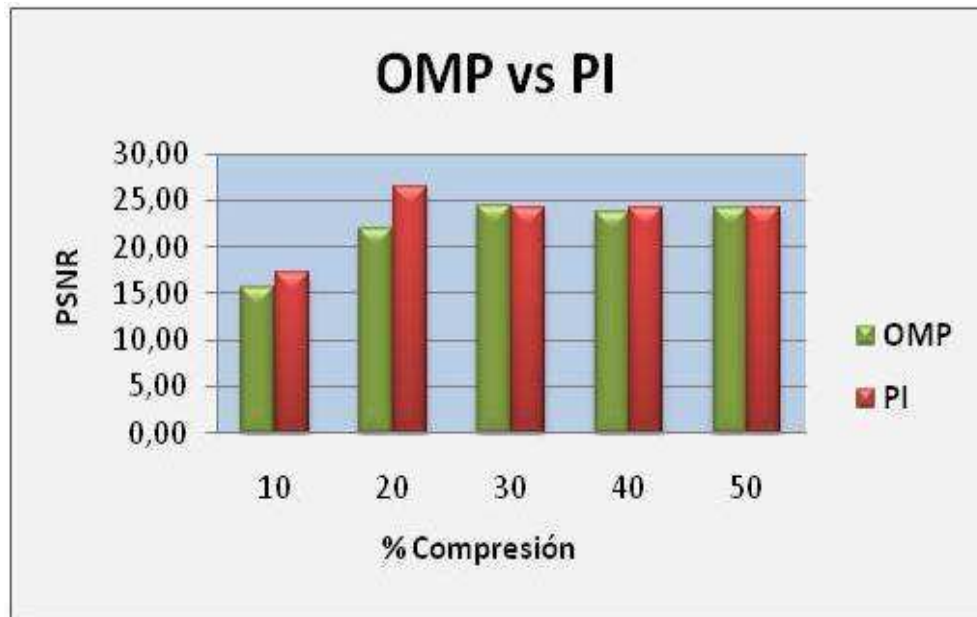




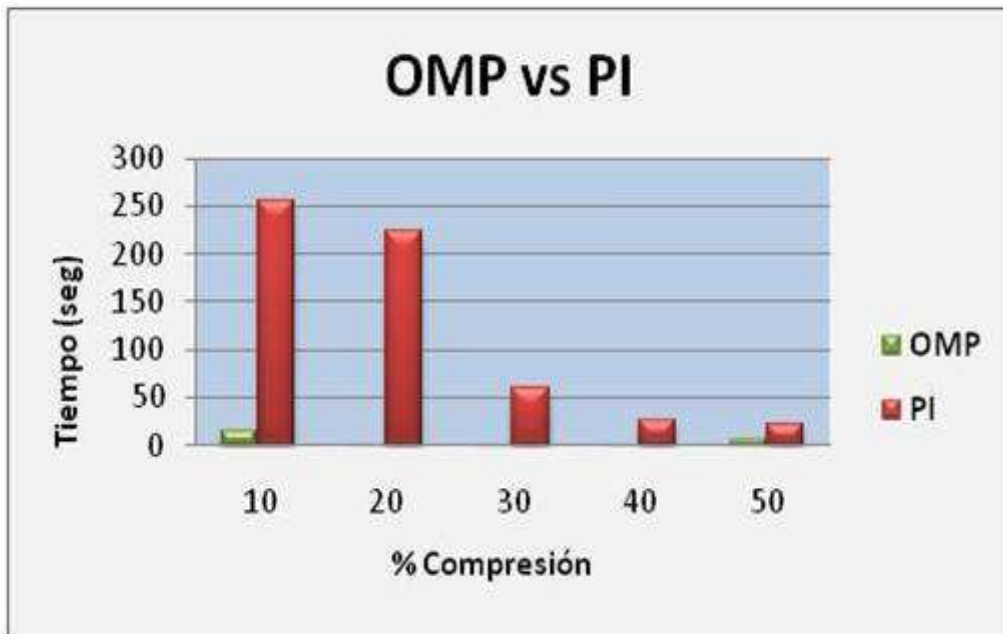
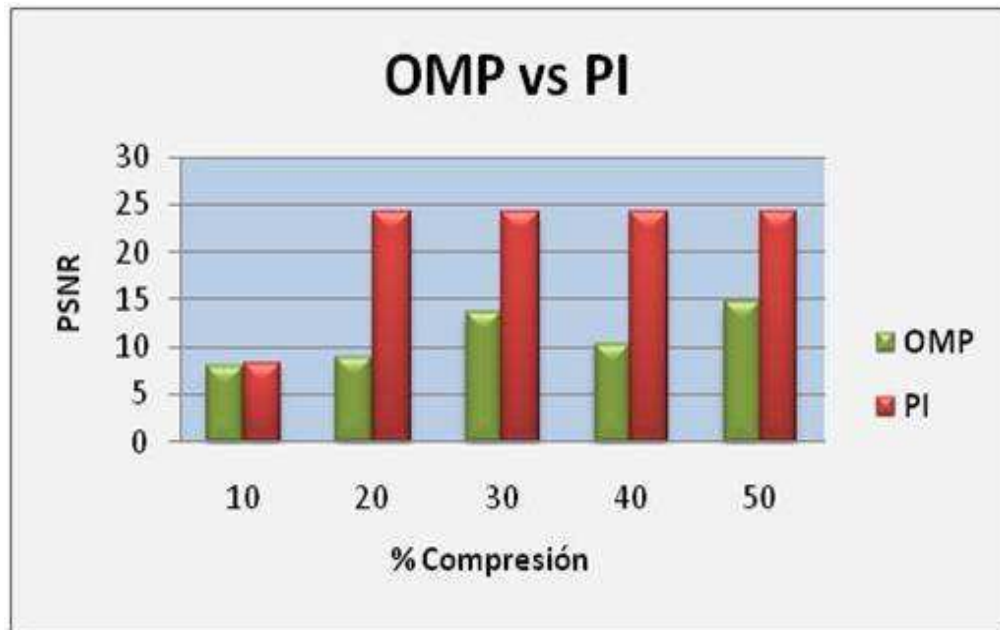
Con estos resultados se puede observar que para un mayor nivel de dispersión el algoritmo PI presenta mejores resultados que el algoritmo OMP y este a su vez presenta mejores tiempo de computo, este realizando la comparación para los diferentes niveles de reconstrucción.

Análisis Comparativo para señales Bidimensionales

- Resultados utilizando la base de muestro Gaussiana



- Resultados utilizando la base de muestreo Hadamard



- **ANÁLISIS COMPARATIVO CON ESTUDIOS PREVIOS**

Los resultados obtenidos en la simulación de la figura 10, tienen un comportamiento similar a los observados en el estudio realizado por Joel A. Tropp y Anna C. Gilbertx("Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit", 2007), donde se trabajó en la reconstrucción de una señal dispersa de longitud=256, y se obtuvo que los mejores porcentajes de recuperación se logran cuando $k \ll m$. Además se demuestra en ambos trabajos que cuando el número de elementos diferentes de cero de la señal aumenta, son necesarias más mediciones para la reconstrucción. En la siguiente tabla se muestran algunos de los resultados encontrados haciendo variaciones en el nivel de dispersión, empleando la base de muestreo Gaussiana y tomando un número de mediciones de 32.

Tabla 33. Análisis comparativo

Nivel de Dispersión	Porcentaje de Recuperación Estudio Publicado	Porcentaje de Recuperación Proyecto
4	80	74,1461
12	0	13,3027
20	0	0,816
28	0	0.0637
36	0	0,0166

Tropp y Gilbert(2007) afirman que Basis Pursuit reconstruye señales dispersas generando mejores resultados en el porcentaje de recuperación que el algoritmo Orthogonal Matching Pursuit, pero que este último es más rápido. En este proyecto se comprueba la validez de esta afirmación. (Ver tabla 19)

Tabla 34. Resultados obtenidos empleando la base de muestreo Gaussiana y porcentaje de compresión de 4%.

m	32	64	96	128	160	192	224	256
ORTHOGONAL MATCHING PURSUIT								
PromPR	74,1461	79,4491	81,509	82,2999	82,6774	81,8498	83,3806	83,4509
Tiempo Seg.	0,00036	0,00041	0,00045	0,00049	0,00056	0,00062	0,00067	0,00073
BASIS PURSUIT								
PromPR	99,7818	99,9999	99,9999	99,9999	99,9999	100	99,9999	99,7818
Tiempo Seg.	0,0365	0,0413	0,0534	0,0659	0,0797	0,0968	0,1229	0,0365

En Lobato Polo A., Ruiz Coral R, Quiróga Sepúlveda J. y Recio Vélez A. (Sparse signal recovery using orthogonal matching pursuit (OMP), 2009) se concluye que para la recuperación de una señal unidimensional empleando el algoritmo OMP la mejor base de muestreo es la Hadamard aleatoria, lo cual concuerda con nuestros resultados.(Ver tabla 20).

Tabla 35. Resultados obtenidos empleando el algoritmo Orthogonal Matching Pursuit buscando el número de mediciones necesarias para alcanzar un porcentaje de recuperación del 75% de una señal dispersa $s \in R^{256}$.

	32	64	96	128	160	192	224	256
Porcentaje de recuperación para k = 4								
Hadamard	75,5521	85,7666	82,4739	83,6398	84,5774	87,1754	87,1075	99,9642
Gaussiana	74,1461	79,4491	81,509	82,2999	82,6774	81,8498	83,3806	83,4509
Porcentaje de recuperación para k = 12								
Hadamard	16,4437	68,4219	69,585	75,3807	74,3922	77,073	77,2056	97,7114
Gaussiana	13,3027	64,1132	68,3608	70,8865	71,1105	71,4897	72,4145	72,772
Porcentaje de recuperación para k = 20								
Hadamard	1,0377	46,9742	65,8568	69,395	71,0147	72,6496	73,0401	91,7481
Gaussiana	0,816	37,1159	62,9818	65,8562	67,8297	68,4969	69,2007	70,3566
Porcentaje de recuperación para k = 28								
Hadamard	0,111	17,9098	61,5939	68,4431	68,2074	71,1794	72,7024	81,6106
Gaussiana	0,0637	10,6572	52,0135	62,9948	65,4745	66,6715	66,6655	67,7624
Porcentaje de recuperación para k = 36								
Hadamard	0,0103	3,9953	49,0286	66,6486	67,6159	70,8872	72,1073	69,9802
Gaussiana	0,0166	2,9263	31,4061	57,2923	62,767	64,8471	65,8702	66,7325

CONCLUSIONES

De los resultados experimentales realizados se pueden obtener las siguientes conclusiones:

Para garantizar la reconstrucción de una señal k-dispersa, es necesario tener en cuenta la relación existente entre el nivel de dispersión y el porcentaje de compresión, ya que de las pruebas realizadas se pudo observar que para lograr una buena reconstrucción de una imagen se requiere que el número de muestras que debe tener la base de muestreo debe ser superior al nivel de dispersión, para los diferentes resultados obtenidos entre más próximo era el número de mediciones al nivel de dispersión no se obtenían resultados favorables.

En diferentes situaciones el algoritmo de Punto Interior de la teoría de Basis Pursuit(BP) presenta mejor rendimiento que el algoritmo Greedy Orthogonal Matching Pursuit(OMP) ya que en los resultados obtenidos en las simulaciones para un porcentaje de compresión entre el 20% y el 50%, Punto Interior con las bases de muestreo Gaussiana y Scramble Block Hadamard Ensemble, puede recuperar una imagen dispersa en el dominio de Wavelet con un PSNR entre 21db y 26,53db logrando obtener coeficientes más precisos, mientras que OMP tiene mayor probabilidad de fallo en el proceso de selección de dichos coeficientes, ya que se obtuvo valores de PSNR entre 2,21db y 19,58db para la matriz de Hadamard aleatoria a pesar de que el porcentaje de columnas encontradas por el algoritmo está entre el 94% y el 99% impidiendo recuperar algunos coeficientes necesarios para la reconstrucción. Esto quiere decir que OMP es muy rápido recuperando las posiciones de las componentes de la base de representación de una señal pero requiere refinamientos para encontrar el valor.

La base de muestreo randómica es la mejor opción para reconstruir imágenes dispersas en el dominio de Wavelet, ya que dicha matriz garantiza gran incoherencia con la base de representación. Este proyecto se obtuvieron valores hasta de 26,24 db en la reconstrucción de imágenes de 64x64 pixeles.

El algoritmo Orthogonal Matching Pursuit con la base de muestreo de Hadamard muestra imágenes aceptables, sin embargo, los valores del PSNR son de 6,77db y 2,21db, debido a que el porcentaje de columnas encontradas no fue del 100%. La imagen recuperada es buena ya que en la cantidad de columnas acertadas se encuentra la información relevante de la imagen original.

Los tamaños de las imágenes empleadas para la reconstrucción influyen directamente en el proceso de reconstrucción de los algoritmos seleccionados, debido a las capacidades de memoria en el equipo.

RECOMENDACIONES

Para futuros trabajos en el área de compresión de imágenes utilizando la teoría de Compressive Sensing se plantean las siguientes recomendaciones:

- Realizar un estudio de la aplicación de esta teoría y de posibles algoritmos que permitan la reconstrucción de imágenes a color.
- Debido a problemas de memoria al realizar la práctica en imágenes de tamaños superiores a 128x128, se recomienda aplicar técnicas para optimizar los algoritmos, o implementarlos en dispositivos de mayor procesamiento. Una alternativa es dividir la tarea de recuperación en pequeñas tareas que puedan ser realizadas por múltiples procesadores.
- Realizar la implementación de algoritmos existentes que muestran mejores resultados de reconstrucción de acuerdo al estado del arte presentado al inicio, ya que pueden contar con la teoría base que se encuentra en este libro para su implementación y las características de los algoritmos implementados.
- Realizar estudios sobre algoritmos de reconstrucción que no requieran que la imagen sea exactamente dispersa en un dominio específico (wavelet, Fourier, coseno) que permitan obtener la imagen sin pérdida de información.
- Hacer un estudio de las bases de muestreo que pueden ser utilizadas en el área de reconstrucción de señales en base de la teoría de muestreo Compressive Sensing.

REFERENCIAS BIBLIOGRAFICAS

- [1] Mark D. Plumbley, Marco Bevilacqua, "Sparse reconstruction for compressed sensing using stagewise polytope faces pursuit". IEEE,2009.
- [2] Mihailo Stojnic, Weiyu Xu, Babak Hassibi. "Compressed sensing of approximately sparse signals". IEEE,2008.
- [3] Mihailo Stojnic, Weiyu Xu, and Babak Hassibi. "Compressed Sensing – probabilistic analysis of a null-space characterization". California Institute of Technology, Pasadena, CA. IEEE 2008
- [4] Stéphane G. Mallat, Zhifeng Zhang. "Matching Pursuits With Time-Frequency Dictionaries". IEEE Transactions on Signal Processing Vol 41, no 12, December 1993.
- [5] Mohammad Golbabaee, Pierre Vandergheynst. "Average case analysis of sparse recovery with thresholding: new bounds based on average dictionary coherence". IEEE, 2008.
- [6] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit". Siam Review Vol. 43, no. 1, pp. 129–159, 2001.
- [7] Emmanuel J. Candes, Terence Tao. "Decoding by Linear Programming". IEEE Transactions on Information Theory, Vol. 51, no. 12, December 2005.
- [8] Weiyu Xu, Babak Hassibi. "Further Results on Performance Analysis for Compressive Sensing Using Expander Graphs". IEEE,2007
- [9] Shaobing Scott Chen, David Donoho." Basis Pursuit". IEEE,1995.
- [10] Joel A. Tropp. "Greed is Good: Algorithmic Results for Sparse Approximation". IEEE Transactions on Information Theory, Vol. 50, no. 10, October 2004
- [11] Stéphane G. Mallat, Zhifeng Zhang. "Matching Pursuits With Time-Frequency Dictionaries". IEEE Transactions on Signal Processing Vol 41,no 12, December 1993
- [12] Thomas Blumensath, Mike E. Davies. "Gradient Pursuits". IEEE Transactions on Signal Processing, Vol. 56, no. 6, June 2008

[13] Adriana Patricia Lobato Polo, Rafael Humberto Ruiz Coral, Julián Armando Quiróg

Sepúlveda y Adolfo León Recio Vélez. "Sparse signal recovery using orthogonal matching pursuit (OMP)". Revista Ingeniería e Investigación Vol. 29 no. 2, Agosto de 2009

[14] Joel A. Tropp." Just Relax: Convex Programming Methods for Identifying Sparse Signals in Noise". IEEE Transactions on Information Theory, Vol. 52, no. 3, March 2006

[15] Chinh La, Minh N. Do. "Tree-based Orthogonal Matching Pursuit Algorithm for signal reconstruction". IEEE,2006

[16] Thomas Blumensath and Mike E. Davies . "Stagewise Weak Gradient Pursuits". IEEE Transactions on Signal Processing, Vol. 57, no. 11, November 2009

[17] Deanna Needell and Roman Vershynin. "Signal Recovery From Incomplete and Inaccurate Measurements Via Regularized Orthogonal Matching Pursuit". IEEE Journal of Selected Topics in Signal Processing, vol. 4, no. 2, April 2010.

[18] Deanna Needell , Roman Vershynin." Uniform Uncertainty Principle and signal recovery via Regularized Orthogonal Matching Pursuit". The Journal of the Foundations of Computational Mathematics, 2009

BIBLIOGRAFIA

- ABOLGHASEMI Vahid, JARCHI Delaram, and SANEI Saeid, A Robust Approach for Optimization of The Measurement Matrix in Compressed Sensing. Institute of Electrical and Electronics Engineers (IEEE). 2010.
- BARANIUK Richard G., Compressive Sensing. Institute of Electrical and Electronics Engineers (IEEE). Signal Processing magazine, 2007.
- BLUMENSATH Thomas and DAVIES Mike E. Stagewise Weak Gradient Pursuits. Institute of Electrical and Electronics Engineers (IEEE) Transactions on Signal Processing, Vol. 57, no. 11, November 2009
- BLUMENSATH Thomas, DAVIES Mike E. Gradient Pursuits. Institute of Electrical and Electronics Engineers (IEEE) Transactions on Signal Processing, Vol. 56, no. 6, June 2008.
- BOUFOUNOS Petros T., BARANIUK Richard G. 1-Bit Compressive Sensing. Institute of Electrical and Electronics Engineers (IEEE), 2008.
- CANDÈS E. and TAO T., Near-optimal signal recovery from random projections: Universal encoding strategies?. Institute of Electrical and Electronics Engineers (IEEE). Trans. Inf. Theory, vol.52, no. 12, pp. 5406–5425, 2006.
- CANDÈS Emmanuel J., TAO Terence. Decoding by Linear Programming. Institute of Electrical and Electronics Engineers (IEEE) Transactions on Information Theory, Vol. 51, no. 12, December 2005.
- CANDÈS E. and TAO T., Decoding by linear programming. Institute of Electrical and Electronics Engineers (IEEE). Trans. Inf. Theory, vol. 51, no. 12, pp. 4203–4215, 2005.
- CANDÈS Emmanuel J. and WAKIN Michael B. Nonconvex compressive sensing and reconstruction of gradient-sparse images: Random vs. tomographic Fourier sampling. Institute of Electrical and Electronics Engineers (IEEE), March 2008.
- CHARTRAND Rick. Nonconvex compressive sensing and reconstruction of gradient-sparse images: Random vs. tomographic Fourier sampling. Institute of Electrical and Electronics Engineers (IEEE) ,2008.
- CHEN S., DONOHO D., and SAUNDERS M., Atomic decomposition by basis pursuit. Siam Review Vol. 43, no. 1, pp. 129–159, 2001.
- CHEN Shaobing Scott, DONOHO David.” Basis Pursuit”. Institute of Electrical and Electronics Engineers (IEEE), 1995.

- CHEN Xi, YU Zhuizhuan, HOYOS Sebastian, SADLER Brian M., and SILVA-MARTINEZ Jose. A Sub-Nyquist Rate Sampling Receiver Exploiting Compressive Sensing. Institute of Electrical and Electronics Engineers (IEEE), Transactions on circuits and systems: regular papers, 2010.
- DONOHO D., Compressed sensing. Institute of Electrical and Electronics Engineers (IEEE). Trans. Inf. Theory, vol. 52, no. 4, pp. 1289–1306, 20.
- DONOHO D., For most large underdetermined systems of linear equations, the minimal norm solution is also the sparsest solution, Commun. Pure Appl. Math., vol. 59, no. 6, pp. 797–829, 2006.
- DUARTE Marco F., DAVENPORT Mark A., WAKIN Michael B., and BARANIUK Richard G. Sparse Signal Detection from Incoherent Projections. Institute of Electrical and Electronics Engineers (IEEE) .2006.
- GOLBABAEE Mohammad, VANDERGHEYNSt Pierre. Average case analysis of sparse recovery with thresholding: new bounds based on average dictionary coherence. Institute of Electrical and Electronics Engineers (IEEE) Transactions on Signal Processing, 2008.
- LA Chinh, DO Minh N. Tree-based Orthogonal Matching Pursuit Algorithm for signal reconstruction. Institute of Electrical and Electronics Engineers (IEEE), 2006.
- LOBATO POLO Adriana Patricia, RUIZ CORAL Rafael Humberto, Julián Quiroga Sepúlveda Armando y Adolfo León Recio Vélez. Sparse signal recovery using ortogonal matching pursuit (OMP). Revista Ingeniería e Investigación Vol. 29 no. 2, Agosto de 2009.
- MALLAT StCphane G., ZHANG Zhifeng. Matching Pursuits with Time-Frequency Dictionaries. Institute of Electrical and Electronics Engineers (IEEE) Transactions on Signal Processing Vol 41,no 12, December 1993.
- MALLAT StCphane G., ZHANG Zhifeng. Matching Pursuits with Time-Frequency Dictionaries. Institute of Electrical and Electronics Engineers (IEEE) Transactions on Signal Processing Vol 41, no 12, December 1993.
- MESECHER Dave, CARIN Larry, KADAR Ivan, PIRICH Ron, Exploiting Signal Sparseness for Reduced- Rate Sampling, Institute of Electrical and Electronics Engineers (IEEE), 2009.
- NEEDELL Deanna and VERSHYNIN Roman. Signal Recovery From Incomplete and Inaccurate Measurements Via Regularized Orthogonal Matching Pursuit. Institute of Electrical and Electronics Engineers (IEEE) Journal of Selected Topics in Signal Processing, vol. 4, no. 2, April 2010.

- NEEDELL Deanna, VERSHYNIN Roman. Uniform Uncertainty Principle and signal recovery via Regularized Orthogonal Matching Pursuit. The Journal of the Foundations of Computational Mathematics, 2009.
- OPPENHEIM Alan V., Ronald W. Schafer. Tratamiento de señales en tiempo discreto. 2^{da} Edición.
- PLUMBLEY Mark D., BEVILACQUA Marco. Sparse Reconstruction for Compressed Sensing Using Stagewise Polytope Faces Pursuit. Agosto, 2009.
- SEYEDZADE Seyed Mohammad, MIRZAKUCHAKI Sattar, TAHMASBI Amir. Using subclass discriminant analysis, fuzzy integral and symlet decomposition for face recognition. IEEE, 2010.
- STOJNIC Mihailo, XU Weiyu, and HASSIBI Babak. "Compressed Sensing – Probabilistic Analysis of a null-space characterization". California Institute of Technology, Pasadena, CA. Institute of Electrical and Electronics Engineers (IEEE), 2008.
- STOJNIC Mihailo, XU Weiyu, HASSIBI Babak. Compressed Sensing of Approximately Sparse Signals. Institute of Electrical and Electronics Engineers (IEEE), 2008.
- TROPP Joel A., Greed is good: Algorithmic Results for Sparse Approximation. Institute of Electrical and Electronics Engineers (IEEE) Transactions on Information Theory, Vol. 50, no. 10, October 2004.
- TROPP Joel A., Just Relax: Convex Programming Methods for Identifying Sparse Signals in Noise". Institute of Electrical and Electronics Engineers (IEEE) Transactions on Information Theory, Vol. 52, no. 3, March 2006.
- XU Weiyu, HASSIBI Babak. Further Results on Performance Analysis for Compressive Sensing Using Expander Graphs. Institute of Electrical and Electronics Engineers (IEEE) , 2007.
- YANG Yi, AU Oscar C., FANG Lu, WEN Xing and TANG Weiran. Perceptual Compressive Sensing for image signals, Institute of Electrical and Electronics Engineers (IEEE), 2009.