

UNIVERSIDAD DE PAMPLONA



FACULTAD DE INGENIERÍAS Y ARQUITECTURA  
PROGRAMA DE INGENIERÍA EN TELECOMUNICACIONES

RECONSTRUCCIÓN DE IMÁGENES SPARSE DE RADAR EN ALTA RESOLUCIÓN  
USANDO ALGORITMOS SPGL1 y GPSR

Israel David Navarro Rada

TRABAJO DE GRADO PARA OPTAR POR EL TÍTULO DE  
INGENIERO EN TELECOMUNICACIONES

**Director: MSC. Luis Enrique Mendoza**  
Director del programa: **MSC. William Villamizar rozo**

**Pamplona, Colombia**  
**Junio 2016**  
**UNIVERSIDAD DE PAMPLONA**



FACULTAD DE INGENIERÍAS Y ARQUITECTURA  
PROGRAMA DE INGENIERÍA EN TELECOMUNICACIONES

Reconstrucción de imágenes SPARSE de radar en alta resolución usando algoritmos  
SPGL1 y GPSR  
Trabajo de Grado para Optar por el Título de  
Ingeniero en Telecomunicaciones

**FECHA DE INICIO DEL TRABAJO:** febrero 2012  
**FECHA DE TERMINACIÓN DEL TRABAJO:** junio 2016

NOMBRES Y FIRMAS DE AUTORIZACIÓN PARA LA SUSTENTACIÓN:

\_\_\_\_\_  
ISRAEL DAVID NAVARRO RADA  
**AUTOR**

\_\_\_\_\_  
MSC. LUIS ENRIQUE MENDOZA  
**DIRECTOR**

\_\_\_\_\_  
MSC. WILLIAM VILLAMIZAR ROZO  
**DIRECTOR DEL PROGRAMA**

**JURADO CALIFICADOR:**

\_\_\_\_\_  
MSC. HERNANDO VELANDIA

\_\_\_\_\_  
ESP. LUIS ALBERTO MUÑOZ

\_\_\_\_\_  
MSC .NYDIA SUSANA SANDOVAL

**Pamplona, Colombia**  
**Junio 2016**

## **DEDICATORIA**

A Dios por ser mi principal sustento, fuerza y compañía por darme salud para lograr mis objetivos, además de su infinita bondad y amor.

A mis padres, esposa, hija y hermas por su confianza, por sus consejos, por la motivación constante, amor y apoyo incondicional durante mi formación profesional.

A mis familiares, y amigos por la confianza, y su apoyo que me dieron fuerzas para luchar por mis metas.

## **PENSAMIENTO**

Aprendí que no se puede dar marcha atrás, que la esencia de la vida es ir hacia adelante. En realidad, la vida es una calle de sentido único.

Agatha Christie

## **AGRADECIMIENTOS**

Primeramente a Dios Todopoderoso por darme vida, salud y fortaleza, por permitirme alcanzar siempre mis metas y culminar esta etapa de mi vida para comenzar una nueva.

A mi Mamita hermosa no me equivoco si digo que eres mi mayor bendición, gracias por todo tu esfuerzo, paciencia, tu apoyo y por la confianza que depositaste en mí.

A mi papa por enseñarme que en la vida lo que se consigue con sacrificio es lo que más se valora, por creer en mí pese a las adversidades, te quiero mucho, eres una de las vigas que fortalece la pirámide de mi vida.

A mi esposa e hija gracias por ser un soporte fundamental en mi vida, por tu apoyo incondicional, paciencia, comprensión y sobre todo por tu gran amor, Sobra decir que eres mí complemento.

A mis hermanas Leidy, Yaniluz, Katherine, Mairena y Lorena les doy gracias a DIOS porque de alguna manera han estado allí para apoyarme.

Al profesor Luis Enrique Mendoza por su dedicación en el desarrollo de este trabajo, sus enseñanzas, y su gran apoyo.

A cada uno de mis profesores que contribuyeron con mi formación profesional.

A todos mis amigos y compañeros de universidad sin excluir a ninguno.

Desde los más profundo de mi corazón GRACIAS.

Israel David Navarro Rada

## RESUMEN

En este trabajo de grado se implementaran nuevas técnicas de reconstrucción de señales como lo son SPGL1 y GPSR, a nivel de software, en imágenes de radar de alta resolución, estas imágenes están en un dominio *sparse* (poco densas), y de no ser así, se utilizará la técnica de la transformada discreta del coseno, para convertir estas imágenes de no *sparse* a *sparse*, esto es para tener una reconstrucción exitosa. Seguidamente se muestra que efectivamente la teoría Compressive Sensing (CS) puede ser aplicada para la reconstrucción de imágenes de radar de alta resolución, pues así lo indica la teoría.

La imagen de radar en estado *sparse*, será proyectada sobre una matriz aleatoria, que permitirá comprimir la imagen a 10%,25%,50% y 75% todo este proceso se permitió ver la reconstrucción de la imagen en cada una de estas compresiones y además se analiza con un estudio comparativo el rendimiento de cada uno de los métodos de reconstrucción, usando diferentes criterios como: perdida de información y costo computacional con la finalidad de dar a conocer cuál de estos métodos es el más eficiente en cada tipo de aplicaciones.

Este trabajo presenta conjuntamente la simulación de estas técnicas de reconstrucción en MATLAB a nivel de código, y con la herramienta conocida como guide se verá una interfaz gráfica de usuario, adicionalmente se tiene un manual para el usuario el cual constara de los pasos a seguir para experimentar la reconstrucción de una imagen.

Los métodos fueron efectivos al reconstruir imágenes de radar en alta definición, se presentó perdida de información o error de reconstrucción diferentes para cada método, SPGL1 obtuvo errores para la imagen 1 a distintos grado de compresión de (10%=0.2311, 25%=0.8245, 50%=2.7630, 75%=7.3135), para GPSR se obtuvo (10%=5.2220, 25%=6.8218, 50%=7.3108, 75%=9.7869) el solucionador SPGL1 demostró ser mejor en cuanto a la calidad de reconstrucción.

**Palabras claves:** Compressive Sensing (CS), GPSR, SPGL1, compresión, imágenes de radar.

## ABSTRACT

In this paper grade new reconstruction techniques signals as are SPGL1 and GPSR, software level in radar images of high resolution, these images are in a (sparse) sparse domain is implemented, and otherwise the technique of the discrete cosine transform is used to convert these images not sparse to sparse, this is to have a successful reconstruction. Then it is shown that the theory actually Compressive Sensing (CS) can be applied to image reconstruction of high-resolution radar, as well suggests the theory.

The radar image in sparse state, will be screened on a random matrix, which will compress the image to 10%, 25%, 50% and 75% this process will allow us to see the reconstruction of the image in each of these compressions and also it analyzed with a comparative study the performance of each reconstruction methods, using different criteria such as loss of data and computational cost in order to make known which of these methods is the most efficient for each type of application.

This work jointly presents the simulation of these reconstruction techniques in MATLAB-level code, known as guide tool will be a graphical user interface, additionally we will have a user manual which will consist of the steps to experience reconstruction of an image.

The methods were effective in reconstructing radar images in high definition, was presented loss of data or failure of different reconstruction for each method, SPGL1 obtained errors for image 1 to different degree of compression (10% = 0.2311, 25% = 0.8245 50% = 2.7630, 75% = 7.3135), for GPSR was obtained (10% = 5.2220, 25% = 6.8218, 50% = 7.3108, 75% = 9.7869) solver SPGL1 proved best in terms of quality of reconstruction.

Keywords: Compressive Sensing (CS), GPSR, SPGL1, compression, radar images.

# Contenido

Capítulo 1 .....	1
Fundamento Del Trabajo De Grado.....	1
1.1 Justificación. ....	2
1.2 Motivación. ....	2
1.3 problema.....	3
1.4 Objetivos.....	4
1.4.1 Objetivo General. ....	4
1.4.2 Objetivos Específicos.....	4
1.5 Acotaciones. ....	5
1.6 Antecedentes.....	5
1.6.1 Antecedentes internacionales compressed sensing. ....	5
1.6.2 Antecedentes nacionales compressed sensing.....	6
Capítulo 2 .....	8
Marco teórico.....	8
2.1 ESTADO DEL ARTE .....	9
2.1.1 Compressive Sensing. ....	9
2.1.1.1 Señales Compresibles. ....	9
2.1.1.2 Propiedad De Escasez (sparsity). ....	11
2.1.1.3. Muestreo Incoherente. ....	12
2.1.1.4 Diccionarios. ....	13
2.1.1.4.1 Diccionario En El Dominio Del Tiempo. ....	13
2.1.1.4.2 Diccionario En Bases Cosenos. ....	14
2.1.1.4.3 Diccionario Holográfico.....	15
2.2. Reconstrucción De Una Señal Utilizando La Teoría CS. ....	15
2.2.1 Submuestreo y Recuperación De La Señal Sparse.....	16
2.2.2 Teorema De Muestreo No Lineal.....	17
2.2.2.1 Teorema 1.....	18
2.3. Diseño De La Matriz De Medición $\Phi$ . ....	18
2.3.1 Propiedad De Isometría Restringida.....	19
2.3.1.1 Teorema 2.....	19
2.4. GPSR (Proyección Del Gradiente Para La Reconstrucción Sparse). ....	20
2.4.1 Dimensiones Del Programa Cuadrático Con Restricciones (BCQP).....	21
2.4.2 Técnica del Algoritmo De Proyección Del Gradiente. ....	22
2.4.3 Proyección Del Gradiente Básico: GPSR-Basic Algoritmo. ....	22
2.4.4 Barzilai-Borwein Proyección Del Gradiente: Algoritmo GPSR-BB .....	24
2.4.4. 1 Convergencia.....	25
2.4.4.2 Terminación. ....	25
2.4.4.3 Parámetros Tao (T) De Regularización: Arranque En Caliente.....	27
2.4.4.4. Comparación entre GPSR y la norma $l_1$ .....	28
2.5. SPGL1 (gradiente de proyección espectral para $L_1$ ).....	30
2.5.1. Basis Pursuit Denoise. ....	30
2.5.2 regularización de la norma $l_1$ . ....	31
2.5.3 Enfoque. ....	32
2.5.3.1 Curva de Pareto. ....	32



2.5.3.2 Subproblema doble. ....	33
2.5.3.3 Búsqueda de raíces. ....	35
2.5.3.4 Algoritmo 1: Gradiente Espectral proyectado para (LS $\tau$ ). ....	36
2.5.3.5 Gradiente proyectado para Lasso. ....	37
2.5.3.6 Proyección de la norma uno. ....	37
2.4.3.7 Programas para Lasso y cuadráticos. ....	40
2.6. Teledetección Radar. ....	42
2.6.1. Fundamento de la obtención de la imagen radar. ....	42
2.6.2. Resolución de las imágenes de radar. ....	43
2.6.2.1. Resolución espacial. ....	43
2.6.2.1.1. Aplicación de las imágenes de alta resolución según su resolución espacial. ....	43
2.6.2.1.2. Resolución de 1 a 5 metros ....	43
2.6.2.1.3 Resolución de 20-60 metros ....	43
2.6.2.1.4. Resolución de 100-200 metros ....	44
2.6.2.1.5. Resolución de 1000-2500 metros ....	44
2.7 La interfaz de usuario ....	44
2.7.1 Tipos y criterios de diseño. ....	44
2.7.2 Tipos de interfaz ....	44
2.7.4 Objetivos en el diseño de IU. ....	45
2.7.5 Principios de diseño de interfaces de usuario. ....	46
Capítulo 3. ....	48
Metodología. ....	48
3.1 Proceso de selección de las imágenes de radar en alta resolución. ....	49
3.2. Proceso de compresión de la señales de radar. ....	52
3.2.1 Conversión de la señal o imagen de radar a Sparse. ....	52
3.2.2 Nivel De Esparcidad. ....	54
3.2.3. Creación de la matriz aleatoria de medición. ....	56
3.2.4. Calculo de la señal o vector comprimido de la imagen de radar en (CS). ....	58
3.3. Reconstrucción de la imagen sparse de radar usando el algoritmo SPGL1. ....	58
3.4. Reconstrucción de la imagen sparse de radar usando el algoritmo GPSR. ....	59
3.5. Conversión de señal Sparse a dominio del tiempo. ....	59
3.6 Porcentaje de error generado por los algoritmos de reconstrucción SPGL1 Y GPSR. ....	59
Capítulo 4. ....	61
<b>RESULTADOS</b> ....	61
4.1 Proceso de selección de las imágenes de radar en alta resolución. ....	62
4.2 FIGURA 12 ....	62
4.2 .1 Calculo de la señal comprimida de la figura 12 de radar en (CS). ....	62
4.3 Método de reconstrucción SPGL1. ....	63
4.3.1 Método de reconstrucción SPGL1 para la figura12 de radar en alta resolución. ....	63
4.3.1.1 Conversión de la señal Sparse de la figura 12 al dominio del tiempo utilizando IDCT. ....	63
4.3.1.2 Comparación de la imagen reconstruida con respecto a la figura 12. ....	64
4.3.2 FIGURA 13 ....	65

4.3.2 .1 Método de reconstrucción SPGL1 para la figura 13 utilizando la señal CS. ...	65
4.3.3 FIGURA 14 .....	66
4.3.3.1 Método de reconstrucción SPGL1 para la figura 14 utilizando la señal CS 50%. .....	66
4.3.4 FIGURA 15 .....	67
4.3.4.1 Método de reconstrucción SPGL1 para la figura 15 utilizando la señal CS 50%. .....	67
4.3.5 FIGURA 16 .....	69
4.3.5.1 Método de reconstrucción SPGL1 para la figura 16 utilizando la señal CS 50%. .....	69
4.4 Método de reconstrucción GPSR.....	70
4.4.1 Método GPSR de reconstrucción para las imágenes de radar en alta resolución.	70
4.4.2 FIGURA 12 .....	70
4.4.2.1 Método GPSR de reconstrucción para la figura 12 utilizando las señales CS.	70
4.4.2.2 Comparación de las imágenes reconstruidas con respecto a la figura 12. ....	71
4.4.3 FIGURA 13 .....	72
4.4.3.1 Método GPSR de reconstrucción para la figura 13 utilizando las señales CS.	72
4.4.4 FIGURA 14 .....	73
4.4.4.1 Método GPSR de reconstrucción para la figura 14 utilizando las señales CS de 50%. ....	73
4.4.5 FIGURA 15 .....	74
4.4.5.1 Método GPSR de reconstrucción para la figura 15 utilizando las señales CS de 50%. ....	74
4.4.6 figura 16.....	76
4.4.6.1 Método GPSR de reconstrucción para la figura 16 utilizando las señales CS de 50%. ....	76
4.5 Error generado por los algoritmos de reconstrucción SPGL1 Y GPSR. ....	77
4.6 tiempo de reconstrucción empleados de los métodos SPGL1 y GPSR.....	78
4.7 Interfaz de usuario .....	79
4.7.1 Ventana principal .....	79
4.7.2 Ventana de funcionalidades .....	80
4.7.2.1 Imagen original.....	80
4.7.2.2 Imagen sparse .....	81
4.7.2.3 Matriz comprimida .....	82
4.7.2.4 Imagen reconstruida.....	84
CONCLUSIONES .....	85
BIBLIOGRAFÍA.....	86

## INTRODUCCIÓN

Encontrar una solución a señales poco densas o que tienen pocos puntos diferentes de cero (sparse) en los sistemas lineales indeterminado (son aquellos que tienen un número infinito de soluciones que satisfacen a una o varias ecuaciones) es una de las cuestiones fundamentales en el procesamiento de señales y en otras aplicaciones. En este estudio se presentará a dos nuevos algoritmos iterativos de reconstrucción, que basan su funcionamiento en la proyección del gradiente espectral (GPS), denominado **SPGL1 y GPSR**, para recuperar una señal dispersa a gran escala. Estos algoritmos seleccionan los parámetros de búsqueda de diferentes maneras, incluyendo técnicas basadas en el método de Barzilai-Borwein. En teoría estos enfoques de GPS funcionan muy bien en una amplia gama de aplicaciones, siendo significativamente más rápidos (en términos de tiempo de cálculo). En combinación con **Compressive Sensing (CS)** que es una teoría o paradigma que va en contra de la sabiduría común en la adquisición de datos. CS afirma que uno puede recuperar ciertas señales e imágenes de un número mucho menor de muestras o mediciones que los métodos tradicionales utilizan. CS intenta comprimir una señal y muestrearla a la vez y como consecuencia de esto la reducción de las capacidades de almacenamiento así como también se minimiza el uso de los recursos en el proceso de conversión digital – analógico en cuanto a velocidad de muestreo.

# Capítulo 1

## Fundamento Del Trabajo De Grado

En este capítulo se describe los fundamentos o principios iniciales que motivaron el proyecto, se compone de: justificación, motivación, problema, objetivos, acotaciones y antecedentes.

## **1.1 Justificación.**

Las técnicas **SGPL1 y GPSR** son un enfoque novedoso para la reconstrucción de señales con escasez, y complementadas con la teoría Compressive Sensing (CS), permite una reducción potencialmente grande en los cálculos para la adquisición de señales que tienen una escasa o representación compresible en cierta base, lo cual es una gran justificación para el estudio de estas técnicas que se expanden en varias áreas del conocimiento. En este proyecto se utilizarán los algoritmos antes mencionados, para la reconstrucción de imágenes de radar las cuales tienen un gran tamaño, el estudio de estas imágenes será en el dominio sparse, con los datos obtenidos durante el proceso de estudio en este proyecto se pretenderá dar una solución a la reconstrucción de señales, a un menor costo en este campo de tratamiento de señales.

Así mismo se aprovecharán todas las herramientas físicas y bibliográficas necesarias para concluir el mejor proceso y estudio de las diferentes técnicas que se utilizarán.

## **1.2 Motivación.**

Mi gran motivación es trabajar con GIBUP (Grupo de Investigación de Biomédica de la Universidad de Pamplona), por el gran conocimiento y la experiencia en proyectos de investigación en la rama de procesamiento de señales, pues todo esto se convierte en un atractivo muy fuerte para hacer parte de este grupo, y entorno acogedor.

Me motivó hacer un trabajo de grado enfocado a la reconstrucción de imágenes de radar en alta resolución, para fortalecer mis conocimientos en el tratamiento de señales de la misma forma en las herramientas que se necesiten para este fin.

Por otro lado, es importante mencionar que es importante demostrar que se pueden reconstruir señales usando compressive sensing, además de lograr mejorar el tiempo de reconstrucción de imágenes de alta resolución.

### **1.3 problema.**

Debido a las grandes limitaciones que presentan las antiguas técnicas para la adquisición de las señales, donde el costo y el tiempo para la reconstrucción de señales es muy alto, en este caso específico una imagen de radar en alta resolución, este problema se da por la gran cantidad de muestras necesarias para su solución, aportando así problemas como el espacio de memoria ocupado, mayor ancho de banda para la transmisión de los datos, mayor tiempo de ejecución para la reconstrucción.

Podemos citar un ejemplo claro para la reconstrucción de señales es el teorema Nyquist, donde la frecuencia es muy alta por lo cual tendremos que tomar demasiadas muestras de la señal lo que conlleva a la necesidad de comprimir la información antes de almacenarla o transmitirla. Debido a esto surge la necesidad de estudiar nuevas herramientas de muestreo, comprensión y reconstrucción de señales.

## **1.4 Objetivos.**

### **1.4.1 Objetivo General.**

Reconstruir imágenes sparse de radar en alta resolución usando algoritmos SPGL1 y GPSR.

### **1.4.2 Objetivos Específicos.**

- Hacer un estado del arte de las técnicas SPGL1 y GPSR para la reconstrucción de imágenes.
- Establecer las condiciones mínimas del tipo de señales a reconstruir.
- Implementar una interfaz gráfica de usuario del sistema, con su respectivo manual.
- Desarrollar un software para la reconstrucción de imágenes en el dominio sparse, usando las librerías GPSR y SPGL1.
- Evaluar el funcionamiento de la simulación de las técnicas SPGL1 y GPSR.

## **1.5 Acotaciones.**

- El tiempo de ejecución dependerá de las características de la computadora.
- Se implementará el algoritmo de reconstrucción a las imágenes de radar que cumplan con el criterio sparse.

## **1.6 Antecedentes.**

Por ser técnica nueva en el campo del tratamiento de señales, a nivel regional no existen trabajos donde empleen las técnicas GPSR y SPGL1, pero a nivel internacional, existen universidades con un gran conocimiento, y la experiencia en estas técnicas de reconstrucción, además existen varios algoritmos que se basan en las técnicas del gradiente espectral proyectado (GPS), se debe dar a conocer que en la actualidad esta técnica se encuentra en desarrollo para mejorar su eficiencia, en cuanto al tiempo y exactitud en las reconstrucciones es así que en abril de 2015 se actualizó el software SPGL1 de la versión 1.7 a la versión 1.9, y se puede afirmar que todavía hace falta mucho por hacer en este campo.

### **1.6.1 Antecedentes internacionales compressed sensing.**

En cuanto Compressed Sensing hay trabajos relacionados con la reconstrucción de imágenes.

#### **Investigación acerca de Reconstrucción de imágenes geológicas basadas en la teoría de compressed sensing.**

En la universidad de Chile Hernán Alberto Calderón (2014) habla de estudiar, desarrollar e implementar la técnica compressed sensing para la resolución de problemas inversos en geoestadística, aplicado a la caracterización de perfiles de permeabilidad binario presente en el subsuelo desde un enfoque de reconstrucción de imágenes, consistiendo en modelar la estructura multicanal con imágenes que presentan una descomposición sparse en algún dominio y utilizar esta información para reconstruir la imagen original a partir de un muestreo crítico y no estructurado. Se ha justificado el uso de la DCT como base óptima que permite la cantidad de muestras para hacer un buen uso de compressed sensing y así lograr la reconstrucción de canales de permeabilidad [26].

#### **Reconstrucción de imágenes a través de compressed sensing.**

En la universidad de Rice (Estados Unidos), Justin Romberg (2008), realizó un estudio para la Reconstrucción de imágenes a través de compressed sensing donde convirtió un gran conjunto de datos digitales en un conjunto sustancialmente muy



pequeño para reconstruir una imagen por medio de **compressed sensing** (CS). La elección de la matriz de medición nos permite recopilar la información necesaria sobre la imagen para posteriormente ser reconstruida a través de mínimos cuadrados, donde se tendrá una búsqueda de la imagen más cercana a la original. En lugar de recoger punto de la imagen en X lugares distintos, o promedios en varias zonas pequeñas (píxeles) la reconstrucción de las imágenes se llevó a cabo con éxitos [27].

### **Investigación acerca reducción de los tiempos de adquisición de imágenes por resonancia magnética utilizando técnicas de compressed sensing.**

En la universidad Pontificia Católica (Chile), Carlos Sing, Cristián Tejos N, Pablo Irrázaval, (2009) realizaron un estudio con la finalidad de reducir los tiempos de adquisición de imágenes por resonancia magnética utilizando técnicas de compressed sensing, El proceso de adquisición de imágenes por resonancia magnética (IRM) es lento. Una forma para disminuir los tiempos de adquisición es a través de reconstrucciones de datos submuestreados, es decir tomar menos muestras que las necesarias en aplicaciones estándares y reconstruir las muestras faltantes a través de algoritmos matemáticos, este trabajo se utiliza la técnica de reconstrucción de datos submuestreados basadas en Compressed Sensing (CS) para disminuir los tiempos de adquisición, obteniendo imágenes idénticas a las obtenidas con todas las muestras. Donde se realizaron reconstrucciones de datos submuestreados de fantasmas e IRM con 60%, 55% y 50% de las muestras. Cuando el número de muestras fue mayor al doble del número de píxeles con intensidad cero, las reconstrucciones obtenidas fueron idénticas a las originales. Para las IRM esto se logró con 60% de las muestras, logrando reducciones del 40% en los tiempos de adquisición [28].

### **1.6.2 Antecedentes nacionales compressed sensing.**

#### **Estudio sobre los Algoritmos de reconstrucción en Compressed Sensing.**

En la universidad de los Andes Jesús Alberto Rodríguez (2011), presento un estudio sobre los Algoritmos de reconstrucción en Compressed Sensing, demostrando que efectivamente la teoría compressed Sensing puede ser aplicada para la reconstrucción de señales e imágenes, Debido a la poca esparcidad de algunas imágenes en el dominio del tiempo, se introdujo un nuevo dominio, conocido como el dominio de las derivadas, en pocas palabras lo que se hace es aprovechar la alta coherencia que existe entre las filas o columnas de ciertas imágenes, para aumentar el nivel de esparcidad de la imagen teniendo esto se implementaron distintos algoritmos de reconstrucción como: Matching Pursuit, Orthogonal Matching Pursuit,

Regularized Orthogonal Matching Pursuit para la su reconstrucción, las cuales fueron reconstruidas utilizando solo el 70% de la data original [5].

# Capítulo 2

## Marco teórico

En este capítulo se describe el fundamento matemáticos y teóricos que corresponde a la técnica de compresión llamada Compressive Sensing, y las técnicas de reconstrucción GPSR y SPGL1.

## 2.1 ESTADO DEL ARTE

### 2.1.1 Compressive Sensing.

Compressive Sensing (CS) ha sido propuesta por Candés y Donoho como una teoría de gran potencial que plantea nuevos paradigmas para el procesamiento de señales, pues reduce considerablemente el número de muestras necesarias para la reconstrucción de una señal, reduciendo considerablemente la frecuencia a la cual ésta se debe muestrear, lo cual rompe el paradigma de la teoría clásica [2].

La teoría del Compressed Sensing establece que una señal poco densa o sparse puede ser reconstruida con alta probabilidad a partir de un conjunto de muestras provenientes de su proyección aleatoria, siempre y cuando la señal cumpla con la condición de esparcidad o poca densidad en algún dominio al cual denominaremos diccionario  $\Psi$  y que éste a su vez sea incoherente con la matriz de medición  $\Phi$  [1].

El número de muestras necesarias para reconstruir una señal habitualmente es mucho menor que la señal original, lo que implica:

- ❖ Menor frecuencia de muestreo
- ❖ Menor cantidad de datos
- ❖ Menor uso de los recursos de almacenaje
- ❖ Menor requerimiento de velocidad de los convertidores analógico-digitales.
- ❖ Menor uso de ancho de banda para la transmisión de los datos, es decir, menor tiempo requerido para la transmisión de los datos.

Las siguientes son las herramientas más utilizadas para la conversión de las señales a Sparse [4].

- Transformada discreta del coseno (DCT)
- Transformada rápida de Fourier (FFT)
- Transformada Wavelet

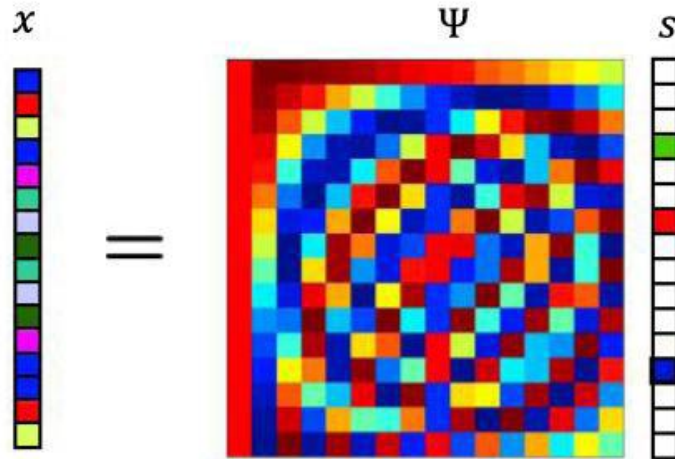
#### 2.1.1.1 Señales Compresibles.

Sea  $x$  una señal real, finita, que representa una señal discreta en el tiempo, Sparse, con valores cero y  $K$ -valores diferentes de cero el cual se pueden expresar como vector columna de longitud  $N$  en  $\mathbb{R}^n$ ,  $x[n]$ , donde  $n=1,2,\dots, N$ . Cualquier señal en  $\mathbb{R}^n$  puede ser representada en términos de una base de  $N \times 1$  vectores  $\{\Psi_i\}_{i=1}^N$ , (suponemos que la base es ortonormal). La señal  $x$  puede reconstruirse a partir de

las combinaciones lineales de la matriz (NxN) y un vector s que contiene los valores significativos o no-cero de x definida matemáticamente como [3]:

$$X = \sum_{i=1}^n s_i \Psi_i = s\Psi \quad (1)$$

La señal x definida en la ecuación (1), es representada en la Figura 1.



**Figura 1.** Representación gráfica de x en términos de  $\Psi$  y s, el uso de colores es homólogo a los valores numéricos.

Donde cada valor de  $S_i$  puede obtenerse a partir de  $\Psi_i^T$  y x con la siguiente expresión:

$$s_i = \langle X, \Psi_i \rangle = \Psi_i^T X \quad (2)$$

Dónde T denota matriz traspuesta de  $\Psi$ . x y s son representaciones equivalentes de la señal, con x en el dominio del tiempo o del espacio y s en el dominio  $\{\Psi_i\}$

La señal x es k-sparse si puede expresarse como combinación lineal de sólo k vectores base, es decir, sólo k de los  $s_i$  coeficientes son distintos de cero y (n-k) coeficientes son cero. El caso de interés es cuando la señal cumple  $k \ll n$ . La señal x es compresible si la representación ecuación (1) tiene un pequeño número de componentes distintos de cero siendo la mayoría de ellos iguales a cero [5].

### 2.1.1.2 Propiedad De Escasez (sparsity).

Se dice que una señal  $x$  es  $k$ -sparse en algún dominio  $\Psi \in \mathbb{R}^m$ , cuando puede ser representada por una combinación lineal de  $k$  átomos que provienen del dominio, dónde los átomos son los vectores columna que conforman  $\Psi$ .

Si se representa la base  $\Psi = [\Psi_1, \Psi_2, \dots, \Psi_m]$ , donde cada componente de  $\Psi$  es un vector columna entonces se podría escribir  $x$  de la siguiente forma:

$$X \approx \sum_{i=1}^k s_i \Psi_i = s\Psi \quad (3)$$

Donde  $s = [s_1, s_2, \dots, s_n]$  es un vector con  $k$  componentes distintas de cero, donde los  $s_i$  representan la contribución del átomo  $\Psi_i$  a la definición de la señal  $x$ .

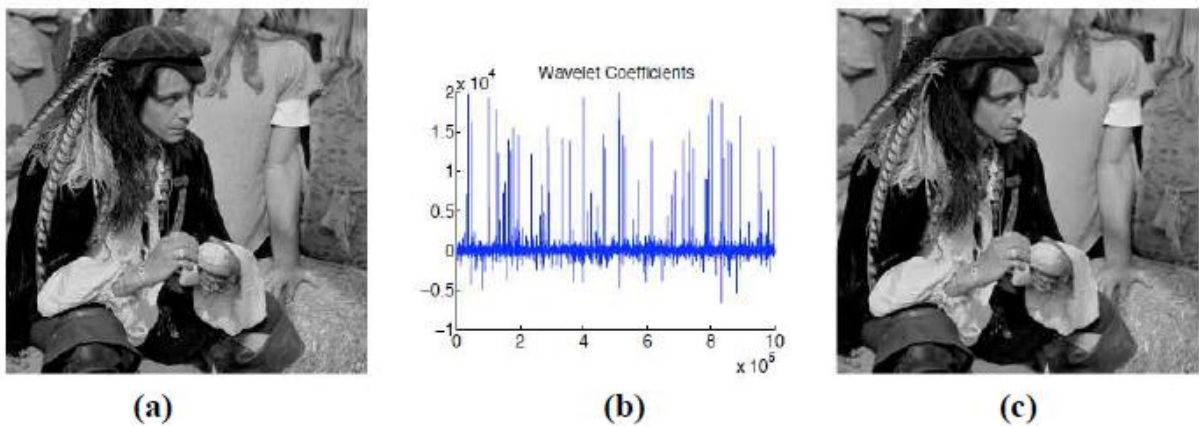
Matemáticamente hablando, tenemos un vector  $x$  en  $\mathbb{R}^n$  que expresamos en una base orto-normal  $\Psi = [\Psi_1, \Psi_2, \dots, \Psi_n]$ , de la siguiente forma:

$$X(t) = \sum_{i=1}^n s_i \Psi_i(t) \quad (4)$$

Donde  $s_i$  representa la contribución del átomo  $\Psi_i$  a la definición de la señal  $x$ ,  $S_i = \langle X, \Psi_i \rangle$ .

El ejemplo expuesto por Emmanuel Candès y Michael Wakin, [2]. CS explota el hecho de que muchas de las señales naturales son sparse o compresibles en el sentido de que tienen representaciones concisas cuando se expresan en la base adecuada.

Consideramos por ejemplo, la figura 5.a) y su transformada wavelet en b). Aunque casi todos los píxeles de la imagen tienen valores distintos de cero los coeficientes wavelet ofrecen una representación concisa: la mayoría de los coeficientes son pequeños, y los pocos coeficientes principales capturan casi toda la información.



**Figura 2. a)** Imagen original con valores de píxeles en el rango de  $[0,255]$  y **b)** sus coeficientes de transformada wavelet, relativamente pocos coeficientes capturan la

mayor parte de la energía de la señal. **c)** Reconstrucción obtenida a partir de los coeficientes wavelet [2].

Matemáticamente hablando, tenemos un vector  $x$  en  $\mathbb{R}^n$  que expresamos en una base ortonormal (por ejemplo una base wavelet)  $\Psi = [\Psi_1, \Psi_2, \dots, \Psi_n]$ , de la siguiente forma:

$$X = \sum_{i=1}^k s_i \Psi_i = s\Psi \quad (5)$$

Donde  $s_i$  representa la contribución del átomo  $\Psi_i$  a la definición de la señal  $x$ ,  $s_i = \langle x, \Psi_i \rangle$ . Ahora la definición de escasez estaría clara: cuando tenemos una señal sparse podemos descartar los coeficientes pequeños sin una pérdida de percepción elevada.

Consideremos  $x_k(t)$  formado sólo por las componentes de  $x(t)$  correspondientes a los  $k$  valores de  $x$  distintos de cero, por definición  $x_k := \Psi s_k$  donde  $s_k$  será un vector con sólo  $k$  componentes distintas de cero (el resto de componentes serán cero). Este vector es sparse ya que todas sus componentes salvo unas pocas,  $k$ , valen cero, se denomina entonces  $k$ -sparse. Dado que  $\Psi$  es una base ortonormal, tenemos que  $\|x - x_k\|_2 \approx \|s - s_k\|_2$ , y si  $s$  es una señal sparse entonces  $s_k$  será una buena aproximación y por tanto, el error  $\|x - x_k\|_2$  será pequeño. En resumen, si la señal es sparse podemos despreocuparnos de una gran cantidad de términos sin que suponga una gran pérdida. La figura 5.c) muestra un ejemplo donde la pérdida de percepción es casi inapreciable tras haber descartado el 97,5% de los coeficientes.

### 2.1.1.3. Muestreo Incoherente.

Por incoherencia entre dos matrices  $\Phi$  y  $\Psi$  se entiende que ningún elemento de la matriz  $\Phi$  puede ser obtenido mediante una combinación lineal de elementos de la matriz  $\Psi$  y viceversa.

Vemos la siguiente definición: La coherencia entre la matriz de mediciones  $\Phi$  y la base  $\Psi$  viene dada por

$$\mu(\Phi, \Psi) = \sqrt{n} \cdot \max_{1 \leq k, j \leq n} |\langle \phi_k, \Psi_j \rangle| \quad (6)$$

Si  $\Phi$  y  $\Psi$  contienen elementos correlacionados, la coherencia entre ellos será grande. De lo contrario, será pequeña. En ambos casos,  $\mu(\Phi, \Psi) \in [1, \sqrt{n}]$ .

Compressive sensing está basado en la incoherencia entre las dos matrices, veamos un ejemplo:  $\Phi$  es la base canónica  $\phi_k(\tau) = \delta(\tau - \kappa)$  y  $\Psi$  es la base de Fourier  $\Psi_j(t) = n^{-1/2} e^{j\omega_j t}$ . En este caso la pareja tiempo-frecuencia obedece a  $\mu(\Phi, \Psi) = 1$ , y por lo tanto tenemos la máxima incoherencia.

Las matrices aleatorias son altamente incoherentes con respecto a cualquier base fija  $\Psi$ . Seleccionamos una base ortonormal uniforme aleatoria que podemos construir ortonormalizando  $n$  vectores muestra independientes y uniformes. Luego,

con alta probabilidad la coherencia entre  $\Phi$  y  $\Psi$  será  $\sqrt{2 * \log n}$  . Por extensión, formas de ondas aleatorias con entradas i.i.d, por ejemplo entradas Gaussianas o binarias  $\pm 1$ , presentarán una baja coherencia con respecto a cualquier base  $\Psi$  [9].

#### 2.1.1.4 Diccionarios.

Un diccionario es un conjunto de funciones básicas mediante el cual se pueden representar señales como una combinación lineal de varias funciones de ese conjunto. A cada elemento del diccionario se le denomina átomo de diccionario.

##### 2.1.1.4.1 Diccionario En El Dominio Del Tiempo.

En el dominio del tiempo el diccionario se reduce simplemente a la matriz identidad puesto que se puede expresar una señal *sparse* en dicho dominio como una combinación lineal de los elementos contenidos en ese diccionario. En la Fig.4 se observa una señal *sparse* en el dominio del tiempo con solo tres componentes que son diferentes de cero. Dicha señal se puede expresar como una combinación lineal de los elementos de la matriz identidad que en este caso es el diccionario en el dominio del tiempo. Esta descomposición se analiza con más claridad en la ecuación (7) donde  $\Psi$  (es el diccionario) es la matriz identidad ( $I_{10 \times 10}$ ) y  $[0 \ 0.5 \ 0 \ 0.8 \ 0 \ 0 \ 0 \ 0 \ 0.3 \ 0]$  es el vector fila cuyas componentes corresponden a la amplitud de la señal en cada posición [5].

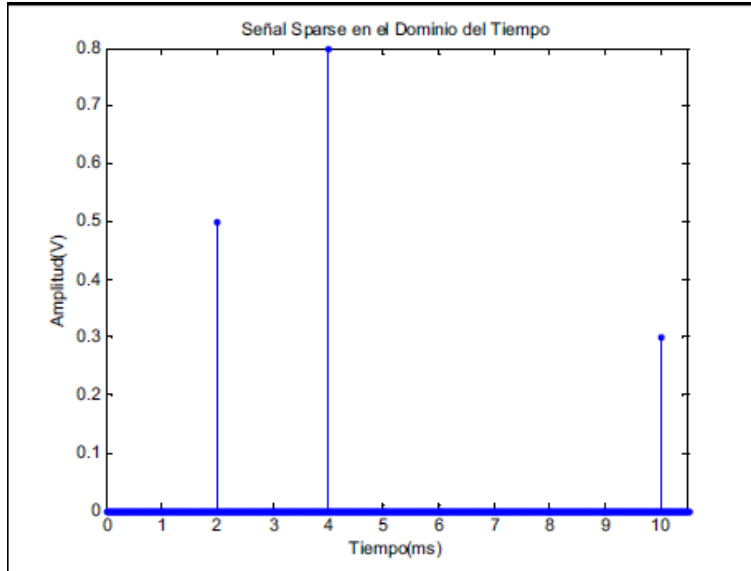
$$X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0.5 \\ 0 \\ 0.8 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.3 \\ 0 \end{bmatrix} \quad (7)$$

Observe que la señal *sparse*  $x$  se puede definir matemáticamente como:

$$X = \Psi \Phi = [0 \ 0.5 \ 0 \ 0.8 \ 0 \ 0 \ 0 \ 0 \ 0.3 \ 0]$$

El vector  $X$  genera la gráfica de la Fig. 4 quedando demostrado que una señal *sparse* se puede expresar en la forma dada en (3), es decir como una combinación de pocos elementos (átomos) contenidos en un diccionario previamente definido[5].





**Figura 3.** Señal sparse en el dominio del tiempo

#### 2.1.1.4.2 Diccionario En Bases Cosenos.

La generación de este diccionario se fundamenta en las funciones bases de la DCT (Transformada Discreta del Coseno), que es una transformada real ampliamente utilizada en Compresión de imágenes y video donde sus vectores bases son generadas a partir de la señal generatriz:

$$\Psi (n; f) =w (f) \cos\left(\frac{2f\Pi n}{N}\right) \quad (8)$$

Donde:

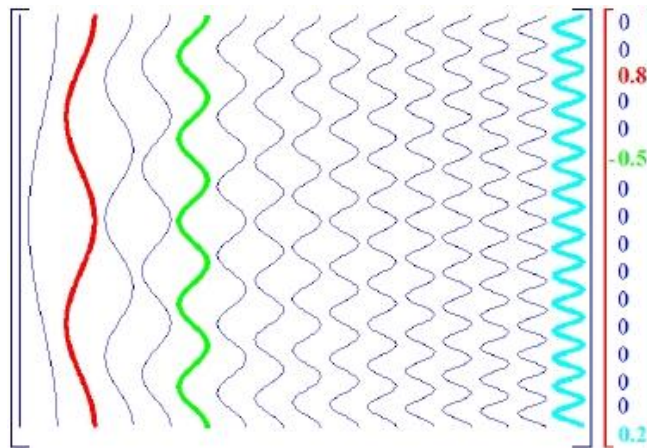
$$w(f)=\begin{cases} \frac{1}{\sqrt{n}} & \text{para } f = c \\ \sqrt{\frac{2}{N}} & \text{para } f > c \end{cases} \quad (9)$$

En la ecuación (8)  $n= 1,2,\dots,N$  es el tiempo discreto y  $N$  el tamaño de la muestra de la señal original  $X$  La frecuencia  $f$ , en general es una variable continua que para efectos de simplificación en el diseño del diccionario es discretizada tomando valores enteros, es decir  $f=1,2,..F_0$  siendo  $F_0$  la frecuencia máxima de la señal en estudio.

El termino  $w (f)$  es un factor de normalización El factor de normalización hace que cada columna del diccionario tenga norma unitaria.

A modo de ilustrar la apariencia del diccionario en bases de cosenos y visualizar mejor la combinación lineal de una señal *sparse* en el dominio de la transformada discreta del coseno (DCT).

En la Figura. 5 se muestran los átomos del diccionario.



**Figura 4.** Diccionario en bases de cosenos.

### 2.1.1.4.3 Diccionario Holográfico.

La definición de diccionario holográfico  $V$  es sencillamente proyectar cada función básica contenida en el diccionario  $\Psi$  la matriz de proyecciones aleatorias  $\Phi$ .

$$V = [V_1 : V_2 : V_3 : \dots : V_Z] = \Phi\Psi \quad (10)$$

Dónde cada componente  $V_i$  es un vector columna.

El objetivo es crear una especie de base de datos para comparar con las proyecciones de la señal que se desea reconstruir. El diccionario holográfico, la matriz de proyecciones aleatorias y el diccionario de funciones básicas son datos primordiales para llevar a cabo el proceso de reconstrucción de señales mediante algoritmos iterativos [5].

## 2.2. Reconstrucción De Una Señal Utilizando La Teoría CS.

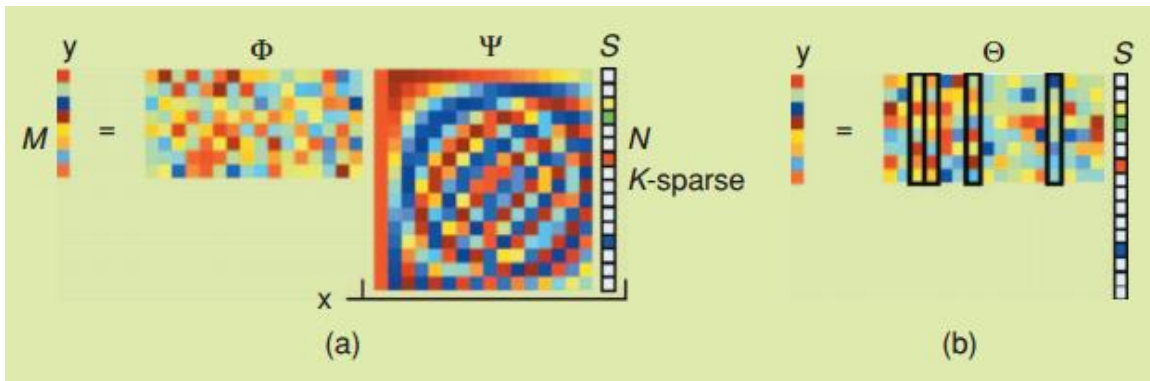
Definición: Sea  $x \in \mathbb{R}^n$  una señal que es  $k$ -Sparse, en algún dominio (tiempo, frecuencia...), ésta puede ser reconstruida con alta probabilidad, usando sólo un reducido conjunto de proyecciones y sobre una base aleatoria  $\Phi \in \mathbb{R}^m$  formada por

una colección de vectores  $\{\Phi_j\}_{j=1}^m$  siempre y cuando la matriz de proyecciones  $\Phi$  sea incoherente con el diccionario.

$$y = \Phi x = \Phi \Psi s = \Theta s \quad (11)$$

Donde  $\Theta = \Psi \Phi$  y  $\Phi$  es una matriz  $m \times n$ .

Dentro de  $y$  se encuentran implícitos como productos de combinaciones y operaciones los valores K-Sparse de la señal  $x$ . El grado de compresión durante el muestreo es controlado por el usuario y está expresado según el valor de  $M$ , que define el número de filas de la matriz aleatoria  $\Phi$ , que para nuestro caso  $M$  va a tomar por contajes con respecto a los valores de  $N$ , que define el número de columnas o longitud la señal Sparse.



**Figura 5. a)** CS con matriz de mediciones Gaussiana aleatoria  $\Phi$  y matriz transformada discreta del coseno (DCT)  $\Psi$ . El vector de coeficientes  $s$  es sparse con  $k=4$ . **b)** Proceso realizado con  $\Theta = \Psi \Phi$ . Existen 4 columnas que se corresponden con los  $s_i$  coeficientes distintos de cero, el vector de medidas  $y$  es una combinación lineal de dichas columnas [3].

### 2.2.1 Submuestreo y Recuperación De La Señal Sparse.

La teoría del CS establece que para la reconstrucción de la señal original se debe encontrar el conjunto de átomos del diccionario holográfico que más contribuyen a la formación del conjunto de mediciones. Esto se puede lograr al resolver el siguiente problema de optimización:

$$\min \|s\|_0 \text{ sujeto a } y = Vs \quad (12)$$

Donde  $\| \cdot \|_0$  representa la norma  $l_0$  y  $V \in \mathbb{R}^{m \times n}$  representa el diccionario holográfico.

Resolver este problema de optimización es bastante difícil, ya que no se puede garantizar que la solución al problema sea única. Para verificar que esta solución sea única, se tiene que barrer todos los subconjuntos que puedan generar subsistemas  $y = Vs$ , y luego verificar si este sistema tiene solución. El costo de realizar esta búsqueda exhaustiva crece exponencialmente a medida que  $n$  crece.

Debido a la dificultad al resolver este problema surge la necesidad de buscar algún tipo de aproximación, que sea lo bastante parecida a la solución del problema anteriormente planteado.

**Norma  $l_p$** , se define la norma  $l_p$  del vector  $S$  como

$$(\|s'\|_p) \sum_{i=1}^N |s_i|^p \quad (13)$$

Para resolver problemas de este tipo, se busca el vector de menor norma  $l_2$  que resuelva:

$$\tilde{s} = \operatorname{argmin} \|s'\|_2 \text{ tal que } \theta s' = y \quad (14)$$

Esta optimización tiene como solución:

$$\tilde{s} = \theta^T (\theta \theta^T)^{-1} y \quad (15)$$

Desafortunadamente, con la norma no se encuentra una solución  $k$ -sparse, devolviendo en cambio un vector no sparse, es decir, con muchos elementos distintos de cero.

**Norma  $l_1$** , la optimización basada en la norma  $l_1$

$$\langle s \rangle = \operatorname{argmin} \|s'\|_1 \text{ tal que } \theta s' = y \quad (16)$$

Podemos recuperar exactamente señales  $k$ -Sparse siempre que la solución al problema de minimización de la norma  $l_0$  sea lo suficientemente sparse, la solución a la norma  $l_1$  será única e igual a la solución que se obtiene al resolver el problema de minimización de la norma  $l_0$

### 2.2.2 Teorema De Muestreo No Lineal.

Tras la reflexión realizada en el apartado anterior, se decide recuperar la información a partir de la norma  $l_1$ . *Candès, Romberg y Tao* demuestran que casi siempre es posible recuperar la señal  $x$  exactamente resolviendo el siguiente problema de optimización no lineal:

$$\begin{aligned} \operatorname{Min} \quad & \|\tilde{x}\|_1 \text{ sujeto a } \Phi \tilde{x} = y \\ & x \in \mathbb{R}^n \end{aligned} \quad (17)$$

Donde 
$$\|\tilde{x}\|_1 = \sum_{i=1}^n |\tilde{x}_i|$$

Se desarrolla la relación del problema de optimización anterior con la programación lineal, la cual se define, de forma estándar, en términos de la variable  $x \in \mathbb{R}^m$  como:

$$\operatorname{min} \quad c^T x \text{ sujeto a } Ax=b \text{ siendo } x \geq 0 \quad (18)$$

Dónde  $c^T x$  es la función objetivo,  $Ax=b$  es la colección de restricciones y  $x \geq 0$  es la Constante positiva. Aunque la solución al problema (18) requiere una optimización no lineal, puede ser reformulado como un programa lineal en su forma estándar (17).

### 2.2.2.1 Teorema 1.

Fijar  $x \in \mathbb{R}^n$  y suponer que la secuencia de componentes  $k$  de  $x$  expresada en la base  $\Psi$  es  $k$ -sparse. Seleccionamos  $m$  mediciones en el dominio  $\Phi$  de forma uniforme aleatoriamente. Entonces sí

$$m \geq \mu^2(\Phi, \Psi) \cdot k \cdot \log n \quad (19)$$

Para alguna constante  $C$  positivo, la solución de (17) es exacta con probabilidad abrumadora. (Se Demuestra que la probabilidad de éxito supera el valor de  $1-\delta$  si

$$m \geq \mu^2(\Phi, \Psi) \cdot k \cdot \log \left( \frac{n}{\delta} \right)$$

Además, el resultado sólo está garantizada por casi todo rastro  $s$  secuencias con un soporte fijo [21].

Veremos tres observaciones:

1. El papel de la coherencia es imprescindible, cuanto menor sea la coherencia menos muestras necesitamos.
2. No aparece una pérdida de información considerable a pesar de usar simplemente  $m$  mediciones siendo  $m$  mucho menor que el número de muestras de la señal original. Si  $(\Phi, \Psi)$  es aproximadamente 1 entonces el número de muestras necesarias es del orden de  $k \cdot \log(n)$ .
3. La señal  $x$  puede ser recuperada a partir de las  $m$  mediciones (sin que la información importante de la señal  $k$ -sparse se vea dañada por la reducción de dimensión de  $x \in \mathbb{R}^n$  a  $y \in \mathbb{R}^m$ ) sin necesidad de conocer el número de elementos distintos de cero, su posición o su valor [21].

### 2.3. Diseño De La Matriz De Medición $\Phi$ .

La matriz de medición  $\Phi$  debe permitir la reconstrucción de la señal  $x$  de longitud  $n$  a partir de las  $m \ll n$  muestras pertenecientes al vector de proyecciones  $y$ . si  $x$  es  $k$ -sparse y se conocen las posiciones de las  $k$  muestras distintas de cero, el problema se resuelve siempre que  $m \geq k$ . Una condición necesaria y suficiente para esta simplificación del problema es que para cualquier vector  $v$  que comparta las mismas  $k$  muestras distintas de cero que  $s$  y para algún  $\varepsilon > 0$

$$1 - \varepsilon \leq \frac{\|\theta v\|_2}{\|v\|_2} \leq 1 + \varepsilon \quad (20)$$

Es decir, la matriz  $\theta$  debe conservar las longitudes de estos vectores  $k$ -sparse. En general, la posición de las  $k$  muestras distintas de cero no es conocida. De todas Formas, una condición suficiente para conseguir una solución estable es que la matriz  $\theta$  satisfaga (20) para un vector  $k$ -sparse  $v$  arbitrario. Esta condición se refiere a la propiedad de isometría restringida (RIP) [21].

También se requiere que ningún elemento de la matriz  $\Phi$  pueda ser obtenido mediante una combinación lineal de elementos de la matriz  $\Psi$  y viceversa, lo que se corresponde con la propiedad de incoherencia.

### 2.3.1 Propiedad De Isometría Restringida.

En esta sección introducimos una noción clave que ha resultado ser muy útil en el estudio de la robustez del Compressive Sensing, conocida como Propiedad de Isometría Restringida, **RIP**.

Objetivo

Construir matrices capaces de codificar la información del mayor conjunto posible de señales.

**Definición:** Para cada número real  $k = 1, 2, \dots$  se define la constante isométrica  $\delta_k$  de una matriz  $\Phi$  como el número más pequeño que se encuentra en el intervalo  $0 < \delta_k < 1$  verificando [9].

$$(1 - \delta_k) \|x\|_2^2 \leq \|\phi x\|_2^2 \leq (1 + \delta_k) \|x\|_2^2 \quad (21)$$

Implicaciones de la Propiedad de Isometría Restringida:

1) Reconstrucción de la señal a partir de las proyecciones aleatorias.

Si la Propiedad de Isometría Restringida se cumple, entonces la reconstrucción

Obtenida de la resolución del problema:

$$\text{Min}_{x \in \mathbb{R}^n} \|\tilde{x}\|_1 \quad \text{sujeito a} \quad \Phi \tilde{x} = y \quad (22)$$

Es exacta.

#### 2.3.1.1 Teorema 2

Suponemos que  $\delta_{2k} < \sqrt{2} - 1$ . Entonces la solución  $x^*$  a (22)

Obedece:

$$\|x^* - x\|_{l_2} \leq C_0 \|x - x_k\|_{l_1} / \sqrt{s} \quad y \quad \|x^* - x\|_{l_1} \leq C_0 \|x - x_k\|_{l_1} \quad (23)$$

Para alguna constante  $C_0$ , donde  $x_k$  es el vector  $x$  con todas sus componentes puestas a cero excepto el conjunto de las  $k$  componentes mayores. Las conclusiones de este teorema son más fuertes que las del *Teorema 1* en el cual se

expone que si  $x$  es  $k$ -sparse entonces  $x = x_k$  y por lo tanto la recuperación es exacta. Con este nuevo teorema, si  $x$  no es  $k$ -sparse (23) asegura que la calidad de la señal recuperada será tan buena como si se supieran de antemano las posiciones de los  $k$  elementos mayores. En otras palabras la reconstrucción es casi tan buena como la proporcionada por un oráculo que, con conocimiento pleno y perfecto sobre el objeto, pudiera extraer los  $k$  coeficientes más significantes.

Si somos capaces de encontrar una matriz  $\Phi$  que obedezca la hipótesis del teorema, podemos aplicarlo, y tendremos la garantía de recuperar exactamente todos los vectores  $k$ -sparse o de otro modo las  $k$  entradas mayores de todos los vectores, es decir, no existe probabilidad de fallo. Además puesto que la misma matriz  $\Phi$  funciona para todos los vectores de  $\mathbb{R}^n$ , a menudo nos referimos a ella como universal.

#### 2.4. GPSR (Proyección Del Gradiente Para La Reconstrucción Sparse).

El enfoque de GPSR (gradiente de proyección para la reconstrucción escasa) se requiere sólo productos matriz-vectores que implican a  $A$  y  $A^T$  (matriz), en lugar de acceso explícito a  $A$ . Se trata esencialmente de un algoritmo de proyección del gradiente (GP). Aplicado a una formulación de programación cuadrática (24) en la que la ruta de búsqueda de cada iteración se obtiene mediante la proyección de la dirección negativa del gradiente en el conjunto factible de la función objetivo, junto con una cuidadosa selección de criterios de paradas y una fase debiasing final (la que se ajustan sobre el conjunto de soporte de la solución de (24)), GPSR no implica límites en la matriz  $A^T A$  e implica sólo un nivel de iteración [6].

Por mucho ha habido un gran interés en la solución de este problema de optimización con restricciones convexa que se expresa en la siguiente ecuación (24).

$$\min_x \frac{1}{2} \|y - Ax\|_2^2 + T \|x\|_1 \quad (24)$$

Donde la variable  $\mathbf{X} \in \mathbb{R}^n$ ,  $y \in \mathbb{R}^k$ ,  $\mathbf{A}$  es una matriz de medición de  $N \times k$ ,  $T$  es un parámetro no negativo.

Para resolver este problema de optimización convexa, GPSR propone expresar la ecuación (24), como un programa cuadrático, esto se hace mediante el fraccionamiento de la variable  $x$  en sus partes positivas y negativas.

Formalmente, se introducen los vectores  $\mathbf{U}$  y  $\mathbf{V}$ , para hacer la sustitución en la ecuación (24) [12].

$$x = u - v, \quad u \geq 0, \quad v \geq 0 \quad (25)$$

Estas relaciones son satisfechas por  $U_i = (x_i)_+$  y  $V_i = (-x_i)_+$  para todo  $i = 1, 2, \dots, N$ , donde  $(\cdot)_+$  denota la parte positiva del operador definida como  $(x)_+ = \max\{0, x\}$ .

Tenemos así:  $\|x\|_1 = \mathbf{1}_n^T u + \mathbf{1}_n^T v$ . Donde  $\mathbf{1}_n = [1, 1, \dots, 1]^T$  es el vector que consiste en N requeridos, así que (24) se puede escribir como el siguiente programa cuadrático con restricciones (BCQP):

$$\min_{u,v} \frac{1}{2} \|y - A(u - v)\|_2^2 + T \mathbf{1}_n^T u + \mathbf{1}_n^T v \quad (26)$$

Donde  $U$  y  $V \geq 0$

Se puede escribirse (24) BCQP en forma más estándar,

$$\min_z c^T z + \frac{1}{2} z^T B z \equiv F(z) \quad s. t \quad z \geq 0, \quad (27)$$

Donde

$$z = \begin{bmatrix} u \\ v \end{bmatrix}, \quad b = A^T y, \quad c = T \mathbf{1}_{2n} + \begin{bmatrix} -b \\ b \end{bmatrix}$$

$$B = \begin{bmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{bmatrix} \quad (28)$$

#### 2.4.1 Dimensiones Del Programa Cuadrático Con Restricciones (BCQP).

Se puede observar que la dimensión del problema (27) es el doble que la del problema original (24):  $x \in \mathbb{R}^n$ , mientras que  $z \in \mathbb{R}^{2n}$ . Sin embargo, este aumento de la dimensión tiene sólo un impacto menor. Las operaciones con matrices que implican a B se pueden realizar más económicamente que su tamaño sugiere, mediante la explotación de su particular estructura (28). Para un determinado  $z = [u^T \ v^T]^T$  [6].

$$Bz = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} A^T A (u - v) \\ -A^T A (u - v) \end{bmatrix} \quad (29)$$

Lo que indica  $Bz$  es que se puede encontrar mediante el cálculo de la diferencia vectorial de  $u-v$  y a continuación, multiplicando cada vez por  $A$  y  $A^T$ . Desde  $\nabla F(z) = c + Bz$  (es el gradiente de la función objetivo en (26)), llegamos a la conclusión de que la computación de  $\nabla F(z)$  requiere una multiplicación de cada uno de  $A$  y  $A^T$ , suponiendo que  $c$ , depende de  $b = A^T Y$ , es el pre-calculado al principio del algoritmo.

Otra operación común en los algoritmos GP se describe a continuación es encontrar el escalar  $z^T Bz$  para un determinado  $z = [u^T, v^T]^T$ . Es fácil ver que

$$z^T Bz = (u - v)^T A^T A (u - v) = \|A(u - v)\|_2^2$$



Lo que indica que esta cantidad se puede calcular utilizando sólo una única multiplicación por A. Puesto que  $F(z) = (1/2)z^T B z + C^T z$ , se deduce que la evaluación de  $F(z)$  también requiere sólo una multiplicación por A.

Es importante señalar que cuando la solución de (23) se sabe de antemano que cuando la solución es no negativa, podemos reescribir directamente los problemas y se

$$\min_x (T1_n - A^T y)^T x + \frac{1}{2} x^T A^T A x, \quad \text{s. t. } x \geq 0 \quad (30)$$

Este problema es, como (27), una BCQP, y se puede resolver con los mismos algoritmos. Sin embargo la presencia de la restricción  $x \geq 0$  nos permite evitar la división de la variables en partes positivas y negativas.

#### 2.4.2 Técnica del Algoritmo De Proyección Del Gradiente.

En esta sección se discuten las técnicas para la resolución de gradiente proyectado (GP), En nuestro enfoque GPSR, nos movemos de la siguiente manera  $z^{(k)}$  es la iteración en la función objetivo ,donde se incrementan estas iteraciones así  $z^{(k+1)}$  . En primer lugar, elegimos el parámetro escalar  $\alpha^{(k)} > 0$  [6] .

$$w^{(k)} = (z^{(k)} - \alpha^{(k)} \nabla F(z^{(k)}))_+ \quad (31)$$

Donde se encuentra un umbral que permite pasar todos los valores no negativos, y pone a cero el resto, es decir, el gradiente incrementa cada iteración en la dirección de ascenso más agudo, GPSR elige un paso y actualiza la solución [7].

Ahora establecemos un segundo escalar  $\lambda^{(k)} \in [0,1]$  .

$$z^{(k+1)} = (z^{(k)} + \lambda^{(k)}(w^{(k)} - z^{(k)}))_+ \quad (32)$$

Nuestros enfoques descritos a continuación difieren en sus decisiones de  $\alpha^{(k)}$  y  $\lambda^{(k)}$ .

#### 2.4.3 Proyección Del Gradiente Básico: GPSR-Basic Algoritmo.

En el enfoque básico, la búsqueda de cada iteración  $z^{(k)}$  a lo largo del gradiente negativo de  $-\nabla F(z^{(k)})$ , que se proyecta en el ortante no negativo, y la realización de una búsqueda de línea con marcha hacia atrás hasta lograr una disminución suficiente de la función objetivo (F) esta disminución se logra con la estrategia llamada ("regla de Armijo a lo largo del arco de proyección. ") Utilizamos una aproximación inicial para  $\alpha^{(k)}$  eso sería el minimizador exacto de F a lo largo de esta dirección si no hay nuevos límites debían ser encontrado.

En concreto, se define el vector  $g^{(k)}$  por

$$g_i^k = \begin{cases} (\nabla F(z^{(k)}))_i & \text{if } z_i^{(k)} > 0 \text{ o } (\nabla F(z^{(k)}))_i < 0 \\ 0 & \text{de otra manera} \end{cases} \quad (33)$$

Entonces elegimos la estimación o paso inicial para ser

$$\alpha_0 = \arg \min_{\alpha} F(z^{(k)} - \alpha g^{(k)}),$$

A continuación, con la siguiente expresión se consigue el tamaño adecuado de paso hasta alcanzar una disminución suficiente en la función objetivo.

La cual se puede calcular de forma explícita como

$$\alpha_0 = \frac{(g^{(k)})^T g^{(k)}}{(g^{(k)})^T B g^{(k)}} \quad (34)$$

Para protegerse contra los valores de  $\alpha_0$  que son demasiado pequeños o demasiado grande, nos limitamos al intervalo  $[\alpha_{\min}, \alpha_{\max}]$ , donde  $0 < \alpha_{\min} < \alpha_{\max}$ . (En este sentido, definir el operador  $\text{med}(a, b, c)$  para ser el valor medio de sus tres argumentos escalares.) Esta técnica para el ajuste  $\alpha_0$  es aparentemente novedosa, y produce un paso aceptable con mucha más frecuencia que la anterior elección de  $\alpha_0$  como el minimizador de  $F$  a lo largo de la dirección  $-\nabla F(z^{(k)})$ , haciendo caso omiso de los límites [13].

El algoritmo completo se define como sigue.

**Paso 0** (inicialización): dado  $z^{(0)}$  elegir los parámetros  $\beta \in (0, 1)$  y  $\mu \in (0, 1/2)$ ; establecer  $k = 0$ .

**Paso 1**: Calcule  $\alpha_0$  a partir de (24), y reemplazar por  $\alpha_0$   $\text{mid}(\alpha_{\min}, \alpha_0, \alpha_{\max})$ .

**Paso 2** (búsqueda de línea de marcha atrás): Elija  $\alpha^{(k)}$  siendo el primer número de la secuencia de  $\alpha_0, \beta\alpha_0, \beta^2\alpha_0, \dots$  de tal manera que

$$F(z^{(k)} - \alpha^{(k)} \nabla f(z^{(k)})) \leq F(z^{(k)})$$

$$-\mu \nabla F(z^{(k)})^T (z^{(k)} - (z^{(k)} - \alpha^{(k)} \nabla f(z^{(k)}))),$$

Y establecer  $z^{(k+1)} = (z^{(k)} - \alpha^{(k)} \nabla f(z^{(k)}))$ .

**Paso 3**: Realizar la prueba de convergencia y terminar con  $z^{(k+1)}$  solución aproximada si se cumple; en caso contrario hacer  $k \leftarrow k + 1$  y volver a Paso 1.

El cálculo en cada iteración consiste en multiplicaciones de matriz-vector que involucran  $A$  y  $A^T$ , junto con algunos productos internos (los menos significativos) la participación de los vectores de longitud  $n$ . El Paso 2 requiere una evaluación de  $F$  para cada valor de  $\alpha^{(k)}$  tratado, donde cada evaluación requiere una única multiplicación por  $A$ . Una vez que se determina el valor de  $\alpha^{(k)}$ , podemos encontrar

$z^{(k+1)}$  y luego  $\nabla f(z^{(k+1)})$  con una más la multiplicación por  $A^T$ . Otra multiplicación por  $A$  es suficiente para calcular el denominador de (34) al inicio de cada iteración. En total, el número de multiplicaciones por  $A$  o por iteración  $A^T$  son dos más el número de valores de  $\alpha^{(k)}$  intentó en el paso 2.

#### 2.4.4 Barzilai-Borwein Proyección Del Gradiente: Algoritmo GPSR-BB

Algoritmo GPSR-Basic asegura que la función objetivo ( $F$ ) disminuye en cada iteración. Recientemente, se ha presentado un enfoque debido a Barzilai y Borwein (BB) [7]. Este enfoque fue originalmente desarrollado en el contexto de la reducción al mínimo sin restricciones de una función no lineal lisa ( $F$ ). Se calcula cada paso por la fórmula  $\delta^{(k)} = -H_k^{-1} \nabla f(z^{(k)})$ , donde  $H_k$  es una aproximación a la matriz de Hesse de  $F$  en  $z^{(k)}$ . Barzilai y Borwein proponen una simple elección particular de la aproximación de  $H_k$ : Establecen que ser un múltiplo de la identidad  $H_k = \eta^{(k)}I$ , donde  $\eta^{(k)}$  se elige de manera que esta aproximación tiene similares al comportamiento de la verdadera de la matriz de Hesse sobre el más reciente paso, esto es [24].

$$\nabla F(z^{(k)}) - \nabla F(z^{(k-1)}) \approx \eta [z^{(k)} - z^{(k-1)}] \quad (35)$$

Con  $\eta^{(k)}$  elegido para satisfacer esta relación en el mínimos cuadrados sentido. En la configuración no constreñida, la actualización fórmula que [7].

$$z^{(k+1)} = z^{(k)} - (\eta^{(k)})^{-1} \nabla F(z^{(k)});$$

Este paso se toma incluso si se produce un aumento en  $F$ . para ser eficaz en problemas sencillos. Numerosas variantes tienen ha propuesto recientemente, y se sometió a un buen acuerdo de evaluación teórica y computacional [13]. El enfoque BB se ha extendido a BCQPs, donde elegimos  $\lambda^k$  en (33) como el minimizador exacto sobre el intervalo  $[0, 1]$  y elija  $\eta^{(k)}$  en cada iteración de la manera descrito anteriormente, excepto que  $\alpha^{(k)} = (\eta^{(k)})^{-1}$  es restringido al intervalo  $[\alpha_{\min}, \alpha_{\max}]$ . En la definición del valor de  $\alpha^{(k+1)}$  en el paso 3 a continuación, hacemos uso del hecho de que para  $F$  se define en (26), tenemos

$$\nabla f(z^{(k)}) - \nabla f(z^{(k-1)}) = B(z^{(k)} - z^{(k-1)})$$

**Paso 0** (inicialización): dado  $z^{(0)}$ , elegir los parámetros  $\alpha_{\min}, \alpha_{\max}, \alpha^{(0)} \in [\alpha_{\min}, \alpha_{\max}]$ , y establece

$k = 0$ .

**Paso 1:** Calcular el paso:

$$\delta^{(k)} = (z^{(k)} - \alpha^{(k)} \nabla f(z^{(k)}))_+ - z^{(k)} \quad (36)$$

**Paso 2** : (búsqueda en línea): Encuentra el escalar  $\lambda^{(k)}$  que reduce al mínimo

$F(z^{(k)} + \lambda^{(k)} \delta^{(k)})$  en el intervalo  $\lambda^{(k)} \in [0, 1]$ , y ajuste  $z^{(k+1)} = z^{(k)} + \lambda^{(k)} \delta^{(k)}$ .

**Paso 3** (actualización  $\alpha$ ): calcular

$$\gamma^{(k)} = (\delta^{(k)})^T B \delta^{(k)}; \quad (37)$$

Si  $\gamma^{(k)} = 0$ , y dejar  $\alpha^{(k+1)} = \alpha_{\max}$ , de lo contrario

$$\alpha^{(k+1)} = \text{mid} \left\{ \alpha_{\min}, \frac{\|\delta^{(k)}\|_2^2}{\gamma^{(k)}}, \alpha_{\max} \right\}$$

**Paso 4:** Realizar la prueba de convergencia y terminar con la solución aproximada  $z^{(k+1)}$  si se cumple; en caso contrario hacer  $k \leftarrow k + 1$  y volver a Paso 1.

Como  $F$  es cuadrática, el parámetro de búsqueda de línea  $\lambda^{(k)}$  en el paso 2 puede calcularse simplemente utilizando la siguiente expresión cerrada [13]:

$$\lambda^{(k)} = \min \left\{ 0, \frac{(\delta^{(k)})^T \nabla F(z^{(k)})}{(\delta^{(k)})^T B \delta^{(k)}}, 1 \right\}$$

(Cuando  $\delta^{(k)T} B \delta^{(k)} = 0$ , fijamos  $\lambda^{(k)} = 1$ .) El uso de este parámetro  $\lambda^{(k)}$  elimina una de las propiedades sobresalientes del enfoque de Barzilai-Borwein, a saber, la posibilidad de que  $F$  puede aumentar en algunas iteraciones. Sin embargo, en nuestros problemas, que apareció para mejorar el rendimiento en la variante estándar no monótono, que establece  $\lambda^{(k)} \equiv 1$  [24].

#### 2.4.4. 1 Convergencia.

Se Resume propiedades de convergencia de los dos algoritmos descrito anteriormente, suponiendo que se produce la terminación sólo cuando  $z^{(k+1)} = z^{(k)}$  (que indica que  $z^{(k)}$  es óptimo).

- **Teorema 3:** La secuencia de iteración  $\{z^{(k)}\}$  generada por el bien de la GPSR-Basic y GPSR-BB algoritmos o bien termina en una solución o más converge a una solución de (27).
- **Teorema 4.** Se puede utilizar para mostrar que todos los puntos de acumulación de  $\{z^{(k)}\}$  son puntos estacionarios. (Este resultado se aplica a un algoritmo en el que la  $\alpha^{(k)}$  son elegidos por un esquema diferente, pero el único requisito pertinente sobre estos parámetros en la prueba que se encuentran en el rango  $[\alpha_{\min}, \alpha_{\max}]$ , como es el caso aquí. Desde el objetivo en (26) está delimitada claramente por debajo (por cero), podemos aplicar para deducir la convergencia a una solución de (27) a una tasa de R-lineal.

#### 2.4.4.2 Terminación.

La decisión sobre cuándo una solución aproximada es de una calidad suficientemente alta para terminar los algoritmos es muy difícil. Deseamos para la

aproximación de z solución sea razonablemente cerca de una solución  $z^*$  y / o que el valor de la función  $F(z)$  sea razonablemente cerca de  $F(z^*)$ , pero al mismo tiempo deseamos evitar el cálculo excesivo implicados en la búsqueda de una solución demasiado precisa. Para el problema de (26), teniendo en cuenta que la selección de variables es la principal motivación de la formulación (24) y que un paso debiasing puede llevarse a cabo en una fase de procesamiento posterior, deseamos que los componentes no nulos de la solución  $z$  aproximada sea una verdadera solución de  $z^*$ .

Se ha utilizado previamente para los problemas BCQP es

$$\|Z - (Z - \alpha \nabla F(Z))\| \leq \text{tolp} \quad (38)$$

Donde TolP es un parámetro pequeño y  $\alpha$  es una constante positiva). Este criterio está motivada por el hecho de que el lado izquierdo es cero si y sólo si  $z$  es óptima. Un segundo criterio, similar está motivado por los resultados de perturbación de complementariedad lineal estas consideraciones motivan una serie de posible criterio de terminación. Un criterio estándar de problemas (LCP). Hay una constante tal que  $C_{LCP}$  [14].

$$\text{dist}(z, S) \leq C_{LCP} \|\min(z, \nabla F(z))\|$$

Donde  $S$  denota el conjunto solución de (27),  $\text{dist}(\cdot)$  es el operador de la distancia, y el mín del lado derecho se toma componente a componente. Con este criterio de convergencia, podemos definir lo siguiente:

$$\|\min(Z, \nabla F(Z))\| \leq \text{tolp} \quad (39)$$

Un tercer criterio propuesto recientemente en es basado en la teoría de la dualidad de la formulación original (23) y se puede demostrar que el doble es

$$\begin{aligned} \max_s \quad & -\frac{1}{2} s^T s - y^T s \\ \text{s.t.} \quad & -\tau \mathbf{1}_n \leq \mathbf{A}^T s \leq \tau \mathbf{1}_n. \end{aligned} \quad (40)$$

Si  $S$  es factible para (40), entonces

$$\frac{1}{2} \|y - Ax\|_2^2 + \frac{1}{2} s^T s + y^T s \geq 0 \quad (41)$$

Con igualdad obtenida si y sólo si  $x$  es una solución de (24) y  $S$  es una solución de (36). Para definir un criterio de terminación, invertimos la transformación en (18) para obtener un candidato  $x$ , y a continuación la construcción de una  $S$  factible es el siguiente:

$$s \equiv \tau \frac{Ax - y}{\|\mathbf{A}^T(Ax - y)\|_\infty}$$

(42)

Sustituyendo estos valores en el lado izquierdo de (41), que puede declarar la terminación cuando esta cantidad es inferior a un umbral TolP. Tenga en cuenta que esta cantidad es un límite superior en la brecha entre  $F(z)$  y el valor objetivo óptimo de  $F(z^*)$ .

Tener en cuenta los índices distintos de cero de  $Z$  o de la forma en que éstos cambian en iteraciones recientes. En el cuarto criterio, la terminación se declara cuando el conjunto índices distinto de cero de una iteración  $z^{(k)}$  cambia por una cantidad inferior a un umbral determinado TolA.

Específicamente, definimos

$$\begin{aligned} \mathcal{I}_k &= \{i \mid z_i^{(k)} \neq 0\}, \\ \mathcal{C}_k &= \{i \mid (i \in \mathcal{I}_k \text{ and } i \notin \mathcal{I}_{k-1}) \text{ or} \\ &\quad (i \notin \mathcal{I}_k \text{ and } i \in \mathcal{I}_{k-1})\}, \end{aligned}$$

Y terminará si

$$|\mathcal{C}_k| / |\mathcal{I}_k| \leq \text{TolA} \quad (43)$$

Este criterio se adapta bien a la clase de problemas abordado en este trabajo (donde se espera que la cardinalidad de  $\mathcal{I}_k$ , en etapas posteriores del algoritmo, a ser mucho menor que la dimensión de  $z$ ), y para Los algoritmos del tipo de proyección del gradiente, los cuales genera una iteración en la frontera del conjunto factible. O para los algoritmos que generan una iteración que permanece en el interior del conjunto factible.

Es difícil elegir un criterio de terminación entre estas opciones que se desempeña bien en todo conjunto de datos y en todos los contextos, usamos (39), con Tolp=10<sup>-2</sup>, que pareció dar bastante resultados consistentes [15].

#### 2.4.4.3 Parámetros Tao (T) De Regularización: Arranque En Caliente.

El enfoque de proyección del gradiente se beneficia de un buen punto de partida se inicializa GPSR en el que T se cambia a un valor cercano o se suele tomar menos iteraciones que el primero; el número de iteraciones depende de la proximidad de los valores de T y la cercanía de las soluciones.

El uso de la técnica arranque en caliente, donde podemos resolver una secuencia de valores de T, el cual se desplazan a un valor siguiente en la secuencia, ya sea en el aumento o disminución. En general, es mejor aumentar para solucionar el fin de T, ya que el número de componentes distintos de cero de  $x$  es la solución de (24) normalmente disminuye a medida que T aumenta, por lo que los pasos adicionales

necesarios en la adopción entre dos soluciones sucesivas suelen adoptar la forma de traslado de algunos componentes  $x$  distintos de cero a la contorno y el ajuste de los otros valores distintos de cero.

Para la solución de una gama de valores de  $\tau$  es que a menudo es difícil determinar un apropiado valor a priori. El análisis teórico puede sugerir un cierto valor, pero es beneficioso para explorar una gama de soluciones en torno a este valor y examinar la escasez de las soluciones resultantes, posiblemente usando alguna prueba basada en la escasez y la solución de ajuste de mínimos cuadrados para elegir la "mejor" solución entre estas posibilidades.

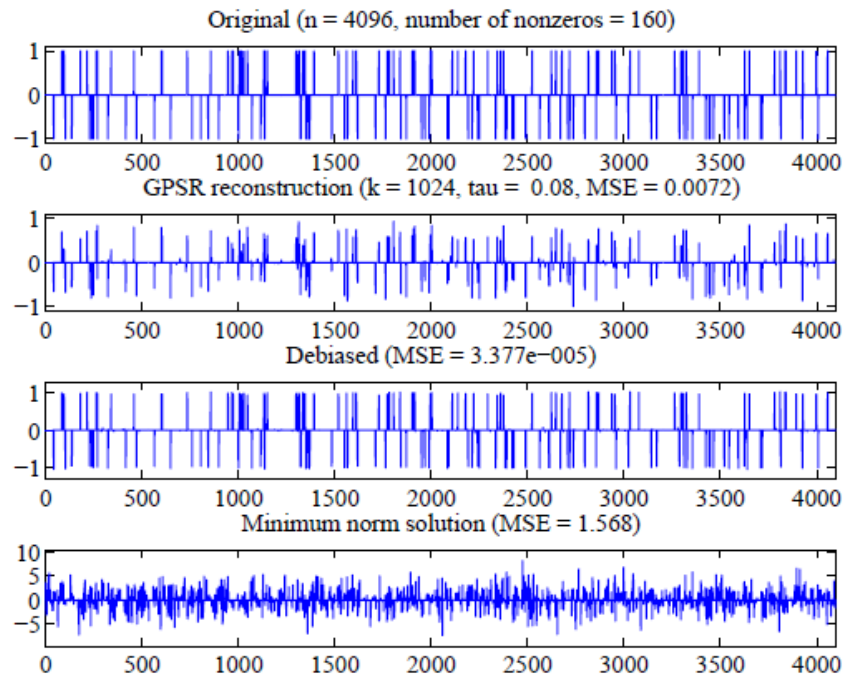
#### 2.4.4.4. Comparación entre GPSR y la norma l1.

En nuestro primer experimento consideramos un típico ejemplo de (CS). Donde el objetivo es reconstruir una señal de longitud- $n$  que es escasa (en la base canónica) a partir de  $k$  observaciones, donde  $k < n$ . En este caso, la matriz  $A$  es de longitud  $k \times n$  se obtiene llenándola con muestras de una distribución gaussiana estándar y luego ortonormalizando las filas.

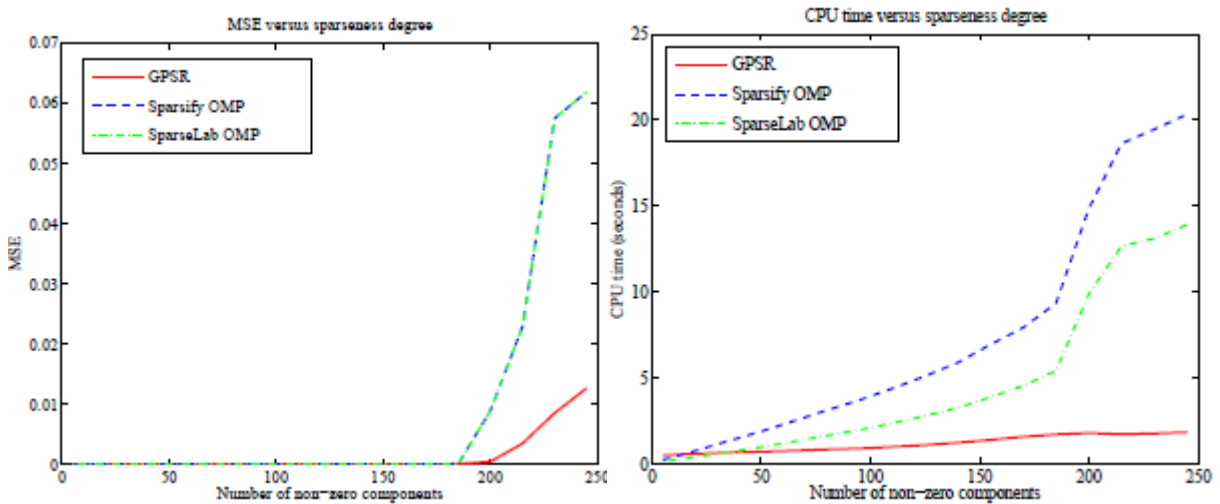
En este ejemplo,  $n = 4096$ ,  $k = 1024$ , la señal original  $x$  contiene 160 picos colocados al azar, con  $\sigma^2 = 10^{-4}$ . Y  $T$  parámetro se elige como se sugiere:

$$\tau = 0.1 \|A^T y\|_\infty; \tag{44}$$

Este valor puede ser entendido por darse cuenta de que para  $T \geq \|A^T Y\|_\infty$  es el mínimo único de (24) es el vector cero. La señal original y la estimación obtenida mediante la resolución de (24) el uso de la versión de la monótona GPSR-BB (que es esencialmente el mismo que el producido por el nonmonotone GPSR-BB y GPSR Básico) se muestra en la Figura 6. También se muestra la reconstrucción obtenida después de la debiasing; a pesar de que GPSR-BB hace un excelente trabajo en la localización de los picos, la reconstrucción debiased exhibe un menor error cuadrático medio (MSE) con respecto a la señal original. Finalmente también representa la solución de mínimos  $\ell_2$ -norma al sistema indeterminado de  $y = Ax$ , obtenida con el `mldivide` función de MATLAB [6].



**Figura 6.** Arriba: señal original, abajo la reconstrucción a través de la minimización de (24) obtenido por GPSR-BB, después la reconstrucción con debiasing, más abajo la estimación de norma mínima obtenida utilizando la función de MATLAB mldivide..



**Figura 7.** Evolución de la función objetivo y la reconstrucción MSE, en función del tiempo de la CPU, lo que corresponde para el experimento ilustrado en la Figura 6.



A continuación, se compara la eficiencia computacional del algoritmo GPSR contra OMP, que a menudo es considerado como un método altamente eficiente, y que es especialmente adecuado para casos muy dispersos.

El MSE de reconstrucción y el tiempo de la CPU requerido por GPSR-BB y OMP, como una función del número de componentes distintos de cero en  $x$ . se observa que todos los métodos básicamente al obtener una reconstrucción para  $m$  casi igual 200, con la soluciones de OMP que comienza a degradarse antes y más rápido.

Sobre la eficiencia computacional, que es nuestro principal objetivo, podemos observar que GPSR-BB es claramente más rápido que ambas implementaciones de OMP, excepto en el caso de escasez extrema ( $m < 50$  elementos distintos de cero en el 4096-vector  $x$  minimizando (23) o cualquier otra función objetivo, se fijan tamaño de la matriz ( $1024 \times 4096$ ) y se considera una gama de grados de poca densidad: el número  $m$  de los picos no ceros en  $x$  (al azar Los valores situados de  $\pm 1$ ) oscila entre 5 y 250.

## 2.5. SPGL1 (gradiente de proyección espectral para L1).

Es Un programa para dar solución a problemas escasos a gran escala que se basa en la norma l1 de mínimos cuadrados regularizados. Está diseñado para resolver los tres problemas siguientes [16]:

- **basis pursuit.**

$$(BP) \quad \underset{x}{\text{minimice}} \|x\|_1 \text{ sujeto a } Ax = b$$

- **basis pursuit denoise.**

$$(BP_\sigma) \quad \underset{x}{\text{minimice}} \|x\|_1 \text{ sujeto a } \|Ax - b\|_1 \leq \sigma$$

- **Lasso.**

$$(LS_\tau) \quad \underset{x}{\text{minimice}} \|Ax - b\|_2 \text{ sujeto a } \|x\|_1 \leq \tau,$$

### 2.5.1. Basis Pursuit Denoise.

Este problema tiene como objetivo encontrar una solución escasa al sistema de ecuaciones indeterminado  $Ax = b$ , donde  $A$  es una matriz  $m \times n$  y  $B$  es un vector  $m$ . donde típicamente  $m \ll n$ .

El planteamiento propuesto es resolver el problema de optimización convexa

$$(BP) \quad \underset{x}{\text{minimice}} \|x\|_1 \text{ sujeto a } Ax = b$$

En presencia de datos con ruido o imperfectos, no es deseable encajar exactamente en el sistema lineal. En su lugar, la restricción en (BP) es relajado para obtener el problema basis pursuit denoise (BPDN).

$$(BP_{\sigma}) \quad \underset{x}{\text{minimice}} \|x\|_1 \text{ sujeto a } \|Ax - b\|_1 \leq \sigma$$

Donde  $\sigma$  es un parámetro positivo y una estimación del nivel de ruido en los datos. El caso  $\sigma = 0$  corresponde a una solución de **basis pursuit** (BP).

problema que escala bien y trabaja eficazmente en un contexto libre de la matriz se presenta un algoritmo, adecuado para aplicaciones a gran escala, que es capaz de la búsqueda de soluciones de  $(BP_{\sigma})$  para cualquier valor de  $\sigma \geq 0$ .

Nuestro enfoque se basa en la refundición de  $(BP_{\sigma})$  como un problema de encontrar la raíz de una ecuación no lineal de una sola variable. En cada iteración de nuestro algoritmo, una estimación de dicha variable se utiliza para definir un problema convexo de optimización, cuya solución produce información de derivación de que se puede utilizar por un método de Newton basado en la raíz de investigación [17].

## 2.5.2 regularización de la norma l1.

El problema de optimización convexa  $(BP_{\sigma})$  es sólo una de la posible declaración del problema de la norma-regularizado de mínimos cuadrados, la etiqueta BPDN se aplica típicamente a los problemas de mínimos cuadrados.

$$(QP_{\lambda}) \quad \underset{x}{\text{minimice}} \|Ax - b\|_2^2 + \lambda \|x\|_1$$

Que es el planteamiento del problema propuesto por Chen, Donoho, y Saunders Una tercera formulación sería [8].

$$(LS_{\tau}) \quad \underset{x}{\text{minimice}} \|Ax - b\|_2 \text{ sujeto a } \|x\|_1 \leq \tau,$$

Este problema tiene una limitación explícita de una-norma y se llama a menudo el problema Lasso. El parámetro ( $\lambda$ ) está relacionado con el multiplicador de LaGrange de la restricción en  $(LS_{\tau})$  y para el recíproco del multiplicador de la restricción en  $(BP_{\sigma})$ . Así, hay opciones de parámetros de  $\sigma$ ,  $\lambda$ , y  $\tau$ , las soluciones de  $(BP_{\sigma})$ ,  $(QP_{\lambda})$ , y  $(LS_{\tau})$  coinciden, y estos problemas son en cierto sentido equivalente. Sin embargo, a excepción de los casos especial tal como son los ortogonales: los parámetros que hacen que estos problemas no pueden ser equivalente y conocido a priori.

Para el caso en el que se conoce a  $\sigma$  que es una estimación del nivel de ruido, Chen, Donoho, y Saunders, sostienen que la elección de  $\lambda = \sigma \sqrt{2 \log n}$  tiene importantes propiedades de optimalidad. Sin embargo este argumento gira en torno a la ortogonalidad de  $A$  [19].

Nos centramos en la situación en la que  $\sigma$  es de aproximadamente conocidas, como cuando podamos estimar los niveles de ruido inherentes a un sistema subyacente o en las mediciones tomado. En este caso, es preferible resolver  $(BP\sigma)$ , y aquí este es nuestro objetivo principal. Una consecuencia importante de nuestro enfoque es que también se puede utilizar para resolver de manera eficiente los problemas estrechamente relacionados  $(BP)$  y  $(LS_T)$ . Nuestro algoritmo se aplica también a los tres problemas en el dominio complejo, que puede surgir en aplicaciones de procesamiento de señales.

### 2.5.3 Enfoque.

En el corazón de nuestro enfoque es la capacidad de resolver eficazmente una (secuencia de los problemas  $LS_T$ ) utilizando la técnica del gradiente espectral proyectado (SPG) Al igual que con  $(QP_\lambda)$ , este problema se ha parametrizado por un escalar; la diferencia crucial, sin embargo, es que la solución dual de  $(LS_T)$  proporciona información vital sobre cómo actualizar a  $T$  para que la siguiente solución de  $(LS_T)$  es mucho más cercana a la solución de  $(BP\sigma)$ . Vemos que  $X_T$  denota la solución óptima de  $(LS_T)$ . La función de un solo parámetro.

$$\varphi(\tau) = \|r_\tau\|_2, \text{ con } r_\tau := b - Ax_\tau \varphi(\tau) \quad (45)$$

Da el valor óptimo de  $(LS_T)$  para cada  $T \geq 0$ , la derivada está dada por  $-\lambda_T$ , Donde  $\lambda_T \geq 0$  es la única solución de  $(LS_T)$ . Es importante destacar que esta doble solución se puede obtener fácilmente como un subproducto de la reducción al mínimo de  $(LS_T)$ . Este enfoque se basa en aplicar El método de Newton para encontrar una raíz de la ecuación no lineal

$$\varphi(\tau) = \sigma \quad (46)$$

Que define una secuencia de parámetros de regularización  $T_k \rightarrow T_\sigma$ , donde  $X_{T_\sigma}$  es una solución de  $(BP\sigma)$ . En otras palabras,  $T_\sigma$  es el parámetro de causas de  $(LS_T)$  y  $(BP\sigma)$  que comparten la misma solución. El algoritmo encuentra las raíces de  $(BP\sigma)$  y  $(LS_T)$  evaluando las función  $\varphi$  y su derivada  $\varphi'$ [17].

#### 2.5.3.1 Curva de Pareto.

La curva de Pareto. En la función  $\varphi$  definida en (46) se obtiene el valor óptimo del problema limitado  $(LS_T)$  para cada valor del parámetro  $T$  de regularización. Su gráfica traza el equilibrio óptimo entre 1-norma de la solución  $x$  y la 2-norma de la residual  $r$ , que define la curva de Pareto.

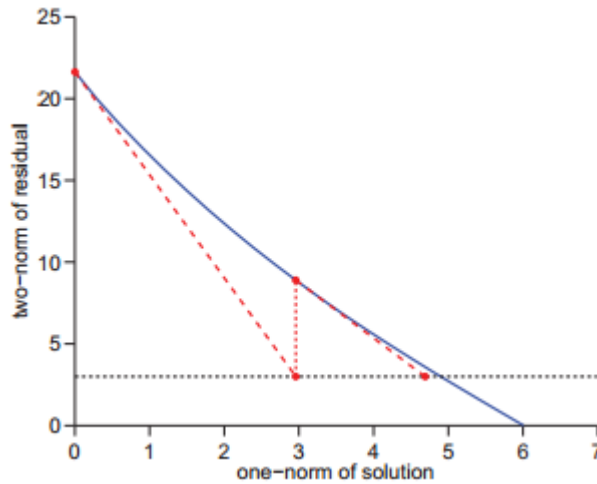
La figura 13 muestra el gráfico de  $\varphi$  para un problema típico. El procedimiento de búsqueda de raíz con el método de Newton, se propone para la localización específica de los puntos en la curva de Pareto, por ejemplo la búsqueda de las raíces en (41) en varias propiedades importantes de la función  $\varphi$ , que es convexa y diferenciable a la función de  $T$ . La diferenciable de  $\varphi$  es tal vez poco intuitivo,

dado que la 1-norma restricción en (LST) no es diferenciable. Para hacer frente a la nonsmoothness de la restricción de una sola norma, se hace un llamamiento a la teoría de la dualidad de LaGrange. Se discute la mayor parte propiedades importantes a continuación.

### 2.5.3.2 Subproblema doble.

El dual del problema Lasso (LST) desempeña un papel impórtate en la comprensión de la curva de Pareto. Con el fin de derivar el doble de (LST), lo primero es la refundición de (LST) como el problema equivalente

$$\underset{r,x}{\text{minimice}} \ ||r||_2 \quad \text{sujeto a } Ax + r = b, ||x||_1 \leq T \quad (47)$$



**Figura 8.** Una curva de Pareto típica (línea continua) que muestra dos iteraciones del método de Newton.

El dual de este problema convexa está dada por

$$\max_{y,\lambda} L(y,\lambda) \text{ sujeto a } \lambda \geq 0 \quad (48)$$

Donde

$$L(y,\lambda) = \inf_{x,r} \{ ||r||_2 - y^T(Ax + r - b) + \lambda(||x||_1 - T) \}$$

Es la doble función de LaGrange, y el m-vector, y el escalar  $\lambda$  son los multiplicadores de LaGrange (por ejemplo, variables duales) correspondientes a las limitaciones en (47). Usamos la separabilidad del ínfimo en r y x para reorganizar términos y llegar a la equivalente declaración

$$L(y,\lambda) = b^T y - T\lambda - \sup_r \{ y^T r - ||r||_2 \} - \sup_x \{ y^T Ax - \lambda ||x||_1 \}.$$

Reconocemos las funciones anteriormente como suprema de las conjugadas de  $\|r\|_2$  y de  $\lambda\|x\|_1$ , respectivamente. Para una norma arbitraria  $\|\cdot\|$  con la norma dual  $\|\cdot\|^*$ , La función de la conjugada de  $f(x) = \alpha\|x\|$  para cualquier  $\alpha \geq 0$  viene dada por

$$f^*(y) := \sup_x \{y^T x - \alpha\|x\|\} = \begin{cases} 0 & \text{si } \|y\|^* \leq \alpha, \\ \infty & \text{de otra manera;} \end{cases} \quad (49)$$

Se deduce que (49) permanece acotado si y sólo si las variables duales  $Y$  y  $\lambda$  satisfacen las restricciones  $\|y\|_2 \leq 1$  y  $\|A^T Y\|_\infty \leq \lambda$ . El dual de (48), y por lo tanto de (LST), viene dada por

$$\max_{y, \lambda} b^T y - T\lambda \text{ sujeto a } \|y\|_2 \leq 1, \|A^T y\|_\infty \leq \lambda; \quad (50)$$

La restricción de no negatividad de  $\lambda$  se hace cumplir de manera implícita por la segunda restricción. Es importante destacar que el doble variables  $Y$  y  $\lambda$  pueden ser fácilmente calculadas a partir de la óptima soluciones primarias. Para derivar  $Y$ , primera nota de que, a partir de (44),

$$\sup_r \{y^T r - \|r\|_2\} = 0 \text{ si } \|y\|_2 \leq 1. \quad (51)$$

Por lo tanto,  $Y = r / \|r\|_2$ , y podemos, sin pérdida de generalidad tomar  $\|Y\|_2 = 1$  en (50). Para derivar  $\lambda$ , tenga en cuenta que, siempre que  $T > 0$ ,  $\lambda$  debe estar en su límite inferior, como se deduce de la restricción  $\|A^T Y\|_\infty \leq \lambda$ . Por lo tanto, tomamos  $\lambda = \|A^T Y\|_\infty$ . (Si  $r = 0$  o  $T = 0$ , la elección de  $Y$  o  $\lambda$ , respectivamente, es arbitraria.) La variable dual  $Y$  a continuación puede ser eliminado, y llegamos a las siguientes condiciones de optimización necesarias y suficientes para la solución primal-dual  $(r, x, \lambda)$  de (46):

$$Ax_T + r_T = b, \|x_T\|_1 \leq T \quad (\text{factibilidad primal}) \quad (52a)$$

$$\|A_T r_T\| \leq \lambda \|r\| \quad (\text{dual de viabilidad}) \quad (52b)$$

$$\lambda T (\|x_T\|_1 - T) = 0 \quad (\text{complementariedad}). \quad (52c)$$

### 2.5.3.3 Búsqueda de raíces.

La búsqueda de raíces se da por cada iteración del algoritmo basándose en (46) que requiere la evaluación de  $\varphi$  y  $\varphi'$  en algún  $T$ , y por lo tanto dará la minimización de  $(LS_T)$ .

Este es un subproblema potencialmente caro, y la eficacia de nuestro enfoque depende de la capacidad de resolver este subproblema sólo aproximándose. El rateo de resultados convergen para el caso en que  $\varphi$  y  $\varphi'$  son sólo aproximadamente conocidos. Esto está en contraste con el habitual análisis inexacto-Newton, que supone que se conoce a  $\varphi$  con exactitud [16] [19].

El algoritmo genera una secuencia de parámetros de regularización  $T_k \rightarrow T_\sigma$  basado en la iteración de Newton

$$T_{k+1} = T_k + \Delta T_k, \quad \text{con} \quad \Delta T_k := (\sigma - \varphi(T_k)) / \varphi'(T_k), \quad (53)$$

De tal manera que las soluciones correspondientes de  $x_{T_k} (LS_{T_k})$  convergen para  $x_\sigma$ . Para los valores de  $\sigma \in (0, \|b\|_2)$ , el Teorema 2 implica que  $\varphi$  es convexa, estrictamente decreciente, y continuamente diferenciable. En ese caso, es evidente que  $T_k \rightarrow T_\sigma$  superlinearly para todos los valores iniciales  $T_0 \in (0, T_{BP})$ .

La eficiencia de nuestro método, al igual que con muchos métodos de tipo Newton para grandes problemas, se basa en la capacidad de llevar a cabo la iteración descrito por (53) con sólo una aproximación de  $\varphi_{(T_k)}$  y  $\varphi'_{(T_k)}$ . Aunque la ecuación no lineal (46) que se desea resolver implica sólo una única variable de  $T$ , la evaluación de  $\varphi(T)$  implica la solución de  $(LS_T)$ , que puede ser un gran problema de optimización que es caro de resolver con la máxima precisión [17].

Para los sistemas de ecuaciones no lineales en general, los métodos de Newton inexactos asumen que el sistema de Newton análoga a la ecuación

$$\varphi'(\tau_k) \Delta \tau_k = \sigma - \varphi(\tau_k)$$

Se resuelve sólo aproximadamente, con un residual que es una fracción del lado derecho. Una fracción constante produce una tasa de convergencia lineal, y una fracción que tiende a cero produce una tasa de convergencia superlinear. Sin embargo, el método inexacta-Newton análisis no se aplica al caso en que el lado derecho (es decir, la función en sí) se conoce sólo aproximadamente, y es Por consiguiente, no es posible saber a priori la precisión requerida para lograr una inexacta velocidad de convergencia de tipo Newton. Esta es la situación que nos encontramos ante si  $(LS_T)$  Se resuelve aproximadamente. Como se muestra a continuación, con sólo un conocimiento aproximado del valor de la función  $\varphi$ , esta versión inexacta del método de Newton converge aún, aunque la velocidad de convergencia es sublinear. La tasa se puede hacer arbitrariamente cerca de superlinear mediante el aumento de la precisión con la que se calcula a  $\varphi$  [18].

### 2.5.3.4 Algoritmo 1: Gradiente Espectral proyectado para (LS $\tau$ ).

Entrada:  $X, \tau, \bar{\delta}$

Salida:  $x_\tau, r_\tau$

Conjunto de mínimo y el máximo paso longitudes entre  $0 < \alpha_{\min} < \alpha_{\max}$ . Establecer la longitud del paso inicial  $\alpha_0 \in [\alpha_{\min}, \alpha_{\max}]$  y los parámetros de descenso suficiente  $\gamma \in (0, 1)$ .

Establecer una línea indicada por el historial de búsqueda  $M \geq 1$ .

Establecer una iteración iniciales:  $x_0 \leftarrow P_\tau[x], r_0 \leftarrow b - Ax_0, g_0 \leftarrow -A^T r_0$ .

Empezar

```

begin
1   $\delta_\ell \leftarrow \|r_\ell\|_2 - (b^T r_\ell - \tau \|g_\ell\|_\infty) / \|r_\ell\|_2$ 
2  if  $\delta_\ell < \bar{\delta}$  then break
3   $\alpha \leftarrow \alpha_\ell$ 
   begin
4   $\bar{x} \leftarrow P_\tau[x_\ell - \alpha g_\ell]$ 
5   $\bar{r} \leftarrow b - A\bar{x}$ 
6  if  $\|\bar{r}\|_2^2 \leq \max_{j \in [0, \min\{k, M-1\}]} \|r_{\ell-j}\|_2^2 + \gamma(\bar{x} - x_\ell)^T g_\ell$ 
7  | break
   else
8  |  $\alpha \leftarrow \alpha/2$ 
   end
9   $x_{\ell+1} \leftarrow \bar{x}, r_{\ell+1} \leftarrow \bar{r}, g_{\ell+1} \leftarrow -A^T r_{\ell+1}$ 
10  $\Delta x \leftarrow x_{\ell+1} - x_\ell, \Delta g \leftarrow g_{\ell+1} - g_\ell$ 
11 if  $\Delta x^T \Delta g \leq 0$  then
12 |  $\alpha_{\ell+1} \leftarrow \alpha_{\max}$ 
   else
13 |  $\alpha_{\ell+1} \leftarrow \min\{\alpha_{\max}, \max[\alpha_{\min}, (\Delta x^T \Delta x) / (\Delta x^T \Delta g)]\}$ 
    $\ell \leftarrow \ell + 1$ 
end
return  $x_\tau \leftarrow x_\ell, r_\tau \leftarrow r_\ell$ 

```

**Paso 1** brecha de la dualidad de cómputo.

**Paso 2** sale si converge

**Paso 3** Calcular la longitud inicial paso

**Paso 4** Búsqueda de candidatos en la línea del iterador

**Paso 5** actualizar el residual correspondiente

**Paso 7** Búsqueda línea de salida

**Paso 8** disminución de la longitud de los pasos

**Paso 9** iteración de actualización

**Paso 11** Actualización de la longitud del paso Barzilai-Borwein

La aplicación SPGL1 se estructura en torno a mayores y menores iteraciones. Cada iteración principal es responsable de determinar el siguiente elemento de la secuencia de  $\{T_k\}$  y para invocar el método SPG se describe en el algoritmo 1 para determinar valores aproximados de  $\varphi(\tau_k)$  y  $\varphi'(\tau_k)$ . Para cada iteración  $k$  importante, hay un correspondiente conjunto de iteraciones menores que convergen al  $(x_{\tau_k}, r_{\tau_k})$  que componen la estructura y se repite en el Algoritmo 1. A menos que el usuario puede proporcionar una buena estimación para el  $T_\sigma$  solución de (41), el algoritmo de búsqueda de raíz elige  $T_0 = 0$  [23].

### 2.5.3.5 Gradiente proyectado para Lasso.

El gradiente espectral proyectado (SPG) es un algoritmo para resolver (LST) se perfila en el Algoritmo 1 Donde Cada iteración de este método requiere una proyección ortogonal de un vector con longitud  $n$  sobre un conjunto factible  $\{x \mid \|x\|_1 \leq T\}$ . Esto se logra a través del operador

$$P_T [c] := \{\arg \min \|c - x\|_2 \text{ sujeto a } \|x\|_1 \leq T\} \quad (54)$$

Que da la proyección de un  $n$ -vector  $c$  sobre la bola de una norma con  $T$  radio. Cada iteración del algoritmo busca en el camino del gradiente proyectado  $P_T [x_l - \alpha g]$ , donde  $g$  es el gradiente actual de la función  $\|Ax-b\|_2^2$  (Que es el cuadrado de la objetivo (LST)); consulte los pasos 4-8. Debido a que el conjunto factible es poliédrica, la proyectada ruta gradiente es lineal por tramos. El criterio utilizado en el paso 6 resultados en una nonmonotone de búsqueda de línea, lo que asegura que por lo menos cada  $M$  iteraciones resulten en una bajada suficiente en la función objetivo.

La iteración candidata inicial en el paso 4 se determina por la longitud del paso computado en los pasos 10-13. Relacionan este paso de longitud, introducido por Barzilai y Bornean, a los valores propios de la de Hesse del objetivo. (En este caso, el grupo de acción es  $A^T A$  [24].

### 2.5.3.6 Proyección de la norma uno.

Hay tres pasos potencialmente caros en el algoritmo1: los pasos 5 y 9 Calcular el vector de productos de matriz  $Ax$  y  $A^T r$ , y el paso 4 calcula la proyección  $P_T [-]$  iteración de un candidato. En esta sección damos un algoritmo para el cálculo de la proyección se define en (54).

El algoritmo tiene una etapa donde se presenta el peor de los casos que es la complejidad de  $O(n \log n)$ . Con el fin de simplificar la discusión siguiente, se supone que las entradas de la  $n$ -vector  $c$  son no negativos. Esto no conduce a ninguna



pérdida de generalidad: tenga en cuenta que si las entradas de  $c$  tenían diferentes signos, entonces sería posible sustituir el objetivo en (54) con el objetivo equivalente  $\|Dc - Dx\|_2$ , donde la matriz diagonal  $D = \text{diag}(\text{sgn}(c))$ . La verdadera solución puede entonces ser recuperado mediante la aplicación de  $D^{-1}$ .

Este algoritmo da la solución a (54) está motivado como sigue. Comenzamos con el juicio de solución  $x \leftarrow c$ . Si esto es factible para (54), y luego salimos inmediatamente con  $P_T[c] = x^* = c$ . De lo contrario, se intenta disminuir la norma del juicio por  $x$

$$v := \|x\|_1 - T, \quad (55)$$

Que es la cantidad de inviabilidad. Por lo tanto, tenemos que encontrar un vector  $d$  tal que  $\|x - d\|_1 = T$  y, con el fin de minimizar el posible aumento en el valor objetivo, elegir  $d$  de manera que  $\|d\|_2$  es mínima. Por tanto, la corrección  $d$  debe resolver

$$\text{Minimice } \|d\|_2 \quad \text{sujeto a} \quad d \geq 0, \|d\|_1 = v.$$

Es sencillo verificar que

$$d^* = \gamma e, \text{ con } \gamma = v/n$$

Es una solución de este subproblema. Sin embargo, no puede salir con  $x \leftarrow c - d^*$  si algunas de estas entradas son negativos porque al hacerlo aumenta el valor de  $\|x\|_1$ -es decir., La proyección debe preservar la firma patrón de  $c$ . Por lo tanto,

$$\text{si cada } d_i^* < c_{\min} := \min_i c_i, \text{ conjunto } x \leftarrow c - d^* \quad (56)$$

Y la salida con la solución de (55). De lo contrario, hacemos cumplir

$$x_i = 0 \quad \text{para todo } i \in I := \{i \mid d_i^* \geq c_{\min}\}, \quad (57)$$

Y después de forma recursiva repetir el proceso descrito anteriormente para las variables restantes  $\{1, \dots, N\} \setminus I$ .

Algoritmo 2 es una destilación de este procedimiento. Con el fin de hacer que sea eficiente y para reducir la sobrecarga debida a la contabilidad, se aplica el procedimiento a una secuencia de subelementos de  $c$ : la primera iteración comienza con un solo elemento que es más grande de

Algoritmo 2: Proyección posterior en el conjunto  $\{x \in \mathbb{R}^n \mid \|x\|_1 \leq T\}$ .

**Input:**  $c, \tau$   
**Output:**  $x$

```

1 if  $\|c\|_1 \leq \tau$  then return  $c$ 
2  $\gamma \leftarrow 0, \delta \leftarrow 0, \nu \leftarrow -\tau$ 
3  $\bar{c} \leftarrow \text{BuildHeap}(|c|)$ 
4 for  $j \leftarrow 1$  to  $n$  do
5    $c_{\min} \leftarrow \bar{c}[1]$ 
6    $\nu \leftarrow \nu + c_{\min}$ 
7    $\gamma \leftarrow \nu/j$ 
8   if  $\gamma \geq c_{\min}$  then break
9    $\bar{c} \leftarrow \text{DeleteMax}(\bar{c})$ 
10   $\delta \leftarrow \gamma$ 
11  $x \leftarrow \text{SoftThreshold}(c, \delta)$ 
    return  $x$ 

```

**Paso 1** salida rápida si  $c$  es factible

**Paso 2** inicialización

**Paso 3**  $\bar{c}$  es una copia del archivo del valor absoluto de  $c$

**Paso 5** se extrae el siguiente elemento más grande de  $c$

**Paso 6** se acumula la inviabilidad; ver (55)

**Paso 7** se define la solución actual de (55); ver (56)

**Paso 8** iteraciones restantes responden todos a (56)

**Paso 9** eliminar el máximo de la pila y volver a heapify

**Paso 10** Grabar la última solución de (56)

**Paso 11** suave-umbral del vector de entrada  $c$ ; ver (57)

La magnitud, y cada iteración posterior añaden un elemento más que se encuentra al lado más grande en magnitud. Este enfoque evita tener que ordenar todo el vector  $c$ . En su lugar, el paso 3 construye una estructura binomial en que el primer elemento de la pila es mayor en magnitud. El costo de BuildHeap es  $O(n)$ , donde  $n$  es la longitud del vector  $c$ . El costo en posteriores iteraciones está dominado por el paso 9, donde la función DeleteMax elimina el elemento más grande actual de la pila y restaura la propiedad del montículo. El escalar  $C_{\min}$ , ajustado en el paso 5, es el elemento más pequeño del subvector actual (En la iteración  $j$ ,  $C_{\min}$  corresponde al elemento del vector general  $c$ , que es mayor en magnitud  $j$ .) Si el paso 8 la pruebas es verdadera, entonces el algoritmo puede salir del bucle de iteración ya que todos los restantes deben satisfacer (57). La etapa de conclusión utiliza las reglas (56) y (57) para generar la solución final. Esta se lleva a cabo en el paso 11 mediante la aplicación de la operación suave de umbral

$$x \leftarrow \text{SoftThreshold}(c, \delta) \iff x_i \leftarrow \text{sgn}(c_i) \cdot \max\{0, |c_i| - \delta\} \quad (58)$$

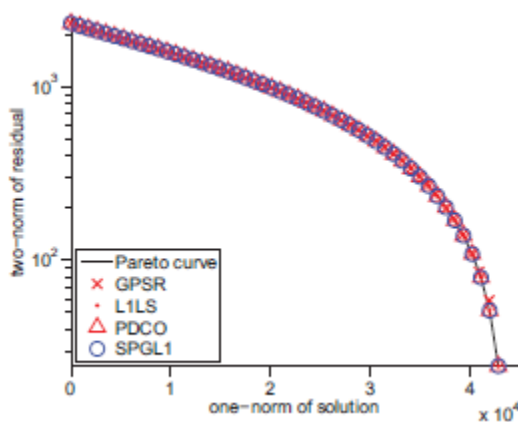
Componente a componente para el vector original  $c$ ; el escalar  $\delta$  se obtiene en el paso 10. Esta operación amortigua elementos que son más grandes en magnitud que  $\delta$ , y establece a cero cualquier elemento que son de una magnitud menor  $\delta$ . En el peor de los casos, el algoritmo 2 procederá de las  $n$  iteraciones completos, y el coste dominante es  $n$  llamadas a DeleteMax. Por consiguiente, el coste global del algoritmo es  $O(n \log n)$  en el peor de los casos. Tenga en cuenta que el operador suave umbral que acabamos de definir produce la solución del problema de optimización convexa separable

$$\min_x \frac{1}{2} \|x - c\|_2^2 + \delta \|x\|_1 \quad (59)$$

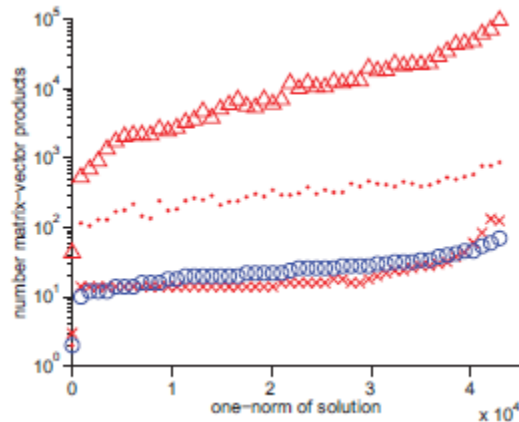
En este sentido, el algoritmo 2 puede ser interpretado como un procedimiento para encontrar la variable dual óptima  $\delta$  asociada a la El confinamiento en (55).

### 2.4.3.7 Programas para Lasso y cuadráticos.

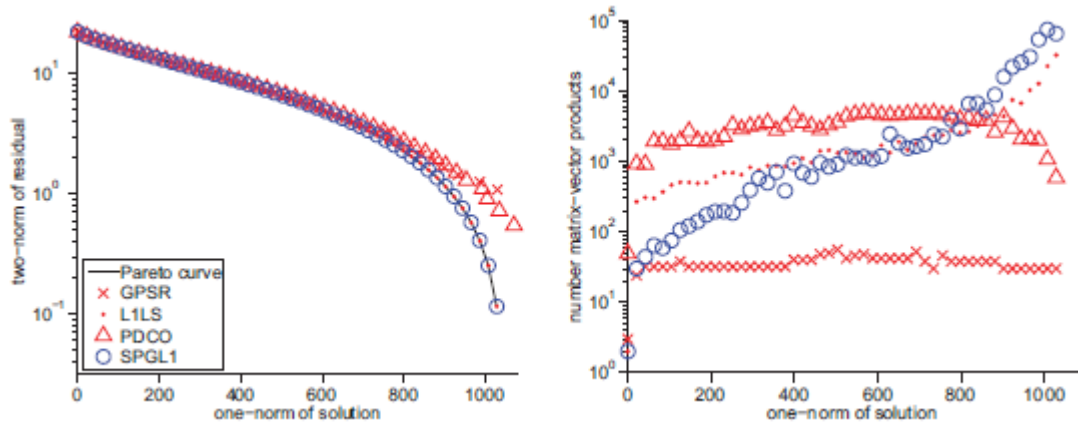
El objetivo principal de este trabajo es proporcionar un algoritmo eficiente para  $(BP\sigma)$ . Sin embargo, para el interés, se muestra aquí cómo SPGL1 también se puede utilizar para resolver de manera eficiente una sola instancia del problema Lasso  $(LST)$ . Esta corresponde a la solución de un solo caso de  $(QP\lambda)$ , donde  $\lambda$  se establece en el multiplicador LaGrange de la restricción Lasso.



A) dcthdr: calidad de la solución



B) dcthdr: rendimiento del solucionador.



C) srcsep1: calidad de la solución

D) srsecp1: rendimiento del solucionador

**Figura 9.** Se muestra el rendimiento y la calidad de los solucionadores equivalente a problemas  $(QP\lambda)$  y  $(LS_T)$ . Las gráficas superior es para el problema dcthdr, y la gráfica inferior es para problema srcsep1. Las gráficas de la izquierda muestran las normas calculadas de los residuales y las soluciones para cincuenta valores de los parámetros; las grafica de la derecha muestra el cálculo correspondiente que es el número de productos matriz-vector requerido para la solución del problema. Tenga en cuenta que las curvas de la izquierda cifras no son convexas, ya que se representan en una escala semilogarítmica.

En la Figura 14 se comparan la calidad de soluciones y el rendimiento computado de los solucionadores GPSR, L1LS, PDCO, y SPGL1 aplican a los problemas dcthdr y srcsep1. Para cada uno de estos dos problemas, generamos cincuenta valores de  $\lambda$  y  $T$  para los que las soluciones de  $(QP\lambda)$  y  $(LS_T)$  coinciden, y por lo tanto generan cincuenta casos de prueba para cada uno de los tipos  $(QP\lambda)$  y  $(LS_T)$ . Aplicamos GPSR, L1LS, y PDCO para cada caso de prueba  $(QP\lambda)$  y aplicar SPGL1 para cada caso de prueba  $(LS_T)$ .

Los puntos de la curva de Pareto son soluciones precisas; soluciones imprecisas se encuentran por encima de la curva, lo que indica que no resuelven el problema correspondiente  $(QP\lambda)$  o  $(LS_T)$ .

En la figura 14 (a), casi todos los puntos se encuentran en la curva de Pareto, y por lo tanto la mayor parte de las soluciones calculadas son exactos. Los puntos correspondientes en la figura 14 (b) indican que SPGL1 y el uso GPSR sólo una pequeña fracción de los productos matriz-vector requerido por PDCO y L1LS, que son métodos de segundo orden. Observamos que las soluciones para problema dcthdr puede extenderse más de cuatro órdenes de magnitud.

En la figura 14 (c), las soluciones calculadas devueltos por PDCO y GPSR son progresivamente peor a medida que  $\lambda$  tiende a cero ( en la solución el de una sola norma aumenta). Curiosamente, L1LS produce constantemente soluciones muy precisa de todos los valores de  $\lambda$ ; cómo nunca, como podría esperarse de un

método de punto interior basado en un gradiente conjugado solver lineal, se puede requerir muchos productos matriz-vector.

## 2.6. Teledetección Radar.

Es una parcela de la teledetección en la cual se busca obtener información de los objetos a partir de las ondas electromagnéticas emitidas o reflejadas en la bandas de las microondas. Estas bandas están situadas en el espectro electromagnético aproximadamente entre las longitudes de ondas de 1 mm y 1m

Para obtener información con las microondas se puede utilizar sensores pasivos (recogen la radiación emitida por los objetos) o sensores activos que emiten la radiación, la dirigen al objeto y recogen los ecos a estos sensores activos se les denomina **radares formadores de imágenes** [11].

### 2.6.1. Fundamento de la obtención de la imagen radar.

La formación de la imagen mediante un radar se basa en iluminar con microondas una porción de la superficie terrestre desde un avión o un satélite. Los objetos situados en la superficie terrestre llamados blancos devuelven unos ecos o señales, el sensor radar recogerá estas señales y las grabara, a continuación se procesaran estos ecos y se formara con el mismo una imagen digital.

En la figura se puede ver un esquema, con la toma de la imagen que se ha iluminado varias veces y sucesivamente una porción de la superficie terrestre [19].

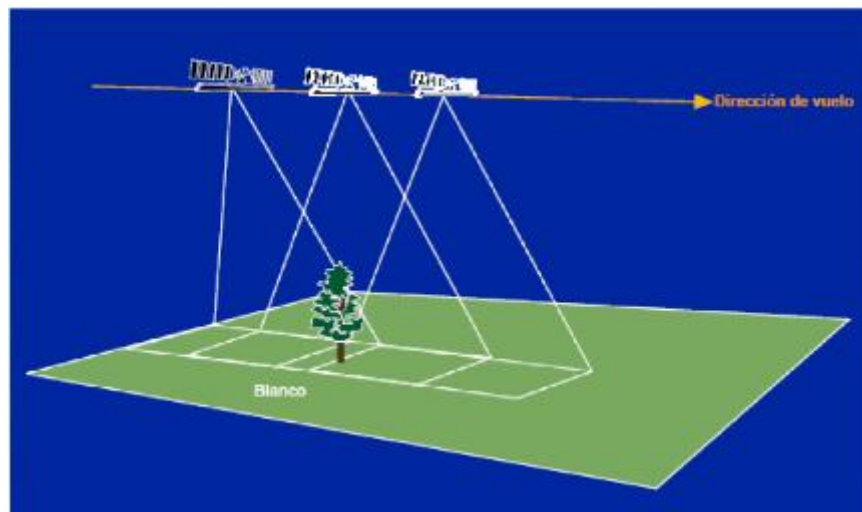


Figura 10 .Toma de una imagen radar

## **2.6.2. Resolución de las imágenes de radar.**

El número de píxeles que integran un sensor de radar o de cámara digital, definen su poder de resolución. Es decir la capacidad de discernir objetos o detalles de un determinado tamaño en las imágenes captadas. A mayor número de píxeles por unidad de superficie, mayor resolución de la foto sensor, **pero también mayor es el volumen del archivo informático generado**. Es lo que se denomina **resolución espacial** [19].

### **2.6.2.1. Resolución espacial.**

Este concepto designa al objeto más pequeño que se puede distinguir en la imagen. Está determinada por el tamaño del píxel, medido en metros sobre el terreno, esto depende de la altura del sensor con respecto a la Tierra, el ángulo de visión, la velocidad de escaneado y las características ópticas del sensor.

#### **2.6.2.1.1. Aplicación de las imágenes de alta resolución según su resolución espacial.**

Las imágenes de radar se han convertido en los últimos años en una herramienta de uso frecuente para la extracción de información para el estudio de la superficie terrestre. A continuación se mencionan las aplicaciones según su resolución.

##### **2.6.2.1.2. Resolución de 1 a 5 metros**

Son utilizadas para las siguientes funciones:

- Distinción de características de objetos tamaño de coche
- Estudio de la vegetación de pequeñas parcelas agrícolas, maduración o enfermedades de las plantas.
- Diferenciación de tipos de edificios
- Modelos digitales de elevación (MDT)

##### **2.6.2.1.3 Resolución de 20-60 metros**

Son utilizadas para las siguientes funciones:

- Localización de grandes infraestructuras (aeropuertos, ciudades, instalaciones deportivas).
- Cartografía de edificios, carreteras, terrenos agrícolas.
- Estudio de la vegetación en parcelas de extensión media

#### **2.6.2.1.4. Resolución de 100-200 metros**

Son utilizadas para las siguientes funciones:

- Cartografía de estructuras geológicas regionales.
- Salud fitosanitaria de zonas extensas.

#### **2.6.2.1.5. Resolución de 1000-2500 metros**

Son utilizadas para las siguientes funciones:

- Análisis de la vegetación regional a nivel continental.
- Estudios medio ambientales y meteorológicos

Debido a la gran cantidad de información que genera la obtención de una imagen de radar, el tiempo de reconstrucción de las imágenes de radar depende de la resolución espacial que va desde semana e incluso de meses.

## **2.7 La interfaz de usuario**

### **2.7.1 Tipos y criterios de diseño.**

La interfaz de usuario, en adelante IU, es el conjunto de elementos hardware y software de un ordenador que proporcionan información al usuario y le permiten interactuar con él.

Las funciones del IU son:

- Presentación de datos al usuario (Salida de información)
- Obtención de datos del usuario (Entrada de información)

### **2.7.2 Tipos de interfaz**

En un sistema de información, podemos encontrar dos tipos de interfaces:

**Interfaces genéricos:** Son los interfaces usados en la mayoría de aplicaciones, y son válidos para la mayoría de funciones y de usuarios.

**Interfaces específicos:** Son los interfaces usados de forma más minoritaria, ya sea porque están eficaces en unas aplicaciones específicas, para un grupo reducido de usuarios o son interfaces desarrollados recientemente.

### **2.7.3 Criterios de diseño**

El proceso de diseño de una IU guarda similitud al proceso de diseño de otros componentes software. Su objetivo es que la aplicación desarrollada tenga el mayor grado de “usabilidad” posible. Se entiende como usabilidad el grado de eficacia y satisfacción de la aplicación teniendo en cuenta:

1. los requisitos funcionales de la aplicación
2. el tipo o tipos específicos de usuarios que tendrá
3. el contexto de la comunicación usuario-aplicación

El desarrollo de los IUs se distingue del desarrollo de otros componentes software en:

1. Debe estar especialmente centrado en los usuarios. Según el tipo de usuario, variará el diseño de la interfaz. No existe un interfaz válido para todos los usuarios y todas las tareas.
2. Tiene un alto componente cognitivo y psicológico. Los criterios de diseño han de tener en cuenta como son los procesos cognitivos de las personas.
3. Para validar los interfaces es necesario la ayuda de los usuarios finales. No basta con los métodos de prueba de otros procesos de diseño.

### **2.7.4 Objetivos en el diseño de IU.**

Los objetivos a tener en cuenta en el diseño de IU son:

1. Efectividad: El diseño de la interfaz debe permitir al usuario acceder a todas las funcionalidades de la aplicación. No deben darse casos en que haya interfaces con funcionalidad limitada, por ejemplo los programas que permiten más acciones cuando se invocan mediante el interfaz de comandos que por su interfaz gráfico.
2. Eficiencia: En el caso de dispositivos de pocas prestaciones, el diseño del IU tiene que buscar minimizar el consumo de recursos por el IU. Aunque en la mayoría de los casos, el recurso a consumir eficientemente es el tiempo de de los usuarios.
3. Seguridad: El interfaz de usuario debe tener en cuenta los posibles errores de los usuarios y las incidencias que puedan surgir durante su ejecución. Por ejemplo, el típico mensaje de aviso cuando se quiere cerrar una aplicación sin haber guardado el trabajo realizado.
4. Facilidad en el uso: El diseño del interfaz debe reducir el esfuerzo realizado por los usuarios. Se tiene que tener en cuenta las capacidades y experiencia de cada tipo de usuarios. Por ejemplo, un interfaz en 3D puede hacer muy difícil su uso por parte de usuarios con una baja capacidad de orientación espacial.



5. **Facilidad en el aprendizaje:** El objetivo de cualquier IU es que sus usuarios consigan dominarlo lo más rápidamente posible. Para ello se tiene en cuenta la experiencia del usuario y se añaden componentes al mismo como ayuda en línea, tutoriales.

6. **Flexibilidad:** La IU debe permitir un uso satisfactorio por parte de diferentes tipos de usuario. Por ejemplo, permitir diferentes vías de interacción: mediante el teclado, uso de barra de botones, menús contextuales.

Todos estos objetivos buscan una sola cosa: maximizar la productividad del usuario. Es decir, que tarde menos en hacer el mismo trabajo o, visto al revés, que en el mismo tiempo haga más trabajo [29].

### **2.7.5 Principios de diseño de interfaces de usuario.**

Existe un factor común al diseño de todos los IU, que es las características comunes del proceso cognitivo de las personas, es decir, como piensan y actúan generalmente las personas al interactuar con un equipo informático. Es por ello, que existen una serie de principios aplicables en el diseño de todas las IU. Estos principios son:

1. **Anticipación:** El interfaz, en la medida de lo posible, debe anticiparse a las necesidades del usuario. Por ejemplo, el interfaz de muchos editores de código muestra la línea de texto donde se encuentra el cursor, evitando que el usuario tenga que llamar a una opción de menú para saberlo.

2. **Autonomía:** El usuario debe controlar la aplicación y no al contrario. El usuario debe disponer de indicadores del estado del sistema para estar informado y poder mantener el control de sus acciones. Es conveniente que el usuario pueda deshacer sus propias acciones, en caso de error.

4. **Consistencia del interfaz:** El comportamiento del interfaz debe ser coherente con lo esperado con el usuario, de forma que facilite su aprendizaje por los usuarios y reduzca los errores humanos. Existen tres niveles de consistencia:

a. **Consistencia interna:** Las decisiones de diseño en una IU deben ser las mismas en todos sus componentes. Por ejemplo, en las ventanas de una aplicación, los botones deben estar situados en un mismo sitio, para que el usuario no tenga que ir buscándolos en cada ventana.

b. **Con el entorno:** Es conveniente que la IU de una aplicación sea coherente con las interfaces de otras aplicaciones y del SO donde se ejecuta, independientemente del fabricante. De forma que lo aprendido para una aplicación valga para las otras..

c. **Con el mundo real:** Los conceptos usados por el IU no deben contradecirse con los conceptos del mundo real.

6. Ergonomía: EL diseño de la IU debe proporcionar a los usuarios un entorno adecuado a su uso. El principal problema de ergonomía en los IU es la legibilidad de la información que se muestra. La información debe ser fácil de ubicar y leer. Para ello debe utilizarse texto con alto contraste, combinaciones de colores que no cansen la vista (fondo oscuro con letras claras o viceversa), uso moderado de colores intensos, tamaños de las fuentes de texto acordes al tamaño de los monitores que se usarán los usuarios.

7. Tiempos de respuesta: Las aplicaciones deben dar respuesta a las acciones del usuario en menos de dos segundos para conseguir la concentración del usuario en la aplicación. En caso de tiempos de respuesta largos, hay que señalarlo e informar del estado de la transacción [29].

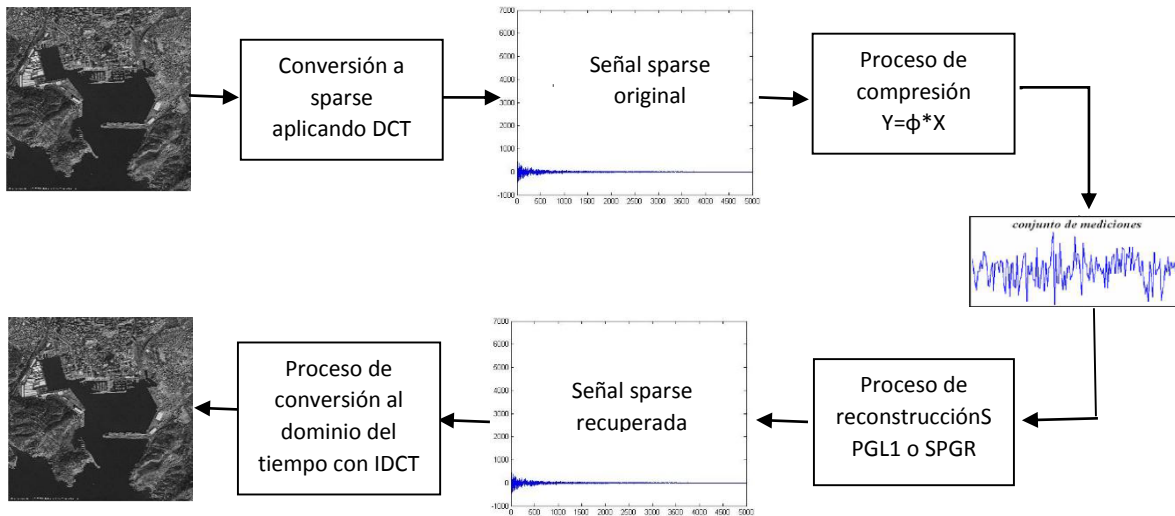
# Capítulo 3

## Metodología

En este capítulo se describe la metodología que se llevó a cabo durante la ejecución de este proyecto de grado.

En primer lugar se hace el proceso de selección de las imágenes de radar en alta resolución las cuales tendrán dimensiones de gran tamaño seguidamente se origina la conversión a dominio sparse, finalmente se emplea el proceso de compresión con la técnica Compressive Sensing (CS) a (diferentes porcentajes 10%,25%,50% y 75%).

En este paso se define la parte en la cual se contempla el proceso de reconstrucción de la imagen a partir de dos archivos (el vector CS y la matriz aleatoria) los cuales van hacer datos de entrada para los solucionador (SPGR Y SPGL1). Después de la reconstrucción de la señal sparse, se pasara al dominio del tiempo con la función IDCT para poder finalmente iniciar a validar el sistema y contemplar la calidad de la reconstrucción.



**Figura 11 .** diagrama de bloque de la metodología

A continuación se describe cada proceso:

### 3.1 Proceso de selección de las imágenes de radar en alta resolución.

Para el inicio de este proyecto de investigación se seleccionaron cinco (5) imágenes de radar de alta resolución las cuales tienen grandes dimensiones y resolución espacial de 20-60 metros, son las más apropiadas para nuestra aplicación y análisis, estas imágenes tienen objetos que podemos localizar fácilmente como (ciudades, instalaciones deportivas, edificios, carreteras, terrenos agrícolas y vegetación en parcelas de extensión media) esto para comparar en la reconstrucción de la imagen la calidad, brillo, forma, visibilidad de dichos objetos.



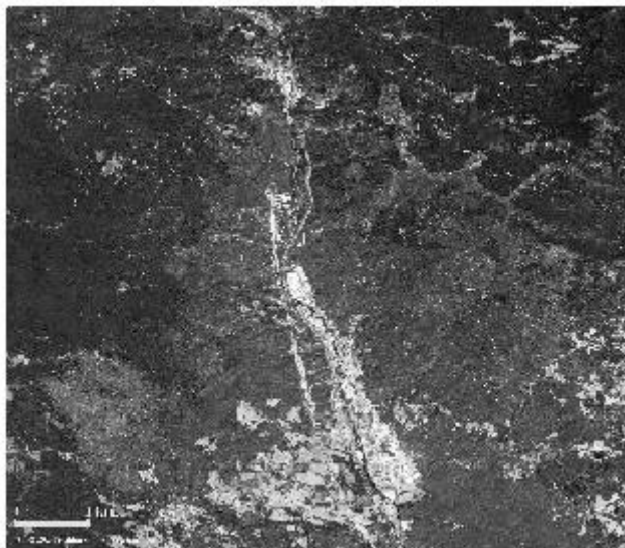
**Figura 12.** Ciudad prohibida, Satélites Pléiades. [10]

**Detalle de la imagen**  
**Localización:** Pekín, China  
**Resolución:** 50 cm  
**Dimensiones:** 5000x5000



**Figura 13.** Cartagena España, Satélites Pléiades [10]

**Detalle de la imagen**  
**Localización:** Cartagena, España  
**Resolución:** 50 cm  
**Dimensiones:** 5000x5009



**Figura 14.** Incendio de la jonquera, Satélites Pléiades [10].

**Detalle de la imagen**

**Localización:** La Jonquera, Empurdà, Catalunya, España

**Resolución:** 50cm

**Dimensiones:** 5000x5000



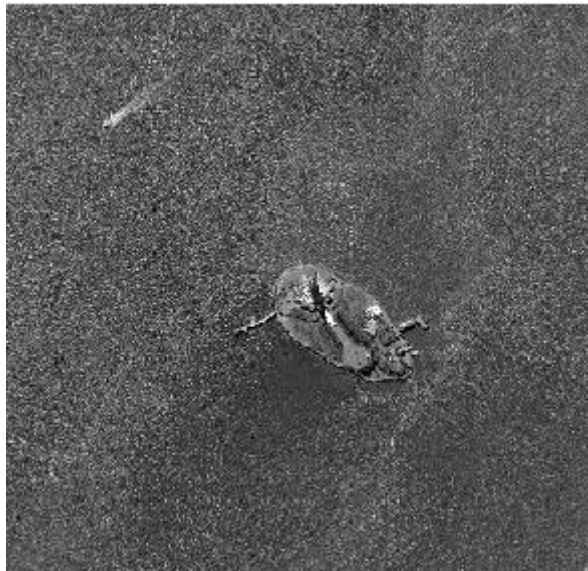
**Figura 15.** Bora Bora, Satélites Pléiades [10].

**Detalle de la imagen**

**Localización:** Bora Bora, Polinesia Francesa

**Resolución:** 50 cm

**Dimensiones:** 4500x5000



**Figura 16.** Estatua de la Libertad [10].

**Detalle de la imagen**  
**Localización:** Estados Unidos  
**Fuente:** Pléiades  
**Resolución:** 50cm  
**Dimensiones:** 4500x5000

Fue necesario trabajar en tonos de grises para que se facilite el procesamiento de reconstrucción de las imágenes ya que los equipos disponibles no cuentan con características de procesamiento necesarias para un óptimo rendimiento.

### **3.2. Proceso de compresión de la señales de radar.**

En esta etapa se realizó la compresión de la imagen de radar utilizando la técnica COMPRESSIVE SENSING (CS). Para esto se hace necesario definir lo siguiente:

#### **3.2.1 Conversión de la señal o imagen de radar a Sparse.**

COMPRESSIVE SENSING (CS) requiere que las señales a tratar se encuentren bajo un dominio Sparse. Debido a que la imagen se encuentra en un dominio del tiempo esta imagen primeramente es vectorizada, luego se hace necesario llevar a cabo una conversión de esta señal  $X$  a dominio Sparse, como se observa figura 17.

Para concebir esta conversión se seleccionaron las dos herramientas más utilizadas llamadas:

- La Transformada discreta del coseno (DCT).
- Transformada discreta de Wavelet (DWT)

La Transformada discreta del coseno (DCT).

Está compuesta de una secuencia de  $u [m, n]$ ,  $0 \leq N-1$  matemáticamente se define así:

$$(60) \quad V [k, l] = \alpha (k) \alpha (l) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} u [m, n] \cos\left[\frac{(2m+1)k\pi}{2N}\right] \cos\left[\frac{(2m+1)l\pi}{2N}\right]$$

Donde el coeficiente  $\alpha (\psi)$  ( $\psi=k, l$ ) toman valores

$$\alpha (\psi) = \begin{cases} \sqrt{\frac{1}{N}} & \psi = 0 \\ \sqrt{\frac{2}{N}} & \psi = 1, \dots, N - 1 \end{cases}$$

En cuanto a la Transformada discreta de Wavelet (DWT)

Matemáticamente se define de la siguiente manera:

$$(61) \quad x(t) = \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} (W_{\psi} x)_{j,k} \Psi_{j,k}(t)$$

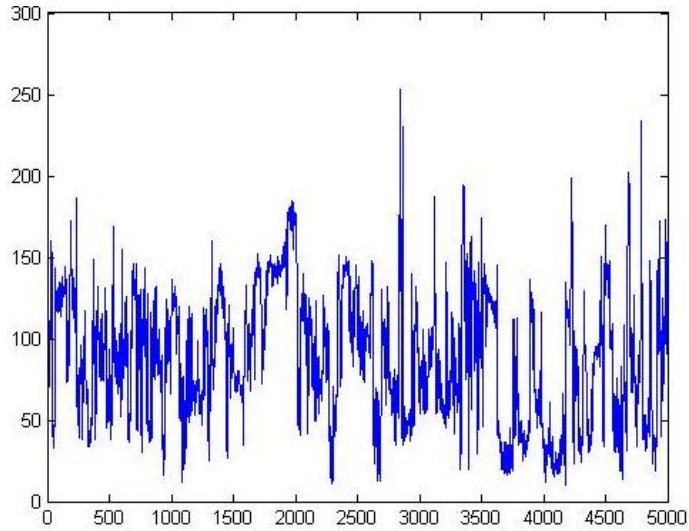
Donde

$$\Psi_{j,k}(t) = \frac{1}{\sqrt{a_0^j}} \Psi(a_0^{-j} t - kb_0)$$

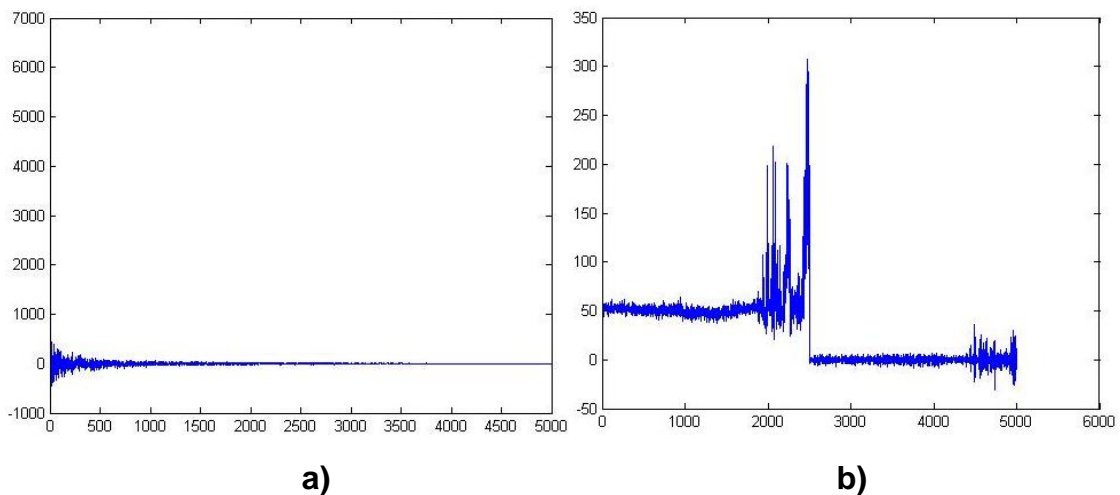
Es el conjunto de funciones wavelet.

Estas herramientas se seleccionaron porque nos permite pasar del dominio del tiempo al dominio sparse, el cual necesitamos para nuestro estudio. Y la segunda característica por la que escogió, es porque pertenecen al conjunto de funciones definidas en los paquetes de MatLab.





**Figura 17.** Representa la señal de una imagen en el dominio del tiempo



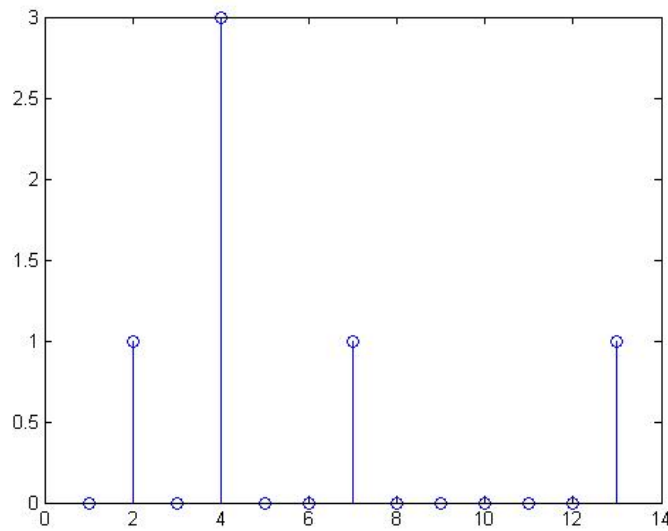
**Figura 18.** Representa dos señales Sparse de radar calculada con a) DCT y b) DWT

### 3.2.2 Nivel De Esparcidad.

Para calcular el nivel de esparcidad a cualquier señal  $x$  sparse se aplica la norma  $l_1$  representada en la ecuación (62), esta norma funciona sumando las magnitudes de todos los coeficientes diferentes de cero, como ejemplo ilustrativo se observa en la figura 14 una señal sparse de longitud de trece puntos los cuales cuatro son diferentes de cero con magnitudes de (1, 3, 1 y 1) al sumar estas magnitudes su resultado de esparcidad es igual a seis.

(62)

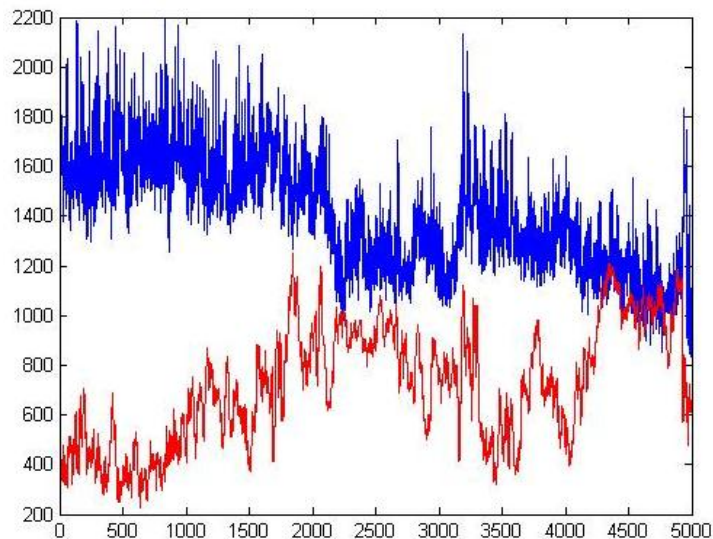
$$\|x\|_1 = \sum_{i=1}^n |x_i|$$



**Figura 19.** Nivel sparse de la señal igual a seis

Después que las imágenes fueron convertidas a señales sparse con las herramientas DCT Y DWT, se tomó cada conversión y se le aplico la norma l1 siendo DCT la más adecuada para aplicar compressed sensing debido a que la esparsidad fue menor algo que resulta muy conveniente para este estudio.

La figura 15 muestra que al aplicarle la norma l1 a la señal sparse de la imagen 1 las amplitudes en la base de DWT (color azul) son mayores que los de DCT (color rojo). Arrojando valores para DWT = 7.0165e+006 y DCT= 3.6004e+006.



**Figura 20.** Nivel de esparsidad DCT vs DWT.

### 3.2.3. Creación de la matriz aleatoria de medición.

Uno de los requisitos para llevar a cabo la compresión de la señal Sparse tiene que ver con el hecho de generar una matriz aleatoria  $\phi$  de dimensiones  $m \times n$ . Para ello, se recurrió a una función definida en MatLab llamada **rand (m, n)**, donde  $m$  representa el número de filas y  $n$  el número de columnas.

El valor  $n$  (cantidad de columnas) de la matriz aleatoria debe ser igual al tamaño de la longitud de la señal Sparse, y el valor  $m$  (cantidad de las filas) de la matriz aleatoria depende de la siguiente analogía, donde  $m=n/2$  corresponde al 50% de compresión, y  $m=n/4$  que corresponde al 75% de compresión, constituyó en el valor mínimo para trabajar estas señales sparse de  $n$  el valor máximo,

Para calcular el valor optimo  $m$  de compresión para la matriz de medición, la ecuación (19)  $m \geq \mu^2(\phi, \Psi) \cdot k \cdot \log n$  fundamentara este resultado.

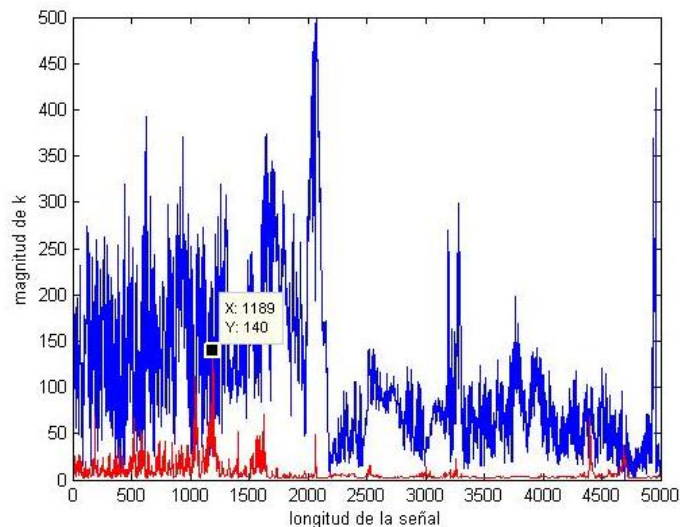
Para la figura 12,  $n=5000$  el mínimo de  $m$  seria  $m = 5000/4 = 1250$  este número corresponde a el número mínimo de filas que tendrá la matriz  $\phi$  de medición, este valor de  $m$  satisface a la desigualdad de la ecuación (19).

$\mu^2(\phi, \Psi) = 1$  , como se definió en la sección (2.1.1.3. Muestreo Incoherente).

En la figura 21, ilustra los valores de los  $k$  coeficientes distintos de cero que pertenece a las conversiones de DWT (color azul) y DCT (color rojo) para cada vector  $x$  sparse que conforman la longitud de la figura 12, se observa que los coeficientes que pertenecen a la DCT se ajustan con mayor fidelidad a la desigualdad de la ecuación (19), se tomara el  $k=140$  por ser el pico de mayor amplitud en DCT y satisface la desigualdad, se sustituye  $k, n$  y  $m$  en la ecuación (19) obtenemos.

$$1250 \geq 1 \cdot (140) \cdot \log(5000)$$

$$1250 \geq 1192$$



### Figura 21. magnitud de k en DCT y DWT.

Se decidió trabajar con  $n/2$  por motivos de calidad y la tasa de datos necesarios para una buena reconstrucción donde  $m=n/2$ .

Por ultimo hay que aplicar la ecuación (21)  $(1 - \delta k)\|x\|_2^2 \leq \|\phi x\|_2^2 \leq (1 + \delta k)\|x\|_2^2$  a la matriz de medición que expresa **a la propiedad de isometría restringida** que tiene como objetivo demostrar que tan buena es la matriz codificando la información del mayor conjunto posible de señales para una reconstrucción exitosa.

La figura 22 a) muestra como la desigualdad de la ecuación (21) se cumple para la matriz de medición y los vectores sparse de la figura 12. La señal verde corresponde a la parte derecha, la señal roja corresponde a la parte del centro y la señal azul corresponde a la parte izquierda de la desigualdad, b) muestra un zoom a la señal para poder apreciar la desigualdad de la RIP.

Donde  $\delta k$  son constantes que están en el intervalo  $0 < \delta k < 1$  para este estudio las constantes escogidas están en el intervalo  $[0,8 \ 0,9]$  las cuales satisfacen a la desigualdad.

$x$  es el vector sparse y  $\phi$  es la matriz de medición.

Se aplicó a la figura 12 la RIP y el resultado obtenido fue el siguiente.

$$(1 - 0.8)\|x\|_2^2 \leq \|\phi x\|_2^2 \leq (1 + 0.8)\|x\|_2^2$$

$$2.0973e+003 \leq 2.6135e+003 \leq 1.8876e+004$$

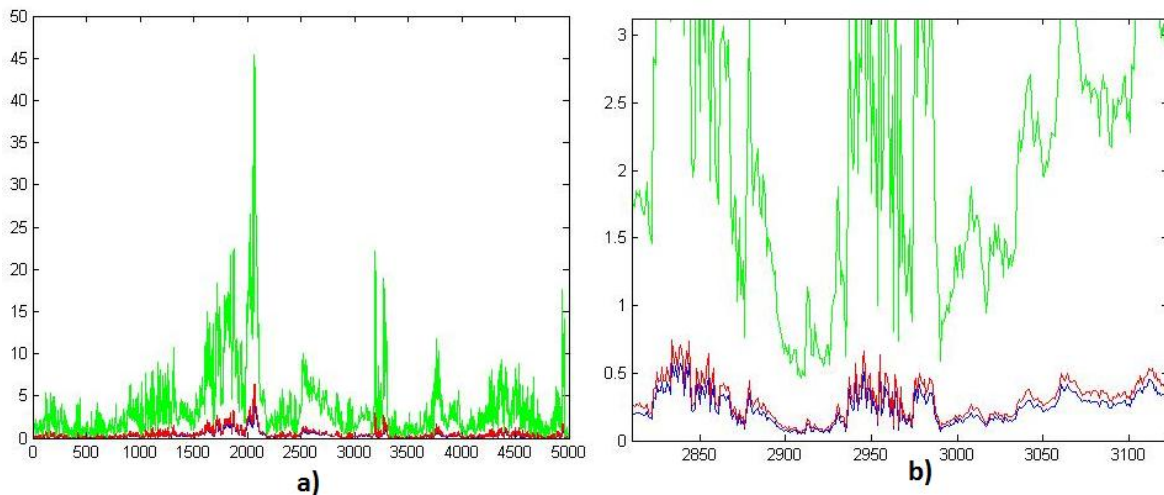


Figura 22. a) Representa la RIP, b) representa un zoom a la RIP.

### 3.2.4. Cálculo de la señal o vector comprimido de la imagen de radar en (CS).

Una vez se tiene la Matriz Aleatoria  $\phi$  de dimensiones  $m \times n$  y la señal calculada en un dominio Sparse de longitud  $n$ , se realiza el cálculo de la señal comprimida con Compressive Sensing (CS).

Según la teoría CS, el cálculo de la señal o vector  $\mathbf{b}$  comprimido, se define como la sumatoria del producto de cada punto correspondiente a las filas columnas  $\mathbf{n}$  de la Matriz Aleatoria, con cada uno de los puntos del vector de la señal  $\mathbf{X}$  Sparse de longitud  $\mathbf{n}$ . Cada sumatoria pasará a componer un valor de la señal o vector  $\mathbf{b}$  comprimido en CS. De esta forma, el proceso se repite para cada una de las filas de la Matriz Aleatoria. Esto significa que la longitud del vector  $\mathbf{b}$  calculado en CS va a ser igual a la cantidad de filas de la Matriz Aleatoria.

De acuerdo a lo anterior, el valor de las filas  $m$  de la Matriz Aleatoria  $\phi$  es la variable más importante en este proceso el cual constituye el grado de compresión de la señal Sparse, pues este depende de la cantidad de filas que tiene la Matriz Aleatoria.

Los valores de  $m$  son calculados a partir de porcentajes del tamaño de  $n$  de la siguiente manera,  $m = n \cdot \text{porcentaje}$ , donde se hicieron pruebas con porcentajes de 75%, 50%, 25% y 10%.

En el cálculo del vector de compresión CS para un  $m$  de 100 %, la reconstrucción es efectiva, por que recupera la señal de forma exacta, pero el vector calculado en CS no tuvo ninguna compresión, debido a que no hubo una reducción de puntos. Es decir que la longitud  $m$  de la señal calculada en CS es igual a la longitud  $n$  de la señal a comprimir (sparse).

Para otros porcentajes, se realiza compresión o reducción de puntos, por ejemplo, para un porcentaje de 25%, se produce la mayor compresión de puntos en este trabajo, la reconstrucción no es de buena calidad, debido a que se envían muy pocos puntos y se puede perder información importante.

De esta forma se probó la técnica de compresión CS usando la señal sparse a partir de la Transformada Discreta del Coseno (DCT), para cada porcentaje descrito anteriormente.

### 3.3. Reconstrucción de la imagen sparse de radar usando el algoritmo SPGL1.

Atendiendo a los fundamentos teóricos se conoce que SPGL1 es un software novedoso que da solución a problemas escasos a gran escala, SPGL1 basa en la norma  $l_1$  de mínimos cuadrados regularizados, para nuestro caso el problema a solucionar es:

- **basis pursuit.**

$$(BP) \quad \text{minimice } \|X\|_1 \text{ sujeto a } AX = b$$

Ya que se ajusta al problema de escasos de las imágenes de radar en alta resolución

Donde  $X$  es mi función objetivo o señal sparse la cual se minimizara sujeto a la restricción  $AX = b$ , donde  $A$  será la matriz aleatoria y  $b$  es mi señal escasa comprimida.

El solucionador SPGL1 recibirá a  $b$  y  $A$ , para buscar en ellos mediante la técnica de iteración del gradiente espectral proyectado el conjunto óptimo para así dar una solución acertada.

### **3.4. Reconstrucción de la imagen sparse de radar usando el algoritmo GPSR.**

GPSR es un algoritmo capaz de dar solución a señales escasas que están en un sistema lineal de ecuaciones indeterminado. (A partir de matriz aleatoria  $A$  y el vector  $b$  comprimido con CS).

Se refiere a un conjunto de muchas soluciones, estos problemas o ecuaciones no siempre pueden ser resueltos directamente con la información dada, por eso se hace necesario algoritmos o artilugios matemáticos que busquen dichas soluciones, como lo es el gradiente proyectado el cual toma una función objetivo o una señal sparse donde encuentran mínimos local para trabajar sobre ellos a lo largo del gradiente negativo y dar así las solución más apropiada.

### **3.5. Conversión de señal Sparse a dominio del tiempo.**

Una vez la señal en CS ha sido reconstruida se obtiene una señal en un dominio Sparse, es por ello que es necesario la conversión de un dominio Sparse al dominio del tiempo, y de esta manera recuperar la señal original para luego iniciar validar el sistema, la forma en que se hizo esta conversión, Fue utilizando la función **IDCT** definida en los paquetes de MatLab, donde esta representa la transformada inversa discreta del coseno.

### **3.6 Porcentaje de error generado por los algoritmos de reconstrucción SPGL1 Y GPSR.**

Por último se calcula el porcentaje de error obtenido de cada imagen reconstruida a partir de los algoritmos. Definido por la siguiente ecuación.

(63) 
$$\text{Error} = \frac{[\sum_{i=0}^n (y_i^2 - r_i^2)]}{sum} * 100\%$$

Donde **y** es la imagen original, **r** es la imagen reconstruida **sum** es la sumatoria al cuadrado de cada punto de la imagen original y **n** es la cantidad de puntos de las imágenes.

# Capítulo 4

## RESULTADOS

Con el fin de lograr los objetivos planteados al inicio de esta tesis en este capítulo se muestran los resultados obtenidos en las diferentes etapas planteadas y ejecutadas en este proyecto.

Se llevará a cabo un estudio comparativo usando varios criterios que son importantes a la hora de decidir cuál de los algoritmos de reconstrucción es el más adecuado para este estudio. En primero lugar se compara el error generado por cada método de reconstrucción con respecto a la imagen original, para determinar que algoritmo es más eficaz en cuanto al tiempo y calidad de reconstrucción.



#### 4.1 Proceso de selección de las imágenes de radar en alta resolución.

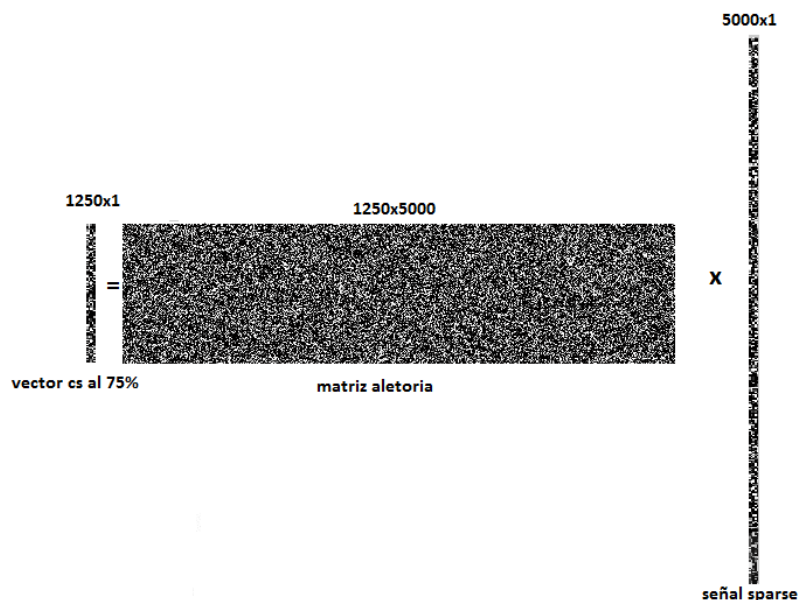
Se tomaran cinco imágenes de radar en alta definición, con las características mencionadas en el capítulo 3, primeramente tomaremos la imagen 1 para nuestro primer experimento seguidme se escogen las imágenes restantes para un total de 5 experimentos a cada imagen se le aplicaran los dos métodos de reconstrucción para identificar que método es mejor, se hará énfasis en la imagen 1 donde se establecerán diferentes grados de compresión (10%,25%,50% y 75%).

#### 4.2 FIGURA 12

##### 4.2 .1 Calculo de la señal comprimida de la figura 12 de radar en (CS).

Antes del proceso de la compresión de la figura 12 esta fue trasladada del dominio del tiempo al dominio sparse con la herramienta del paquete de MATLAB llamada **DCT**, esta se le aplico a cada fila de la imagen original resultando una señal sparse como se observa en la figura 19, donde a) es la señal original en el dominio del tiempo) señal en el dominio sparse.

En la figura 18 representa gráficamente el proceso matemático de compresión que se llevó acabo con la figura 12 donde se toman vectores de 5000x1 los cuales van hacer multiplicados por la matriz aleatoria para resultar un vector comprimido al 75% el que va llevar la información necesaria para que los algoritmos de reconstrucción lleguen a la solución más adecuada.



**Figura 23.** Proceso de compresión de la figura 12.

### 4.3 Método de reconstrucción SPGL1.

#### 4.3.1 Método de reconstrucción SPGL1 para la figura12 de radar en alta resolución.

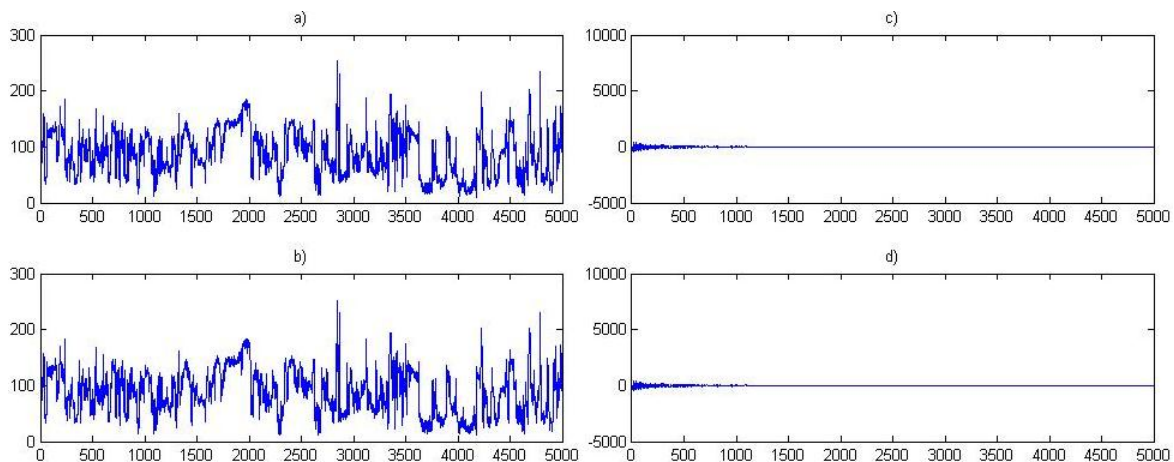
En el proceso de reconstrucción de la imagen será de la siguiente manera:

Se aplicara SPGL1 a cada señal sparse o vector comprimido en los diferentes grados de compresión (10%,25%,50% y 75%), guardando cada una de las señales reconstruidas en el mismo orden, así se garantiza mantener la correlación que guardan las columnas de la imagen.

##### 4.3.1.1 Conversión de la señal Sparse de la figura 12 al dominio del tiempo utilizando IDCT.

Después que la imagen es reconstruida se le aplica IDCT.

En la figura 19. Se ilustra lo siguiente: d) señal reconstruida en el dominio sparse con el algoritmo SPGL1 esto es a partir del veinticinco por ciento (25%) de los datos de la imagen o en otras palabras con un grado de compresión del 75%. la cual se le aplico **IDCT** herramienta que está en el paquete de MATLAB esto para trasladar de dominio, y por ultimo b) señal reconstruida en el dominio del tiempo.



**Figura 24.** Representa: a) señal original en el dominio del tiempo, b) señal reconstruida en el dominio del tiempo, c) señal original en el dominio sparse, d) señal reconstruida en el dominio sparse.

***Estos procesos se hicieron para todas las imágenes y para cada grado de compresión se ha mostrado una sola vez para no ser repetitivo con el RESULTADO.***

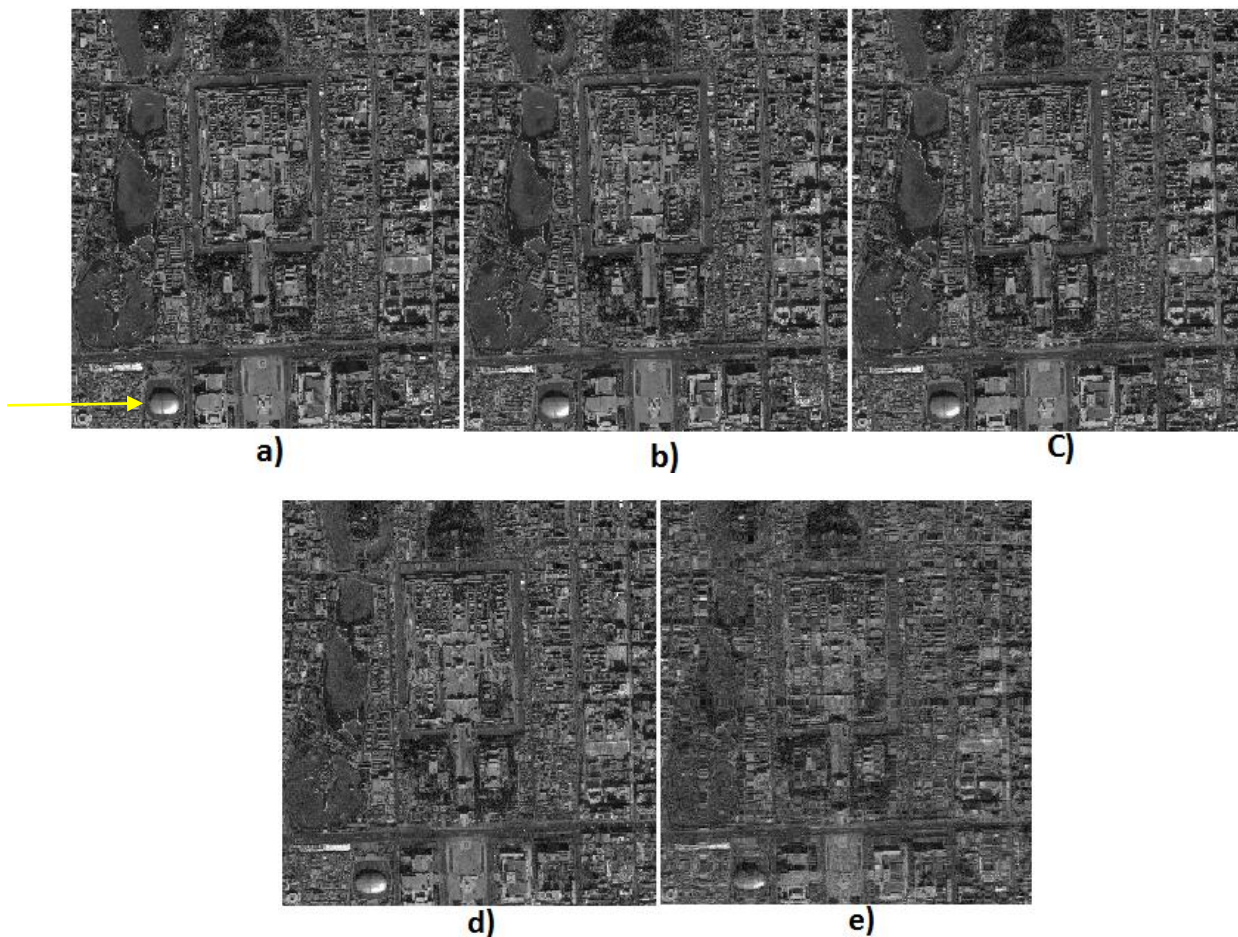
#### 4.3.1.2 Comparación de la imagen reconstruida con respecto a la figura 12

A continuación se presentan las imágenes reconstruidas a partir del algoritmo SPGL1.

En la Figura 20. Se ilustra la figura 12 con sus respectivas reconstrucciones a partir de los diferentes grados de compresión, podemos ver que en las imágenes, se pueden identificar varias edificaciones a pesar de la pérdida de información al momento de reconstruir son visibles, como lo es el estadio nacional de Pekín señalado con la flecha amarilla en la figura (a) podemos observar que en todas las imágenes está presente en forma y brillo lo que indica unos buenos resultados.

Estas imágenes reconstruidas sirven para aplicaciones en las que hay que identificar edificios, calles etc. en otras palabras vigilancia satelital.

Con esto podemos decir que el algoritmo tiene resultados satisfactorios ya que al ser probado en el máximo grado de compresión las formas de las edificaciones se mantienen.



**Figura 25.** Ilustra, **a)** imagen original, **b)** imagen reconstruida con un cs de 10%,**c)** imagen reconstruida con un cs de 25%, **d)** imagen reconstruida con un cs de 50% **e)** imagen reconstruida con un cs de 75%.

#### 4.3.2 FIGURA 13

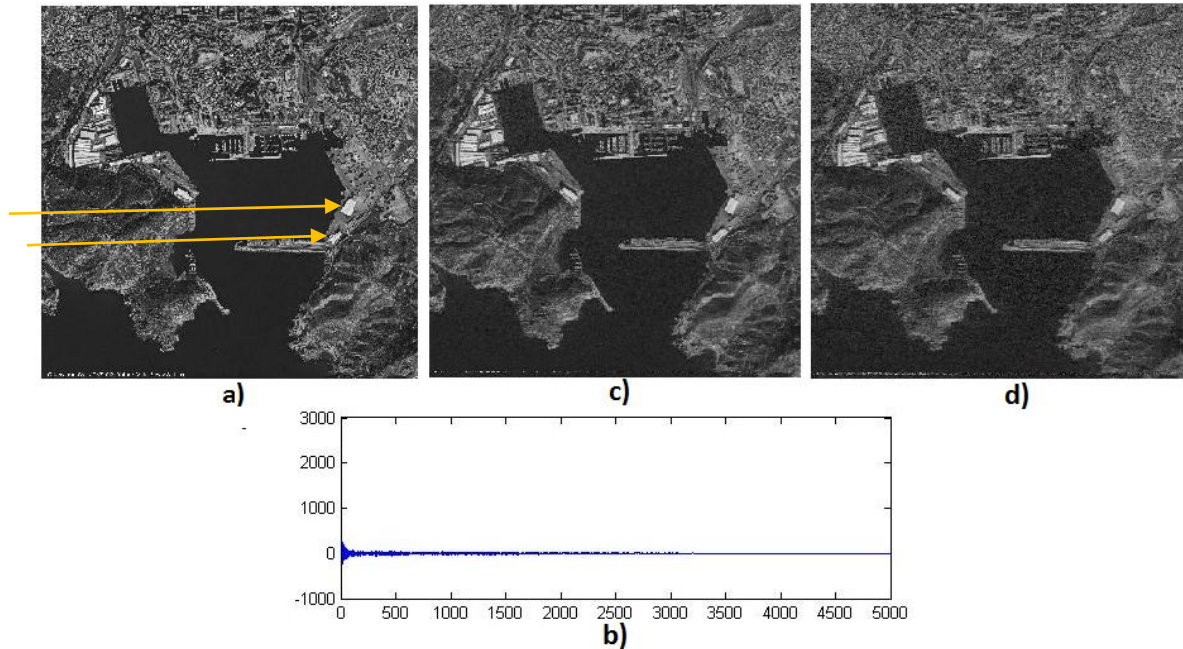
##### 4.3.2 .1 Método de reconstrucción SPGL1 para la figura 13 utilizando la señal CS.

De acuerdo con la Figura 21. Se aprecia. a) La imagen original de radar a procesar que posteriormente fue convertida al dominio sparse como se observa en b) seguidamente se comprimió al 75% y 50% como se resolvió en la sección (4.1.2).

Con las imágenes comprimidas se procede a reconstruirlas con el algoritmo SPGL1 posteriormente se aplica IDCT herramienta que está en el paquete de MATLAB esto para trasladar las imágenes sparse al dominio del tiempo para que se pueda observar la reconstrucción.

c) ilustra la imagen reconstruida a partir del cincuenta por ciento (50%) de los datos de la imagen original y d) presenta la imagen reconstruida a partir del veinticinco por ciento (25%) de los datos de la imagen original, podemos ver que en las imágenes, se pueden identificar varias edificaciones a pesar de la pérdida de información al momento de reconstruir, como lo son los dos edificios señalados con la flecha amarilla en la figura (a) se observa que en todas las imágenes está presente en forma y brillo lo que indica buenos resultados. Estas imágenes reconstruidas sirven para aplicaciones en las que hay que identificar edificios, calles etc. en otras palabras vigilancia satelital.

Con esto podemos decir que el algoritmo tiene resultados satisfactorios ya que al ser probado en el máximo grado de compresión las formas de las edificaciones se mantienen.



**Figura 26.** Ilustra, a) imagen original, b) señal sparse de la imagen original c) imagen reconstruida con un cs de 50% d) imagen reconstruida con un cs de 75%.

### 4.3.3 FIGURA 14

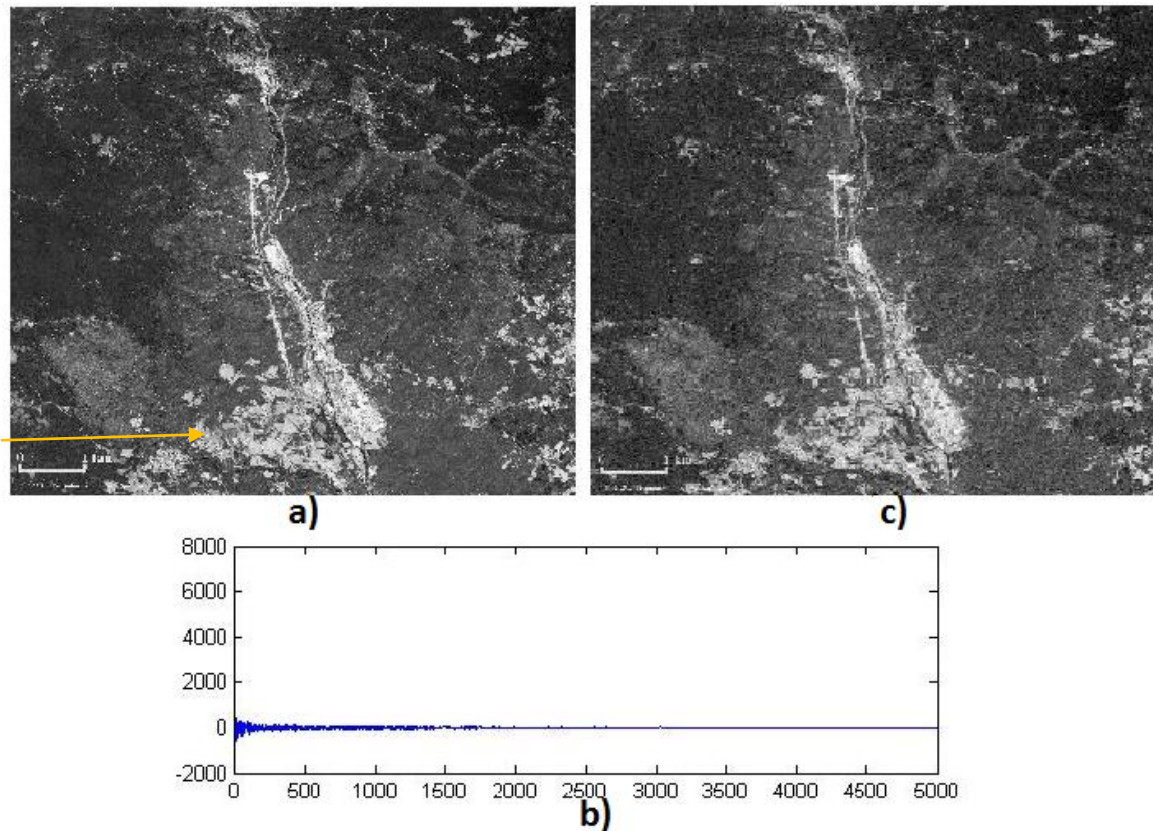
#### 4.3.3.1 Método de reconstrucción SPGL1 para la figura 14 utilizando la señal CS 50%.

De acuerdo con la Figura 22. Se aprecia. a) La imagen de radar a procesar que posteriormente fue convertida al dominio sparse como se observa en b) seguidamente se comprimió al 50% como se resolvió en la sección (4.1.2) con matriz aleatoria de mediciones con dimensiones de 2500x 5000.

Con la imagen comprimida se procede a reconstruirlas con el algoritmo SPGL1 posteriormente se aplica IDCT herramienta que está en el paquete de MATLAB esto para trasladar las imágenes sparse al domino del tiempo para que se pueda observar la reconstrucción.

c) ilustra la imagen reconstruida a partir del cincuenta por ciento (50%) de los datos de la imagen original, podemos ver que en las imágenes, se pueden identificar la mancha blanca señalados con la flecha amarilla en la figura (a) corresponde al incendio que hubo en la jonquera la cual fue tomada por el satélite Pléiades a pesar de la perdida de información al momento de reconstruir se puede visualizar perfectamente, y que en las dos imágenes está presente en forma y brillo lo que indica buenos resultados.

Estas imágenes reconstruidas sirven para aplicaciones en las que hay que hacerle seguimiento a determinado grupo de cultivo. En otras palabras vigilancia satelital a la agricultura en general.



**Figura 27.** Ilustra, a) imagen original, b) señal sparse de la imagen original c) imagen reconstruida con un cs de 50%.

#### 4.3.4 FIGURA 15

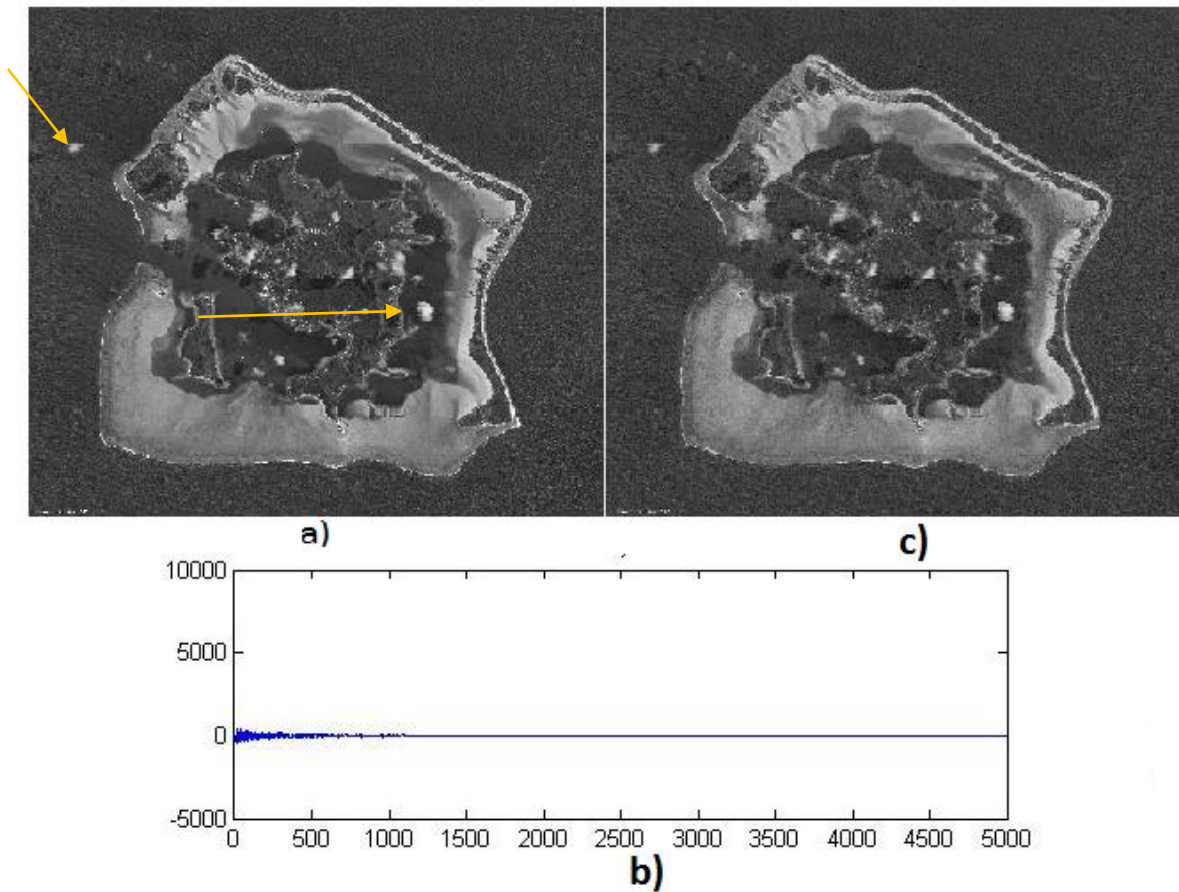
##### 4.3.4.1 Método de reconstrucción SPGL1 para la figura 15 utilizando la señal CS 50%.

De acuerdo con la Figura 23. Se aprecia. a) La imagen de radar a procesar que posteriormente fue convertida al dominio sparse como se observa en b) seguidamente se comprimió al 50% como se resolvió en la sección (4.1.2) con matriz aleatoria de mediciones con dimensiones de 2500x 5000.

Con la imagen comprimida se procede a reconstruirlas con el algoritmo SPGL1 posteriormente se aplica IDCT herramienta que está en el paquete de MATLAB esto para trasladar las imágenes sparse al dominio del tiempo para que se pueda observar la reconstrucción.

c) ilustra la imagen reconstruida a partir del cincuenta por ciento (50%) de los datos de la imagen original, podemos ver que en las imágenes, se pueden identificar la mancha blanca señalados con la flecha amarilla en la figura (a) corresponde al pequeñas islas que están al redor de la isla Bora Bora cual fue tomada por el satélite Pléiades a pesar de la perdida de información al momento de reconstruir se puede visualizar perfectamente, y que en las dos imágenes está presente en forma y brillo lo que indica buenos resultados.

Estas imágenes reconstruidas sirven para aplicaciones en las que hay que hacerle seguimiento protuberancias en la superficie terrestre, fallas geológicas nacimiento de pequeñas islas .En otras palabras usarlas en las ramas de la geología.



**Figura 28.** Ilustra, a) imagen original, b) señal sparse de la imagen original c) imagen reconstruida con un cs de 50%.

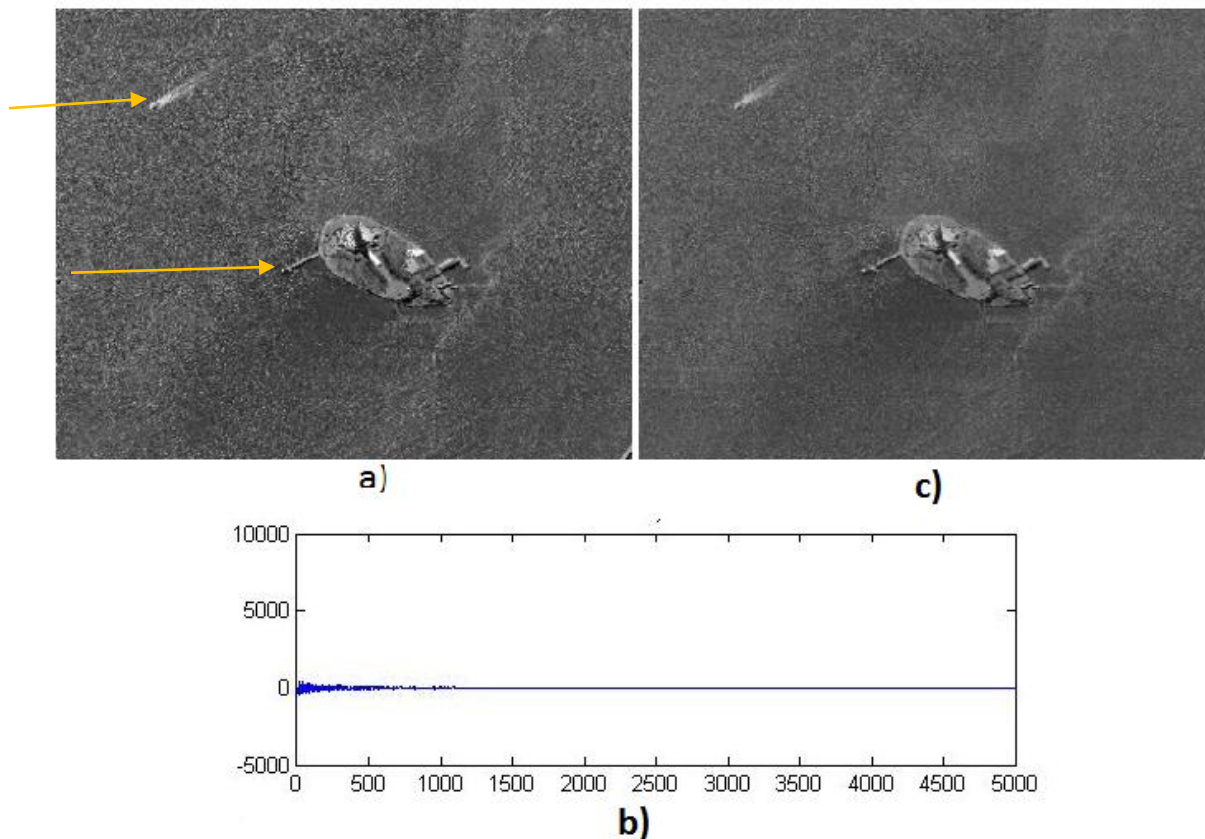
### 4.3.5 FIGURA 16

#### 4.3.5.1 Método de reconstrucción SPGL1 para la figura 16 utilizando la señal CS 50%.

De acuerdo con la Figura 24. Se aprecia. a) La señal de la imagen de radar a procesar que posteriormente fue convertida al dominio sparse como se observa en b) seguidamente se comprimió al 50% como se resolvió en la sección (4.1.2) con matriz aleatoria de mediciones con dimensiones de 2500x 5000.

Con la imagen comprimida se procede a reconstruirlas con el algoritmo SPGL1 posteriormente se aplica IDCT herramienta que está en el paquete de MATLAB esto para trasladar las imágenes sparse al dominio del tiempo para que se pueda observar la reconstrucción.

c) ilustra la imagen reconstruida a partir del cincuenta por ciento (50%) de los datos de la imagen original, podemos ver que en las imágenes, se pueden identificar un pequeño barco y el muelle con la flecha amarilla en la figura (a) corresponde a la estatua de la libertad la cual fue tomada por el satélite Pléiades a pesar de la pérdida de información al momento de reconstruir se puede visualizar perfectamente, y que en las dos imágenes está presente en forma y brillo lo que indica buenos resultados.





**Figura 28.** Ilustra, a) imagen original, b) señal sparse de la imagen original c) imagen reconstruida con un cs de 50%.

#### 4.4 Método de reconstrucción GPSR.

##### 4.4.1 Método GPSR de reconstrucción para las imágenes de radar en alta resolución.

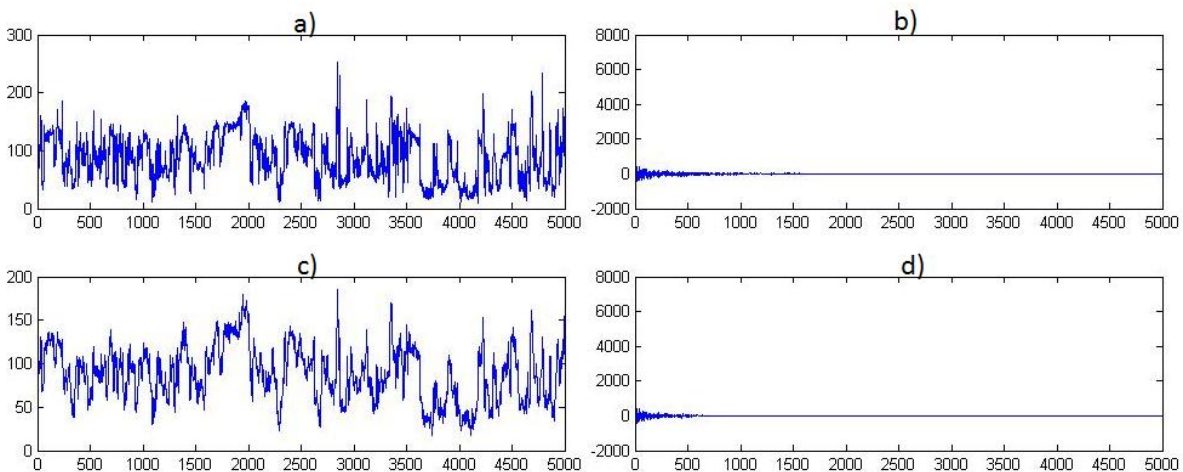
Se aplicara GPSR a todas las cinco imágenes.

Se enfatizara en la imagen 1 donde se empleara GPSR a cada vector comprimido en los diferentes grados de compresión (10%,25%,50% y 75%) que fue sometida la imagen, e ir guardando cada una de las señales reconstruidas en el mismo orden, así se garantiza mantener la correlación que guardan las columnas de la imagen.

##### 4.4.2 FIGURA 12

##### 4.4.2.1 Método GPSR de reconstrucción para la figura 12 utilizando las señales CS.

En la siguiente imagen de la Figura 24. Se aprecia. a) La señal de la imagen de radar a procesar en el dominio del tiempo que posteriormente fue convertida al dominio sparse como se observa en b) seguidamente se comprimió como se desarrolló en la sección (4.1.2) con matrices aleatorias de compresión de (10%,25%,50% y 75%) teniendo las imágenes comprimidas se procede a reconstruirlas con el algoritmo GPSR, Donde d) es la señal reconstruida en el dominio sparse. c) señal reconstruida en el dominio del tiempo que posteriormente se le aplico la **IDCT** como se vio en la sección (4.1.4).



**Figura 29.**Representa: a) señal original en el dominio del tiempo, b) señal original en el dominio sparse, c) señal reconstruida en el dominio del tiempo, d) señal reconstruida en el dominio sparse.

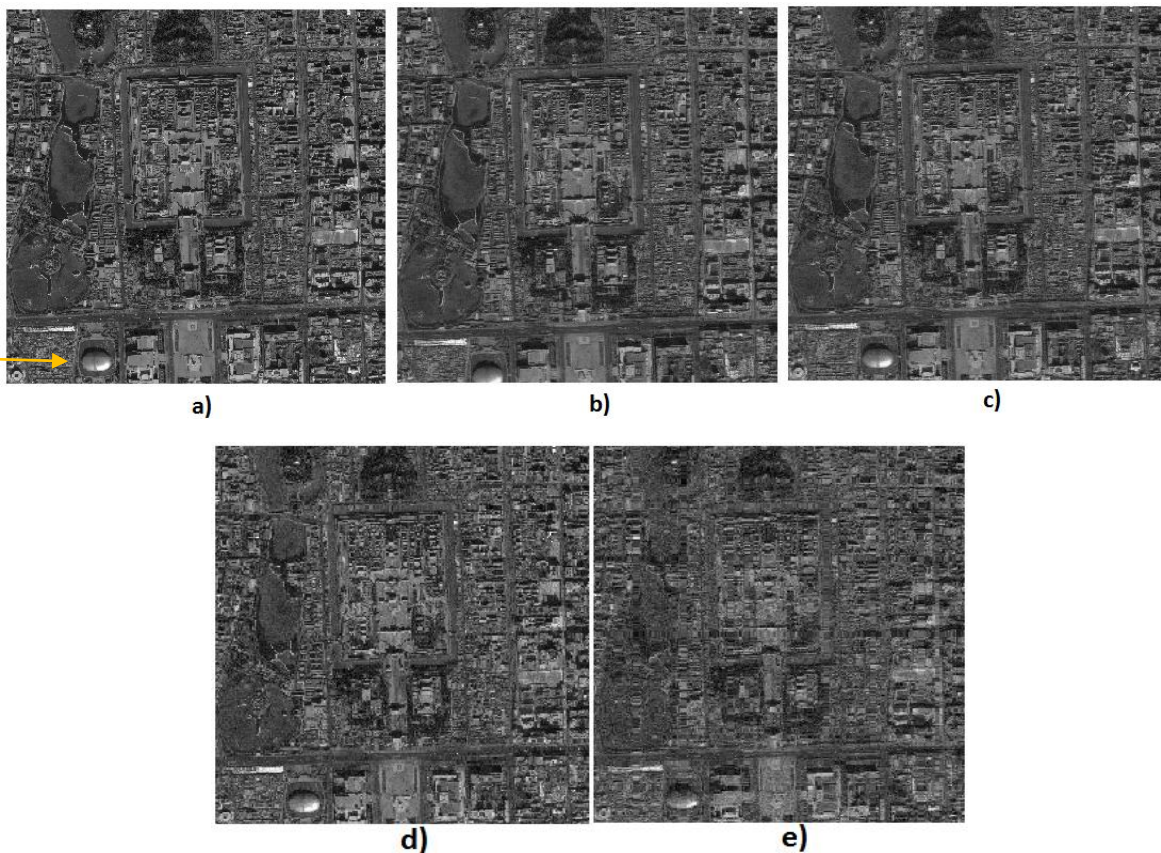
#### 4.4.2.2 Comparación de las imágenes reconstruidas con respecto a la figura 12.

En la Figura 25. Se ilustra la imagen 1 con sus respectivas reconstrucciones a partir de los diferentes grados de compresión, podemos ver que en las imágenes, se pueden identificar varias edificaciones a pesar de la perdida de información al momento de reconstruir son visibles, como lo es el estadio nacional de Pekín señalado con la flecha amarilla en la figura (a) podemos observar que en todas las imágenes está presente en forma y brillo lo que indica unos buenos resultados.

Estas imágenes reconstruidas sirven para aplicaciones en las que hay que identificar edificios, calles etc. Pero con grados de compresión no tan extremos.

Con esto podemos decir que el algoritmo tiene resultados satisfactorios con grados de compresión al 50%.

Mientras menos sea el grado de compresión mejor será nuestra reconstrucción.



**Figura 30.** Ilustra, **a)** imagen original, **b)** imagen reconstruida con un cs de 10%, **c)** imagen reconstruida con un cs de 25%, **d)** imagen reconstruida con un cs de 50% **e)** imagen reconstruida con un cs de 75%.

#### 4.4.3 FIGURA 13

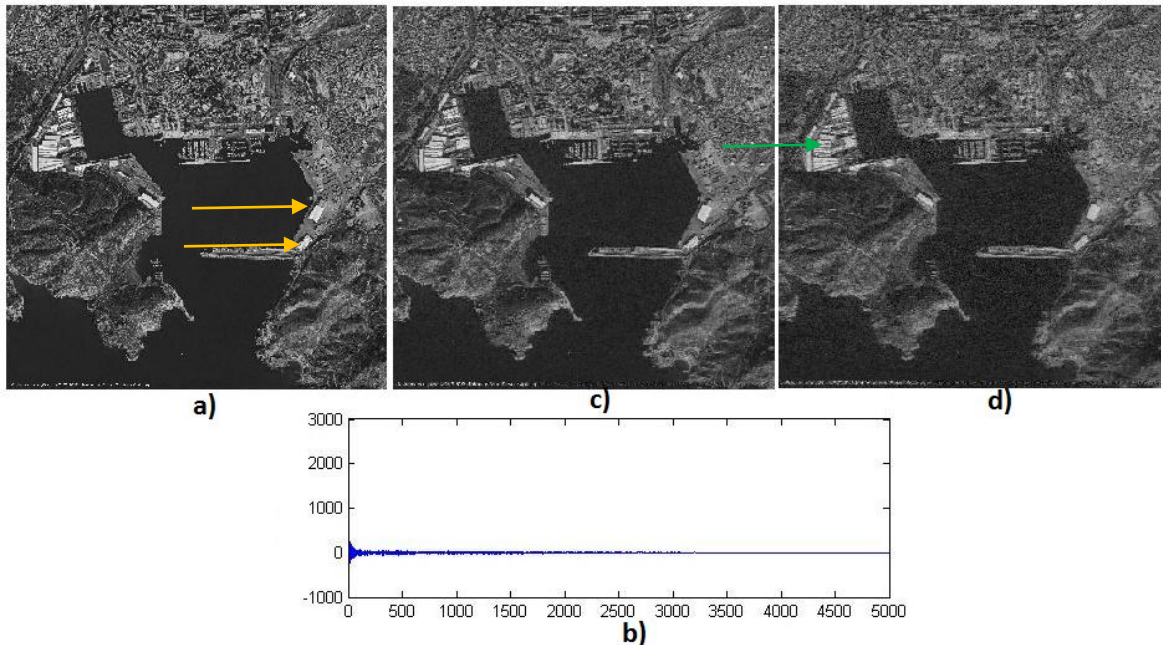
##### 4.4.3.1 Método GPSR de reconstrucción para la figura 13 utilizando las señales CS.

De acuerdo con la Figura 26. Se aprecia. a) La imagen original de radar a procesar que posteriormente fue convertida al dominio sparse como se observa en b) seguidamente se comprimió al 75% y 50% como se resolvió en la sección (4.1.2).

Con las imágenes comprimidas se procede a reconstruirlas con el algoritmo GPSR posteriormente se aplica IDCT herramienta que está en el paquete de MATLAB esto para trasladar las imágenes sparse al dominio del tiempo para que se pueda observar la reconstrucción.

c) ilustra la imagen reconstruida a partir del cincuenta por ciento (50%) de los datos de la imagen original y d) presenta la imagen reconstruida a partir del veinticinco por ciento (25%) de los datos de la imagen original, podemos ver que en las imágenes, se pueden identificar varias edificaciones a pesar de la pérdida de información al momento de reconstruir, como lo son los dos edificios señalados con la flecha amarilla en la figura (a) la flecha verde señala un área la cual no es reconstruida con mucha precisión en la imagen d) esto por el grado de compresión que fue sometida la imagen .

Pero se observa que en todas las imágenes está presente las formas y brillo lo que de las estructuras esto nos indica buenos resultados. Estas imágenes reconstruidas con GPSR sirven para aplicaciones en las que hay que identificar edificios, calles etc. Pero con grados de compresión no tan extremos.



**Figura 31.** Ilustra, a) imagen original, b) señal sparse de la imagen original c) imagen reconstruida con un cs de 50% d) imagen reconstruida con un cs de 75%.

Con esto podemos decir que el algoritmo tiene resultados satisfactorios con grados de compresión al 50%.

#### 4.4.4 FIGURA 14

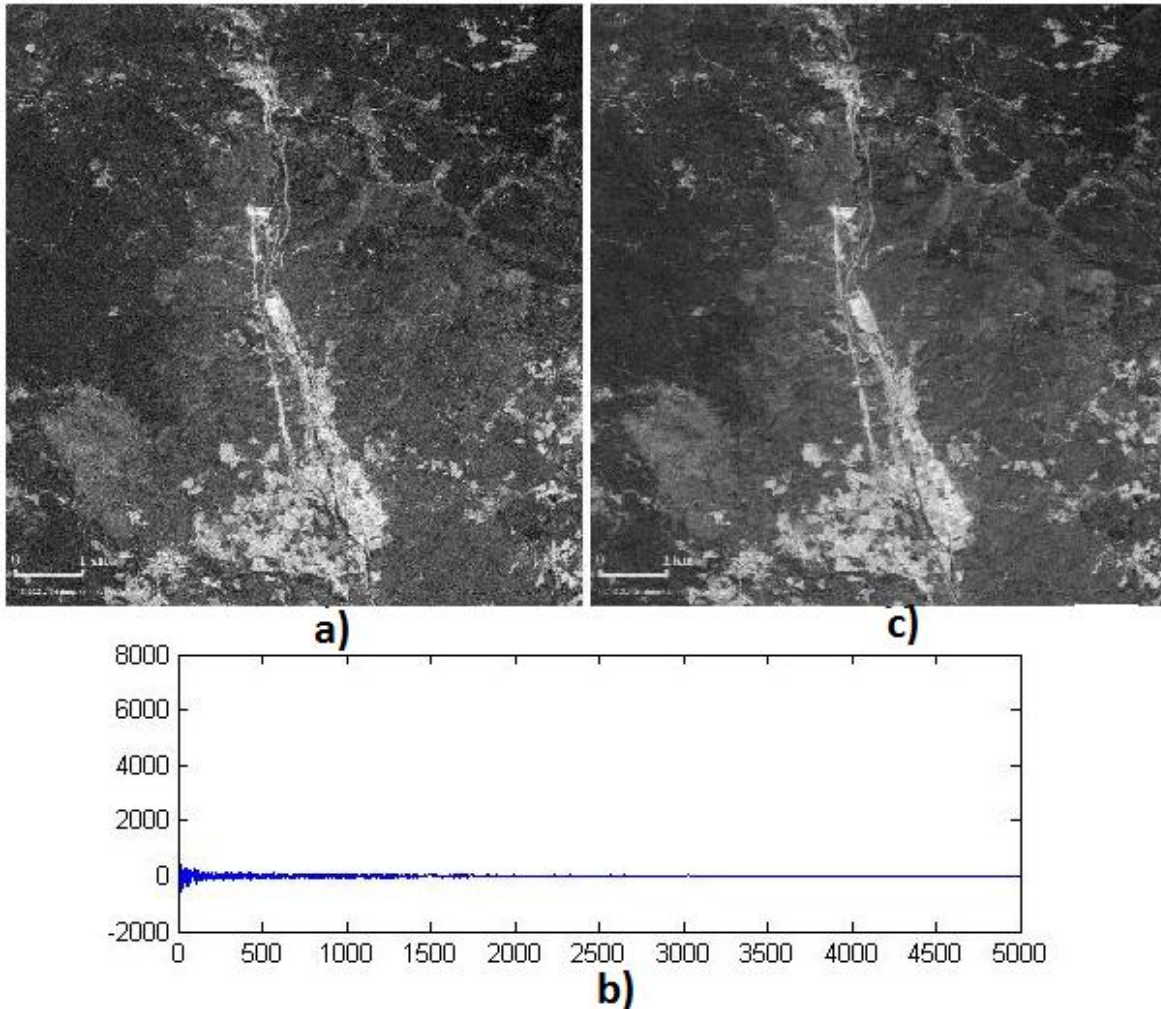
##### 4.4.4.1 Método GPSR de reconstrucción para la figura 14 utilizando las señales CS de 50%.

De acuerdo con la Figura 27. Se aprecia. a) La imagen de radar a procesar que posteriormente fue convertida al dominio sparse como se observa en b) seguidamente se comprimió al 50% como se resolvió en la sección (4.1.2) con matriz aleatoria de mediciones con dimensiones de 2500x 5000.

Con la imagen comprimida se procede a reconstruirlas con el algoritmo GPSR posteriormente se aplica IDCT herramienta que está en el paquete de MATLAB esto para trasladar las imágenes sparse al dominio del tiempo para que se pueda observar la reconstrucción.

c) ilustra la imagen reconstruida a partir del cincuenta por ciento (50%) de los datos de la imagen original, podemos ver que en las imágenes, se pueden identificar la mancha blanca señalados con la flecha amarilla en la figura (a) corresponde al incendio que hubo en la jonquera la cual fue tomada por el satélite Pléiades a pesar de la pérdida de información al momento de reconstruir se puede visualizar

perfectamente, y que en las dos imágenes está presente en forma y brillo lo que indica buenos resultados.



**Figura 32.** Ilustra, a) imagen original, b) señal sparse de la imagen original c) imagen reconstruida con un cs de 50%.

#### 4.4.5 FIGURA 15

##### 4.4.5.1 Método GPSR de reconstrucción para la figura 15 utilizando las señales CS de 50%.

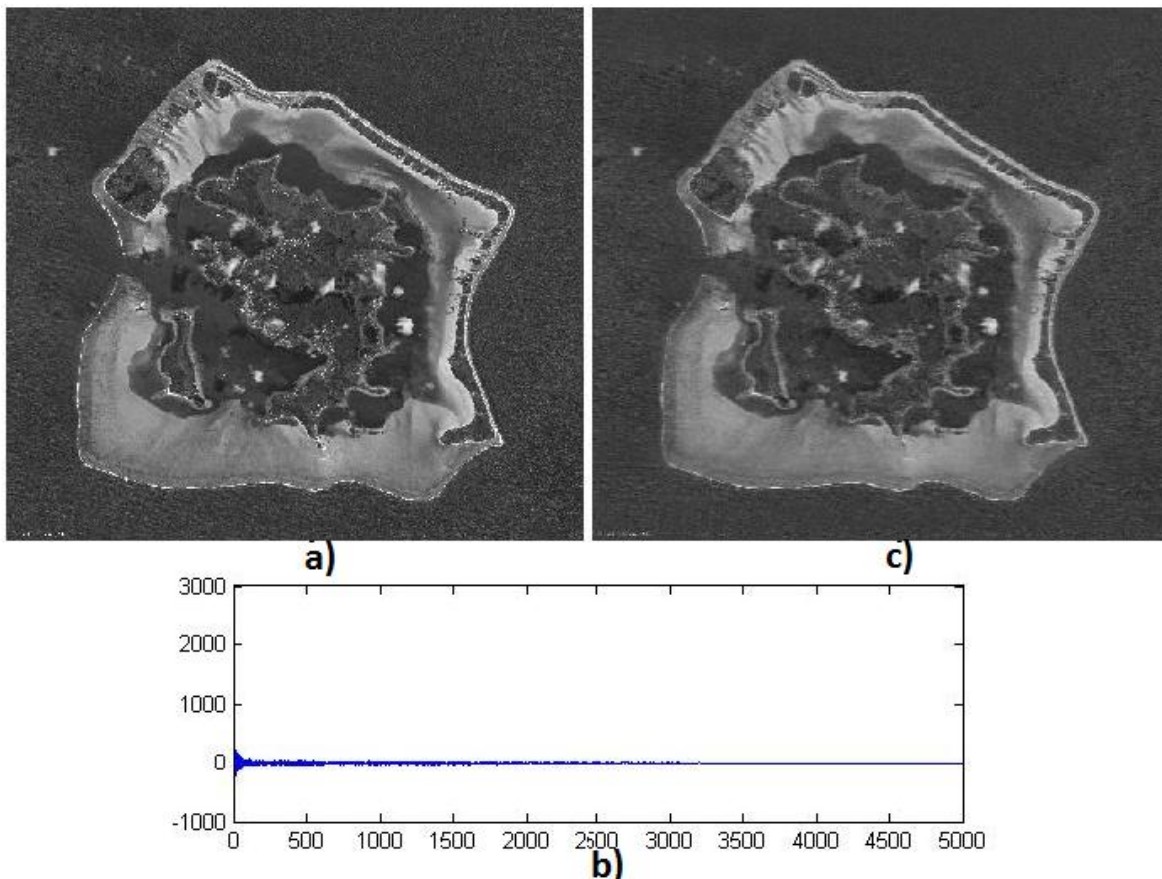
De acuerdo con la Figura 28. Se aprecia. a) La imagen de radar a procesar que posteriormente fue convertida al dominio sparse como se observa en b) seguidamente se comprimió al 50% como se resolvió en la sección (4.1.2) con matriz aleatoria de mediciones con dimensiones de 2500x 5000.

Con la imagen comprimida se procede a reconstruirlas con el algoritmo GPSR posteriormente se aplica IDCT herramienta que está en el paquete de MATLAB esto

para trasladar las imágenes sparse al dominio del tiempo para que se pueda observar la reconstrucción.

c) ilustra la imagen reconstruida a partir del cincuenta por ciento (50%) de los datos de la imagen original, podemos ver que en las imágenes, se pueden identificar la mancha blanca señalados con la flecha amarilla en la figura (a) corresponde al pequeñas islas que están al redor de la isla Bora Bora cual fue tomada por el satélite Pléiades a pesar de la perdida de información al momento de reconstruir se puede visualizar perfectamente, y que en las dos imágenes está presente en forma y brillo lo que indica buenos resultados.

Estas imágenes reconstruidas sirven para aplicaciones en las que hay que hacerle seguimiento protuberancias en la superficie terrestre, fallas geológicas nacimiento de pequeñas islas .En otras palabras usarlas en las ramas de la geología.



**Figura 33.** Ilustra, a) imagen original, b) señal sparse de la imagen original c) imagen reconstruida con un cs de 50%.

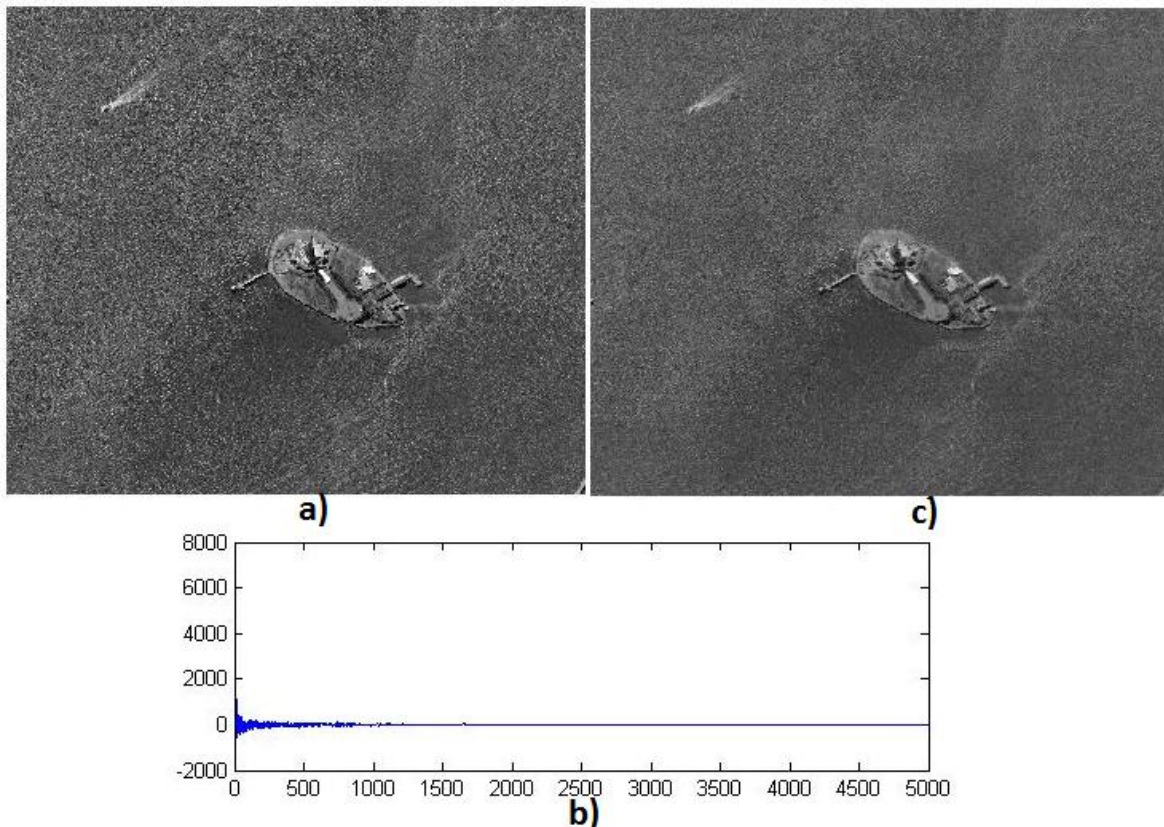
#### 4.4.6 figura 16

##### 4.4.6.1 Método GPSR de reconstrucción para la figura 16 utilizando las señales CS de 50%.

De acuerdo con la Figura 29. Se aprecia. a) La señal de la imagen de radar a procesar que posteriormente fue convertida al dominio sparse como se observa en b) seguidamente se comprimió al 50% como se resolvió en la sección (4.1.2) con matriz aleatoria de mediciones con dimensiones de 2500x 5000.

Con la imagen comprimida se procede a reconstruirlas con el algoritmo SPGL1 posteriormente se aplica IDCT herramienta que está en el paquete de MATLAB esto para trasladar las imágenes sparse al dominio del tiempo para que se pueda observar la reconstrucción.

c) ilustra la imagen reconstruida a partir del cincuenta por ciento (50%) de los datos de la imagen original, podemos ver que en las imágenes, se pueden identificar un pequeño barco y el muelle con la flecha amarilla en la figura (a) corresponde a la estatua de la libertad la cual fue tomada por el satélite Pléiades a pesar de la pérdida de información al momento de reconstruir se puede visualizar perfectamente, y que en las dos imágenes está presente en forma y brillo lo que indica buenos resultados.



**Figura 34.** Ilustra, a) imagen original, b) señal sparse de la imagen original c) imagen reconstruida con un cs de 50%.

#### 4.5 Error generado por los algoritmos de reconstrucción SPGL1 Y GPSR.

A modo de comparar el error generado por los dos métodos en la reconstrucción, de las imágenes de radar en alta definición. Las mismas serán reconstruidas utilizando los dos algoritmos de reconstrucción que fueron presentados en el capítulo 2. Luego se calcula el porcentaje de error que existe entre las imágenes originales y las imágenes reconstruidas.

Como se puede ver en la tabla 1 el error de reconstrucción de la figura 12 para el grado de compresión de 75% es el más alto para cada método, a lo que conlleva a una reconstrucción de baja calidad y para un porcentaje de la señal muestreada al 10% el error es bajo y bastante bueno para el solucionador SPGL1 cuya reconstrucción va hacer mejor.

Muestrear un porcentaje más alto de la señal ya no tiene sentido ya que no se logra ninguna reducción el conjunto de datos necesarios para la reconstrucción de la imagen, haciendo que la teoría del CS no tenga sentido aplicar.

**Tabla 1:** comparación del error obtenido de cada método usado.

IMÁGENES	PORCENTAJE DE COMPRESION	SPGL1 (% DE ERROR)	GPSR (% DE ERROR)
FIGURA 12	10%	0.2311	5.2220
	25%	0.8245	6.8218
	50%	2.7630	7.3108
	75%	7.3135	9.7869
FIGURA 13			
	50%	4.2511	9.4352
	75%	9.9561	12.7020
FIGURA 14	50%	3.3798	7.8817
FIGURA 15	50%	2.0124	5.1322
FIGURA 16	50%	1.9124	6.5310



Se puede observar el error generado por el solucionador GPSR es más alto con respecto SPGL1, lo cual se llega a la conclusión que el algoritmo más eficiente en cuanto a la calidad SPGL1.

#### 4.6 tiempo de reconstrucción empleados de los métodos SPGL1 y GPSR.

A continuación se presenta los tiempos empleados de reconstrucción de los algoritmos para cada imagen.

Estos tiempos de reconstrucción son largos debido al gran tamaño que tienen las imágenes satelitales y por ende tendrá una gran data para su procesamiento estamos hablando de millones de pixeles que pasaran por varias etapas de procesamiento.

En la tabla 2 muestra de manera comparativa los dos algoritmos de reconstrucción, basándonos en esta tabla observamos que GPSR es más eficaz en cuanto a velocidad de reconstrucción.

**Tabla 2:** tiempos de reconstrucción empleados por los métodos

IMÁGENES	PORCENTAJE DE COMPRESION	SPGL1 TIEMPO DE RECONSTRCCION (HORA:MIN)	GPSR TIEMPO DE RECONSTRCCION (HORA:MIN)
FIGURA 12	10%	91:23	80:56
	25%	78:10	69:43
	50%	46:05	39:11
	75%	18:16	13:59
FIGURA 13			
	50%	45:57	35:02
	75%	20:22	12:29
FIGURA 14	50%	44:08	38:32
FIGURA 15	50%	47:19	36:43
FIGURA 16	50%	67:02	54:27

## 4.7 Interfaz de usuario

La interfaz de usuario está conformada por la ventana principal y la ventana de funcionalidades cada ventana cumple con una función específica que serán definidas a continuación.

### 4.7.1 Ventana principal

La figura 35 presenta la ventana principal del usuario está compuesta por los siguientes ítems:

- Título: RECONSTRUCCIÓN DE IMÁGENES SPARSE DE RADAR EN ALTA RESOLUCION USANDO LOS ALGORITMOS SPGL1 Y GPSR.
- Autor : Israel Navarro Rada
- Semillero de investigación en ingeniera biomédica y telecomunicaciones.
- Programa : ingeniería en telecomunicaciones
- Grupo al que pertenece: grupo de investigación de ingeniería biomédica (GIBUP).

Para dar acceso a las funciones y contenido de la interfaz se presionara el botón inicio  resaltado en la figura 35.



**Figura 35.** Ilustra ventana principal de la interfaz de usuario

Seguidamente desplegara una ventana con todas las funcionalidades de la interfaz como se observa en la figura 36.

## 4.7.2 Ventana de funcionalidades

La ventana de funcionalidad está conformada por cuatro subventanas llamadas:

- 1) imagen original
- 2) imagen sparse
- 3) matriz comprimida
- 4) imagen reconstruida

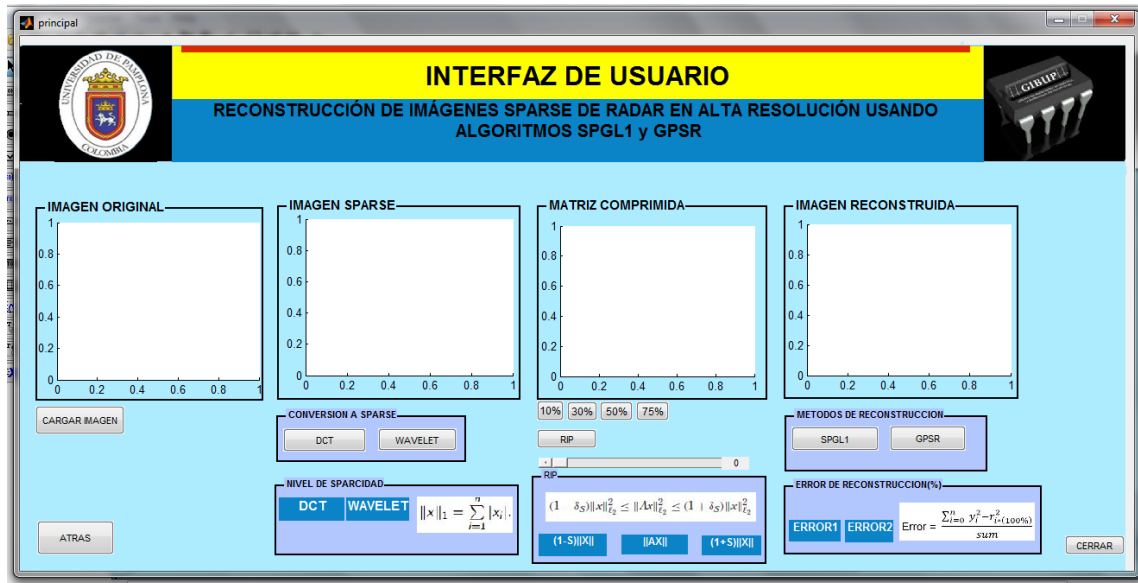
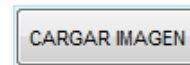


Figura 36. Ventana de funcionalidades

### 4.7.2.1 Imagen original

Para cargar la imagen original a procesar se presiona el botón resaltado en la figura 37, la cual se mostrara en la primera subventana como se ilustra en la siguiente figura.



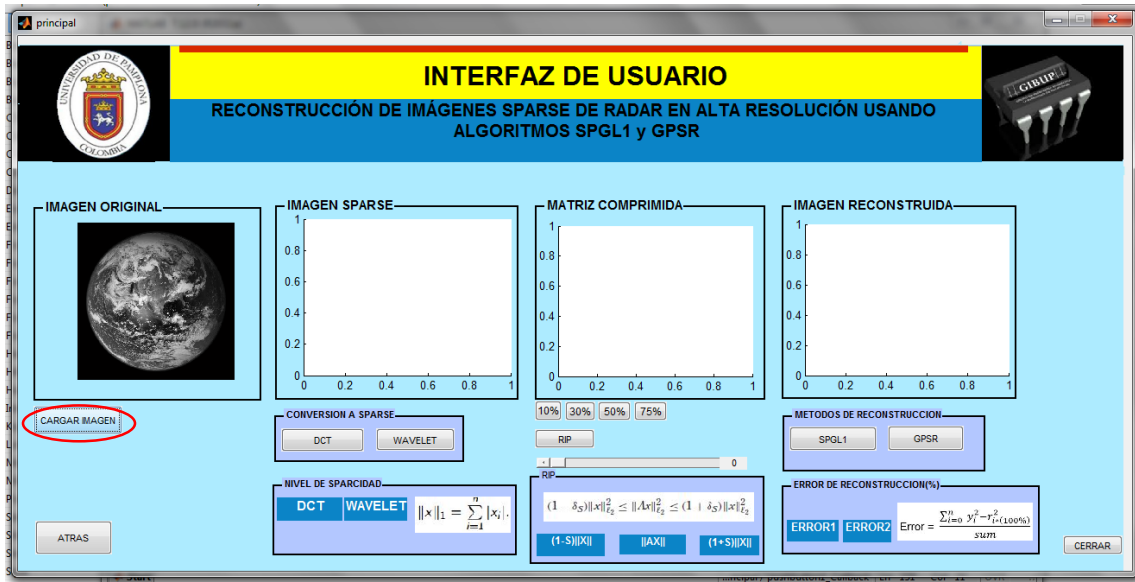


Figura 37. Proceso de cargar una imagen en la primera subventana.

#### 4.7.2.2 Imagen sparse

Para la conversión de una imagen que está en el dominio del tiempo al dominio sparse la interfaz cuenta con una herramienta llamada conversión a sparse que consta de dos botones que corresponden a las técnicas llamada transformada discreta del coseno (DCT) (botón izquierdo) y la trasformada de wavelet (botón derecho) resaltada en la figura 38.

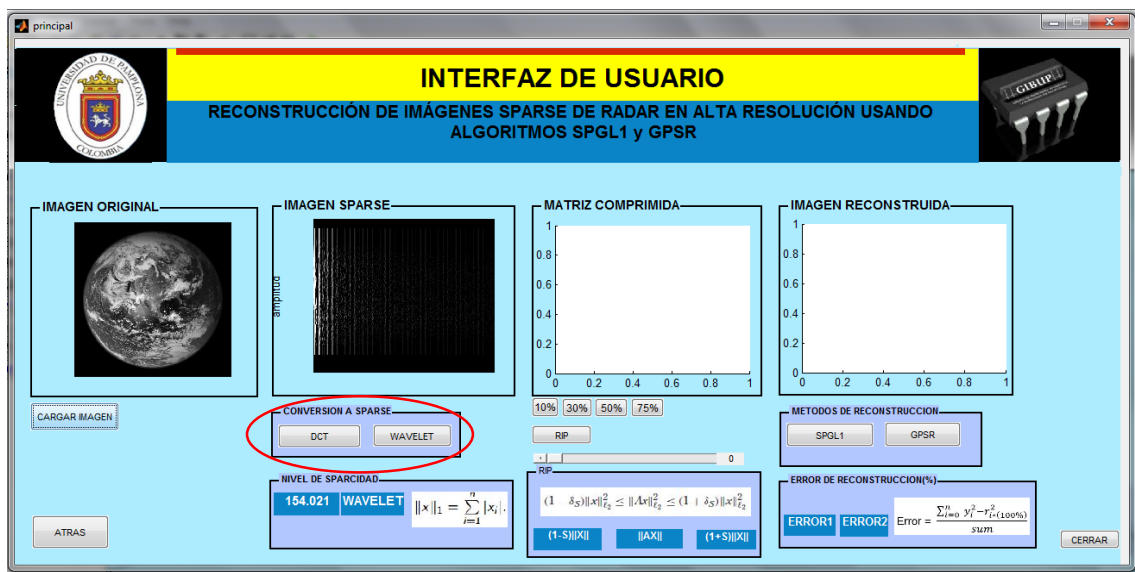
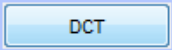
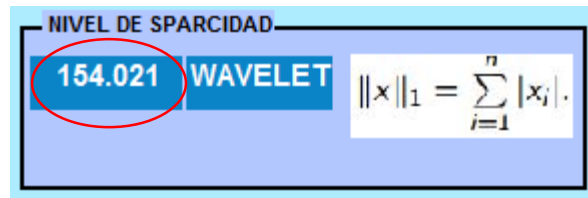
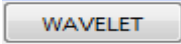


Figura 38. Herramienta resaltada para la conversión a sparse

Al presionar el botón  de esta herramienta se graficará la imagen sparse calculada con la (DCT) en la segunda subventana de la interfaz, además se calcula el nivel de esparcidad de la imagen, este valor será mostrado en la ventanilla llamada nivel de esparcidad como se presenta en la figura 39.







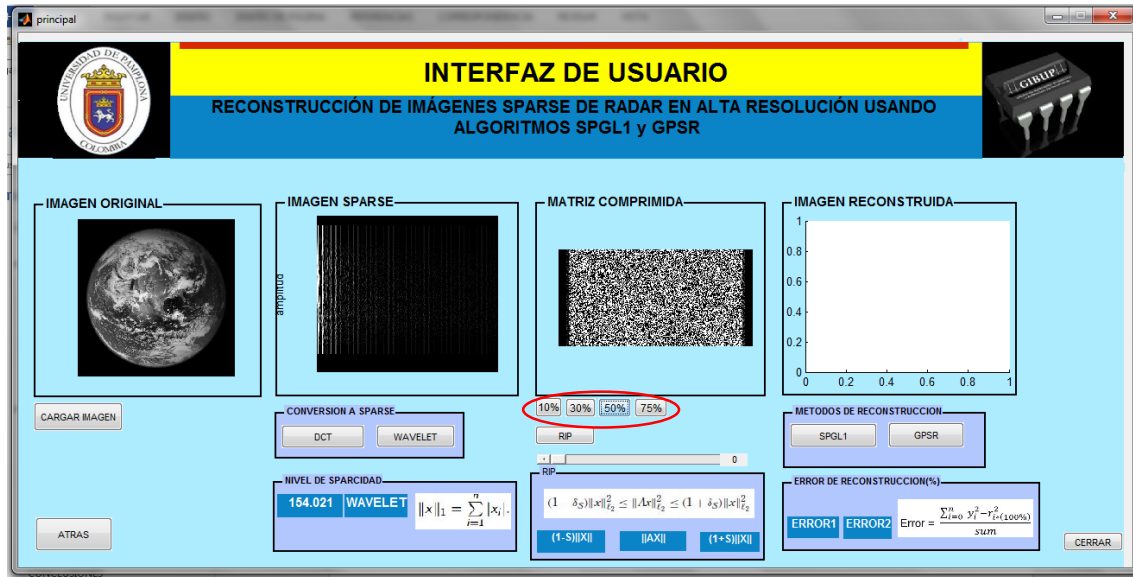
**Figura 39.**ventanilla resaltada, muestra el valor de esparcidad calculado con la DCT

Para graficar la imagen sparse y calcular el nivel de esparcidad con la transformada de wevelet solo hay que presionar el botón 

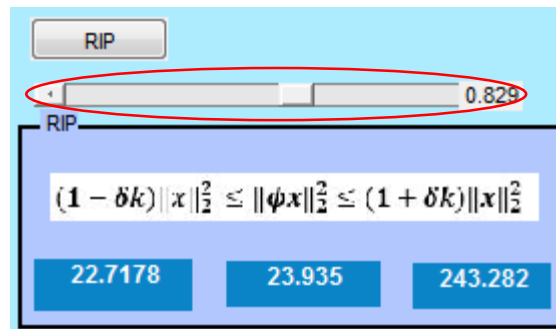
La ecuación mostrada en la figura 39 corresponde a la norma L1, la cual calcula el nivel de esparcidad.

#### 4.7.2.3 Matriz comprimida

En esta subventana se aplica la teoría Compressive Sensing representada en cuatro niveles de compresión y simbolizada por los botones     resaltados en la figura 40, al presionar un nivel, la información será comprimida a dicho nivel y posteriormente graficada en esta subventana en forma de una matriz digitalizada, como se observa en la figura 40.



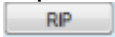
**Figura 40.** Niveles de compresión resaltados y matriz comprimida



**Figura 41.** Ventanilla de la propiedad de isometría restringida (RIP)

La ventanilla de propiedad de isometría restringida (RIP) depende de la matriz de medición tomada para crear la matriz comprimida que se muestra en la subventana tres, básicamente si la desigualdad que se muestra en la figura 41 se cumple la matriz de medición es apta para la compresión.

Con el slider encerrado en un ovalo rojo se varia los valores de delta que se encuentran en el intervalo (0.8 a 0.9).

Después que se ha escogido el delta se procede a calcular la RIP presionando el botón,  más abajo se mostrara los valores correspondientes a la desigualdad.

#### 4.7.2.4 Imagen reconstruida

Para la reconstrucción de la imagen comprimida la interfaz cuenta con dos métodos de reconstrucción llamados SPGL1 Y GPSR encerrados en un ovalo rojo como se ilustra en la figura 42 es solo seleccionar el método deseado.

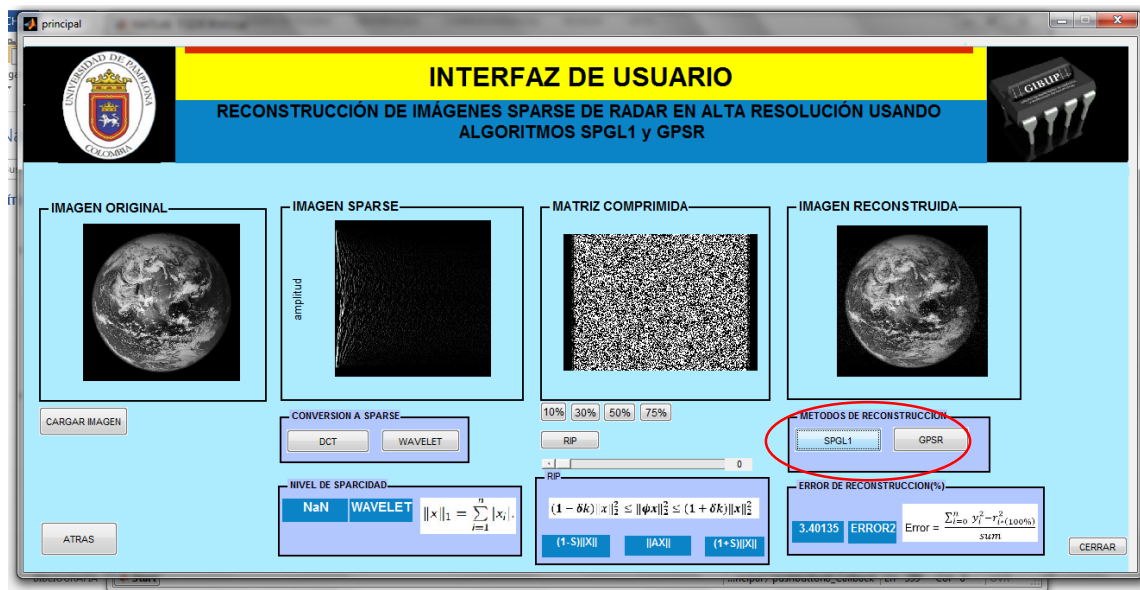


Figura 42. ilustra imagen reconstruida en la subventana cuatro

Al seleccionar el método también se calcula el porcentaje de error obtenido en la reconstrucción de la imagen como se observa en la figura 43 encerrado en un ovulo rojo, la variable error2 pertenece al error de GPSR y la formula representa el cómo se calcula el porcentaje de error que se obtiene.

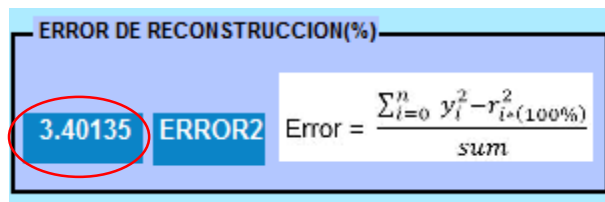


Figura 43. Ilustra error obtenido

## CONCLUSIONES

- En este trabajo implementó los algoritmos de reconstrucción SPGL1 Y GPSR, con los cuales se pudo reconstruir señales en el dominio sparse. Se realizó un estudio comparativo del rendimiento de cada uno de los algoritmos de reconstrucción en base a los siguientes criterios: error de reconstrucción y tiempo de reconstrucción.
- Los métodos arrojaron errores pequeños en la reconstrucción para distintos porcentajes de compresión para la figura 12 el algoritmo PGL1 se tuvieron los siguientes (10%=0.2311, 25%=0.8245, 50%=2.7630, 75%=7.3135) esto conlleva a una buena reconstrucción el solucionador SPGL1 demostró ser mejor en cuanto a la calidad de reconstrucción.
- Los tiempos de reconstrucción fueron largos por la gran cantidad de información que tienen las imágenes de radar, debido a esto se observó que el solucionador GPSR es más eficaz en tiempo de reconstrucción si se comparan los dos algoritmos con un grado de compresión de 50% para GPSR 54H:27M y para SPGL1 67H:02M para la figura 16.
- se logró reconstruir las imágenes utilizando sólo un pequeño conjunto de proyecciones gracias a Compressive Sensing(CS) .A medida que el nivel de esparcidad aumenta el número de proyecciones necesarias para la reconstrucción también lo hace, sin embargo el número de muestras necesarias para la reconstrucción es bastante reducido.
- Se observó que la conversión a sparse de una imagen con DCT tiene un menor nivel de esparcidad que la DWT. lo que es una propiedad importante para que CS pueda explotar su potencial de codificar y muestrear señales o imágenes para que posteriormente sea una reconstrucción exitosa a menor cantidad de cálculo.
- La propiedad RIP tiene como Objetivo garantizar y construir matrices capaces de codificar la información del mayor conjunto posible de señales.
- El porcentaje ideal de compresión para las imágenes fue de un 50% para para el algoritmo SPGL1 y 75% para GPSR, estas reconstrucciones a estos porcentajes nos permiten observar con mucha claridad edificaciones, calles autos etc, reconstrucciones que sirven para cualquier rama se la teledetección.



## BIBLIOGRAFÍA

- [1] emmanuel j. candès and michael b. wakin, an introduction to compressive sampling, stanford university in 1998.
- [2] Richard G. Baraniuk , IEEE signal processing magazine july 2007, Electrical and Computer Engineering at Rice University ,pag 2-3.
- [3] Baraniuk G. R. (2007). Compressive Sensing
- [4] Lenis M. Meriño y Msc. Luis E. Mendoza,compresión robusta usando Compressive Sensing (cs),Universidad de Pamplona – Colombia, página 5.
- [5] Br. Jesús Alberto Rodríguez De Sarrazin, Estudio de los Algoritmos de reconstrucción en Compressed Sensing, universidad de los andes Colombia, pág. 8.
- [6] M. Figueiredo, R. Nowak, and S. J. Wright, Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems, in IEEE J. Selected Topics in Signal Process., IEEE Press, Piscataway, NJ, 2007.
- [7] J. Barzilai and J. Borwein. “Two point step size gradient methods,” IMA Journal of Numerical Analysis, vol. 8, pp. 141–148, 1988.
- [8] S. Chen, D. Donoho, and M. Saunders. “Atomic decomposition by basis pursuit,” SIAM Journal of Scientific Computation, vol. 20, pp. 33–61, 1998.
- [9] Compressed Sensing with Coherent and Redundant Dictionaries (2010) Emmanuel J. Candes , Yonina C. Eldar , Deanna Needell , Department of Electrical Engineering, Technion - Israel Institute of Technology pag 2.
- [10] <http://www.infoterra.es>.
- [11] García rodríguez, Rafael y de la vega panizo Rogelio, Teledetección: nociones y aplicaciones, paginas 221-232.
- [12] J. J. Fuchs. “Multipath time-delay detection and estimation,” IEEE Transactions on Signal Processing, vol. 47,pag. 237–243, 1999.
- [13] Y.-H. Dai and R. Fletcher. “Projected Barzilai-Borwein methods for large-scale box-constrained quadratic program.
- [14] R. W. Cottle, J.-S. Pang, and R. E. Stone. The Linear Complementarity Problem, Academic Press, 1993.
- [15] D. Donoho. “De-noising by soft-thresholding,” IEEE Transactions on Information Theory, vol. 41, pag. 613–627.

- [16] E. J. Candès, J. Romberg, and T. Tao, Stable signal recovery from incomplete and inaccurate measurements, *Comm. Pure Appl. Math.*, 59 (2006), pp. 1207–1223.
- [17] ewout van den berg and michael p. friedlander ,probing the pareto frontier for basis pursuit solutions, *siam j. en computación científica*, 31 (2): 890-912, noviembre de 2008.
- [18] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, Least angle regression, *Ann. Statist.*, 32 (2004),
- [19] Matthew A. Herman and Thomas Strohmer, High-Resolution Radar via Compressed Sensing, State University of New York at Buffalo, Stanford University and then joined the Department of Mathematics,pag 2-3, 2008 IEEE.
- [20] Boufounus T. P. y Baraniuk G. R. (2008). 1-Bit Compressive Sensing.
- [21] Scott Shaobing Chen, David L. Donoho and Michael A. Saunders, Atomic decomposition by basis pursuit, SIAM 1998.
- [22] X. Huo, Sparse Image representation Via Combined Transforms, PhD thesis, Stanford, 1999.
- [23] Wotao Yin Department of Mathematics, UCLA Spring , Optimization Barzilai-Borwein Method pagina 3-7, 2015.
- [24] Andrzej Cichocki ,Laboratory for Advanced Brain Signal Processing, Riken Brain Science Institute, Japan; and Warsaw University of Technology and Systems Research Institute, PAN, PolandNonnegative matrix and tensor factorizations applications to exploratory multi-way data analysis andblind source separation pag 272-276.
- [25] Dr. Martin Hurtado, diseño de arreglos y procesamiento de señales en sistemas de radar presentada ante la facultad de ingeniería de la universidad nacional de la plata, 2015
- [26] Hernán Alberto calderón Investigación acerca de Reconstrucción de imágenes geológicas basadas en la teoría de compressed sensing, universidad de chile ,2014.
- [27] Imaging via Compressive Sampling, Justin Romberg, attended Rice University, 2006.
- [28] Carlos Sing, Cristián Tejos N, Pablo Irrázaval, universidad Pontificia Católica (Chile), Investigación acerca reducción de los tiempos de adquisición de imágenes por resonancia magnética utilizando técnicas de compressed sensing (2009).
- [29] jose Patricio Bovet Derpich, La interfaz de usuario: tipos y criterios de diseño, 2011.pag 1-9.

## **MARCO LEGAL**

Para el desarrollo de este proyecto se simuló el funcionamiento de la teoría de compressive sensing y los métodos de reconstrucción SPGL1 Y GPSR en imágenes de radar en alta definición fue un proceso único del autor sin hacer copia de algún trabajo realizado, para la estructura del marco teórico ha sido debidamente referenciada, por lo tanto del marco legal este se encuentra libre de requerimientos que comprometan la propiedad intelectual y material de este proyecto.

Para la simulación del funcionamiento de cada método se utilizó el software MATLAB con una versión de prueba 2011a.