

**TRANSMISIÓN DE IMÁGENES DIGITALES
APLICANDO LA TÉCNICA DE MUESTREO Y
COMPRESIÓN A TRAVÉS DE UN SISTEMA DE
COMUNICACIÓN INALÁMBRICO**

IRENE LIZETH MANOTAS GUTIÉRREZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECAÑICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA
2013

TRANSMISIÓN DE IMÁGENES DIGITALES APLICANDO LA TÉCNICA DE MUESTREO Y COMPRESIÓN A TRAVÉS DE UN SISTEMA DE COMUNICACIÓN INALÁMBRICO

IRENE LIZETH MANOTAS GUTIÉRREZ

Monografía de grado para optar por el título de:
Master en Ingeniería de Sistemas

Director:

Ph.D.(c), MPE. HENRY ARGUELLO FUENTES

Codirector:

MSc. FERNANDO A. ROJAS MORALES

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECAÑICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA
2013

Índice general

Introducción	9
1. Técnica de Muestreo y Compresión Aplicada en Imágenes	12
1.1. Compressive Sensing (CS)	12
1.2. Matrices de Muestreo	13
1.3. Algoritmos de Reconstrucción	16
1.3.1. Algoritmo Iterative Hard Thresholding	18
1.3.2. Algoritmo Orthogonal Matching Pursuit	19
2. Algoritmo de Reconstrucción CS Basado en Bloques y Transmisión de Datos CS	22
2.1. Algoritmo de Reconstrucción Basado en Bloques Sobrepuestos	22
2.2. Transmisión de Datos CS en un Sistema de Comunicación	30
2.2.1. Sistemas de Comunicación Inalámbricos	31
2.2.2. Transmisión de Datos con Bluetooth	33
3. Resultados	37
3.1. Compresión y Reconstrucción con Compressive Sensing sobre Dispositivos Móviles	37
3.1.1. Calidad de la Reconstrucción	38
3.1.2. Tiempo de Reconstrucción	44
3.1.3. Consumo de Energía de los Algoritmos de Reconstrucción	45
3.2. Simulación de la Transmisión de Datos Tomados con CS	46
3.2.1. Tasa de Error por Bit	47
3.2.2. Rendimiento en el Canal	47

4. Conclusiones	50
4.1. Contribuciones	51

Índice de figuras

1.1. Vector de mediciones $y = \Phi f$	14
1.2. Etapas de la técnica de CS sobre imágenes en un sistema utilizando dispositivos Smartphone. a) La imagen es detectada/comprimida por la cámara CS del dispositivo; b) La imagen detectada es recuperada por un algoritmo de reconstrucción en el dispositivo móvil.	14
2.1. Estructura de un bloque sobrepuesto $F_{m,n}$ con un tamaño de bloque b y un factor de sobreposición Δ	25
2.2. Diagrama de bloques para el proceso de a) Compresión OBB-CS y b) Reconstrucción OBB-CS	26
2.3. Tipos de bloques para una imagen con tamaño de bloque b y factor de sobreposición Δ	27
2.4. Diagrama de bloques para el proceso de transmisión de datos muestreados y comprimidos con CS.	32
3.1. Resultados PSNR para la imagen Lena utilizando dos matrices de muestreo diferentes, varios niveles de dispersión k/N de la imagen y diferentes porcentajes M/N para el total de muestras. a) Recuperación de la imagen Lena con la matriz Gaussiana; b) Recuperación de la imagen Lena con la matriz SBHE.	39

3.2. Resultados PSNR para la imagen Barco utilizando dos matrices de muestreo diferentes, varios niveles de dispersión k/N de la imagen y diferentes porcentajes M/N para las mediciones. a) Recuperación de la imagen Barco con la matriz Gaussiana; b) Recuperación de la imagen Barco con la matriz SBHE.	40
3.3. Resultados de reconstrucción para la imagen Lena de 48×48 píxeles; a) Imagen original; b) recuperación con el algoritmo IHT en Matlab; c) recuperación IHT en el dispositivo iOS; d) recuperación con el algoritmo OMP en Matlab; e) recuperación OMP en el dispositivo iOS.	41
3.4. Resultados de reconstrucción para la imagen Barco de 48×48 píxeles; a) Imagen original; b) recuperación con el algoritmo IHT en Matlab; c) recuperación IHT en el dispositivo iOS; d) recuperación con el algoritmo OMP en Matlab; e) recuperación OMP en el dispositivo iOS.	42
3.5. Resultados de reconstrucción para la imagen Lena utilizando el enfoque OBB-CS con el algoritmo OMP; (Columna izquierda) Imágenes originales de 128×128 y 256×256 píxeles; (Columna Central) Imagen Recuperada en la Plataforma PC; (Columna Derecha) Imágenes recuperadas en la Plataforma iOS utilizando un factor de sobreposición $\Delta = 8$ para imágenes de 128×128 y $\Delta = 4$ para imágenes de 256×256 . El porcentaje de muestras M/N es del 60%.	43
3.6. Resultados de reconstrucción para la imagen Barco utilizando el enfoque OBB-CS con el algoritmo OMP; (Columna izquierda) Imágenes originales de 128×128 y 256×256 píxeles; (Columna Central) Imagen Recuperada en la Plataforma PC; (Columna Derecha) Imágenes recuperadas en la Plataforma iOS utilizando un factor de sobreposición $\Delta = 8$ para imágenes de 128×128 y $\Delta = 4$ para imágenes de 256×256 . El porcentaje de muestras M/N es del 60%.	44

3.7. Tiempo de recuperación requerido por (a) el algoritmo OMP y (b) el algoritmo IHT para recuperar bloques de tamaño $b = 32$ píxeles para imágenes de 128×128 y 256×256 píxeles.	45
3.8. Diagrama de cajas de los niveles de consumo de energía para los algoritmos OMP y IHT sobre la plataforma iOS tomando diferentes porcentajes de muestras de las imágenes.	46
3.9. Tasa de Error por Bit del canal de comunicación utilizando modulación FSK.	47
3.10. Rendimiento del sistema de comunicación (Throughput) en bits por segundo [bps] para diferentes valores de la calidad del canal cuando se transmiten los datos de una imagen de 256×256 en un sistema de comunicación utilizando la modulación FSK. . . .	49
3.11. Rendimiento del sistema de comunicación (Throughput) en paquetes por segundo [pps] para diferentes valores de la calidad del canal cuando se transmiten los datos de una imagen de 256×256 en un sistema de comunicación utilizando la modulación FSK.	49

Resumen

TITULO: TRANSMISIÓN DE IMÁGENES DIGITALES APLICANDO LA TÉCNICA DE MUESTREO Y COMPRESIÓN A TRAVÉS DE UN SISTEMA DE COMUNICACIÓN INALÁMBRICO ¹

AUTOR: IRENE LIZETH MANOTAS GUTIÉRREZ ²

PALABRAS CLAVE: Compressive Sensing, Dispositivos Móviles, Transmisión de Datos

El presente trabajo analiza la técnica *Compressive Sensing* (CS) cuando se procesan imágenes muestreadas y comprimidas con dicha técnica sobre dispositivos móviles y cuando las imágenes previamente procesadas son enviadas a través de un enlace de comunicación inalámbrico. Compressive Sensing (CS) es una nueva técnica que simultáneamente comprime y muestrea una imagen tomando un conjunto de proyecciones aleatorias de una escena. Un algoritmo de optimización es empleado para reconstruir la imagen utilizando las proyecciones aleatorias. Diferentes algoritmos de optimización se han diseñado para obtener de manera eficiente una correcta reconstrucción de la señal original. En la práctica estos algoritmos se han restringido a implementaciones de CS en arquitecturas de alto rendimiento computacional, como computadores de escritorio o unidades de procesamiento gráfico, debido a el gran número de operaciones requeridas por el proceso de reconstrucción. En este trabajo se presenta el análisis y resultados de implementación de la técnica CS sobre dispositivos de recursos limitados como dispositivos móviles celulares. Los resultados se evalúan en términos de la calidad de reconstrucción de la imagen, el tiempo de reconstrucción y el consumo de energía en el dispositivo. Además se analiza el efecto en el canal de comunicación cuando la técnica CS es usada para los datos transmitidos en un sistema de comunicación, los resultados se evalúan en términos de la relación señal/ruido y el rendimiento del canal.

¹Trabajo de Grado

²Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas. Director, Henry Arguello Fuentes. Codirector, Fernando A. Rojas Morales.

Abstract

TITLE: TRANSMISSION OF DIGITAL IMAGES APPLYING THE COMPRESSIVE SENSING TECHNIQUE THROUGH A WIRELESS COMMUNICATION SYSTEM ¹

AUTHOR: IRENE LIZETH MANOTAS GUTIÉRREZ ²

KEYWORDS: Compressive Sensing, Mobile Devices, Data Transmission.

This work analyzes the compression technique called Compressive Sensing (CS) on images that are sampled and compressed using this technique on mobile phones and when the previously processed images are sent via a communication link. Compressive Sensing (CS) is a new technique that simultaneously senses and compresses an image by taking a set of random projections from the underlying scene. An optimization algorithm is then used to recover the initial image. In practice, these optimization algorithms have restricted CS techniques to be implemented on high performance computational architectures, such as personal computers or graphical processing units (GPU) due the huge number of operations required for the image recovery. This work presents the analysis and results in the implementation of the Compressive Sensing (CS) technique on resource-constrained devices such as mobile phones. The results are evaluated in terms of the quality of image reconstruction, reconstruction time and power consumption on the device. Furthermore, this work analyzes the effect in the communication channel when the CS technique is applied in the data that is transmitted in a communication system, the results are evaluated in terms of the signal-to-noise ratio and the throughput of the channel.

¹Research Work

²Faculty of Physical-Mechanical Engineering. Advisor, Henry Arguello Fuentes. Co-advisor, Fernando Antonio Rojas Morales.

Introducción

En la adquisición de señales, tales como imágenes o vídeos, la tasa de muestreo de Nyquist impone que una gran cantidad de datos necesita ser adquirida para que una señal pueda ser reconstruida. Tradicionalmente, el enorme volumen de información recolectado mediante el cumplimiento de los criterios de Nyquist debe ser comprimido para posteriormente poder almacenar y transmitir los datos adquiridos. Este proceso de primero adquirir una gran cantidad de información para luego eliminar una gran parte de ésta es ineficiente. El proceso de muestreo tradicional de Nyquist se puede sustituir por la técnica de muestreo y compresión Compressive Sensing (CS) [1], [2], que sólo adquiere los componentes pertinentes de la señal subyacente a través del muestreo y compresión de la señal en un solo paso. CS explota la propiedad de dispersión o *sparsity* de una señal (por ejemplo, una imagen) para adquirir su información más relevante utilizando los productos internos de la señal subyacente con vectores aleatorios. En CS, solo se requiere que $M \ll N$ muestras de la señal $\mathbf{f} \in \mathbb{R}^N$ sean adquiridas. Posteriormente, la señal se reconstruye mediante la solución de un problema inverso, utilizando un programa lineal [3] o un algoritmo codicioso [4]. En general, la técnica de CS toma una imagen \mathbf{F} en su representación vectorial \mathbf{f} , la cual es dispersa en alguna base de representación Ψ , de tal manera que se puede aproximar por una combinación lineal de k vectores de Ψ con $k \ll N$, donde N representa la dimensión de la imagen y k representa la dispersión, es decir, el número de elementos distintos de cero en la base de representación de la imagen. La teoría de CS establece que \mathbf{f} se puede recuperar con alta probabilidad tomando M proyecciones aleatorias de la imagen, donde $M \ll k \log(N) \ll N$ [5], [6]. Las proyecciones están dadas por $\mathbf{y} = \Phi \mathbf{f} = \Phi \Psi \boldsymbol{\theta}$, donde Φ es una matriz de

muestreo aleatoria de $M \times N$, cuyas filas son incoherentes (o estadísticamente independientes) con las columnas de la matriz Ψ , es decir, todos los productos internos entre las matrices Φ y Ψ son pequeños. En la literatura, algunas matrices de muestreo Φ utilizadas incluyen la configuración de Gauss (GE), la configuración de Signos Simétricos (SSE) y la configuración de Bloques Hadamard Aleatorios (SBHE), entre otras. Por otro lado, para el proceso de recuperación de la imagen se busca el vector de aproximación f que satisface la ecuación $y = \Phi f$. Dado que el vector de proyecciones, o vector de muestras y tiene dimensión M , donde $M \ll N$, existe un número infinito de soluciones que satisfacen ésta ecuación. Por lo tanto, para buscar un vector solución es común optimizar la medida de dispersión k de la imagen, donde el problema de encontrar un vector con el menor número de elementos distintos de cero está dado por:

$$\min_f \|f\|_0 \text{ Sujeto a } \|y - \Phi f\|_2 < \varepsilon, \quad (1)$$

donde la norma l_0 , representada por $\|f\|_0$, cuenta el número de elementos distintos de cero en f , y ε es un valor de tolerancia. Sin embargo, el problema presentado por (1) es un problema de optimización no lineal, que se ha demostrado es NP duro. Por lo tanto, diferentes estrategias sub-óptimas se han utilizado para resolver el problema descrito en (1). En la práctica, las estrategias más comunes para resolver (1) incluyen las técnicas de relajación convexa, las técnicas de optimización local no convexa y las estrategias de búsqueda codiciosa. Estos métodos de reconstrucción utilizan algoritmos de orden $O(MN)$ o $O(N \log M)$, dependiendo del enfoque seleccionado, para encontrar una solución aproximada del problema original. En la literatura se han propuesto varios algoritmos que utilizan las estrategias mencionadas anteriormente para la reconstrucción de imágenes dispersas. En concreto, los enfoques de búsqueda codiciosa, cuyo funcionamiento básico es encontrar los soportes de la señal desconocida secuencialmente, han demostrado ser relativamente rápidos en comparación con otros enfoques y, a menudo se consideran como la única forma práctica de resolver grandes problemas de aproximación dispersa. Los algoritmos Iterative Hard Thresholding (IHT)

y Orthogonal Matching Pursuit (OMP) [1] son dos ejemplos de algoritmos de recuperación que han demostrado su gran potencial para recuperar una señal comprimida a través de la técnica de CS. Por otro lado, las características de la técnica CS ha permitido que ésta sea ampliamente aplicada en diferentes campos, como en la captura de imágenes (por ejemplo, en el diseño de la cámara de un solo píxel [2]), en óptica (por ejemplo, en imágenes hiperespectrales [3, 4]), y en comunicaciones (por ejemplo, en transmisiones sobre canales inalámbricos [5]). En aplicaciones involucrando dispositivos embebidos o móviles en conjunto con la técnica CS, se encuentran aquellas que incluyen la transmisión de datos en redes de sensores inalámbricos [6], redes de sensores de telecardiología [7] y redes inalámbricas de sensores corporales (por ejemplo, en [8] utilizando un dispositivo iPhone). Sin embargo, los trabajos que involucran el uso de dispositivos móviles se limitan a analizar señales unidimensionales y se centran principalmente en dos tareas: recoger datos procedentes de diferentes nodos de una red y transmitir los datos comprimidos. Por el contrario, existen pocas investigaciones sobre CS aplicado en dispositivos embebidos como teléfonos inteligentes. Por lo tanto, teniendo en cuenta las ventajas de CS en la compresión de datos y el creciente interés en las tecnologías móviles y sus aplicaciones, es importante analizar las implicaciones en tiempo de ejecución y la calidad de la recuperación de CS mediante una implementación en un dispositivo móvil. En este trabajo se analizan los requerimientos y la implementación de los procesos de compresión y reconstrucción de imágenes en una plataforma de teléfonos inteligentes usando dos algoritmos de reconstrucción de señales dispersas y una estrategia de CS basada en bloques sobrepuestos denominada OBB-CS. Adicionalmente, se presenta un análisis del proceso de transmisión de los datos comprimidos con la técnica CS sobre un canal de comunicación y los efectos en la transmisión de datos comprimidos con la técnica de CS a través de simulaciones de un canal de comunicación dado.

Capítulo 1

Técnica de Muestreo y Compresión Aplicada en Imágenes

1.1. Compressive Sensing (CS)

La técnica de muestreo y compresión o Compressive Sensing (CS) aprovecha el hecho de que un gran número de señales encontradas en la práctica son dispersas, o *sparse*, en algún dominio de transformación como Wavelet o Curvelet, lo que significa que la mayor parte de la energía de la señal está concentrada en un porcentaje reducido de sus componentes k , k . CS explota esta dispersión, es decir, el número de elementos distintos de cero representado por k , al dictar que se necesita una cantidad mucho menor de muestras para sensar la imagen (Ver Figura 1.1) que cuando se utilizan enfoques tradicionales como el establecido por el criterio de Shannon/Nyquist, el cual requiere que la tasa de muestreo sea superior a dos veces la máxima frecuencia de la imagen k .

Para el muestreo y compresión de imágenes, la técnica CS establece que una imagen F , o su representación vectorial $f \in \mathbb{R}^N$, se puede recuperar con alta probabilidad cuando se utilizan solo M proyecciones aleatorias, donde $M \ll k \log(N) \ll N$. Una matriz de muestreo $\Phi \in \mathbb{R}^{M \times N}$ toma las proyecciones sobre el vector de la imagen, las cuales definen las muestras y datos

comprimidos de la imagen, y la razón M/N representa el porcentaje de muestras tomadas de la imagen.

Las matrices de muestreo se deben seleccionar de manera que se ajusten a la Propiedad Isométrica Restrictiva (RIP) [1], la cual proporciona las condiciones suficientes para garantizar un rendimiento casi óptimo utilizando algoritmos de reconstrucción, y además determina el número mínimo de mediciones necesarias para este proceso de reconstrucción. La literatura ha propuesto varias matrices que cumplen la condición de RIP, incluyendo las matrices cuyas entradas son generadas independiente e idénticamente distribuidas (i.i.d) de una distribución Normal o Bernoulli. La principal ventaja de estas matrices es que son incoherentes con cualquier señal dispersa, y, por tanto, el número de muestras necesarias para la reconstrucción exacta es mínima [2]. Sin embargo, las matrices de muestreo tienen varios inconvenientes relacionados con el tiempo de almacenamiento y costo computacional, por lo que deben ser analizadas antes de ser implementadas en dispositivos con recursos limitados como dispositivos móviles celulares.

Las principales características de las matrices de muestreo se explican con más detalle en la sección 1.2. En la Figura 1.2 se muestra un ejemplo de un sistema de captura/recuperación de imágenes utilizando la técnica de CS; el sistema comprende la captura y la reconstrucción de la imagen en el dispositivo móvil. Este sistema CS asume la incorporación de sensores CS, tales como el utilizado en la cámara de un solo píxel [3], en el dispositivo móvil para obtener las mediciones de la imagen. Los detalles sobre la aplicación óptica de sensores CS se pueden encontrar en [4].

1.2. Matrices de Muestreo

La matriz de muestreo $\Phi \in \mathbb{R}^{M \times N}$, también llamada matriz de Compressive Sensing, es un elemento clave en el proceso de reconstrucción dado que es la encargada de tomar las proyecciones de la imagen subyacente. En gene-

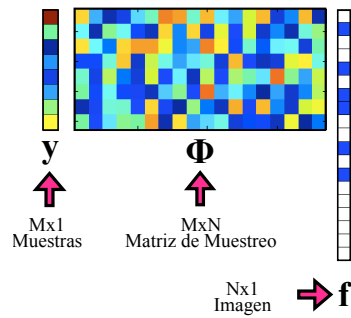


Figura 1.1: Vector de mediciones $y = \Phi f$

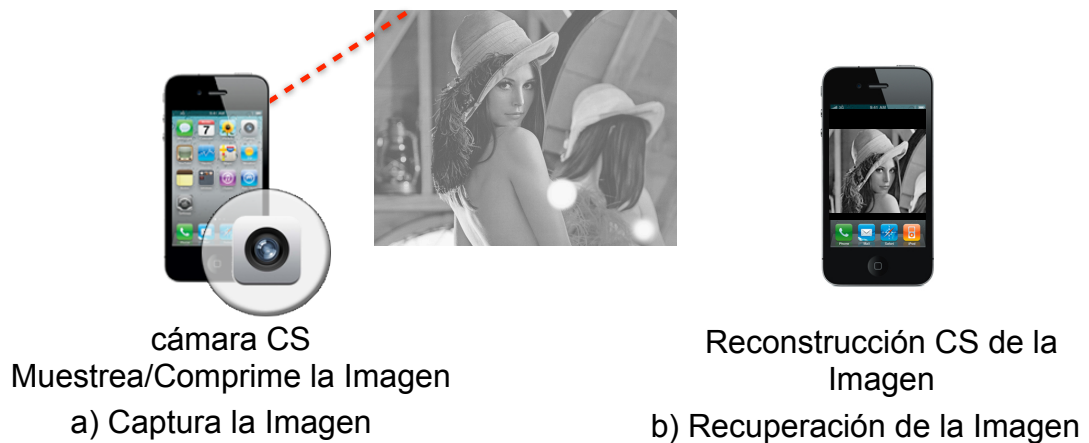


Figura 1.2: Etapas de la técnica de CS sobre imágenes en un sistema utilizando dispositivos Smartphone. a) La imagen es detectada/comprimida por la cámara CS del dispositivo; b) La imagen detectada es recuperada por un algoritmo de reconstrucción en el dispositivo móvil.

ral, las matrices de muestreo tienen que satisfacer la *Propiedad de Isometría Restrictiva* (RIP) definida en [1], [2], dado que es una condición suficiente para la reconstrucción de la imagen. Las matrices aleatorias, generadas con entradas i.i.d. de una distribución Normal o un proceso de Bernoulli, satisfacen la condición suficiente para la reconstrucción.

Algunas configuraciones populares de matrices de muestreo incluyen la *configuración Gaussiana aleatoria*, la *configuración parcial de Fourier* (PFE), la *configuración de Signos Simétricos* (SSE), la familia de matrices Bernoulli (*semi-Hadamard*, *parcial Hadamard*), la *matriz binaria sparse* (SBM) y la *matriz Rademacher* [3]. En el cuadro 1.1 se muestran las matrices de muestreo más comunes junto con sus descripciones, condición de dispersión, costo computacional y costo de almacenamiento. Dentro de este conjunto, las matrices Gaussiana y Bernoulli tienen dos grandes inconvenientes en su implementación: primero, los requerimientos de memoria por los elementos de la matriz y, segundo, la alta complejidad computacional debido a que éstas son matrices estructuradas.

La matriz PFE explota la propiedad de cálculo rápido de la Transformada Rápida de Fourier (FFT) y por lo tanto, reduce significativamente la complejidad del sistema de muestreo. Sin embargo, PFE es solo incoherente con señales dispersas en el dominio del tiempo, lo que reduce considerablemente su ámbito de aplicación. Las matrices de muestreo como la configuración de bloques Hadamard mezclados o *Scramble Block Hadamard Ensemble* (SBHE) [4], [5] o la SBM [6] se consideran más adecuadas para su aplicación debido al hecho de que éstas configuraciones son más dispersas y su cálculo es más rápido en comparación con la configuración Gaussiana aleatoria.

Teniendo en cuenta los recursos computacionales limitados de los dispositivos móviles y las propiedades de las matrices de medición descritas anteriormente, en este trabajo se realiza la comparación entre las matrices Gaussiana y SBHE, las cuales son analizadas para la implementación del sistema de imagen CS utilizando dispositivos móviles.

Cuadro 1.1: Descripción Matrices de Muestreo Φ

Nombre	Matriz Φ	Descripción	Dispersión	Costo Computacional	Almacenamiento
Gaussiana	$N(0, 1/m)$	Los elementos $\phi_{i,j}$ se escogen independientemente de una distribución Normal. Esta es una base incoherente universal.	$k \leq C(M/\log(N/M))$	$O(MN)$	$O(MN)$
SBHE	$Q_M W P_N$	Q_M es un vector aleatorio con las filas seleccionadas de la matriz Hadamard en bloques W de orden B , cuyas filas son reordenadas utilizando el operador aleatorio P_N .	$k \leq C(M/(\sqrt{N/B}(\log N)^2))$	$O(M \log(N))$	$O(M)$
PFE	$Q_M Y$	Selecciona M filas de la matriz de Fourier Parcial Y aleatoriamente usando Q .	$k \leq CM/(\log N)$	$O(M \log(N))$	$O(M)$
SBM	$\phi_{i,j} = 1/\sqrt{d}$	Cada columna tiene d elementos no cero igual a $1/\sqrt{d}$, donde $d < N$ y depende de la dispersión de la señal.	$k \leq M/\log(N/M)$	$O(N)$	$O(M)$

1.3. Algoritmos de Reconstrucción

La teoría de CS utiliza algoritmos para la recuperación de señales dispersas con el fin de reconstruir la imagen original. Dependiendo del algoritmo de recuperación utilizado se emplean diferentes estrategias para encontrar la reconstrucción de la imagen que mejor se aproxima a la imagen original. Los algoritmos *Greedy*, también conocidos como algoritmos codiciosos, son un tipo de algoritmos de aproximación de señales dispersas diseñados para encontrar la solución al siguiente problema de optimización k -sparse:

$$\min_{\hat{\theta}} \|y - \Phi \Psi \hat{\theta}\|_2 \quad \text{Sujeto a} \quad \|\hat{\theta}\|_0 < k. \quad (1.1)$$

La solución del problema (1.1) obtiene la mejor aproximación o estimador del vector de coeficientes θ de la imagen f , utilizando solo k columnas de la matriz de muestreo Φ , donde k representa la dispersión de la señal. Las es-

trategias Greedy constituyen una clase de técnicas subóptimas utilizadas en la práctica para encontrar el vector $\hat{\theta}$ con el menor número de elementos distintos de cero, donde $\hat{f} = \Psi\hat{\theta}$ es el estimador o vector de aproximación de la imagen f , y Ψ es la matriz de representación donde la imagen es dispersa y usualmente se toma como una matriz Wavelet cuando se trabaja con imágenes ?.

En general, la complejidad de los algoritmos de reconstrucción Greedy es más baja que los métodos de minimización l_1 ?. Algunos de los inconvenientes con los algoritmos Greedy están relacionados con el costo computacional implicado en el cálculo de la operación de proyección, lo que limita su aplicación a problemas pequeños. Sin embargo, las estrategias Greedy han mostrado reconstruir de manera eficiente las señales de muestras adquiridas con la técnica CS y además se caracterizan por tener éxito con un número mínimo de observaciones ?.

Por otro lado, los algoritmos de umbralización iterativa, como el algoritmo de umbral fuerte iterativo (*Iterative Hard Thresholding-IHT*), es otra clase de algoritmo Greedy que ha demostrado tener un buen rendimiento y reconstrucción, y además es un algoritmo capaz de encontrar la solución al problema con un número mínimo de observaciones ?. El algoritmo IHT relaja la penalización de la norma l_0 y la sustituye por la penalización de norma l_1 . Este algoritmo se ha diseñado para resolver el siguiente problema convexo:

$$\min_{\hat{\theta}} \|\hat{\theta}\|_1 : \|\mathbf{y} - \Phi\Psi\hat{\theta}\|_2^2 \leq \xi, \quad (1.2)$$

donde ξ es la tolerancia del error. En este trabajo, los métodos Greedy son estudiados debido a que éstos métodos tienen implementaciones rápidas y son computacionalmente menos exigentes que otras estrategias, como el método de búsqueda base con eliminación de ruido (*Basis Pursuit De-noising - BPDN*) ? y, por tanto, los hace más adecuados para una implementación sobre dispositivos móviles. Se escogieron dos algoritmos para ser analizados: El algoritmo IHT ? y el algoritmo *Orthogonal Matching Pursuit* (OMP) ?. Por

consiguiente, la descripción de esos algoritmos se da a continuación.

1.3.1. Algoritmo Iterative Hard Thresholding

El algoritmo Iterative Hard Thresholding (IHT) es un método de búsqueda codiciosa iterativo que no requiere inversión de la matriz y proporciona garantías de error en un tiempo óptimo $\mathcal{O}(k \log(n/k))$. El algoritmo calcula la solución $\hat{\theta}$ en la iteración (t) como:

$$\hat{\theta}^{(t+1)} = H_k(\hat{\theta}^{(t)} + \lambda \mathbf{A}^T(\mathbf{y} - \mathbf{A}\hat{\theta}^{(t)})) \quad (1.3)$$

donde la matriz \mathbf{A} es el producto de las matrices de muestreo y representación $\mathbf{A} = \Phi\Psi$, y $H_k(\cdot)$ es el operador de umbral fuerte, el cual es un operador no lineal que transforma todos los k elementos de un vector a cero sin incluir los k elementos más grandes en magnitud. El parámetro λ representa el tamaño del paso, y $\hat{\theta}^{(0)}$ es un vector de ceros. El Algoritmo 1 muestra los pasos requeridos por el algoritmo IHT para encontrar la solución $\hat{\theta}$. Dos características principales de este algoritmo incluyen que converge en tiempo lineal, debido al hecho de que se basa en una estrategia de gradiente descendente, y que la selección del tamaño del paso necesita ser elegido apropiadamente $\lambda, \lambda, \lambda$.

Algorithm 1 Algoritmo IHT.

Require: $\mathbf{y}, \mathbf{A}, k, \lambda, \xi, \text{nIter}$

Ensure: $\hat{\theta}^{(t+1)}$
 $\hat{\theta}^{(0)} \leftarrow \mathbf{0}$
for $t = 0$ hasta nIter **do**
 $\mathbf{v}^{(t)} = \mathbf{y} - \mathbf{A}\hat{\theta}^{(t)}$
 $\mathbf{u}^{(t+1)} = \hat{\theta}^{(t)} + \lambda \mathbf{A}^T \mathbf{v}^{(t)}$
 $\hat{\theta}^{(t+1)} = H_k(\mathbf{u}^{(t+1)})$
 if $\|\mathbf{v}^{(t)}\| \leq \xi$ **then**
 return $\hat{\theta}^{(t+1)}$
 end if
end for

1.3.2. Algoritmo Orthogonal Matching Pursuit

Orthogonal Matching Pursuit (OMP) es una estrategia Greedy que selecciona iterativamente los elementos en el vector de aproximación de una imagen resolviendo un problema de mínimos cuadrados. La aproximación de la imagen representada por $\hat{\mathbf{f}} = \Psi \hat{\boldsymbol{\theta}}$, donde Ψ es la base de representación dispersa de la imagen y $\hat{\boldsymbol{\theta}}$ es el estimador del vector de coeficientes de la imagen en el dominio Ψ , se calcula en la iteración (t) utilizando:

$$\hat{\boldsymbol{\theta}}^{(t)} = \mathbf{A}_{\Gamma^{(t)}}^{\dagger} \mathbf{y}, \quad (1.4)$$

donde Γ representa un conjunto de índices seleccionados en base al producto interior entre el residuo actual $\mathbf{r}^{(t)} = \mathbf{y} - \hat{\mathbf{y}}$, y las columnas de la matriz de muestreo $\mathbf{A} = \Phi \Psi$, y el vector $\hat{\mathbf{y}}$ está definido como $\hat{\mathbf{y}} = \mathbf{A} \hat{\boldsymbol{\theta}}^{(t)}$. La submatriz $\mathbf{A}_{\Gamma^{(t)}}$ se construye utilizando solo aquellas columnas de \mathbf{A} cuyos índices se encuentren en el conjunto Γ en la iteración (t); y la pseudo-inversa de la matriz $\mathbf{A}_{\Gamma^{(t)}}$ está representada por $\mathbf{A}_{\Gamma^{(t)}}^{\dagger}$. El Algoritmo 2 muestra el pseudocódigo de la estrategia utilizada por el algoritmo OMP, el cual identifica cuáles columnas de la matriz \mathbf{A} participan en el vector de muestras y para encontrar una mejor aproximación del vector $\boldsymbol{\theta}$ en cada iteración resolviendo el problema de mínimo cuadrados dado por $\hat{\boldsymbol{\theta}}^{(t)} = \arg \min_{\boldsymbol{\theta}^{(t)}} \|\mathbf{y} - \mathbf{A}_{\Gamma^{(t)}} \boldsymbol{\theta}^{(t)}\|_2$. En cada iteración, el algoritmo OMP se selecciona la columna de \mathbf{A} que está más correlacionada con el residuo \mathbf{r} . Al finalizar, la aproximación $\hat{\boldsymbol{\theta}}^{(t)}$ de los coeficientes de imagen, tiene los índices de los elementos diferentes de cero en los componentes listados en $\Gamma^{(t)}$.

En el cuadro 1.2 se describen los algoritmos OMP y IHT en términos del número de operaciones, el almacenamiento y la complejidad computacional requerida, donde q y p representan el número de iteraciones en los algoritmos IHT y OMP respectivamente; κ denota la complejidad de la aplicación del operador \mathbf{A} y \mathbf{A}^T , la cual puede ser $O(MN)$ o $O(N \log M)$ dependiendo de si el operador es estructurado, por ejemplo al utilizar la configuración Gaussiana, o no estructurado, por ejemplo al utilizar la configuración de bloques Hadamard mezclados (SBHE), y c representa el número de columnas seleccionadas que

Algorithm 2 Algoritmo OMP.**Require:** $\mathbf{y}, \mathbf{A}, k, e, \text{nIter}$ **Ensure:** $\hat{\boldsymbol{\theta}}^{(t)}$ $\mathbf{r}^{(0)} \leftarrow \mathbf{y}, \Gamma^{(0)} \leftarrow \emptyset, \hat{\boldsymbol{\theta}}^0 \leftarrow \mathbf{0}$ **for** $t = 1$ hasta nIter **do** $\mathbf{g}^{(t)} = \mathbf{A}^T \mathbf{r}^{(t-1)}$ $j = \max_j |\mathbf{g}^{(t)}|$ $\Gamma^{(t)} = \Gamma^{(t-1)} \cup j$ $\hat{\boldsymbol{\theta}}^{(t)} = \mathbf{A}_{\Gamma^{(t)}}^\dagger \mathbf{y}$ $\hat{\mathbf{y}} = \mathbf{A} \hat{\boldsymbol{\theta}}^{(t)}$ $\mathbf{r}^{(t)} = \mathbf{y} - \hat{\mathbf{y}}$ **if** $\|\mathbf{r}^{(t)}\| \leq e$ **then****return** $\hat{\boldsymbol{\theta}}^{(t)}$ **end if****end for****return** $\hat{\boldsymbol{\theta}}^{(t)}$

Algoritmo	IHT	OMP
Número de Operaciones	$M + N + k + O(\kappa) + [N \log N]$	$2M + N + 4O(\kappa) + [N \log N]$
Costo de Almacenamiento	$M + N + MN + k$	$[M + N + MN] + Mc + c$
Complejidad Computacional	$O(q\kappa)$	$O(p\kappa)$

Cuadro 1.2: Requisitos computacionales de los algoritmos de recuperación IHT y OMP. El operador $O(\kappa)$ representa la complejidad de aplicar el operador \mathbf{A} y \mathbf{A}^T , y c representa el número de columnas seleccionadas para el algoritmo de OMP.

varía de 1 a k para el algoritmo OMP.

Como el cuadro 1.2 indica, el algoritmo OMP tiene mayores requerimientos de almacenamiento y número de operaciones que el algoritmo IHT. Sin embargo, el número de iteraciones requeridas por cada algoritmo al final determina el costo computacional de cada uno; el número de iteraciones está determinado para el algoritmo OMP igual que la dispersión k de la imagen, y para el algoritmo IHT por el error en la aproximación. Además del costo computacional de cada algoritmo, es importante también conocer la calidad de la reconstrucción

de la imagen obtenida por cada uno de los algoritmos, lo cual se discute en el capítulo 3.

Capítulo 2

Algoritmo de Reconstrucción CS Basado en Bloques y Transmisión de Datos CS

En este capítulo se presenta la descripción de los principales aportes de este documento, que incluyen el algoritmo propuesto para la reconstrucción de imágenes en dispositivos móviles cuando éstas han sido muestreadas y comprimidas con la técnica de Compressive Sensing (CS), y la simulación del sistema de comunicación para una transmisión de datos utilizando CS.

2.1. Algoritmo de Reconstrucción Basado en Bloques Sobrepuestos

Este trabajo describe un nuevo algoritmo de recuperación de imágenes muestreadas y comprimidas con la técnica de CS a través de un enfoque basado en bloques sobrepuestos para su aplicación en dispositivos móviles ?. Para la implementación del sistema de recuperación de una imagen muestreada y comprimida con CS, este trabajo asume que el dispositivo móvil tiene incorporado un sensor de CS, como el descrito en la cámara de pixel sencillo ?, que es el encargado de adquirir las muestras de la escena o imagen.

El algoritmo de reconstrucción de CS propuesto, de ahora en adelante denominado enfoque OBB-CS, se basa en una estrategia de bloques sobrepuestos y se diseñó con el propósito de reconstruir imágenes de gran tamaño en un dispositivo móvil cuando se utiliza la técnica de CS para tomar las muestras de la imagen. El enfoque OBB-CS explota la estructura de la imagen y de la matriz de muestreo durante el proceso de reconstrucción de la imagen. Dadas las características de almacenamiento de memoria limitadas del dispositivo móvil, el enfoque OBB-CS permite analizar imágenes de gran tamaño y extender los resultados obtenidos en [?], en donde se realizó la reconstrucción, sobre un dispositivo móvil, de imágenes de 48×48 píxeles adquiridas con CS.

En la literatura, algunos trabajos relacionados con la reconstrucción CS de imágenes utilizando un enfoque basado en bloques incluyen el algoritmo OMP basado en bloques propuesto por Parichat et al [?], en donde el algoritmo propuesto reconstruye los bloques de una imagen de 64×64 píxeles y luego reorganiza la imagen utilizando cada uno de los bloques reconstruidos. Sin embargo, la descripción del enfoque utilizado por Parichat et al utiliza $M = N$ muestras, lo cual se traduce en la no compresión de la imagen dado que el número de muestras es igual al tamaño de la imagen original. Similarmente, Gan propone un enfoque de reconstrucción CS de imágenes basado en bloques en [?], el cual divide la imagen original en bloques pequeños y no sobrepuestos de 32×32 píxeles. Sin embargo, el algoritmo de reconstrucción utilizado por Gan utiliza un filtro Wiener de 3×3 para reducir el efecto de artefactos¹ y así suavizar la imagen, lo cual involucra un mayor requerimiento computacional para hallar la solución y por lo tanto, no es apropiado para su implementación en un dispositivo móvil.

Por otro lado, la reconstrucción de imágenes con el enfoque OBB-CS utiliza un modelo diferente cuya característica principal es dividir la imagen en bloques pequeños de $b \times b$ píxeles. Cada bloque es comprimido para luego ser transmitido/almacenado en otro dispositivo, el cual se encarga de reconstruir

¹El efecto de artefactos representa la distorsión en la imagen producida por la división en bloques.

la imagen original con un algoritmo de reconstrucción Greedy que recupera los bloques de la imagen y obtiene una aproximación de ésta a través de una reorganización de los bloques recuperados. A continuación se describe con más detalle el enfoque de reconstrucción basado en bloques sobrepuestos denominado OBB-CS.

Enfoque de Bloques Sobrepuestos: OBB-CS

El modelo del enfoque OBB-CS consiste en explotar la estructura de la matriz de muestreo Φ para recuperar una aproximación de la imagen original utilizando las muestras tomadas con CS de varios bloques extraídos de la imagen original. En OBB-CS, la imagen \mathbf{F} se expresa como una configuración de $q \times q$ submatrices de dimensión $b \times b$ representadas por $\mathbf{F}_{m,n}$ y definida por

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{0,0} & \mathbf{F}_{0,1} & \cdots & \mathbf{F}_{0,q-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{F}_{q-1,0} & \mathbf{F}_{q-1,1} & \cdots & \mathbf{F}_{q-1,q-1} \end{bmatrix} \quad (2.1)$$

donde cada submatriz, representada por $\mathbf{F}_{m,n}$, es un bloque sobrepuesto perteneciente a la imagen \mathbf{F} .

El tamaño b define la dimensión de cada bloque de la imagen y es un parámetro que se puede escoger cumpliendo dos criterios: i) b debe ser un número no primo; y ii) b debe ser un divisor del número total de píxeles N de la imagen. La selección de b permite dividir la imagen en un número fijo de bloques sobrepuestos de igual tamaño, y además evita añadir ceros para que los bloques tengan el mismo tamaño (proceso conocido como *zero padding*). El número de bloques q de la imagen depende del número total de píxeles N de la imagen, y del tamaño b de los bloques; este valor es calculado como $q = \lceil \frac{\sqrt{N}}{b} + 1 \rceil$.

Los bloques tomados de la imagen son bloques sobrepuestos, es decir, los bloques comparten píxeles entre sí de acuerdo a un factor de sobreposición

Δ . El factor de superposición entre bloques Δ , define la cantidad de píxeles superpuestos entre bloques consecutivos en la imagen. Este factor de superposición se calcula como $\Delta = (qb - \sqrt{N}) / (q - 1)$. La Figura 2.1 muestra la estructura de un bloque superpuesto $F_{m,n}$ de la imagen F . Las secciones del bloque superpuesto marcadas como $A_{m,n}$, $B_{m,n}$, $C_{m,n}$, $D_{m,n}$, $Z_{x_{m,n}}$, $Z_{y_{m,n}}$, $Z_{w_{m,n}}$ y $Z_{z_{m,n}}$, representan las secciones superpuestas o compartidas entre bloques consecutivos de la imagen, mientras que la sección $E_{m,n}$ representa la región no compartida del bloque.

Cada bloque superpuesto $F_{m,n}$ puede ser representado como el vector $f_{m,n} \in \mathbb{R}^{b^2 \times 1}$; y el vector de muestras $y_{m,n}$ tomado de cada bloque superpuesto se calcula como $y_{m,n} = A_{m,n}^\gamma f_{m,n}$, donde $A_{m,n}^\gamma \in \mathbb{R}^{h \times b^2}$ es la matriz de muestreo del bloque con $h = M/q$.

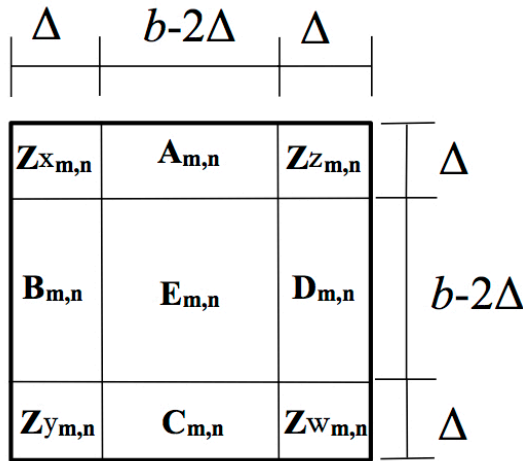


Figura 2.1: Estructura de un bloque superpuesto $F_{m,n}$ con un tamaño de bloque b y un factor de superposición Δ .

Reconstrucción con el Enfoque OBB-CS

El vector de reconstrucción $y_{m,n}$, correspondiente a cada bloque superpuesto $F_{m,n}$, es utilizado para obtener el correspondiente bloque de aproximación $\hat{F}_{m,n}$ utilizando los algoritmos OMP e IHT descritos en el capítulo 1. El bloque reconstruido $\hat{F}_{m,n}$ tiene también las secciones $\hat{A}_{m,n}$, $\hat{B}_{m,n}$, $\hat{C}_{m,n}$, $\hat{D}_{m,n}$,

$\hat{Z}_x, \hat{Z}_y, \hat{Z}_w$ y \hat{Z}_z , similar a como se muestra en la Figura 2.1. La imagen completa reconstruida F' es formada utilizando las versiones reorganizadas $F'_{m,n}$ para cada uno de los bloques de aproximación $\hat{F}_{m,n}$. La Figura 2.2 muestra el diagrama de bloques que representa los procesos de compresión y reconstrucción de una imagen utilizando los bloques de aproximación $\hat{F}_{m,n}$ de la imagen.

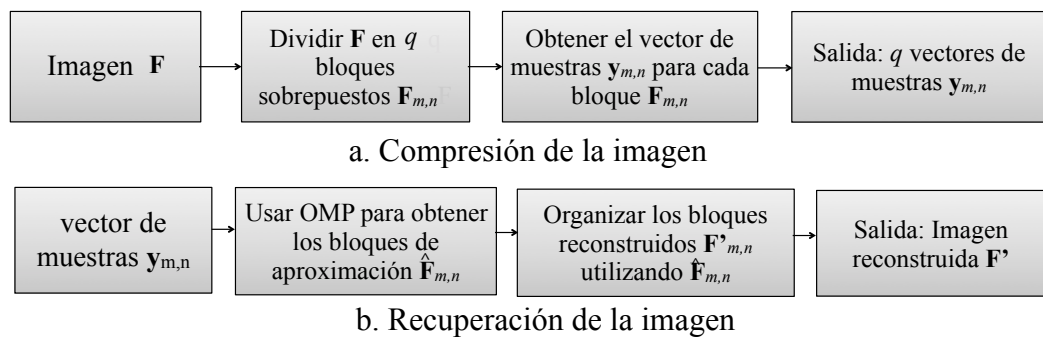


Figura 2.2: Diagrama de bloques para el proceso de a) Compresión OBB-CS y b) Reconstrucción OBB-CS

Un ejemplo de la división en bloques de una imagen se presenta en la Figura 2.3, donde la imagen se divide en nueve bloques que representan los tipos de bloques definidos en la estructura propuesta de bloques sobrepuestos. Los tipos de bloques son enumerados desde I hasta IX , donde cada bloque tiene un factor de sobreposición Δ y un tamaño de bloque $b \times b$. Para una dimensión reducida del tamaño del bloque b , el número total de bloques en la imagen aumenta pero los tipos de bloques ilustrados en la Figura 2.3 se mantienen. Para hallar cada una de las versiones organizadas $F'_{m,n}$ de los bloques de la imagen, se tiene en cuenta el cálculo del promedio de las regiones compartidas entre bloques consecutivos, la cual depende del tipo de bloque y de la posición del bloque de aproximación relacionado $\hat{F}_{m,n}$. La estimación de los bloques reorganizados $F'_{m,n}$ se realiza de la siguiente manera:

- Para los bloques tipo I , II , IV y V , el bloque reorganizado $F'_{m,n}$ se ex-

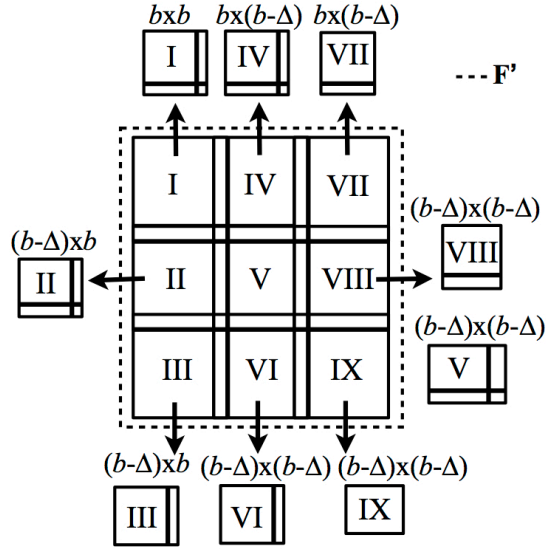


Figura 2.3: Tipos de bloques para una imagen con tamaño de bloque b y factor de sobreposición Δ .

presa como:

$$\mathbf{F}'_{m,n} = \begin{bmatrix} \mathbf{E}'_{m,n} & \mathbf{D}'_{m,n} \\ \mathbf{C}'_{m,n} & \mathbf{Z}'_{m,n} \end{bmatrix} \quad (2.2)$$

donde $\mathbf{E}'_{m,n}$, $\mathbf{D}'_{m,n}$, $\mathbf{C}'_{m,n}$ y $\mathbf{Z}'_{m,n}$ son las regiones del bloque no compartida, inferior compartida, lateral derecha compartida, e esquinera compartida respectivamente, las cuales están definidas por las ecuaciones (2.3) - (2.6) para el bloque tipo I .

$$\mathbf{E}'_{m,n} = \hat{\mathbf{A}}_{m,n} + \hat{\mathbf{B}}_{m,n} + \hat{\mathbf{E}}_{m,n} + \hat{\mathbf{Z}}\mathbf{x}_{m,n}. \quad (2.3)$$

$$\mathbf{C}'_{m,n} = (\hat{\mathbf{C}}_{m,n} + \hat{\mathbf{Z}}\mathbf{y}_{m,n} + \hat{\mathbf{A}}_{m+1,n} + \hat{\mathbf{Z}}\mathbf{x}_{m+1,n})/2. \quad (2.4)$$

$$\mathbf{D}'_{m,n} = (\hat{\mathbf{D}}_{m,n} + \hat{\mathbf{Z}}\mathbf{z}_{m,n} + \hat{\mathbf{B}}_{m,n+1} + \hat{\mathbf{Z}}\mathbf{x}_{m,n+1})/2. \quad (2.5)$$

$$\mathbf{Z}'_{m,n} = (\hat{\mathbf{Z}}\mathbf{w}_{m,n} + \hat{\mathbf{Z}}\mathbf{z}_{m+1,n} + \hat{\mathbf{Z}}\mathbf{y}_{m,n+1} + \hat{\mathbf{Z}}\mathbf{x}_{m+1,n+1})/4. \quad (2.6)$$

Para las regiones $\mathbf{C}'_{m,n}$ y $\mathbf{D}'_{m,n}$ se toma el promedio entre los dos bloques sobrepuestos que están compartiendo la misma sección de píxeles de la imagen, mientras que para la región $\mathbf{Z}'_{m,n}$ se calcula el promedio de los píxeles compartidos entre 4 bloques debido a que esta región esquinera se sobrepone entre los cuatro bloques adyacentes, como se muestra en la Figura 2.3.

- El bloque tipo *II* tiene una región no compartida $\mathbf{E}'_{m,n}$ más pequeña en comparación con el bloque tipo *I*, por lo tanto ésta región y la región lateral derecha compartida $\mathbf{D}'_{m,n}$ se calculan por medio las siguientes ecuaciones:

$$\mathbf{E}'_{m,n} = \hat{\mathbf{B}}_{m,n} + \hat{\mathbf{E}}_{m,n}, \quad (2.7)$$

$$\mathbf{D}'_{m,n} = (\hat{\mathbf{D}}_{m,n} + \hat{\mathbf{B}}_{m,n+1})/2; \quad (2.8)$$

mientras las regiones $\mathbf{C}'_{m,n}$ y $\mathbf{Z}'_{m,n}$ se calculan para el bloque tipo *II* como se muestra en la ecuaciones (2.4) y (2.6).

- Para el bloque tipo *IV* las regiones no compartida $\mathbf{E}'_{m,n}$ e inferior compartida $\mathbf{C}'_{m,n}$ se calculan mediante

$$\mathbf{E}'_{m,n} = \hat{\mathbf{A}}_{m,n} + \hat{\mathbf{Z}}\mathbf{z}_{m,n} + \hat{\mathbf{E}}_{m,n} + \hat{\mathbf{D}}_{m,n} \quad (2.9)$$

$$\mathbf{C}'_{m,n} = (\hat{\mathbf{C}}_{m,n} + \hat{\mathbf{B}}_{m+1,n})/2; \quad (2.10)$$

y las regiones lateral derecha compartida $\mathbf{D}'_{m,n}$ y esquinera $\mathbf{Z}'_{m,n}$ se calculan como se muestra en las ecuaciones (2.5) y (2.6).

- El bloque tipo *V* tiene la misma región no compartida que el bloque de aproximación, es decir, $\mathbf{E}'_{m,n} = \hat{\mathbf{E}}_{m,n}$, y las regiones compartidas esqui-

nera $\mathbf{Z}'_{m,n}$, letral derecha compartida $\mathbf{D}'_{m,n}$ e inferior compartida $\mathbf{C}'_{m,n}$ se calculan a través de las ecuaciones (2.6), (2.8) y (2.10) respectivamente.

- Para los tipos de bloques *III* y *IV*, el bloque reconstruido solo comparte la sección lateral derecha $\mathbf{D}'_{m,n}$ con el bloque adjacente. Por lo tanto, el bloque reorganizado $\mathbf{F}'_{m,n}$ se expresa como

$$\mathbf{F}'_{m,n} = \begin{bmatrix} \mathbf{E}'_{m,n} & \mathbf{D}'_{m,n} \end{bmatrix} \quad (2.11)$$

donde las regiones lateral derecha compartida $\mathbf{D}'_{m,n}$ y no compartida $\mathbf{E}'_{m,n}$ se obtienen por medio de las siguientes ecuaciones para el bloque tipo *III*:

$$\mathbf{D}'_{m,n} = (\hat{\mathbf{D}}_{m,n} + \hat{\mathbf{Z}}\mathbf{w}_{m,n} + \hat{\mathbf{B}}_{m,n+1} + \hat{\mathbf{Z}}\mathbf{y}_{m,n+1})/2. \quad (2.12)$$

$$\mathbf{E}'_{m,n} = \hat{\mathbf{B}}_{m,n} + \hat{\mathbf{Z}}\mathbf{y}_{m,n} + \hat{\mathbf{C}}_{m,n} + \hat{\mathbf{E}}_{m,n} \quad (2.13)$$

y éstas mismas regiones están definidas para el bloque tipo *IV* por las ecuaciones (2.12) para la sección lateral derecha compartida, y para la sección no compartida por la siguiente ecuación:

$$\mathbf{E}'_{m,n} = \hat{\mathbf{A}}_{m,n} + \hat{\mathbf{Z}}\mathbf{z}_{m,n} + \hat{\mathbf{E}}_{m,n} + \hat{\mathbf{D}}_{m,n} \quad (2.14)$$

- Para los bloques tipo *VII* y *VIII*, el bloque reorganizado $\mathbf{F}'_{m,n}$ se halla por medio de la siguiente ecuación

$$\mathbf{F}'_{m,n} = \begin{bmatrix} \mathbf{E}'_{m,n} \\ \mathbf{C}'_{m,n} \end{bmatrix} \quad (2.15)$$

donde para el bloque tipo *VII* la región inferior compartida $\mathbf{C}'_{m,n}$ se cal-

cula por medio de la siguiente ecuación:

$$\mathbf{C}'_{m,n} = (\hat{\mathbf{C}}_{m,n} + \mathbf{Z}\hat{\mathbf{w}}_{m,n} + \hat{\mathbf{A}}_{m+1,n} + \mathbf{Z}\hat{\mathbf{z}}_{m+1,n})/2. \quad (2.16)$$

y la región no compartida $\mathbf{E}'_{m,n}$, se obtiene por medio de la ecuación (2.14), mientras que para el bloque tipo *VIII* éstas regiones se hallan a través de las ecuaciones (2.16), para la región inferior compartida, y por la siguiente ecuación para la región no compartida del bloque:

$$\mathbf{E}'_{m,n} = \hat{\mathbf{E}}_{m,n} + \hat{\mathbf{D}}_{m,n}. \quad (2.17)$$

- Finalmente, para el bloque tipo *IX* no es necesario realizar el cálculo de los promedios entre regiones compartidas debido a que este bloque representa una porción de la imagen que no tiene secciones compartidas con bloques consecutivos, como se muestra en la Figura 2.3, es decir $\mathbf{F}'_{m,n} = \hat{\mathbf{F}}_{m,n}$.

Los resultados de reconstrucción de imágenes, referentes a la calidad de la imagen y el tiempo de reconstrucción, obtenidos con el enfoque de bloques sobrepuestos OBB-CS previamente descrito, se presentan en el Capítulo 3.

2.2. Transmisión de Datos CS en un Sistema de Comunicación

Las técnicas de compresión de datos son de gran interés en el almacenamiento y transmisión de datos debido a su capacidad de reducir el consumo de recursos como el disco duro o el uso del canal de comunicación. Esto se debe principalmente a que éstas técnicas permiten obtener una representación de la información (audio, imagen, video, etc) con menos datos, de tal manera que el espacio requerido para su almacenamiento o transmisión sea menor que si se utilizan todos los datos de la fuente de información original.

En el Capítulo 1 y la sección 2.1 se mostró como la técnica de Compressive Sensing (CS) puede ser empleada para muestrear y comprimir imágenes, y cómo esta técnica puede ser implementada en dispositivos móviles con el enfoque propuesto, OBB-CS. En esta sección se presenta la descripción de las simulaciones para la transmisión de datos adquiridos con la técnica de Compressive Sensing, y se presentan las técnicas de modulación y demodulación empleadas durante la transmisión.

2.2.1. Sistemas de Comunicación Inalámbricos

El sistema de comunicaciones es el responsable por la transmisión de la información desde el transmisor hacia el receptor. Un sistema está conformado básicamente por tres elementos ?:

- **Transmisor**, quien convierte los datos en una forma adecuada para ser transmitida sobre el canal.
- **Canal**, que es el enlace físico entre el transmisor y el receptor, y es por donde los datos se propagarán y se verán afectados por el ruido.
- **Receptor**, el cual detecta los datos, los cuales son una versión aproximada de los datos transmitidos por el canal, y reconstruye una aproximación de los datos originales con los datos recibidos para luego entregar la información a su destino final con la menor cantidad de errores y distorsiones como sea posible.

La Figura 2.4, muestra el diagrama de bloques del sistema de comunicación el cual incluye la técnica de CS para muestrear y comprimir los datos siendo transmitidos por el canal de comunicación. Para medir la calidad de los datos recibidos por el receptor en el sistema de comunicación se utiliza la razón señal a ruido o *Signal-to-Noise Ratio* (SNR), y el rendimiento *throughput* (T_h) en la transmisión de los datos describe la tasa promedio de mensajes recibidos sin error.

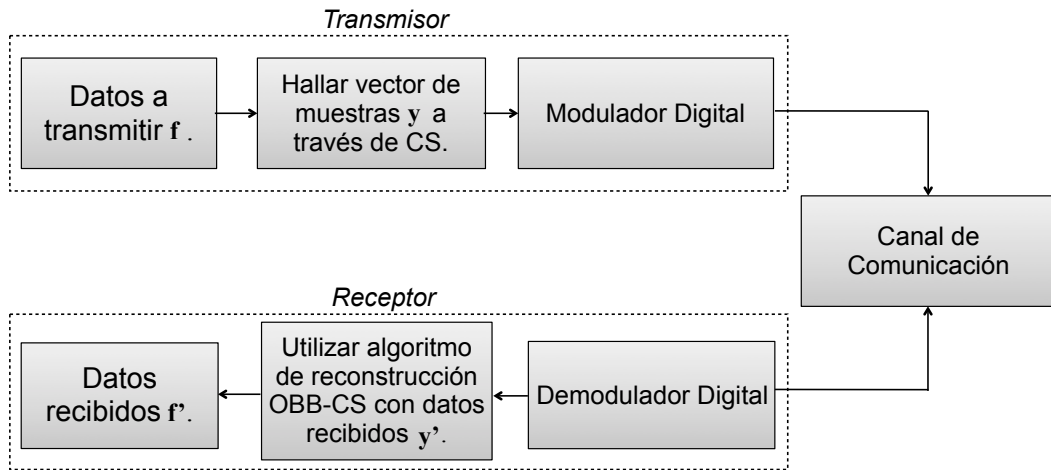


Figura 2.4: Diagrama de bloques para el proceso de transmisión de datos muestreados y comprimidos con CS.

En un sistema de comunicación es inevitable la presencia de ruido, el cual se presenta debido a ondas que tienden a perturbar la transmisión y el procesamiento de señales en un sistema de comunicación inalámbrico. Para medir el efecto del ruido se introduce en la simulación de la transmisión de datos un parámetro que controla el SNR en el sistema de comunicación. Este parámetro se utiliza en el receptor para controlar la razón entre el promedio de potencia de la señal y el promedio de potencia de ruido, donde el SNR se define como:

$$SNR = 10 \log_{10} \left(\frac{P_s}{P_n} \right), \quad [dB] \quad (2.18)$$

donde P_s es la potencia de la señal, y P_n es la potencia del ruido. Por medio del cálculo del SNR se puede comparar en qué porcentaje la señal transmitida sobre el canal ha sido afectada por el ruido en el canal. A continuación se describe el proceso de simulación de la transmisión de datos utilizando una aproximación de un enlace de comunicación Bluetooth, la cual utiliza una aproximación de los procesos de modulación y demodulación Bluetooth en el transmisor y receptor respectivamente, como se muestra en la Figura 2.4. Por otro lado, el rendimiento o *throughput* es una medida de la información que fluye a través del sistema y se toma como la tasa promedio de éxito de la en-

trega de mensajes en el canal de comunicación. Ésta medida se calcula por medio de la siguiente ecuación:

$$T_h = \frac{\text{total de bits transmitidos}}{\text{tiempo total en segundos}} \quad (2.19)$$

2.2.2. Transmisión de Datos con Bluetooth

La tecnología de rango corto Bluetooth ha sido ampliamente acogida por los dispositivos móviles inalámbricos para realizar transmisión de datos entre dispositivos electrónicos y puede ser utilizada para la transmisión de imágenes de un dispositivo móvil a otro sobre un canal de comunicación inalámbrico. Sin embargo, la tecnología Bluetooth emplea diferentes técnicas de comunicación las cuales la convierten en una tecnología compleja para su estudio. Por ende, en este trabajo se realiza la simulación de una transmisión de datos sobre un canal de comunicación utilizando una aproximación de las técnicas de modulación y demodulación empleadas por una transmisión Bluetooth como una manera para conocer los efectos en la transmisión de datos comprimidos con la técnica de CS. En las simulaciones de la transmisión de datos se emplea una aproximación de la pila del protocolo de comunicación Bluetooth, la cual incluye las capas de radio y banda base.

La primera capa en la pila del protocolo Bluetooth, es decir la capa física, es la responsable de transmitir los datos entre dos dispositivos Bluetooth conectados por un enlace físico, mientras que la capa de banda base, los datos se envían en paquetes desde un dispositivo actuando como *maestro*, el cual representa el transmisor, a otro dispositivo actuando como *esclavo*, el cual representa el receptor. Antes de armar los paquetes de datos, la especificación Bluetooth define dos modos de modulaciones que convierten los datos en una forma adecuada para su transmisión sobre el canal de comunicación. Este proceso de modulación se realiza en la capa de radio de Bluetooth, donde el primer modo de modulación definido se denomina *Basic Rate* (BR) y utiliza una onda binaria modulada en frecuencia, la cual minimiza la complejidad del transmisor/receptor; el segundo modo de modulación, el cual es opcional y se

denominada Enhanced Data Rate (EDR), utiliza una modulación de desplazamiento en fase (PSK).

Bluetooth además emplea la técnica de Espectro Ensanchado de Salto de Frecuencia (*Frequency Hopping Spread Spectrum- FHSS*) para evitar interferencias y utiliza 79 frecuencias de salto (en algunos países solo 23 frecuencias), cada una con un ancho de banda de 1 MHz. La técnica de frecuencia de salto es combinada con solicitudes de parada/espera y de repetición automática (ARQ), pruebas de redundancia cíclica (CRC) y corrección de errores de reenvío (FEC), las cuales ayudan a lograr una alta confiabilidad en los enlaces inalámbricos. En la capa de radio, este trabajo utiliza la modulación de desplazamiento en frecuencia (FSK) como una aproximación a la modulación utilizada en el modo de modulación Basic Rate (BR) de Bluetooth para las simulaciones de la transmisión de datos. En la capa de banda base, las simulaciones consideran la transmisión de datos en paquetes, donde el formato de un paquete en el modo BR consiste de tres entidades: 1) el código de acceso, de 72 o 68 bits; 2) la cabecera de 54 bits; y 3) la carga útil de 0 a 2745 bits máximo.

Por otra parte, las medidas de tasa de error de bit (BER) y rendimiento se utilizan para analizar los efectos del ruido en el canal de comunicación y la media del flujo de información que fluye a través del sistema cuando se utilizan los datos siendo transmitidos sin y con compresión. A continuación se describen con más detalle estas dos medidas.

Tasa de Error de Bit (BER)

La tasa de error de bits (BER) se mide en el sistema de comunicación utilizando como datos de entrada los bits de una imagen de 256×256 píxeles. El transmisor representa un dispositivo enviando la imagen y el receptor representa otro dispositivo recibiendo la imagen transmitida. El transmisor se encarga de modular los bits de los datos en una señal digital la cual se envía por un canal simulado. El canal es simulado añadiendo ruido gaussiano a los

datos de la imagen siendo transmitida. El receptor se encarga de demodular la señal recibida y produce una secuencia de bits recuperados. Al finalizar la transmisión, se comparan los bits recibidos y los bits enviados para calcular los errores obtenidos durante la transmisión. El BER se calcula como el número de bits en error dividido por el número total de bits transmitidos durante el intervalo de tiempo estudiado, como se muestra en a continuación:

$$BER = \frac{1}{2} \sqrt{\frac{E_b}{E_0}}, \quad (2.20)$$

donde E_b representa la energía por bit y E_0 es la densidad espectral del ruido de la señal que contiene los datos adquiridos en el receptor.

Rendimiento (Throughput)

El rendimiento de la transmisión de datos, también conocido como Throughput, es el número de bits útiles¹ transmitidos por segundo a través de un sistema de comunicación, como se muestra en la ecuación (2.19). El rendimiento es normalizado y expresado en porcentajes para medir la utilización del canal y la tasa de pérdida de bits en porcentajes. La utilización del canal en porcentaje, es el rendimiento alcanzado y está relacionado con el ancho de banda en bits por segundo [bit/s] de un canal de comunicación digital. En las simulaciones de la transmisión de datos se tiene en cuenta el esquema ARQ donde los paquetes son retransmitidos hasta que un paquete de reconocimiento (ACK) es recibido o el tiempo de espera del enlace de comunicación es excedido.

Características de las Simulaciones de Transmisión de Datos

El propósito de las simulaciones realizadas es mostrar cómo se pueden afectar los datos enviados por el transmisor dependiendo de la calidad del canal utilizado, y al mismo tiempo cómo se afecta el rendimiento T_h debido a la calidad del canal y la cantidad de datos enviados. En las actuales versiones de

¹bits sin error, se cuentan los bits del paquete que corresponden a la carga útil.

aplicaciones que utilizan una baja tasa para la transmisión de datos con Bluetooth, se emplea la modulación gaussiana de desplazamiento en frecuencia (GFSK). Esta modulación implementa la función gaussiana como un filtro de forma de pulso para reducir la transmisión de banda ancha.

En este trabajo, las simulaciones de transmisión de datos se realizan siguiendo una aproximación de ésta modulación a través de la modulación binaria de desplazamiento en frecuencia (BFSK). Asimismo, Los datos de las imágenes seleccionadas se toman como la fuente de datos enviada por el transmisor hacia el receptor en el sistema de comunicación.

Capítulo 3

Resultados

Este capítulo presenta los resultados de la simulación e implementación de la técnica de CS en imágenes y su posterior reconstrucción con los algoritmos presentados en los capítulos 1 y 2 sobre un computador personal y un dispositivo móvil. Se describen las principales características de los datos y plataforma utilizados. Asimismo, este capítulo presenta los resultados de las simulaciones de transmisión de datos, cuya descripción se presentó en el capítulo 2.

3.1. Compresión y Reconstrucción con Compressive Sensing sobre Dispositivos Móviles

Para los algoritmos de reconstrucción seleccionados, la metodología de evaluación que se utilizó se basó en la calidad de reconstrucción de las imágenes, el tiempo de reconstrucción y el consumo de energía en el dispositivo móvil. Las simulaciones e implementación de la técnica de CS incluyen la compresión y recuperación de dos imágenes estándar: Lena y Barco¹ (Figuras 3.5 y 3.6 - Columna Izquierda), utilizando los algoritmos IHT, OMP y OBB-CS presentados en los capítulos 1 y 2.

¹ Imágenes estándares utilizadas en el campo de procesamiento de imágenes. Disponibles en Internet en: <http://sipi.usc.edu/database/>

Se empleó una resolución para las imágenes de 48x48, 128x128 y 256x256 píxeles. Las imágenes fueron aproximadas utilizando diferentes niveles de dispersión k/N de la imagen, donde solo se conservaron los k coeficientes más representativos en la base de representación Ψ Wavelet. Dos plataformas computacionales diferentes se utilizaron para implementar el esquema de CS: Un iPhone 4 con 512 MB de memoria que tiene un procesador ARM Cortex-A8 de 1 GHz (plataforma iOS), y un computador personal con 2GB de memoria y procesador de 2.4 GHz, utilizando el software Matlab (plataforma PC) y el simulador del dispositivo iPhone (Simulador iOS) disponible en la ambiente de desarrollo integrado Xcode¹. Para medir la calidad de las imágenes reconstruidas en comparación con las imágenes originales, se utilizó la métrica estándar del pico de señal a ruido (PSNR), la cual se da en decibelios [dB] y está definida como se muestra a continuación:

$$\text{PSNR} = 20\log_{10}\left(\frac{\text{MAX}_{\mathbf{F}}}{\text{MSE}}\right), \quad (3.1)$$

donde el error cuadrático medio (MSE) entre la imagen original \mathbf{F} y la imagen recuperada \mathbf{F}' se calcula a través la siguiente ecuación:

$$\text{MSE} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \|\mathbf{F}_{(i,j)} - \mathbf{F}'_{(i,j)}\|^2. \quad (3.2)$$

3.1.1. Calidad de la Reconstrucción

Dos conjuntos aleatorios fueron estudiados como matrices de muestreo para fines de comparación, los cuales incluyen la matriz de muestreo de configuración Gaussiana y la matriz de muestreo de configuración de bloques Hadamard mezclados (SBHE), ambas descritas en el capítulo 1. Las Figuras 3.1 y 3.2 muestran los resultados del PSNR para las imágenes reconstruidas de Lena y Barco cuando se muestrean y comprimen usando la matrices de

¹Xcode es el ambiente de desarrollo intergrado (IDE) para desarrollo de aplicaciones basadas en el sistema operativo iOS.

muestreo Gaussiana y SBHE respectivamente en la plataforma PC, usando el software Matlab, y con los algoritmos IHT y OMP.

Los resultados se muestran para diferentes niveles de dispersión k/N y para diferentes porcentajes de muestras M/N de la imágenes. Basándose en los resultados, cuando se utiliza la configuración SBHE como la matriz de muestreo, ésta proporciona casi la misma calidad de la reconstrucción que la configuración Gaussiana, pero al mismo tiempo requiere menos espacio de almacenamiento debido a su naturaleza dispersa, como se mostró en el cuadro 1.1 del capítulo 1.

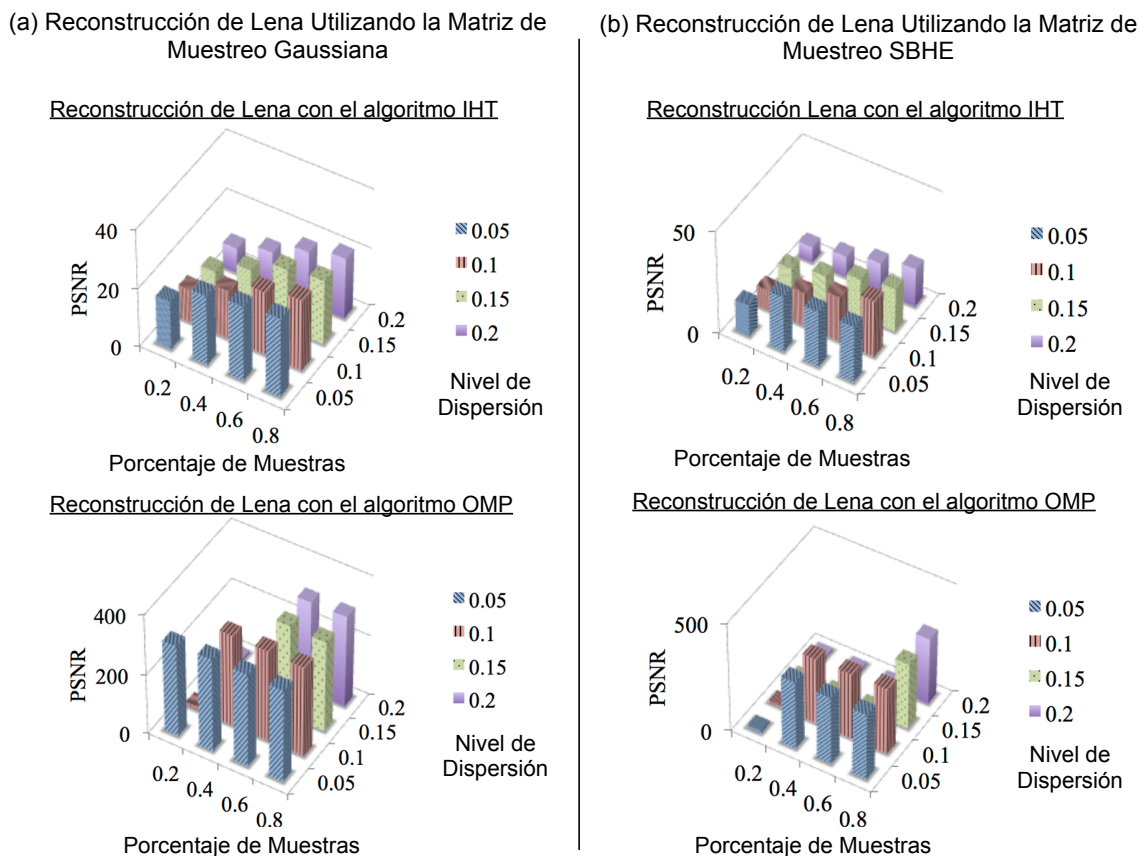


Figura 3.1: Resultados PSNR para la imagen Lena utilizando dos matrices de muestreo diferentes, varios niveles de dispersión k/N de la imagen y diferentes porcentajes M/N para el total de muestras. a) Recuperación de la imagen Lena con la matriz Gaussiana; b) Recuperación de la imagen Lena con la matriz SBHE.

Para la imagen Lena, Figura 3.1, se puede observar que la reconstrucción obtenida con el algoritmo OMP es en general mejor que la reconstrucción obtenida con el algoritmo IHT cuando se utiliza la misma matriz de muestreo.

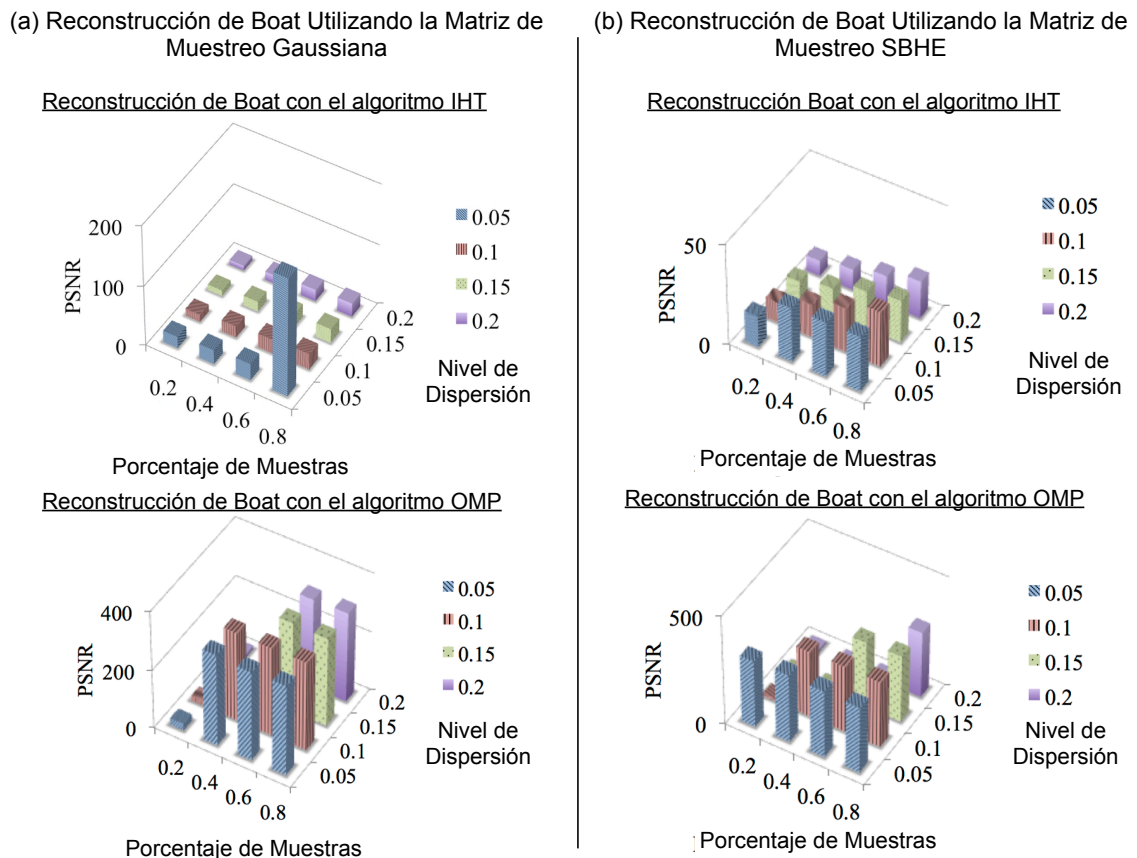


Figura 3.2: Resultados PSNR para la imagen Barco utilizando dos matrices de muestreo diferentes, varios niveles de dispersión k/N de la imagen y diferentes porcentajes M/N para las mediciones. a) Recuperación de la imagen Barco con la matriz Gaussiana; b) Recuperación de la imagen Barco con la matriz SBHE.

Para la imagen Barco, Figura 3.2, los resultados de reconstrucción obtenidos con el algoritmo OMP son similares que los obtenidos con la imagen Lena, donde éste algoritmo obtiene una mejor calidad en la reconstrucción de la imagen que el algoritmo IHT. De la misma manera para ambas imágenes, y para un nivel dado de la dispersión k/N , cuando el porcentaje de muestras

M/N aumenta se obtiene una mejor calidad en la reconstrucción de la imagen. Sin embargo, para porcentajes de muestras iguales o mayores a 60 %, se alcanza una buena calidad de la imagen, es decir, un PSNR igual o mayor a 30 dB.

Debido a las características observadas para la matriz de muestreo SBHE en el proceso de reconstrucción de las imágenes, ésta es escogida como la matriz de muestreo más apropiada para su implementación en dispositivos de recursos limitados y por lo tanto, se emplea para la implementación en el dispositivo móvil de la técnica CS. Las Figuras 3.3 y 3.4 muestran los resultados obtenidos para la imagen Lena y Barco respectivamente. Estas pruebas de reconstrucción de las imágenes se realizaron con imágenes de 48×48 píxeles y utilizando los dos algoritmos de reconstrucción IHT y OMP. El tamaño de las imágenes se tomó de 48×48 píxeles debido a que el dispositivo móvil no podía procesar imágenes de tamaño superior dadas las restricciones de memoria y el requerimiento computacional de los algoritmos implementados.

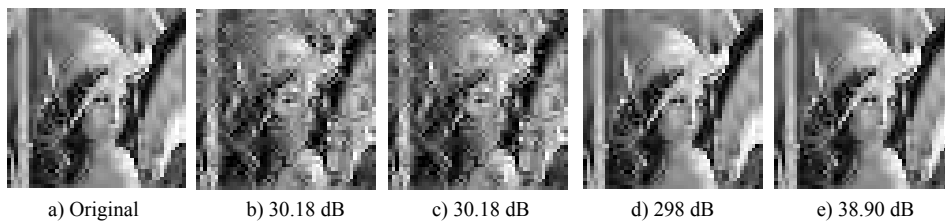


Figura 3.3: Resultados de reconstrucción para la imagen Lena de 48×48 píxeles; a) Imagen original; b) recuperación con el algoritmo IHT en Matlab; c) recuperación IHT en el dispositivo iOS; d) recuperación con el algoritmo OMP en Matlab; e) recuperación OMP en el dispositivo iOS.

Como se muestra en las Figuras 3.3 y 3.4, los resultados en la calidad de la reconstrucción con el algoritmo OMP, imágenes (d) y (e) en las figuras, es superior que la calidad obtenida con el algoritmo de reconstrucción IHT, imágenes (b) y (c) en las figuras. Aunque la calidad de las imágenes en la reconstrucción CS es adecuada, el tamaño de las imágenes reconstruidas en el dispositivo móvil es muy pequeño. Sin embargo, y como se analizó en la

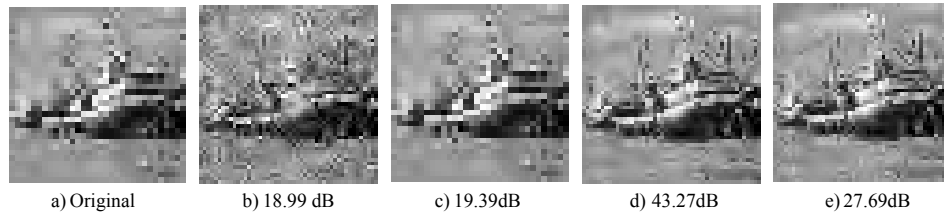


Figura 3.4: Resultados de reconstrucción para la imagen Barco de 48×48 píxeles; a) Imagen original; b) recuperación con el algoritmo IHT en Matlab; c) recuperación IHT en el dispositivo iOS; d) recuperación con el algoritmo OMP en Matlab; e) recuperación OMP en el dispositivo iOS.

sección 2, si se emplea el enfoque basado en bloques sobrepuestos OBB-CS, es posible la reconstrucción de imágenes de mayor tamaño en el dispositivo móvil.

El diseño del enfoque OBB-CS permite reconstruir imágenes de mayor tamaño en el dispositivo móvil utilizando muestras tomadas con la técnica de CS y con el algoritmo OMP, el cual proporciona mejores resultados en la calidad de reconstrucción de las imágenes. La Figuras 3.5 y 3.6 muestran las versiones original y reconstruida para las imágenes Lena y Barco respectivamente, las cuales fueron obtenidas con el enfoque OBB-CS. De igual forma éstas figuras muestran los resultados obtenidos en la plataforma PC, el simulador iOS y en la plataforma iOS.

El tamaño para los bloques utilizado en el enfoque OBB-CS, se definió como $b = 32$, es decir, cada bloque sobrepuesto en cada imagen de las Figuras 3.5 y 3.6, corresponde a un bloque de 32×32 píxeles, los cuales se pueden procesar fácilmente en el dispositivo móvil por el algoritmo de reconstrucción. Para las imágenes de 128×128 píxeles, se calculó un factor de sobreposición $\Delta = 8$, mientras que para las imágenes de 256×256 , el factor de sobreposición entre bloques adyacentes es $\Delta = 4$; estos valores fueron calculados como se explicó en la sección 2.1 del capítulo 2.

Como se puede apreciar en las Figuras 3.5 y 3.6, los resultados en la reconstrucción de las imágenes son exitosos debido a que un PSNR igual o

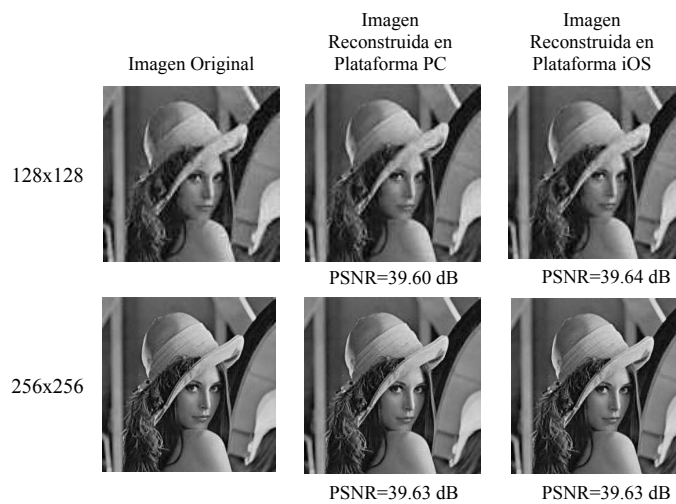


Figura 3.5: Resultados de reconstrucción para la imagen Lena utilizando el enfoque OBB-CS con el algoritmo OMP; (Columna izquierda) Imágenes originales de 128×128 y 256×256 píxeles; (Columna Central) Imagen Recuperada en la Plataforma PC; (Columna Derecha) Imágenes recuperadas en la Plataforma iOS utilizando un factor de sobreposición $\Delta = 8$ para imágenes de 128×128 y $\Delta = 4$ para imágenes de 256×256 . El porcentaje de muestras M/N es del 60%.

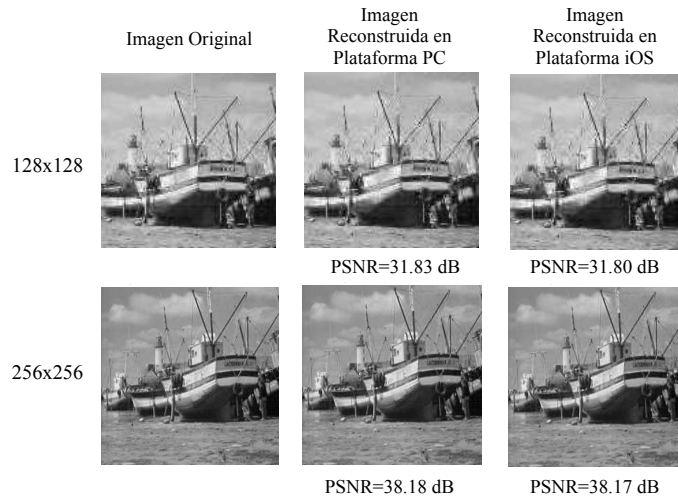


Figura 3.6: Resultados de reconstrucción para la imagen Barco utilizando el enfoque OBB-CS con el algoritmo OMP; (Columna izquierda) Imágenes originales de 128×128 y 256×256 píxeles; (Columna Central) Imagen Recuperada en la Plataforma PC; (Columna Derecha) Imágenes recuperadas en la Plataforma iOS utilizando un factor de sobreposición $\Delta = 8$ para imágenes de 128×128 y $\Delta = 4$ para imágenes de 256×256 . El porcentaje de muestras M/N es del 60%.

mayor a 30dB está relacionado con una buena calidad de la imagen reconstruida. Estos resultados comprueban que el enfoque OBB-CS es apropiado para la reconstrucción de imágenes utilizando la técnica de CS.

3.1.2. Tiempo de Reconstrucción

El tiempo de reconstrucción de las imágenes fué medido en segundos para los algoritmos OMP e IHT. La Figura 3.7 muestra el tiempo de reconstrucción requerido cuando se utiliza el enfoque OBB-CS descrito en el capítulo 2. El tiempo utilizado por iteración para un bloque de 32×32 píxeles es 4.9 segundos en el simulador iOS, y 30.5 segundos en el dispositivo móvil utilizando la plataforma iOS. Cuando la dimensión de la imagen incrementa de 128×128 a 256×256 píxeles, el tiempo de cómputo aumenta también debido a que más bloques necesitan ser procesados.

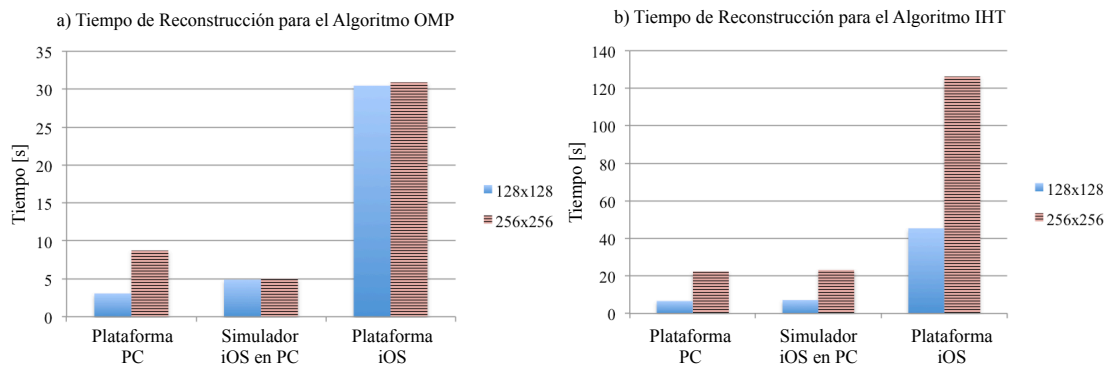


Figura 3.7: Tiempo de recuperación requerido por (a) el algoritmo OMP y (b) el algoritmo IHT para recuperar bloques de tamaño $b = 32$ píxeles para imágenes de 128×128 y 256×256 píxeles.

3.1.3. Consumo de Energía de los Algoritmos de Reconstrucción

El consumo de energía del dispositivo móvil iOS cuando se utilizan los algoritmos de reconstrucción OMP y IHT fue medido utilizando la herramienta software Instruments proporcionada por el IDE Xcode de Apple¹. La aplicación Instruments posee un instrumento denominado Instrumento de Uso de Energía o *Energy Usage Instrument* (EUI), el cual mide el consumo de energía de las aplicaciones corriendo en el dispositivo móvil iOS.

El instrumento EUI toma muestras del consumo de energía en el dispositivo móvil comparando la cantidad de batería cargada y disponible antes y después de la ejecución de una aplicación sobre el dispositivo móvil. Este instrumento proporciona los valores de consumo de energía por cada aplicación siendo ejecutada en el dispositivo móvil, donde el rango de valores para el consumo de energía varía desde 0, lo que significa que no hay consumo de energía, hasta un máximo de 20, el cual indica que el máximo consumo de energía está siendo disipado por la aplicación en el dispositivo móvil.

¹IDE Xcode proporcionado por Apple. Disponible en Internet en: <https://developer.apple.com/technologies/tools/>

La Figura 3.8 muestra el diagrama de cajas¹ de los niveles de consumo de energía requeridos por los algoritmos de reconstrucción implementados para el proceso de compresión y reconstrucción de las imágenes. Analizando el consumo de energía en la plataforma iOS, el algoritmo OMP es el algoritmo con la mayor variabilidad en el consumo de energía. Sin embargo, el consumo de energía del algoritmo OMP y el algoritmo IHT tienden a tener resultados, en promedio, similares.

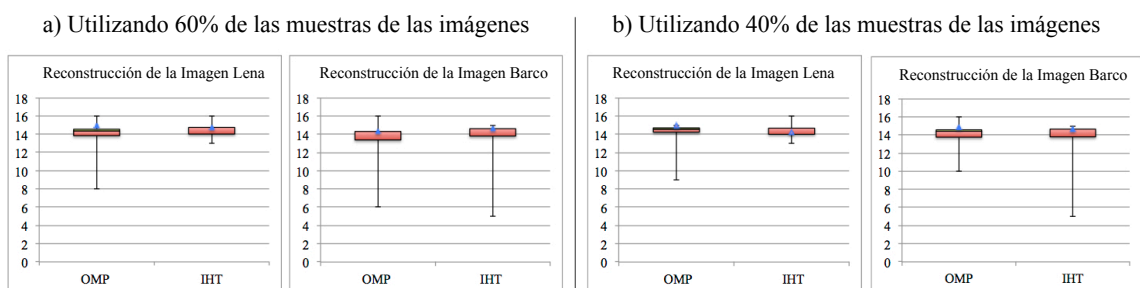


Figura 3.8: Diagrama de cajas de los niveles de consumo de energía para los algoritmos OMP y IHT sobre la plataforma iOS tomando diferentes porcentajes de muestras de las imágenes.

3.2. Simulación de la Transmisión de Datos Tomados con CS

Para analizar la tasa de error por bit y el rendimiento en el canal, se realizaron las simulaciones para la transmisión de los datos correspondientes a las imágenes de 256×256 píxeles cuando los datos estaban sin compresión, y cuando los datos eran muestreados y comprimidos con la técnica de Compressive Sensing (CS). A continuación se presentan los resultados de las simulaciones realizadas.

¹El diagrama de cajas es un gráfico que muestra un resumen de un conjunto de datos medidos en un intervalo dado, el cual además indica el valor central y la variabilidad de los datos.

3.2.1. Tasa de Error por Bit

La Figura 3.9 muestra los resultados de la tasa de error por bit (BER) para diferentes valores del SNR, donde la ordenada es la relación señal a ruido (SNR) normalizado y expresado como E_b/N_0 , y la absisa es el error por bit el cual normalmente es expresado en potencias de diez. Como se puede apreciar en la Figura 3.9, la simulación del BER para la modulación FSK es similar a la curva teórica, las pequeñas diferencias entre la simulación y la curva teórica cuando la calidad del canal es mayor se deben principalmente a que el número de bits utilizados en la simulación no es estadísticamente significativo. Sin embargo, la simulación se aproxima al límite teórico el cual reduce la probabilidad de error a medida que la calidad del canal aumenta.

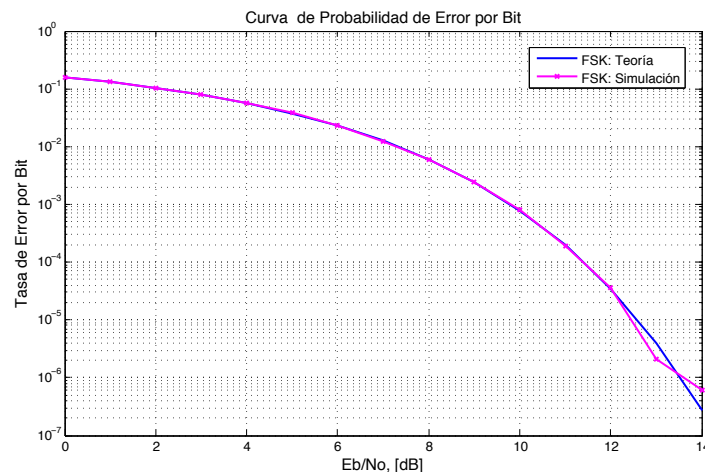


Figura 3.9: Tasa de Error por Bit del canal de comunicación utilizando modulación FSK.

3.2.2. Rendimiento en el Canal

El rendimiento (Throughput) en el canal se calculó para la transmisión de los datos cuando éstos se enviaban sin compresión, y con una compresión del 60% y 40% de los datos de la imagen. La Figuras 3.10 y 3.11 muestran los resultados obtenidos en el rendimiento de la transmisión dado en bits por segundo y en paquetes por segundo respectivamente, para diferentes valores

de la calidad del canal (E_b/N_0). Según estos resultados, al enviar menos datos de la imagen, es decir, reduciendo el número de datos con una compresión del 40 % o 60 % con CS, se puede alcanzar un nivel ligeramente superior, entre el 1 % y 2 %, en el rendimiento de la transmisión en el canal.

Por otro lado, en las simulaciones no se alcanza el límite teórico de 1Mbps en el rendimiento, como se muestra en la Figura 3.10, debido a que sólo se tuvo en cuenta la carga útil de los paquetes enviados, es decir, de los datos contenidos en cada paquete enviado por el canal de comunicación solo se tomaron los datos de carga útil para calcular el rendimiento. Al tomar en cuenta solo la carga útil para cada paquete, se presenta una reducción del rendimiento dado con respecto al límite teórico, el cual tiene en cuenta todos los bits presentes en cada paquete (código, cabecera y carga útil, como se presentó en la sección 2.2.2).

La Figura 3.11 muestra el rendimiento en paquetes por segundo, donde a medida que aumenta la calidad del canal (E_b/N_0), el número de errores en los paquetes se reduce y por ende el rendimiento aumenta hasta llegar al máximo teórico de la transmisión de paquetes $1 \times 10^6/2841$ [pps]¹. Aunque la diferencia en el rendimiento no es muy notoria entre las transmisiones de datos comprimidos y no comprimidos, una gran ventaja de enviar los datos comprimidos con CS se dará en el tiempo requerido para una transmisión, debido a que menos datos serán requeridos para la transmisión y por lo tanto, menos tiempo de espera para su finalización.

¹Límite teórico de 1Mbps dividido entre el tamaño del paquete 2841 bits.

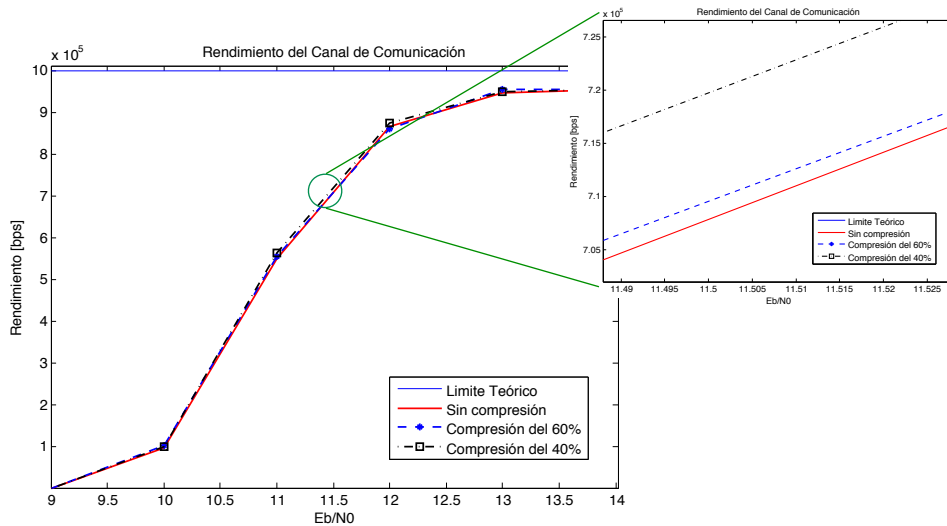


Figura 3.10: Rendimiento del sistema de comunicación (Throughput) en bits por segundo [bps] para diferentes valores de la calidad del canal cuando se transmiten los datos de una imagen de 256×256 en un sistema de comunicación utilizando la modulación FSK.

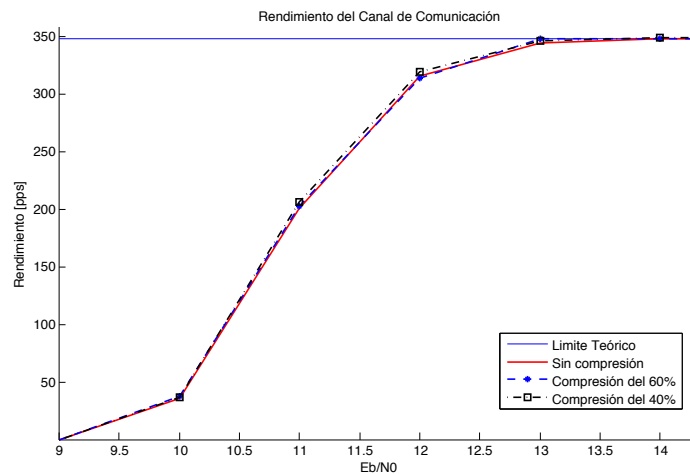


Figura 3.11: Rendimiento del sistema de comunicación (Throughput) en paquetes por segundo [pps] para diferentes valores de la calidad del canal cuando se transmiten los datos de una imagen de 256×256 en un sistema de comunicación utilizando la modulación FSK.

Capítulo 4

Conclusiones

- Se propuso un modelo de reconstrucción basado en bloques para reconstruir imágenes comprimidas con la técnica Compressive Sensing sobre dispositivos de recursos limitados. El modelo propuesto proporciona una alternativa para efectivamente reconstruir imágenes de calidad a través del procesamiento de bloques superpuestos tomados de una imagen.
- Para la reconstrucción de las imágenes en el dispositivo móvil se tuvo en cuenta el consumo de energía de los algoritmos de reconstrucción OMP e IHT utilizados, los cuales tienen en promedio un consumo de energía similar y donde el algoritmo OMP tiene una mayor variabilidad del uso de energía.
- La implementación de dos algoritmos de reconstrucción y del modelo de reconstrucción basado en bloques permitió conocer la viabilidad de implementación de un sistema de compresión y reconstrucción utilizando la técnica Compressive Sensing sobre dispositivos móviles. Los resultados obtenidos muestran que con el enfoque OBB-CS la reconstrucción de imágenes de hasta 256×256 es posible en un dispositivo móvil.
- De los algoritmos analizados para la reconstrucción de las imágenes, el algoritmo OMP junto con la matriz de muestreo SBHE, obtienen una mayor calidad en la reconstrucción de las imágenes, por lo cual son la combinación propuesta para ser utilizada en el enfoque por bloques OBB-CS

sobre dispositivos móviles.

- Las simulaciones realizadas del canal de comunicación dejan ver que la calidad en el canal es un factor clave para alcanzar un mayor rendimiento en la transmisión de datos. El envío de datos comprimidos con la técnica de CS permite alcanzar un ligero aumento en el rendimiento del canal, sin embargo el tiempo en las transmisiones se puede reducir cuando los datos se envían comprimidos debido a que menos datos necesitan ser enviados por el sistema.

4.1. Contribuciones

- Irene Manotas, Henry Arguello y Kristina Winbladh. Implementation of Imaging Compressive Sensing Algorithms on Mobile Handset Devices. Seventh International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), Victoria, BC, Noviembre, 2012.
- Irene Manotas y Henry Arguello. Overlapped Block-Based Compressive Sensing Imaging On Mobile Handset Devices. Revista Facultad de Ingeniería de la Universidad de Antioquia. Enviado en Mayo, 2013. *En Revisión.*

Referencias

- Emmanuel Candès. Compressive sampling. *Proceedings of the International Congress of Mathematicians*, 3:1433–1452, 2006.
- Justin Romberg. Imaging via compressive sampling. *IEEE Signal Processing Magazine*, 2008.
- Richard Baraniuk. Compressive sensing. *IEEE Signal Processing Magazine*, 24(4):118–121, 2007.
- Emmanuel Candès and Michael Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.
- David Donoho. Compressed sensing. *Transactions on Information Theory*, 54(4):1289–1306, 2006.
- Thomas Blumensath and Mike E. Davies. Gradient pursuits. *Transactions on Signal Processing*, 56(6):2370 – 2382, 2006.
- Emmanuel Candès and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.
- Emmanuel Candès, Justin Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Journal on Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.
- Robert Calderbank, Stephen Howard, and Sina Jafarpour. Construction of large class of deterministic sensing matrices that satisfy a statistical isometry

-
- property. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):358–374, 2010.
- Lu Gan, Thong T. Do, and Trac D. Tran. Fast compressive imaging using scrambled block hadamard ensemble. European Signal Processing Conference (EUSIPCO), 2009.
- Thong T. Do, Lu Gan, Nam H. Nguyen, and Trac D. Tran. Fast and efficient compressive sensing using structurally random matrices. *IEEE Transactions on Signal Processing*, 60(1):139–154, 2012.
- Thomas Blumensath. Accelerated iterative hard thresholding. *Signal Process.*, 92(3):752–756, 2012.
- T. Tony Cai, Guangwu Xu, and Jun Zhang;. On recovery of sparse signals via l_1 minimization. *IEEE Transactions on Information Theory*, 55(7):3388–3397, 2009.
- Stephen J. Wright, Robert D. Nowak, and Mario A. T. Figueiredo. Sparse reconstruction by separable approximation. *Signal Processing, IEEE Transactions on*, 57(7):2479–2493, 2009.
- Thomas Blumensath and Mike E. Davies. Iterative hard thresholding for compressed sensing. *Journal on Applied and Computational Harmonic Analysis*, 7(3):265–274, 2009.
- Joel A. Tropp. and Ana C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666, 2007.
- Marco F. Duarte, Mark A. Davenport, Takhar Dharmpal, Jason N. Laska, Sun Ting, Kelly Kevin F., and Richard G. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):83–91, 2008.
- Henry Arguello and Gonzalo Arce. Code aperture optimization for spectrally agile compressive imaging. *Optical Society of America*, 28(11):2400–2413, 2011.

-
- Henry Arguello, Claudia Correa, and Gonzalo Arce. Fast lapped reconstructions in compressive spectral imaging. *Journal of Applied Optics*, 52(10):D32–D45, 2013.
- Mohammadreza Balouchestani, Kaamran Raahemifar, and Sridhar Krishnan. Robust wireless sensor networks with compressed sensing theory. *Journal of Networked Digital Technologies Communications in Computer and Information Science*, 293(10):608–619, 2012.
- Eduardo C. Pinheiro, Octavian A. Postolache, and Pedro S. Girao. Implementation of compressed sensing in telecardiology sensor networks. *International Journal Telemedicine Applications*, 2010:7:1–7:7, 2010.
- K. Kanoun, H. Mamaghanian, N. Khaled, and D. Atienza. A real-time compressed sensing-based personal electrocardiogram monitoring system. In *Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 1–6, 2011.
- Marco F. Duarte, Mark A. Davenport, Dharmpal Takhar, Ting Sun Jason N. Laska, Kevin F. Kelly, and Richard G. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):83–91, 2008.
- Rebecca M. Willett, Roummel F. Marcia, and Jonathan M. Nichols. Compressed sensing for practical optical imaging systems: a tutorial. *SPIE Optical Engineering*, 50, 2011.
- Henry Arguello and Gonzalo Arce. Restricted isometry property in coded aperture compressive spectral imaging. 2012.
- Thomas Blumensath. Accelerated iterative hard thresholding. *Signal Processing*, pages 752–756, 2011.
- Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Rev.*, 43(1):129–159, 2001.
- Irene Manotas y Henry Arguello. Overlapped block-based compressive sensing imaging on mobile handset devices. *Revista Facultad de Ingeniería de la Universidad de Antioquia*, Mayo 2013. En revisión.

Irene Manotas, Henry Arguello, and Kristina Winbladh. Implementation of imaging compressive sensing algorithms on mobile handset devices. In *BWC-CA*, 2012.

Parichat Sermwuthisarn, Supatana Auethavekiat, and Vorapoj Patanavijit. A fast image recovery using compressive sensing technique with block based orthogonal matching pursuit. In *ISPACS: International Symposium on Intelligent Signal Processing and Communication Systems*, pages 212–215, 2009.

Lu Gan. Block compressed sensing of natural images. In *International Conference in Digital Signal Processing*, pages 403–406, 2007.

Simon Haykin. *Communication Systems*. Wiley Publishing, 5th edition, 2009.

Bluetooth. *Specification of the Bluetooth System*, 4 edition, 2010.

Ling-Jyh Chen, Rohit Kapoor, Sewook Jung, M.Y. Sanadidi, and Mario Gerla. An adaptive arq timeout approach for audio streaming over bluetooth. *International Journal Wireless and Mobile Computing*, 3(4), 2009.