



UNIVERSIDAD DE LOS ANDES  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA ELÉCTRICA

DESARROLLO DE HERRAMIENTAS COMPUTACIONALES  
PARA SUPERRESOLUCIÓN DE IMÁGENES SATELITALES  
USANDO REPRESENTACIÓN POCO DENSA DE SEÑALES  
EN DICCIONARIOS REDUNDANTES

Br. Luis Alfredo Portillo Castillo

Mérida, Junio, 2016

UNIVERSIDAD DE LOS ANDES  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA ELÉCTRICA

DESARROLLO DE HERRAMIENTAS COMPUTACIONALES  
PARA SUPERRESOLUCIÓN DE IMÁGENES SATELITALES  
USANDO REPRESENTACIÓN POCO DENSA DE SEÑALES  
EN DICCIONARIOS REDUNDANTES

Trabajo de Grado presentado como requisito parcial para optar al título de Ingeniero  
Electricista

Br. Luis Alfredo Portillo Castillo

Tutor: Prof. José Luis Paredes

Mérida, Junio, 2016

UNIVERSIDAD DE LOS ANDES  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA ELÉCTRICA

**DESARROLLO DE HERRAMIENTAS  
COMPUTACIONALES PARA  
SUPERRESOLUCIÓN DE IMÁGENES  
SATELITALES USANDO REPRESENTACIÓN  
POCO DENSA DE SEÑALES EN DICCIONARIOS  
REDUNDANTES**

Br. Luis Alfredo Portillo Castillo

Trabajo de Grado presentado en cumplimiento parcial de los requisitos exigidos para optar al título de Ingeniero Electricista, aprobado en nombre de la Universidad de Los Andes por el siguiente Jurado.

---

Prof. José Luis Paredes Q.

Tutor

---

Prof. Gerardo Ceballos

Jurado

---

Prof. Nelson Perez

Jurado

*Nuestra recompensa se encuentra en el esfuerzo y no en el resultado. Un esfuerzo total es una victoria completa.*

---

MAHATMA GANDHI

*A mis padres y mi hermano.*

## AGRADECIMIENTOS

A Dios todo poderoso por haberme concedido salud, inteligencia para poder desarrollar mis estudios.

A mi familia: mamá, papá y hermano, quienes me han brindado amor desde mi niñez, valiosos consejos durante mi adolescencia y apoyo incondicional durante la carrera, espero algún día poder retribuirles todo lo que me han dado. ¡Gracias!

A la ilustre Universidad de Los Andes, en especial a la escuela de Ingeniería Eléctrica, que ha servido como herramienta para forjar mis conocimientos y de la que siempre llevaré gratos recuerdos.

A mi amigo y tutor el profesor José Luis Paredes, por haber influenciado mi formación como profesional, siempre en la búsqueda de la excelencia con humildad. Gracias por darme la oportunidad de desarrollar este trabajo, y orientarme en el mismo.

A Karelia Contreras quien con su amor, me ha apoyado siempre en la realización de este trabajo, reconfortando, acompañando y motivando en todo momento y demostrando que con amor todo se puede. Sinceramente ¡Gracias!

Al ingeniero Jerick Órdenes, que gracias a sus consejos se mejoró considerablemente la calidad de este trabajo, además por ser un gran amigo. A los compañeros de estudios, que terminaron convirtiéndose en mis amigos William C., Eddy C., Horfran S., Gladys U., Franklin A., Guillermo S., Roger R., Ivana Z., Robert G., Carlos S. y José F. Gracias por esos buenos ratos durante la carrera, que espero continúen. Dicen que los amigos son los hermanos que eligimos, y así ocurrió con ustedes.

Al Fondo Nacional de Ciencia, Tecnología e Innovación (Fonacit) por el financiamiento parcial de este trabajo de grado bajo el proyecto Nro. *2013001663* titulado: Procesamiento de las Imágenes Satelitales usando técnicas computacionales y de modelado inteligentes, para apuntalar la soberanía nacional.

Luis Alfredo Portillo Castillo.

**Luis Alfredo Portillo Castillo. Desarrollo de Herramientas Computacionales para Superresolución de Imágenes Satelitales Usando Representación Poco Densa de Señales en Diccionarios Redundantes.** Universidad de Los Andes. Tutor: Prof. José Luis Paredes Q. Junio, 2016.

## Resumen

Las imágenes satelitales representan una herramienta de gran ayuda para la exploración de territorios. Hoy en día los países más desarrollados invierten millones de dólares en proyectos para desarrollar tecnologías relacionados con la adquisición de imágenes satelitales. Con el reciente lanzamiento del satélite Miranda, Venezuela también puede disponer de este tipo de imágenes. La presente investigación se centra en mejorar la resolución espacial de las imágenes captadas por el satélite Miranda a partir de técnicas de postprocesamiento, utilizando para ello la teoría de representación poco densa, e implementar una serie de rutinas computacionales que faciliten su uso. Las técnicas estudiadas permiten mejorar la resolución de una imagen objetivo utilizando imágenes con características similares. Esto se logra con el uso de pares de diccionarios redundantes que representan las imágenes usando muy pocas componentes, que es el fundamento de la teoría de representación poco densa. Para adaptar los diccionarios a las características de las imágenes adquiridas por el satélite Miranda se utilizó el conocido algoritmo K-SVD, y para encontrar la mejor aproximación de éstas se usó el algoritmo de búsqueda voraz *Orthogonal Matching Pursuit*. El problema de superresolución de imágenes se resuelve a partir de proyectar el vector de representación poco densa de una imagen de baja resolución sobre un diccionario de alta resolución, previamente entrenados, y luego estimar los píxeles a partir de los datos obtenidos. Se presentan 3 enfoques para llevar a cabo esta tarea variando la forma como se entrenan los diccionarios y como se plantea el problema de optimización para encontrar la representación poco densa. Las técnicas estudiadas fueron probadas ante distintos niveles de contaminación y distorsión de la imagen, generando resultados que superan técnicas comunes como la interpolación.

**Descriptor:** Satélite Miranda, representación poco densa, superresolución, K-SVD, OMP.

## ÍNDICE GENERAL

APROBACIÓN . . . . .	ii
DEDICATORIA . . . . .	iii
AGRADECIMIENTOS . . . . .	iv
RESUMEN . . . . .	v
INTRODUCCIÓN . . . . .	1
<b>Capítulo</b>	<b>pp.</b>
<b>1. MARCO TEÓRICO . . . . .</b>	<b>4</b>
1.1 DESCRIPCIÓN DEL SATÉLITE MIRANDA . . . . .	4
1.1.1 Tipo de órbita . . . . .	5
1.1.2 Tiempo de revista . . . . .	6
1.1.3 Modo de escaneo . . . . .	6
1.2 IMÁGENES CAPTURADAS POR EL SATÉLITE . . . . .	7
1.2.1 Imágenes pancromáticas . . . . .	9
1.2.2 Imágenes Multiespectrales . . . . .	11
1.2.2.1 <i>Imágenes multiespectrales WMC</i> . . . . .	13
1.3 TEORÍA DE SUPERRESOLUCIÓN DE IMÁGENES . . . . .	14
1.3.1 El problema de superresolución . . . . .	15
1.3.2 Superresolución en el dominio de la frecuencia . . . . .	17
1.3.3 Superresolución usando interpolación . . . . .	18
1.3.4 Superresolución usando técnicas estadísticas . . . . .	20
1.3.5 Superresolución basada en ejemplos . . . . .	21
<b>2. TEORÍA DE REPRESENTACIÓN POCO DENSA . . . . .</b>	<b>23</b>
2.1 REPRESENTACIÓN POCO DENSA DE SEÑALES . . . . .	23
2.1.1 Diccionarios para representación poco densa . . . . .	25
2.1.1.1 <i>Diccionario DCT</i> . . . . .	26
2.1.1.2 <i>Diccionarios Entrenados</i> . . . . .	28
2.1.2 Representación poco densa de imágenes . . . . .	29
2.2 ALGORITMOS DE REPRESENTACIÓN POCO DENSA . . . . .	31
2.2.1 Algoritmo OMP para representación poco densa . . . . .	31
2.2.1.1 <i>Optimizando OMP con OMP de conjunto</i> . . . . .	31
2.2.2 Algoritmo K-SVD para entrenamiento de diccionarios redundantes . . . . .	36

2.2.2.1	<i>K-means para cuantización vectorial</i>	36
2.2.2.2	<i>K-SVD como generalización de K-means</i>	37
2.3	SUPERRESOLUCIÓN DE IMÁGENES USANDO REPRESENTACIÓN POCO Densa	38
2.3.1	Diccionarios semiacoplados para superresolución de imágenes	40
2.3.1.1	<i>Entrenamiento de diccionarios semiacoplados</i>	42
2.3.2	Superresolución de imágenes usando diccionarios de pseudoinversa	42
<b>3.</b>	<b>APLICACIÓN DE MÉTODOS DE SUPERRESOLUCIÓN A IMÁGENES SATELITALES</b>	<b>44</b>
3.1	MÉTODOS DE SUPERRESOLUCIÓN DE IMÁGENES SATELITALES	44
3.1.1	Superresolución usando entrenamiento conjunto de diccionarios adaptado a OMP	45
3.1.2	Superresolución usando diccionarios semiacoplados adaptado a OMP	47
3.2	RECONSTRUCCIÓN DE LA IMAGEN DE ALTA RESOLUCIÓN	48
3.2.1	Filtros de reconstrucción	49
3.2.1.1	<i>Selección de las ponderaciones</i>	50
3.2.2	Algoritmo iterativo de <i>Back-Projection</i>	53
3.3	PAQUETE DE SUPERRESOLUCIÓN DE IMÁGENES	54
<b>4.</b>	<b>SIMULACIÓN Y RESULTADOS</b>	<b>56</b>
4.1	DEGRADACIÓN DE LAS IMÁGENES	56
4.2	ANÁLISIS CUANTITATIVO	57
4.2.1	Selección de parámetros	59
4.2.2	Rendimiento variando el factor de escalamiento	59
4.2.3	Rendimiento variando tamaño de bloque	62
4.2.4	Rendimiento variando el filtro de reconstrucción	64
4.2.5	Detección de bordes como medida de rendimiento	66
4.3	ANÁLISIS CUALITATIVO	66
	<b>CONCLUSIONES</b>	<b>74</b>
	<b>RECOMENDACIONES</b>	<b>76</b>
	<b>APÉNDICE A. CÁLCULO DE LA MEDIANA PONDERADA</b>	<b>77</b>
	<b>REFERENCIAS</b>	<b>78</b>
	<b>ANEXOS</b>	<b>82</b>

## ÍNDICE DE TABLAS

<b>Tabla</b>	<b>pp.</b>
1.1 Modos de operación para la adquisición de imágenes . . . . .	7
1.2 Detalles de la imagen pancromática . . . . .	10
1.3 Información de las bandas de las imágenes multiespectrales . . . . .	12
1.4 Detalles de las imágenes multiespectrales . . . . .	14
2.1 Algoritmo OMP . . . . .	32
2.2 Algoritmo OMP <i>Batch</i> . . . . .	35
2.3 Algoritmo K-SVD . . . . .	38
4.1 Medidas de desempeño . . . . .	57

www.bdigital.ula.ve

## ÍNDICE DE FIGURAS

<b>Figura</b>	<b>pp.</b>
1.1 Órbita polar síncrona al sol . . . . .	6
1.2 Composición del satélite Miranda . . . . .	8
1.3 Proceso de adquisición de imágenes del satélite . . . . .	9
1.4 Imagen pancromática tomada por la cámara PMC . . . . .	10
1.5 Imagen multiespectral captada por una cámara PMC . . . . .	11
1.6 Captura de imágenes . . . . .	12
1.7 Imagen multiespectral captada por la cámara WMC . . . . .	13
1.8 Modelo de observación de una imagen, con movimiento en la escena . . . . .	15
1.9 Principio básico para superresolución de imágenes, utilizando múltiples imágenes de baja resolución . . . . .	17
1.10 Técnica de superresolución usando interpolación y posterior deconvolución	19
1.11 Bloques de baja y alta resolución . . . . .	22
2.1 Diccionario 2D-DCT completo de 64 átomos de $8 \times 8$ píxeles . . . . .	27
2.2 Diccionario DCT sobrecompleto, 256 átomos de $8 \times 8$ píxeles . . . . .	28
2.3 Reordenamiento de una imagen en bloques de $2 \times 2$ píxeles . . . . .	30
3.1 Ejemplo de la selección de bloques en la misma región de 2 imágenes . . . . .	46
3.2 Ubicación de los bloques después de un escalamiento por un factor 3 . . . . .	49
3.3 Cantidad de píxeles usados para la estimación . . . . .	50
3.4 Composición de un píxel . . . . .	51
3.5 Función de afinidad para la ponderación de las muestras . . . . .	52
4.1 Imagen de prueba para el análisis cuantitativo . . . . .	58
4.2 Imágenes de entrenamiento para el análisis cuantitativo . . . . .	58
4.3 PSNR usando bloques de $3 \times 3$ y factor de escalamiento = 2 . . . . .	59
4.4 PSNR usando bloques de $3 \times 3$ y factor de escalamiento = 3 . . . . .	60
4.5 PSNR usando bloques de $3 \times 3$ y factor de escalamiento = 4 . . . . .	60
4.6 MAE Variando el tamaño de bloque y factor de escalamiento = 2 . . . . .	62
4.8 MAE Variando el tamaño de bloque y factor de escalamiento = 4 . . . . .	62
4.7 MAE Variando el tamaño de bloque y factor de escalamiento = 3 . . . . .	63
4.9 RMSE usando bloques de $4 \times 4$ y factor de escalamiento = 2 . . . . .	64
4.10 RMSE usando bloques de $3 \times 3$ y factor de escalamiento = 3 . . . . .	65
4.11 RMSE usando bloques de $3 \times 3$ y factor de escalamiento = 4 . . . . .	65
4.12 Resultados de la detección de bordes . . . . .	67

	x
4.13 Imágenes de entrenamiento para el análisis cualitativo . . . . .	68
4.14 Resultados del análisis cualitativo . . . . .	68
4.15 Sierra nevada captada por la cámara PMC 10 m/pixel . . . . .	69
4.16 Imágenes de “La guaira” usadas para entrenar . . . . .	70
4.17 Sierra nevada ampliada por un factor de 4 usando el método de diccionarios conjuntos . . . . .	71
4.18 Sierra nevada ampliada por un factor de 4 usando el método de diccionarios semiacoplados . . . . .	72
4.19 Sierra nevada ampliada por un factor de 4 usando el método de diccionarios de pseudoinversa . . . . .	73

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

# INTRODUCCIÓN

Las imágenes satelitales que son tomadas de la superficie terrestre, tienen una gran variedad de usos entre los que se destacan, meteorología, oceanografía, pesca, agricultura, conservación de la biodiversidad, geología, entre muchos otros. En Venezuela el instituto encargado de capturar y administrar dichas imágenes es, la Agencia Bolivariana para Actividades Espaciales (ABAE). Recientemente con el lanzamiento del satélite “Miranda” el 28 de septiembre de 2012 con propósitos exclusivamente de imagenología Venezuela se encuentra entre los países del mundo con capacidad de generar este tipo de imágenes. Éstas se dividen en, pancromáticas [1], que captan el espectro visible para el ojo humano en un solo canal sin distinguir colores, y multiespectrales [1], que contienen mas información del espectro electromagnético separado en múltiples canales. La resolución de estas imágenes a su vez puede ser clasificada como, resolución espacial, resolución espectral, resolución temporal, resolución radiométrica [2]. La resolución espacial está asociada con la cantidad de píxeles de la imagen, la resolución espectral con las longitudes de onda capturadas por el sensor, la resolución temporal tiene que ver con el tiempo que tomo capturar la imagen, y la resolución radiométrica con los distinto niveles de brillo de la imagen.

Para obtener imágenes de mejor resolución espacial básicamente existen dos alternativas. La primera sería mejorar el aspecto físico de los equipos empleados, es decir, mejorar las cualidades de captura usando cámaras con sensores de mejor calidad que representan un costo mayor y posiblemente un consumo de energía elevado<sup>1</sup>. Otra alternativa es procesar la imagen usando métodos numéricos que buscan resolver el problema a partir del modelo de observación de la imagen dado por:

$$\mathbf{Y} = D\mathbf{H}\mathbf{X} + \boldsymbol{\eta} \quad (1)$$

Donde  $\mathbf{Y}$  es la imagen de baja resolución,  $D$  el operador de diezmado que reduce la

---

<sup>1</sup>En algunos casos una cámara de mejor calidad posee una mayor densidad de receptores por unidad de área, que se traduce en más dispositivos que energizar

dimensión,  $H$  es un filtro paso bajo que degrada la imagen,  $\mathbf{X}$  la escena de alta resolución y  $\boldsymbol{\eta}$  la contaminación existente en el proceso de captura, desde el ruido presente en la atmósfera hasta el que introducen los componente electrónicos, las técnicas que buscan resolver el problema definido en (1) para encontrar  $\mathbf{X}$  son conocidas en la literatura como técnicas de superresolución (SR) de imágenes. El campo de la SR ha sido ampliamente estudiado y se han propuesto varios enfoques para resolver el problema. Uno de los más populares es usar modelos de alta resolución que recobran una imagen de alta fusionando múltiples imágenes de baja sobre las bases de un modelo previamente conocido de alta resolución, utilizando para ello interpolación bilineal o bicúbica. Este enfoque es restringido para imágenes satelitales, ya que, genera un efecto de suavizado. Hoy día los últimos adelantos científicos en esta área generalmente forman parte de un paquete de un *software* privativo, por ejemplo MATLAB®), por eso es necesario implementar estas técnicas en lenguajes de programación libres que faciliten su acceso. En la presente investigación se aborda el problema bajo la óptica de la teoría representación poco densa de señales.

La teoría de representación poco densa de señales es relativamente nueva en el área de procesamiento digital de señales, que ha venido cogiendo auge en los últimos años. Ésta consiste en aproximar una señal con mínimo error usando pocos elementos de un conjunto de bases denominado diccionario, conduciendo al siguiente problema de optimización:

$$\mathbf{y} = \Phi\boldsymbol{\gamma} \text{ tal que } \|\mathbf{y} - \Phi\boldsymbol{\gamma}\|^2 < \epsilon \quad (2)$$

Donde  $\mathbf{y}$  es la señal objetivo,  $\Phi$  el diccionario que agrupa las señales base conocidas como átomos y  $\boldsymbol{\gamma}$  la representación poco densa de la señal  $\mathbf{y}$  en  $\Phi$ . Recientemente el uso de representación poco densa sobre imágenes tuvo resultados satisfactorios en problemas como restauración [3], descontaminación y SR de imágenes debido a la fuerte correlación existente entre píxeles cercanos, razón por la cual se esperan buenos resultados al extrapolarla a imágenes satelitales. El objetivo principal de esta investigación es desarrollar herramientas computacionales basadas en *software* libre que faciliten el empleo de las técnicas de SR que utilicen representación poco densa de señales sobre diccionarios redundantes, y aplicarlas al procesamiento de imágenes captadas por el satélite Miranda. Dentro de los objetivos específicos se encuentran:

- Estudiar los principios básicos que gobiernan la adquisición y precondicionamiento de imágenes satelitales.
- Estudiar las teorías de representación de imágenes en diccionarios redundantes.
- Estudiar las técnicas de SR basadas en representación poco densa y seleccionar las más apropiadas para imágenes satelitales.
- Implementar los algoritmos más adecuados usando herramientas de *software* libre.
- Integrar el módulo desarrollado en una herramienta de visualización de imágenes satelitales.

Este documento se encuentra organizado de la siguiente manera: el capítulo I, aborda las características del satélite y el proceso de adquisición de imágenes, a fin de extraer información de las características de éstas que puedan ser usadas en el proceso de mejora de su resolución, además se estudia el modelo de observación de imágenes usado para simular el proceso de adquisición y a partir del cual se resuelve el problema de SR, que es un problema del tipo inverso, cuya solución no es única y por lo tanto, una variedad de algoritmos se han propuesto para resolverlo, los cuales serán repasados. El capítulo II, está enfocado en la teoría de representación poco densa de señales, abordándola como un problema de optimización, que busca un equilibrio entre el mínimo error y la menor cantidad de componentes que pueden aproximar la señal. En principio se explica el algoritmo OMP utilizado para encontrar la representación poco densa de una señal sobre un diccionario siguiendo unos parámetros de optimización, luego se explica el algoritmo K-SVD utilizado para entrenar diccionarios y por último se explican 3 técnicas de SR basadas en esta teoría. El capítulo III, particulariza las técnicas estudiadas anteriormente para el caso de imágenes satelitales, adaptando las metodologías encontradas a los algoritmos OMP y K-SVD. El capítulo IV corresponde a la parte experimental, donde se hacen una serie de simulaciones para validar la calidad de las técnicas desarrolladas, estudiando su desempeño ante contaminación gaussiana variando distintos parámetros de los algoritmos, además se hacen algunas pruebas para casos reales aprovechando las distintas resoluciones de las imágenes capturadas por el satélite. Por último se plasman las conclusiones basadas en los resultados y realizan algunas recomendaciones para futuros trabajos.

# CAPÍTULO 1

## MARCO TEÓRICO

*La percepción remota constituye hoy día una gran alternativa para estudiar territorios, ya que los avances en tecnología espacial han permitido utilizar satélites con mejores características. Venezuela no se queda atrás en esta iniciativa, razón por la cual, lanzó el satélite “Francisco de Miranda” para una misión de exploración del territorio Nacional, equipado con una serie de cámaras de distintas resoluciones espaciales y espectrales. En general, la resolución de las imágenes habla de su nivel de detalle, permitiendo obtener información mas relevante a mayor resolución, sin embargo, no siempre se tiene disponibilidad de sistemas de alta resolución, razón por la cual, se han desarrollado técnicas de superresolución para mejorar la calidad de las imágenes. En el siguiente capítulo se hablará sobre el satélite Miranda, sus características, y funcionamiento para adquisición de imágenes; además, se expondrá las principales técnicas de superresolución, haciendo un breve repaso desde su aparición en el campo de procesamiento de imágenes, hasta las últimas técnicas desarrolladas hoy día.*

### 1.1 DESCRIPCIÓN DEL SATÉLITE MIRANDA

Los satélites pueden ser clasificados según la misión que vayan a desenvolver (comunicación, teledetección, exploración, entre otras), ésta especifica sus características como, el tipo de órbita, los artefactos empleados para la misión, denominados carga útil, y la plataforma que da el soporte para que la misión pueda ser llevada a cabo. El satélite VRSS-1 (*Venezuelan Remote Sensing Satellite-1*), conocido como satélite “Francisco de Miranda”, es un satélite para teledetección, específicamente para percepción remota con la misión de capturar imágenes del territorio nacional. Sus características y funcionamiento serán descritos en

las siguientes secciones.

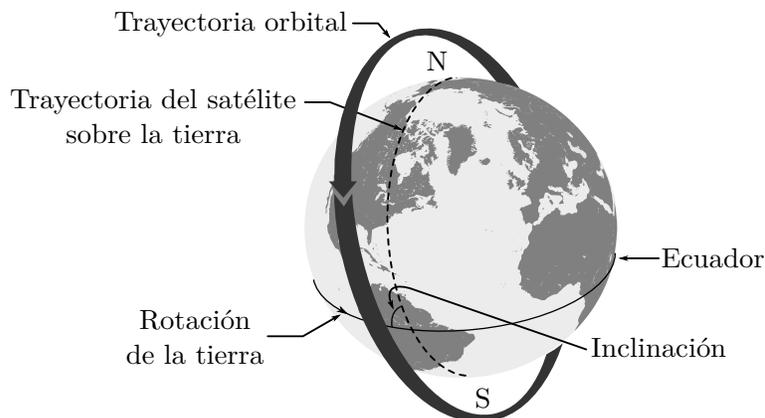
El satélite miranda cuenta con una plataforma tipo CAST2000 [4] y se encuentra equipado con los siguientes elementos:

- Carga útil:
  - Subsistemas de cámaras (PMC *Panchromatic and Multispectral Camera* y WMC *Wide-swath Multispectral Camera*).
  - Subsistema de transmisión de datos (DTS *Data Transfer System*).
- Plataforma:
  - Estructura y mecanismo (SMS *Structure & Mechanism Subsystem*).
  - Altitud y control de órbita (AOCS *Altitude and Orbit Control Subsystem*).
  - Suministro de potencia (PSS *Power Supply Subsystem*).
  - Manejo de datos a bordo (OBDH *On-Board Data Handling*).
  - Control térmico (TCS *Thermal Control Subsystem*)
  - Telemetría y Telecomando (TT&C *Telemetry, Tracking & Command*).

### 1.1.1 Tipo de órbita

Para adquirir imágenes existen 2 tipos de satélites, activos y pasivos; su diferencia radica en que los primeros iluminan la escena artificialmente, mientras que los segundos aprovechan la iluminación del sol; razón por la cual su órbita debe asegurarse que se capturen imágenes a una misma hora, de manera tal que la brillantez de las escenas sea similar, a fin de disminuir errores en la generación de las imágenes.

El satélite Miranda es del tipo pasivo y su órbita que se puede apreciar en la Fig. 1.1 [5] es LEO (*Low Earth Orbit*) del tipo polar síncrona al sol, es decir, que transita cerca de los polos. Orbita a una altura de 639,54 km de norte a sur con una hora de captura al rededor de las 10:30 a.m., su inclinación es de 98° aproximadamente [6].



**Figura 1.1. Órbita polar síncrona al sol.**

### 1.1.2 Tiempo de revista

El tiempo de revista está asociado con lo que tarda el satélite en cubrir una zona que había explorado anteriormente. Para misiones de teledetección es importante que estos tiempos sean cortos, con el fin de monitorear adecuadamente eventos como, inundaciones, incendios, cosechas, entre otros; este tiempo también depende de la cámara que se utilice, ya que, las dimensiones de las áreas a cubrir pueden variar considerablemente.

La velocidad en órbita del satélite Miranda es de 7,504 km/s, dando 14 vueltas al planeta en un día y debido a la inclinación de su órbita pasa 2 o 3 veces sobre el territorio nacional, lo cual ralentiza mucho la cobertura sobre una zona a aproximadamente 57 días utilizando la cámara PMC y 12 días utilizando la cámara WMC. Para disminuir estos tiempos el satélite puede ser girado sobre su eje  $\pm 35^\circ$  con un control de maniobra, reduciéndolos a:

- 3 días usando las cámaras multiespectrales de barrido ancho (WMC)
- 4 días usando las cámaras pancromáticas-multiespectrales (PMC)

### 1.1.3 Modo de escaneo

Para percepción remota existen 2 alternativas para obtener las imágenes. Una es escaneando en la misma dirección de la trayectoria del satélite, la otra es de manera transversal a la trayectoria (conocida como *pushbroom* [5]); esta última es usada por el satélite. Cuando el modo de escaneo es *pushbroom* la imagen se obtiene haciendo un barrido sobre la

zona deseada, componiéndola línea por línea a medida que el satélite se desplaza en su trayectoria.

Dada la velocidad en órbita, el satélite se encuentra aproximadamente 10 min sobre el territorio nacional, por lo cual, presenta 4 modos de operación para la obtención de imágenes. Estos modos, resumidos en la Tabla 1.1 [4], influyen en algunos aspectos de la captura de la imagen como, la razón de compresión, el ancho de banda utilizado para la transmisión y el consumo de memoria de estado sólido (SSM *Solid State Memory*) [4,6].

**Tabla 1.1. Modos de operación para la adquisición de imágenes.**

Modo	Compresion	SSM	Transmisión
CUASI-TIEMPO REAL	PAN 3:1 MS <sup>1</sup> : Sin pérdidas WMC: Sin pérdidas	380 Mbps	–
TIEMPO REAL	PAN 5:1 MS: 4:1 WMC: 4:1	380 Mbps	2x190 Mbps
GRABACIÓN	Cualquiera	–	–
REPRODUCCIÓN	–	380 Mbps	2x190 Mbps

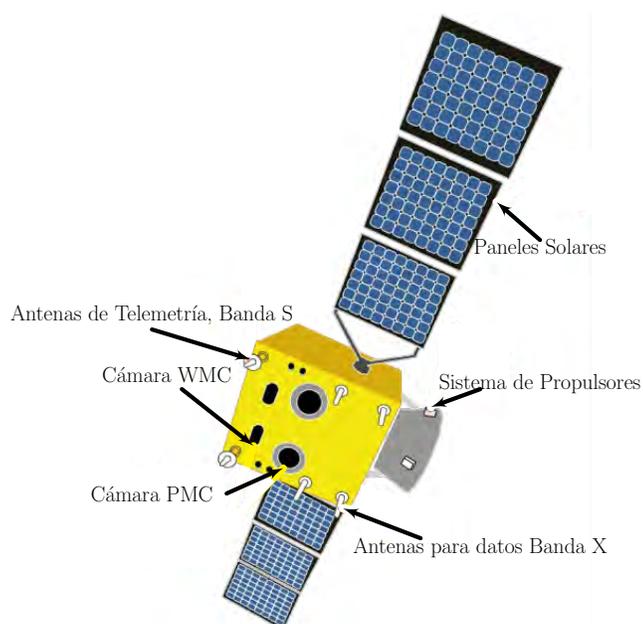
## 1.2 IMÁGENES CAPTURADAS POR EL SATÉLITE

El satélite cuenta con 2 pares de cámaras en su carga útil; estas son, pancromáticas-multiespectrales (PMC) de alta resolución, y multiespectrales de barrido ancho (WMC) de resolución media, cuya disposición se puede apreciar en la Fig. 1.2 [4]. Estas cámaras capturan 3 tipos de imágenes, con distintas resoluciones espectrales y espaciales utilizando arreglos lineales de sensores CCD<sup>2</sup> (*Charged Copuled Device*) con TDI (*Time Delay Integration*). El uso de la tecnología TDI se justifica debido al modo de escaneo (*pushbroom*) permitiendo que los detectores CCD se recarguen proporcionalmente a la proyección de la escena sobre ellos evitando imágenes borrosas y aumentando la relación señal a ruido

<sup>1</sup>Imágenes multiespectrales.

<sup>2</sup>La información obtenida por cada sensor corresponde a un pixel de la imagen

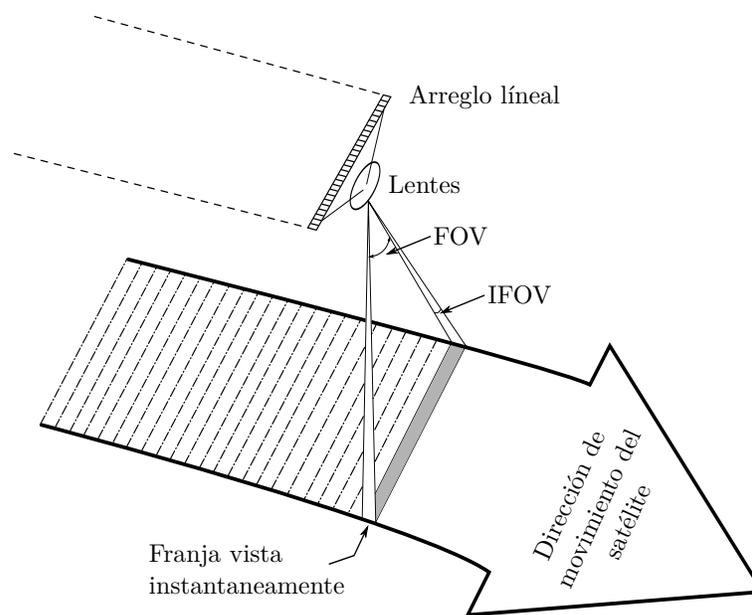
(SNR *Signal-to-Noise Ratio*), esto se logra con una calibración adecuada [7].



**Figura 1.2. Composición del satélite Miranda.**

El proceso de adquisición de imágenes que se muestra en la Fig. 1.3 [5], se produce cuando la radiación emitida por la superficie terrestre incide sobre el arreglo de sensores, dentro del campo de visión de estos conocido como FOV (*Field Of View*), la información captada en un instante de tiempo se encuentra dentro del IFOV (*Instantaneous Field Of View*) [1]. La señal analógica generada por los sensores es muestreada y cuantizada a 10 bits para generar la imagen digital, similar a la grabación de un vídeo donde existe movimiento en la escena; de esta manera cada trama del vídeo correspondería a cada línea de la imagen obtenida durante el escaneo. Finalmente los productos obtenidos son los siguientes:

- Imágenes pancromáticas captadas por la cámara PMC.
- Imágenes multiespectrales captadas por la cámara PMC.
- Imágenes multiespectrales captadas por la cámara WMC.



**Figura 1.3. Proceso de adquisición de imágenes del satélite.**

### 1.2.1 Imágenes pancromáticas

Las imágenes pancromáticas son captadas por el par de cámaras PMC que poseen 4 arreglos de sensores CCD (2 en cada cámara, con distintos planos focales), estas imágenes poseen la mayor resolución espacial y la menor espectral. El término “pancromático” significa “a través de los colores”, indicando que la imagen es generada a partir de la información de todo el espectro visible representado en una sola banda en escala de grises, variando los niveles de brillo sin distinguir entre colores. El uso de éstas se hace necesario cuando se requiere mayor detalle espacial de los objetos de la imagen despreciando la información que puedan aportar los colores; en general existen 2 maneras de componer estas imágenes. Utilizando la información del espectro visible para el ojo humano, es decir, utilizar la información de las bandas rojo, verde y azul (RGB *Red, Green, Blue*). La otra alternativa es sustituir la banda azul con la cercana al infrarrojo (NIR *Near to Infra-Red*), para omitir la información que puede aportar la atmósfera, que en algunos caso de percepción remota se hace innecesaria.

Las imágenes pancromáticas capturadas por el satélite Miranda se componen a partir del espectro visible, cubriendo un ancho de aproximadamente 30 km con respecto al nadir.<sup>3</sup>

<sup>3</sup>El nadir es conocido como el punto central de la escena, ya que se encuentra exactamente debajo de la cámara.

La cantidad de sensores TDI-CCD disponibles para capturar estas imágenes es de 12000 (divididos en 2 arreglos) lo que se traduce en una resolución espacial de aproximadamente 2,5 m/píxel [6]. En la Fig. 1.4 se muestra una imagen pancromática captada por el satélite Miranda sobre el aeropuerto internacional “Simón Bolívar” de Maiquetía; las características de este tipo de imágenes se muestran en la Tabla 1.2.



Figura 1.4. Imagen pancromática tomada por la cámara PMC.

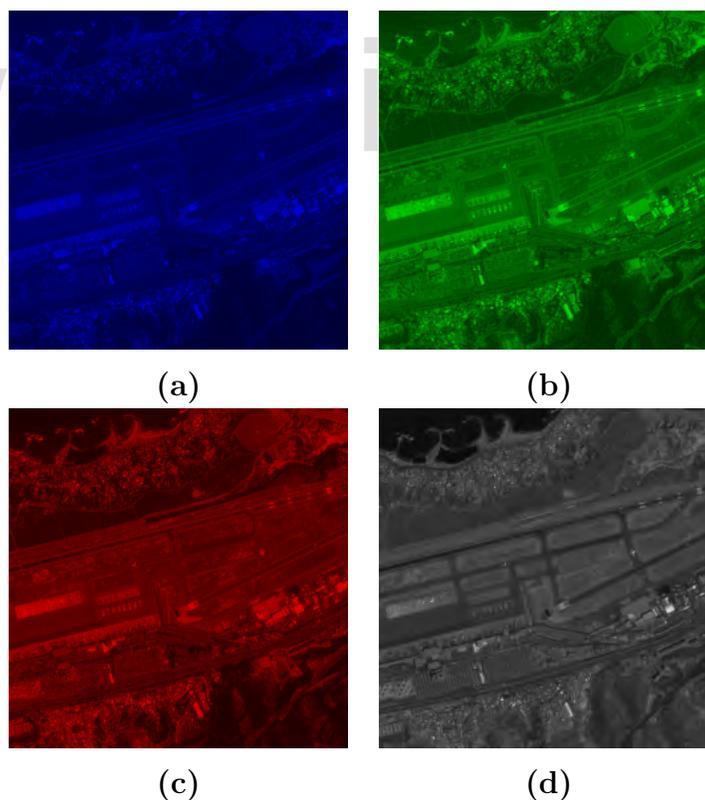
Tabla 1.2. Detalles de la imagen pancromática

TIPO DE ESCANEADO	<i>pushbroom</i>
BANDAS ESPECTRALES	PAN: (0,45 - 0,90) $\mu\text{m}$
RESOLUCIÓN ESPACIAL	2,5 m/píxel
RESOLUCIÓN RADIOMÉTRICA	10 bits
ANCHO DE LA IMAGEN	30 km
DETECTOR CCD	12000 píxeles de 10 $\mu\text{m}$ x 10 $\mu\text{m}$
SNR	48 dB (ángulo solar respecto al <i>Cenit</i> 70°, superficie de albedo <sup>4</sup> 0,65)

<sup>4</sup>Porcentaje de radiación que cualquier superficie refleja respecto a la radiación que incide sobre la misma.

### 1.2.2 Imágenes multispectrales

Las imágenes multispectrales son el otro producto que puede ser capturado con el satélite Miranda. Estas imágenes son captadas tanto por el sistema de cámaras PMC, como por el WMC, con diferencias entre ambos. Las imágenes poseen la información brindada por distintas longitudes de onda dentro del espectro electromagnético. La información de cada banda es guardada en una imagen cuantizada de igual manera a 10 bits. El uso de estas se hace necesario cuando se requiere discriminar superficies en la imagen. En la Fig. 1.5 se muestra una imagen multispectral tomada por el satélite Miranda correspondiente a una porción del aeropuerto internacional “Simón Bolívar” de Maiquetía separada por bandas. Donde la Fig. 1.5(a) es la banda azul, la Fig. 1.5(b) es la banda verde, la Fig. 1.5(c) es la banda roja, la Fig. 1.5(d) es la banda cercana al infrarrojo representada en escala de grises. La información que aporta cada banda se detalla en la Tabla 1.3 [5].



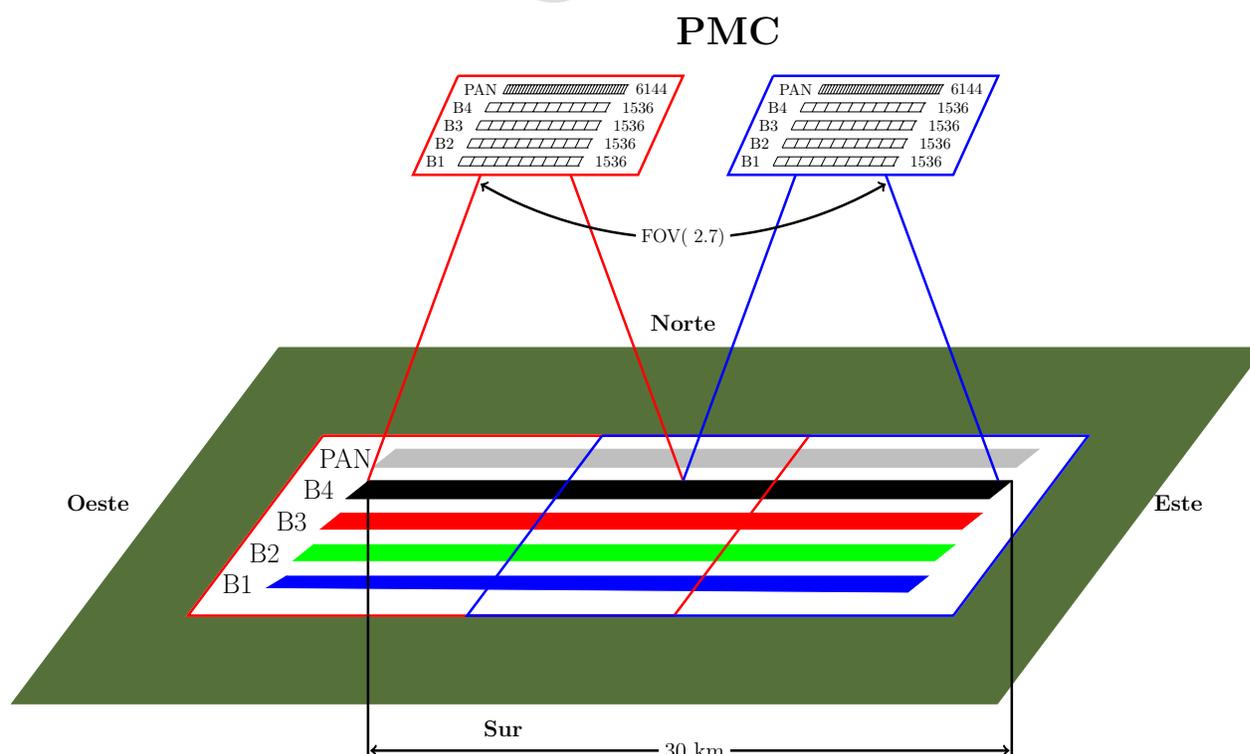
**Figura 1.5. Imagen multispectral captada por una cámara PMC.**

La información que aporta un mismo píxel a lo largo del espectro, se conoce como firma espectral; ésta no es mas que un vector unidimensional que contiene los distintos

**Tabla 1.3. Información de las bandas de las imágenes multiespectrales.**

BANDA 1: AZUL (0,45-0,52) $\mu\text{m}$	Penetración de agua clara, batimetría, cartografía de las aguas costeras, absorción de clorofila, distinción entre la vegetación de coníferas y caducifolio.
BANDA 2: VERDE (0,52-0,59) $\mu\text{m}$	Registra la radiación verde reflejada por la vegetación sana, evalúa vigor de la planta, reflectancia de agua turbia.
BANDA 3: ROJO (0,63-0,69) $\mu\text{m}$	Absorción de clorofila importante para la discriminación del tipo de planta.
BANDA 4: NIR (0,77-0,89) $\mu\text{m}$	Indicador de la estructura de la célula vegetal, biomasa, vigor de la planta, la completa absorción por el agua facilita la delimitación de costas.

valores del píxel en cada banda. Para capturar estas imágenes se usan distintos arreglos de sensores TDI-CCD que reaccionan a una longitud de onda diferente, generando 4 señales diferentes para una misma área captada por los lentes de la cámara. En la Fig. 1.6 [6], se ilustra este proceso. Para el sistema de cámaras PMC se utilizan 3000 sensores por banda arrojando así una resolución espacial de aproximadamente 10 m/píxel [6].



**Figura 1.6. Captura de imágenes.**

### 1.2.2.1 *Imágenes multiespectrales WMC*

El sistema de cámaras WMC permite capturar solamente imágenes multiespectrales, a diferencia del sistema de cámaras PMC; éstas cubren un área mayor y poseen menor resolución espacial. El incremento del ancho de barrido se logra por la óptica refractiva de los lentes, ensanchando el área visible de la cámara. Las imágenes WMC se utilizan cuando extensas zonas del territorio quieren ser capturadas; una aplicación para el uso de estas imágenes es estudiar posibles fenómenos naturales. Cada cámara WMC posee 12000 sensores TDI-CCD distribuidos en 4 arreglos a distintos planos focales, con una resolución de 16 m/píxel. En la Fig. 1.7 se muestra una imagen multiespectral captada por la cámara WMC usando el espectro visible (RGB) que cubre parte del estado Vargas, su espacio marítimo, y una fracción de la ciudad de Caracas. Las características de este tipo de imágenes se muestra en la Tabla 1.4 [4].

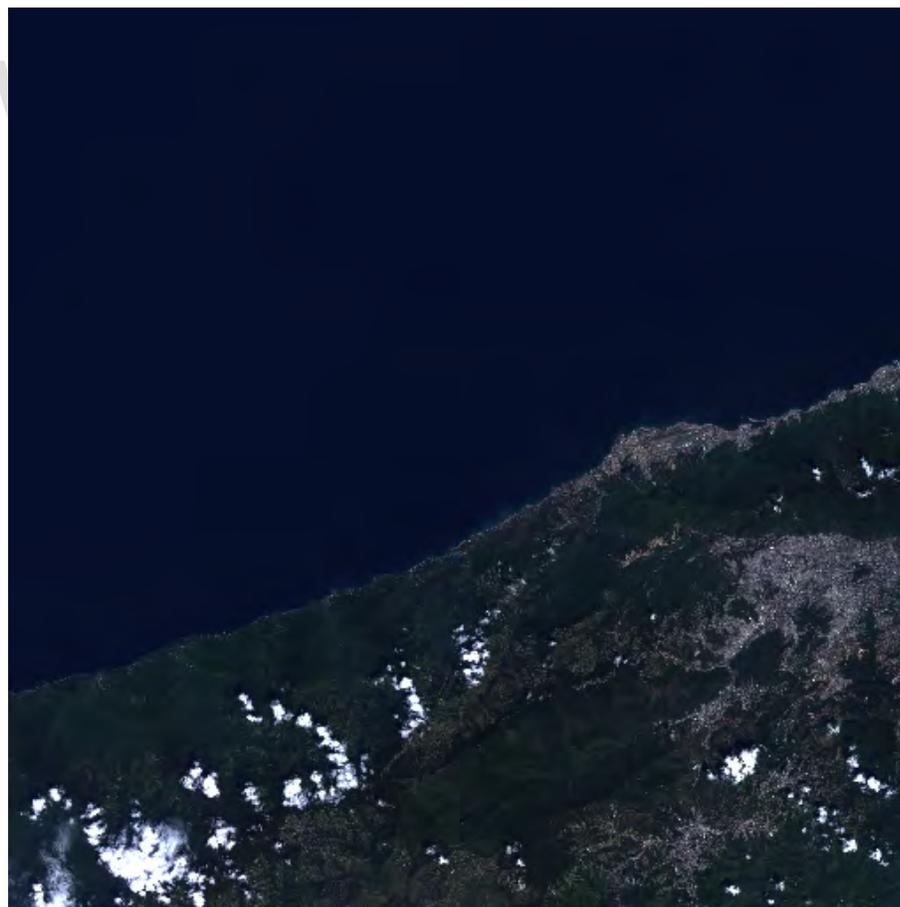


Figura 1.7. Imagen multiespectral captada por la cámara WMC.

Tabla 1.4. Detalles de las imágenes multispectrales.

CÁMARA UTILIZADA	PMC	WMC
TIPO DE ESCANEADO	<i>pushbroom</i>	<i>pushbroom</i>
BANDAS ESPECTRALES	Banda 1: Azul (0,45-0,52) $\mu\text{m}$ Banda 2: Verde (0,52-0,59) $\mu\text{m}$ Banda 3: Rojo (0,63-0,69) $\mu\text{m}$ Banda 4: NIR (0,77-0,89) $\mu\text{m}$	Banda 1: Azul (0,45-0,52) $\mu\text{m}$ Banda 2: Verde (0,52-0,59) $\mu\text{m}$ Banda 3: Rojo (0,63-0,69) $\mu\text{m}$ Banda 4: NIR (0,77-0,89) $\mu\text{m}$
RESOLUCIÓN ESPACIAL	10 m/píxel	16 m/píxel
RESOLUCIÓN RADIOMÉTRICA	10 bits	10 bits
ANCHO DE LA IMAGEN	30 km	192 km
DETECTOR CCD	3000 píxeles de 40 $\mu\text{m}$ x 40 $\mu\text{m}$	12000 píxeles de 6,5 $\mu\text{m}$ x 6,5 $\mu\text{m}$
SNR	48 dB (ángulo solar respecto al <i>Cenit</i> 70°, superficie de albedo 0,65)	20 dB (ángulo solar respecto al <i>Cenit</i> 15°, superficie de albedo 0,65) 46 dB (ángulo solar respecto al <i>Cenit</i> 70°, superficie de albedo 0,65)

### 1.3 TEORÍA DE SUPERRESOLUCIÓN DE IMÁGENES

Como se pudo ver en las secciones anteriores la calidad de una imagen en términos de resolución, está definida por las características físicas de las cámaras que captan la imagen. Mientras más sensores sean utilizados mejor será la resolución de la imagen. Las principales limitaciones para el *hardware* residen en el costo y la energía requerida por la cámara. Un número mayor de sensores incrementa los costos tanto económicos como energéticos; además, el sistema de control para la captura tiene que ser más complejo y los dispositivos de almacenamiento de mayor capacidad. Por estas razones se plantea una solución de postprocesamiento vía *software*, manipulando la imagen con ayuda de un computador. Las modificaciones que se hacen a una imagen con el fin de mejorar la resolución espacial para obtener mayor información, se conocen en la literatura como, superresolución (SR)

de imágenes. Dichas técnicas han sido desarrolladas desde finales del siglo pasado y han ido mejorando a través del tiempo, tanto en robustez como en calidad de reconstrucción.

### 1.3.1 El problema de superresolución

Para poder encontrar un modelo matemático que permita resolver el problema es necesario analizar los elementos que intervienen en el proceso de captura de la imagen. Como se mencionó en las secciones anteriores, una imagen se captura cuando la radiación electromagnética de una escena incide en un grupo de sensores que reaccionan a la luz, generando una señal analógica que luego es digitalizada. Este proceso no es ideal y se ve sometido a una serie de contaminaciones y distorsiones; la contaminación se debe al ruido que pueda introducir la atmósfera, por su composición, junto con los sensores (ruido eléctrico). Las distorsiones que sufre la imagen con respecto a la escena están asociadas con el movimiento, y con la distorsión propia del dispositivo, modelada a través de la función de dispersión de punto (PSF *Point Spread Function*). Finalmente la imagen obtenida es una versión borrosa, diezmada y contaminada de la escena original. Este proceso se muestra en la Fig. 1.8 [8].

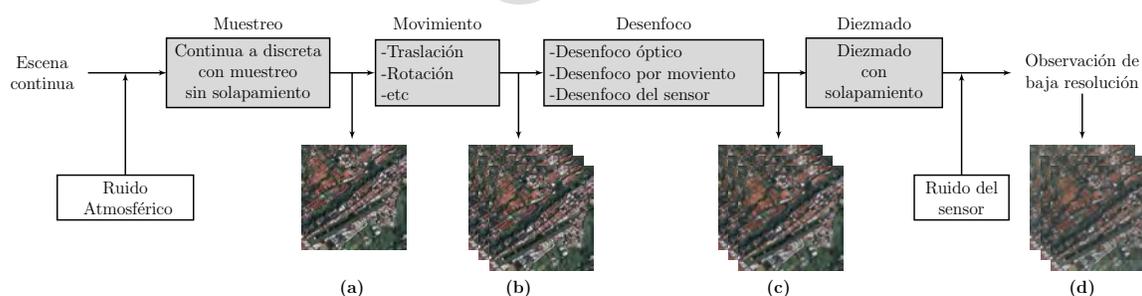


Figura 1.8. Modelo de observación de una imagen, con movimiento en la escena.

**Definición 1.1. Función de dispersión de punto** Es una función lineal que describe la respuesta de un sistema de adquisición de imágenes, cuando la fuente es un objeto con forma de punto; exaltando como éste se dispersa en la imagen. A partir de esta función se puede modelar el efecto de desenfoco del sistema de adquisición, considerando que una escena se compone de puntos distribuidos en el espacio.

Conocidos los elementos que intervienen en la adquisición de una imagen, se puede afirmar, que una gran cantidad de imágenes de baja resolución pueden ser obtenidas de

una misma escena. Denotando la variable  $\mathbf{X}$  como una escena de alta resolución,  $\mathbf{Y}_k$  como la  $k$ -ésima imagen de baja resolución,  $H_k, F_k, D$  y  $V_k$  los operadores que modelan la dispersión del sensor, el movimiento de la escena, el diezmado y el ruido respectivamente, entonces, el problema de superresolución puede ser planteado como [8]:

$$\mathbf{Y}_k = H_k F_k D \mathbf{X} + \mathbf{V}_k \quad (1.1)$$

Debido a que los operadores  $H_k$  y  $F_k$  tienen pocos elementos no nulos (*sparse matrix*), el problema en (1.1) es del tipo inverso de naturaleza mal definido (*ill-posed*), ya que, el sistema de ecuaciones para encontrar su solución es subdeterminado. Como el objetivo de la superresolución es conseguir la mejor aproximación de  $\mathbf{X}$  dada la imagen  $\mathbf{Y}_k$  y los operadores  $H_k$ , y  $F_k$  generalmente son desconocidos, el problema también es mal condicionado (*ill-conditioned*), significando que un pequeño error en los datos iniciales puede ser mucho mayor en la respuesta.

**Definición 1.2. Problemas mal definidos** Los problemas mal definidos (*ill-posed*) en el sentido de Hadamard son aquellos que no cumplen las siguientes condiciones:

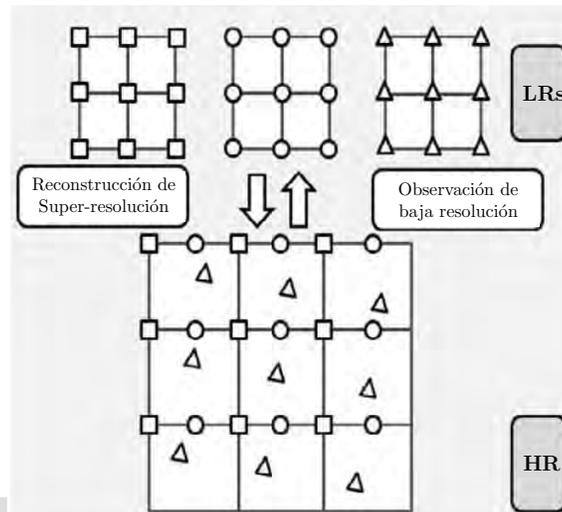
1. La solución existe.
2. La solución es única.
3. El comportamiento de la solución depende de las condiciones iniciales.

Si se agrupan todas las posibles imágenes de baja resolución junto con sus respectivos operadores de degradación en un arreglo matricial, el problema de superresolución se puede generalizar de la siguiente manera [8]:

$$\begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \mathbf{Y}_3 \\ \vdots \\ \mathbf{Y}_k \end{bmatrix} = \begin{bmatrix} D_1 H_1 F_1 \\ D_2 H_2 F_2 \\ D_3 H_3 F_3 \\ \vdots \\ D_k H_k F_k \end{bmatrix} \mathbf{X} + \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \mathbf{V}_3 \\ \vdots \\ \mathbf{V}_k \end{bmatrix} \quad (1.2)$$

$$\mathbf{Y} = \mathbf{M}\mathbf{X} + \mathbf{V} \quad (1.3)$$

El hecho de que muchas imágenes desplazadas entre sí puedan ser obtenidas de una misma escena (1.2), hace pensar, que una imagen de alta resolución puede ser generada a partir de varias imágenes de baja, si existe desplazamiento entre los píxeles [9]. En la Fig. 1.9 [8] se ilustra este proceso.



**Figura 1.9.** Principio básico para superresolución de imágenes, utilizando múltiples imágenes de baja resolución.

En general el gran desafío de las técnicas de superresolución, es recuperar componentes de alta frecuencia perdidas durante la adquisición de la imagen, evitando en lo posible generar artefactos indeseados y amplificar el ruido inicial.

### 1.3.2 Superresolución en el dominio de la frecuencia

Este enfoque fue propuesto por Tsai y Huang [10], quienes fueron pioneros en el área. Se basa en la existencia de componentes de alta frecuencia solapadas debido al diezmado en la adquisición. Aprovechando la propiedad de desplazamiento de la transformada de Fourier, la relación de solapamiento que existe entre la CFT (*Continuous Fourier Transform*) y la DFT (*Discrete Fourier Transform*); asumiendo que las imágenes son señales de banda, el problema de superresolución se resuelve de la siguiente manera.

Sea  $x(t_1, t_2)$  la escena original y  $X(w_1, w_2)$  su respectiva transformada, una versión desplazada de la imagen correspondería a,  $x_k(t_1 + \delta k_1, t_1 + \delta k_2)$ , aplicando la propiedad de desplazamiento de la CFT:

$$X_k(w_1, w_2) = e^{j2\pi(\delta k_1 + \delta k_2)} X(w_1, w_2) \quad (1.4)$$

La correspondiente imagen de baja resolución  $y_k[n_1, n_2] = x_k[n_1 T_1 + \delta k_1, n_2 T_2 + \delta k_2]$ , donde  $T_1$  y  $T_2$  son los períodos de muestreo, con  $n_1 = 0, 1, 2, \dots, N_1 - 1$  y  $n_2 = 0, 1, 2, \dots, N_2 - 1$ , la DFT de imagen de baja resolución corresponde a:

$$Y_k[r_1, r_2] = \frac{1}{T_1 T_2} \sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} X_k \left( \frac{2\pi}{T_1} \left( \frac{r_1}{N_1} - m_1 \right), \frac{2\pi}{T_2} \left( \frac{r_2}{N_2} - m_2 \right) \right) \quad (1.5)$$

Para estimar la imagen de alta resolución a partir de estas suposiciones, se plantea el siguiente sistema de ecuaciones:

$$\mathbf{y} = \Phi \mathbf{x} \quad (1.6)$$

Donde  $\mathbf{y}$  es un vector columna de  $r_1 r_2 \times 1$  que contiene los coeficientes de la transformada,  $\mathbf{x}$  es un vector columna de  $N_1 N_2 \times 1$  con los coeficientes de la escena de alta resolución y  $\Phi$  una matriz rectangular de  $r_1 r_2 \times N_1 N_2$  que relaciona las transformadas. Obsérvese que el sistema (1.6) está propuesto en el dominio de Fourier. Para obtener la imagen en el dominio espacial es necesario aplicar la transformada inversa a la solución. Algunos autores como Kim y col [11, 12] y Bose [13] agregaron términos de regularización usando mínimos cuadrados, y mínimos cuadrados ponderados para considerar distintos modelos de degradación. Para aumentar la eficiencia computacional Rhee y col [14] utilizó la DCT (*Discrete Cosine Transform*). El problema de esta técnica es la poca robustez, ya que, no consigue buenas reconstrucciones ante modelos de degradación muy fuertes y contaminación severa.

### 1.3.3 Superresolución usando interpolación

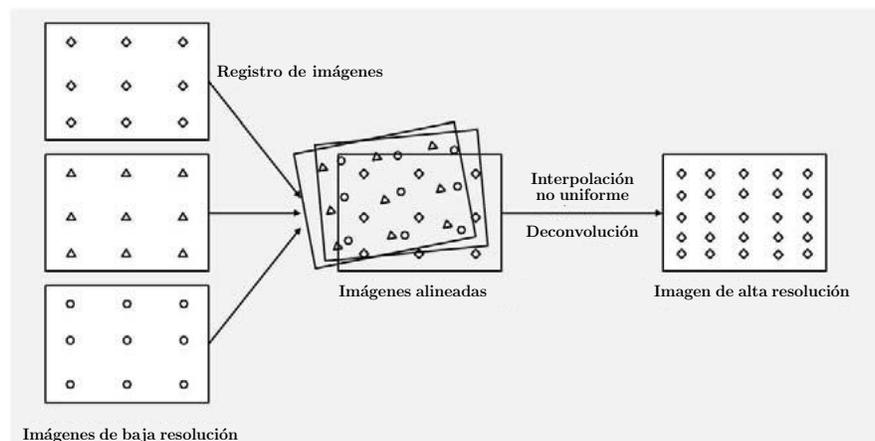
Estas técnicas son aplicadas en el dominio espacial de la imagen a diferencia del anterior que se estudiaba en el dominio de la frecuencia. Para entender como funciona la técnica, se parte, de la ecuación (1.1); se asume que  $H_k$  es un sistema lineal invariante en espacio (LSI *Linearly Spatial Invariant*) y que es el mismo para todas las  $k$  muestras, entonces se

hace la siguiente manipulación  $H\mathbf{X} = \mathbf{Z}$ , para obtener:

$$\mathbf{Y}_k = DF_k\mathbf{Z} + \mathbf{V}_k \quad (1.7)$$

Esta técnica se compone de tres etapas, que se ilustran en la Fig. 1.10 [8]. La primera, es usar un algoritmo de registro de imágenes como en [15] para alinear en una rejilla de alta resolución las imágenes de baja resolución, haciendo la mejor estimación del movimiento existente entre sub-píxeles. El siguiente paso es utilizar un proceso de interpolación no uniforme para corregir el movimiento de sub-píxeles y así obtener una versión borrosa de la imagen, que en (1.7) corresponde a  $\mathbf{Z}$ . El último paso es aplicar un algoritmo de deconvolución para eliminar el efecto borroso y así obtener la imagen de alta resolución. En la literatura se han propuesto varias alternativas, como interpolación triangular [16], interpolación polifásica usando mediana ponderada [17], entre otras.

Computacionalmente son más eficientes que las técnicas en el dominio de la frecuencia, lo que las hace utilizables en aplicaciones de tiempo real, por otro lado, son poco robustas y están limitadas al modelo de degradación, es decir, en la fase de deconvolución es importante tener un buen aproximado del modelo de degradación para obtener una buena reconstrucción.



**Figura 1.10. Técnica de superresolución usando interpolación, y posterior deconvolución.**

### 1.3.4 Superresolución usando técnicas estadísticas

Existen varios enfoques para tratar el problema de superresolución usando técnicas estadísticas, sin embargo, todos se basan en un mismo planteamiento, el cual es, tener una información previa relacionada con la solución, para hallarla utilizando términos de regularización. El uso de métodos de estimación bayesianos provee una manera flexible de modelar el problema cuando una función de densidad de probabilidad a posteriori (FDP) puede ser establecida. El MAP (máximo a posteriori) maximiza dicha función, en otras palabras:

$$x = \arg \max P(x|y_1, y_2, \dots, y_k) \quad (1.8)$$

Donde  $P(x|y_1, y_2, \dots, y_k)$  es la probabilidad de que ocurra  $x$  dado el conjunto de valores  $y_1, y_2, \dots, y_k$ . Usando el logaritmo y aplicando el teorema de *Baye's* a la probabilidad condicional se obtiene:

$$x = \arg \max \ln P(y_1, y_2, \dots, y_k|x) + \ln P(x) \quad (1.9)$$

En este caso el modelo a priori de la imagen  $P(x)$  y la condición de densidad  $P(y_k|x)$  serán definidas por la información relacionada con la imagen de alta resolución. Una buena manera de estimar  $P(x)$  es usando campos aleatorios de *Markov*, que permiten, utilizar una distribución probabilística de *Gibbs* [9]:

$$P(x) = \frac{1}{Z} e^{-\sum_{c \in S} \phi_c(x)} \quad (1.10)$$

Donde  $Z$  es un término de normalización,  $\phi_c(x)$  es una función de potencia no negativa de los píxeles dentro de la región de  $c$ . El problema de optimización usando (1.10) es muy complejo, por lo cual, se hacen ciertas suposiciones [9].

Asumiendo que el modelo a priori es gaussiano, la función de potencia se convierte en  $\phi_c(x) = (D^n x)^2$ , donde  $D^n$  es una derivada enésima, que convierte el problema de optimización en un problema lineal. Luego si el error entre tramas es independiente y el ruido está distribuido uniformemente, el problema de optimización se compacta a:

$$\mathbf{X} = \arg \min \|\mathbf{Y} - M\mathbf{X}\|^2 + \alpha\Phi(\mathbf{X}) \quad (1.11)$$

Donde  $\alpha$  es el parámetro de regularización. Los diferentes enfoques estadísticos para resolver (1.11) varían en la forma como se trata el modelo de degradación  $M$ , el modelo a priori  $\alpha\Phi(\mathbf{X})$ , y las técnicas de inferencia estadística [8,9].

### 1.3.5 Superresolución basada en ejemplos

En algunos casos no se dispone de varias imágenes de una misma escena, por lo cual no es posible aplicar las técnicas anteriores, ya que no existe desplazamiento entre sub-píxeles, y no se puede aplicar regularización para resolver el problema mal definido. Con el fin de solventar esta desventaja se propuso regularizar, usando imágenes de estadística similar, denominadas ejemplos.

La metodología basada en ejemplos consiste en usar pares de bloques, de imágenes de baja y de alta resolución ( $y_i, x_i$ , mostrados en la Fig. 1.11), y explotar la concurrencia entre ellos, usando el modelo de observación (1.1). El primer trabajo en este paradigma fue propuesto por *Freeman* y col [18], quienes usaron bloques de entrenamiento, sobre un campo aleatorio de *Markov*, y lo aplicaron a una imagen objetivo para predecir su correspondiente imagen de alta resolución. Otra alternativa es utilizar los  $k$ -vecinos más cercanos al bloque de baja, esta técnica, requiere grandes cantidades de bloques, razón por lo cual *Chang* y col [19] proponen utilizar un empotramiento de vecindarios, agrupando los bloques de alta y de baja en 2 espacios vectoriales. Para cada bloque de baja de la imagen objetivo  $y_i^t$ , se encuentran los  $k$ -vecinos más cercanos  $N_t$  en  $\{y_i\}_{i=1}^n$  y se calculan las ponderaciones, a través, del siguiente problema de optimización:

$$\hat{w}_s = \arg \min_{w_s} \|y_k^t - \sum_{y_s \in N_t} \omega_s y_s\|^2, \text{ tal que } \sum_{y_s \in N_t} w_s = 1 \quad (1.12)$$

El bloque de alta resolución se genera a partir de la media ponderada usando las ponderaciones obtenidas:

$$\hat{x}_k^t = \sum_{y_s \in N_t} \hat{w}_s x_s \quad (1.13)$$

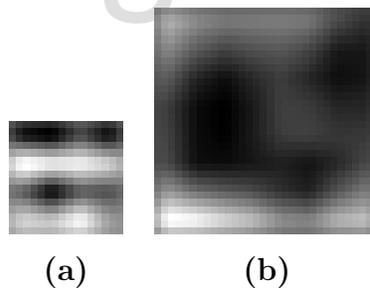
Yang y col [20] proponen otra alternativa derivada de la teoría de sensado comprimido, utilizando 2 diccionarios  $D_h$  y  $D_l$  para representar los bloques de alta y baja resolución. La representación de cada bloque sobre los diccionarios es poco densa, y se obtiene resolviendo el siguiente problema:

$$\hat{\omega} = \arg \min_{\omega} \|\omega\|_1 \text{ tal que } \|y_t - D_l \omega\|^2 < \sigma^2 \quad (1.14)$$

Generando los bloques de manera similar a (1.13), usando el diccionario de alta:

$$x_t = D_h w \quad (1.15)$$

Las técnicas usando ejemplos conducen a reconstrucciones con calidad superior a las técnicas anteriores si se usan imágenes adecuadas, pues brindan mayor robustez al ruido y al modelo de degradación. La desventaja de estos métodos es su alto costo computacional lo que las hace todavía inviables para aplicaciones de tiempo real. En las siguientes secciones de abordará con mayor detalle la propuesta de Yang, y sus variantes.



**Figura 1.11. Bloques de baja y alta resolución.**

# CAPÍTULO 2

## TEORÍA DE REPRESENTACIÓN POCO Densa

*La teoría de representación poco densa, ha cogido auge en los últimos años y ha conseguido relacionarse con el sensado comprimido, permitiendo su aplicación en el área de procesamiento digital de señales, específicamente en imágenes. Esta teoría se basa en el uso de pocas componentes de un diccionario para representar una señal objetivo. Muchos algoritmos se han desarrollado para recobrar las señales, así como, para entrenar diccionarios, haciendo que sea muy utilizable para resolver problemas comunes en aplicaciones como la restauración. Las técnicas de SR están estrechamente ligadas con la restauración como se vió en el capítulo anterior. En este capítulo se abordan los fundamentos básicos de la teoría de representación poco densa, y se describen los algoritmos que serán utilizados en la presente investigación. Además, se aborda el problema de SR desde la óptica de esta teoría, y se explicaron los enfoques más relevantes que han arrojado buenos resultados en la solución de dicho problema.*

### 2.1 REPRESENTACIÓN POCO Densa DE SEÑALES

Una señal puede ser representada, a través, de la combinación lineal de otras, generalmente denominadas señales elementales. Las señales digitales se obtienen muestreando una señal continua a una determinada frecuencia de muestreo, y se representan como un arreglo de datos de dimensión  $n$ . Esta señal digital puede ser compuesta con un máximo de  $n$  señales elementales linealmente independientes. En el procesamiento de señales se ha estudiado que existen bases para estas señales elementales, las cuales representan gran parte de la energía de la señal en su dominio con unas pocas componentes, si la señal no está contaminada, por ejemplo, las bases de la transformada Fourier, las bases de la transformada discreta del

coseno (DCT) y las wavelets entre otras. En el contexto de las transformaciones lineales, la señal en tiempo discreto  $c$  se transforma a  $s$  siguiendo:

$$\mathbf{s} = \Phi \mathbf{c} \quad (2.1)$$

En (2.1) se muestra un ejemplo de una transformación lineal, para cambiar el dominio de una señal  $\mathbf{c} \in \mathbb{R}^n$ ,  $\Phi$  representa las señales elementales ordenadas en una matriz de  $m \times n$  y  $\mathbf{s}$  la señal en el nuevo dominio  $\mathbb{R}^m$ , este modelo se referencia en la literatura como la síntesis y su recíproco es el modelo de análisis (2.2).

$$\mathbf{c} = \Phi^+ \mathbf{s} \quad (2.2)$$

Donde  $\{\cdot\}^+$  denota la pseudoinversa de una matriz. Partiendo de este hecho, se puede afirmar que una señal se puede representar, con unas pocas componentes, en un dominio adecuado, es ahí donde la teoría de representación poco densa (*sparse*) puede ser aplicada. Ésta consiste en encontrar las pocas componentes, en un conjunto de bases denominado diccionario, que mejor representen una señal bajo cierta condición. Formalmente:

$$\min_{\gamma} \|\gamma\|_0 \text{ tal que } \mathbf{D}\gamma = \mathbf{y} \quad (2.3)$$

La ecuación (2.3) es ideal, ya que, las señales naturales no son poco densas exactamente [21], razón por la cual, el problema se plantea como una aproximación:

$$\min_{\gamma} \|\gamma\|_0 \text{ tal que } \|\mathbf{D}\gamma - \mathbf{y}\|_2 \leq \epsilon \quad (2.4)$$

**Definición 2.1. Seudonorma** Dado un espacio vectorial  $V$ , una norma es un función  $p: V \rightarrow \mathbb{R}$ , que cumple:

- Homogenidad absoluta:  $p(a\mathbf{v}) = |a|p(\mathbf{v})$
- Desigualdad triangular:  $p(\mathbf{u} + \mathbf{v}) \leq p(\mathbf{u}) + p(\mathbf{v})$

---

<sup>1</sup>Usualmente en el procesamiento digital de señales, las transformada son de dimensión  $n \times n$ , a fin de conservar la longitud del vector y pasar de un dominio a otro sin mucha complejidad

- *Puntos separados:*  $p(\mathbf{v}) = 0 \iff \mathbf{v} = \mathbf{0}$

Aquellas funciones  $p : V \rightarrow R$  que no cumplen los 3 axiomas anteriores son conocidas como *seudonormas*

El problema (2.4) es conocido como, restringido por error (*error constrained*), donde  $\epsilon$  es el error máximo permitido entre la señal y su aproximación. Este problema requiere minimizar la seudonorma  $l_0$ , que cuenta la cantidad de componentes no nulas (distintas de cero) en un vector. Esta minimización es un problema no convexo combinatorial NP completo, y requiere tiempos exagerados de computo para su solución. La mayoría de los algoritmos, para resolverlo proponen relajar la seudonorma  $l_0$ , a la norma  $l_1$ , de igual manera existe otra forma de aproximar (2.3), usando como restricción el nivel poca densidad, (*sparsity constrained*):

$$\min_{\gamma} \|\mathbf{D}\gamma - \mathbf{y}\|_2 \text{ tal que } \|\gamma\|_0 \leq \tau \quad (2.5)$$

Donde  $\tau$  es el umbral de componentes máximas para representar la señal.

### 2.1.1 Dicionarios para representación poco densa

El término diccionario, que ya fue introducido en la sección anterior, cumple un papel muy importante en la representación poco densa, por definición, es una colección de señales elementales de norma  $l_2$  unitaria, las cuales se conocen como los átomos del diccionario. En términos prácticos el diccionario es representado por una matriz  $m \times n$ , donde cada átomo es un vector columna, generalmente estos son identificados con un subíndice. Sea  $\Phi$  el diccionario, sus correspondientes átomos serían:  $\Phi = [\varphi_1, \varphi_2, \varphi_3, \dots, \varphi_n]$ .

La composición del diccionario, en términos de la geometría entre sus átomos, tiene implicaciones importantes para la representación de la señal. Si el diccionario posee  $n$  átomos linealmente independientes expande todo el espacio vectorial y cualquier señal puede ser representada perfectamente, se dice, que es un diccionario completo, si se cumple que,  $m \leq n$ . Si el diccionario posee átomos dependientes entre sí, una señal tiene infinitas representaciones, entonces se dice que es redundante, si se cumple que,  $m < n$ . En aplicaciones prácticas se escogen diccionarios completos y redundantes (sobrecompletos).

### 2.1.1.1 Diccionario DCT

La transformada discreta del coseno, es una transformada sinusoidal, unitaria, invertible, esto significa que su kernel<sup>2</sup>, está compuesto, por una serie de bases sinusoidales, ortogonales de norma  $l_2$  unitaria. Su desarrollo está basado en la DFT, con la particularidad, que se asume que la señal es antisimétrica en su extensión, lo que permite una mejor aproximación en los bordes, con respecto a la DFT. Existen 4 tipos de transformada DCT, para los efectos de esta investigación solo se hará referencia a la DCT-II y la DCT-III, como la DCT y su inversa respectivamente, se recomienda ver el capítulo 2 de la referencia [22] para mayor información.

Un diccionario DCT unidimensional de  $N$  átomos puede ser definido como:

$$T_{i,j} = \begin{cases} \frac{1}{\sqrt{N}} & \text{si } i = 0, 0 \leq j \leq N - 1 \\ \sqrt{\frac{2}{N}} \cos\left(\frac{\pi(2j+1)i}{2N}\right) & \text{si } 0 < i \leq N - 1, 0 \leq j \leq N - 1 \end{cases} \quad (2.6)$$

El modelo de síntesis de una señal  $x \in \mathbb{R}^N$ , usando el diccionario DCT es:

$$\mathbf{x} = \mathbf{T}\mathbf{y} \quad (2.7)$$

Siendo  $\mathbf{y}$  la representación de  $\mathbf{x}$  en el dominio de la DCT. Dado que  $T$  es ortonormal, se cumple que  $\mathbf{T}^+ = \mathbf{T}^\dagger$ , donde  $\{\cdot\}^\dagger$  denota la matriz transpuesta. De esta manera la DCT de la señal  $\mathbf{x}$  puede obtenerse haciendo:

$$\mathbf{y} = \mathbf{T}^\dagger \mathbf{x} \quad (2.8)$$

Concluyendo que el diccionario DCT completo, no es más, que la transpuesta de su matriz de transformación. En el caso de las imágenes que son señales bidimensionales que varían espacialmente, la deducción del diccionario DCT no es tan directa. Sea  $\mathbf{X} \in \mathbb{R}^{N \times N}$  una imagen, su respectiva transformada bidimensional es:

$$\mathbf{Y} = \mathbf{T}^\dagger \mathbf{X} \mathbf{T} \quad (2.9)$$

y su transformada inversa:

---

<sup>2</sup>Un kernel es una matriz generatriz.

$$\mathbf{X} = \mathbf{T}\mathbf{Y}\mathbf{T}^\dagger \quad (2.10)$$

Para encontrar el diccionario hay que considerar las variaciones horizontales y verticales de la imagen, una manera es hacer un reordenamiento lexicográfico de la matriz en un vector, esta operación se denota con el operador  $\text{vec}()$ . Aplicándolo a la imagen en cuestión se obtiene:

$$\text{vec}(\mathbf{X}) = \text{vec}(\mathbf{T}\mathbf{Y}\mathbf{T}^\dagger) \quad (2.11)$$

El término de la derecha en (2.11) puede reescribirse usando las propiedades del producto *Kronecker* como:

$$\text{vec}(\mathbf{T}\mathbf{Y}\mathbf{T}^\dagger) = (\mathbf{T} \otimes \mathbf{T}) \text{vec}(\mathbf{Y}) \quad (2.12)$$

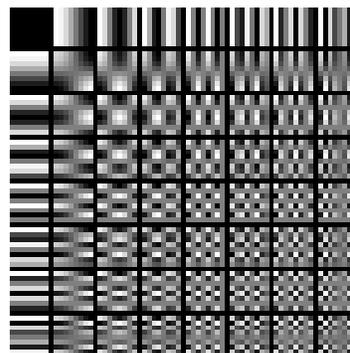
Sustituyendo (2.12) en (2.11), se obtiene una equivalencia similar al modelo de síntesis:

$$\text{vec}(\mathbf{X}) = (\mathbf{T} \otimes \mathbf{T}) \text{vec}(\mathbf{Y}) \quad (2.13)$$

Con esto se demuestra un diccionario DCT bidimensional de  $N \times N$  puede ser generado a partir del producto *Kronecker* de 2 diccionarios unidimensionales DCT de  $\sqrt{N} \times \sqrt{N}$ .

$$\mathbf{T}_{2D} = \mathbf{T} \otimes \mathbf{T} \quad (2.14)$$

En la Fig. 2.1 se muestra las bases bidimensionales del diccionario DCT completo

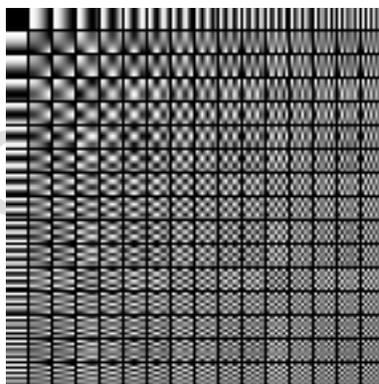


**Figura 2.1.** Diccionario 2D-DCT completo de 64 átomos de  $8 \times 8$  píxeles.

Como se dijo al principio de la sección, el uso de diccionarios redundantes es más efectivo para la aproximación, existe una versión redundante del diccionario DCT, denominado DCT sobrecompleto. Este diccionario es de dimensiones  $M \times N$  con  $N > M$ , y sus átomos ya no son ortogonales. El diccionario 1D-DCT sobrecompleto se puede obtener a partir de:

$$T_{ij} = \begin{cases} \frac{1}{\sqrt{M}} & \text{si } i = 0, 0 \leq j \leq M - 1 \\ \cos\left(\frac{\pi ij}{N}\right) & \text{si } 0 < i \leq M - 1, 0 \leq j \leq N - 1 \end{cases} \quad (2.15)$$

De manera análoga al caso anterior, un diccionario 2D-DCT sobrecompleto de dimensiones  $M \times N$ , se genera como el producto *Kronecker* de dos diccionarios 1D-DCT de  $\sqrt{M} \times \sqrt{N}$ . En la Fig. 2.2 se muestran las bases bidimensionales de un diccionario sobrecompleto.



**Figura 2.2.** Diccionario DCT sobrecompleto, 256 átomos de  $8 \times 8$  píxeles.

### 2.1.1.2 *Diccionarios Entrenados*

El uso de bases de transformadas como diccionarios, conduce a buenos resultados para aproximaciones poco densas de señales, si éstas, se seleccionan apropiadamente, es decir que modelen de la mejor manera la naturaleza de la señal. Encontrar cual es el mejor diccionario  $\Phi$  para representar una señal es una ardua tarea que puede tener infinitas soluciones, por eso, en la literatura se han propuesto varias técnicas para entrenar diccionarios que se adapten apropiadamente a la naturaleza del problema en estudio. El entrenamiento, se refiere a adaptar los átomos de un diccionario base a la estadística de un grupo de señales para mejorar la representación poco densa de éstas. La mayoría de los algoritmos de

entrenamiento busca resolver el siguiente problema de optimización:

$$\arg \min_{\mathbf{D}} \|\mathbf{D}\mathbf{\Gamma} - \mathbf{Y}\|_2 \text{ tal que } \forall i \|\gamma_i\|_0 \leq \tau \quad (2.16)$$

La mayoría de los métodos se dividen en 2 etapas, la primera encontrar la representación poco densa, y la segunda, actualizar el diccionario para disminuir el error. Entre los métodos de entrenamiento más populares destacan, MOD (*Method of Optimal Directions*) el cual resuelve (2.16) solucionando el problema cuadrático directamente  $\mathbf{D} = \mathbf{Y}\mathbf{\Gamma}^+$ . Otro algoritmo bastante popular es K-SVD (*K-times Singular Value Decomposition*), el cual se estudia con mayor detalle mas adelante, a diferencia de MOD, actualiza átomo por átomo además de actualizar el vector de coeficientes poco denso, lo cual, acelera su convergencia.

### 2.1.2 Representación poco densa de imágenes

Como es bien sabido las imágenes son señales bidimensionales en  $\mathbb{R}^{M \times N}$ . Una imagen debe ser reordenada lexicográficamente para poder aplicar las técnicas de representación poco densa, lo que requiere espacios de memoria y un costo computacional elevado. Por ejemplo, una imagen de de  $256 \times 256$  píxeles se convertiría en un vector de 65536 componentes, se necesitaría un diccionario  $\Phi \in \mathbb{R}^{65536 \times 65536}$ , para representar la imagen correctamente. Para que un computador pueda procesar esa información necesita cantidades exorbitantes de memoria, sumando los tiempos de acceso a esta, hace que el procesamiento de una imagen aplicando técnicas de representación poco densa sea inviable. Para solventar esta limitante se propone dividir la imagen en pequeños bloques y procesar cada bloque por separado, alineados en una matriz considerando solapamiento entre ellos.

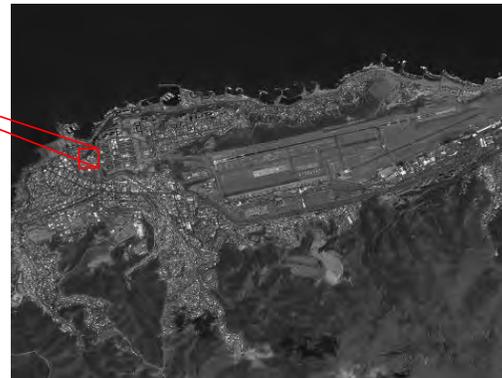
En la Fig. 2.3(a) no se considera solapamiento, mientras que en la Fig. 2.3(b) si se considera; concluyendo que el solapamiento de bloques genera grupos de señales más grande, lo cual es una ventaja, ya que, le agrega redundancia a cada píxel. Como desventaja incrementa los tiempos de procesamiento. Supóngase que se tiene una imagen  $\mathbf{Y} \in \mathbb{R}^{N \times N}$  y el tamaño de bloque es  $n \times n$ , considerando  $N$  múltiplo de  $n$ , la imagen se puede dividir en  $(N/n)^2$  bloques sin considerar solapamiento, y en  $(N - n + 1)^2$  bloques considerando un solapamiento completo. El uso de técnicas de representación poco densa sobre imágenes a llevado al estado del arte problemas comunes como:

26	53	45	34
83	79	69	75
3	74	44	11
34	85	16	45

$$\begin{bmatrix} 26 & 3 & 45 & 44 \\ 83 & 34 & 69 & 16 \\ 53 & 74 & 34 & 11 \\ 79 & 85 & 75 & 45 \end{bmatrix}$$


(a)

26	53	45
83	79	69
3	74	44

$$\begin{bmatrix} 26 & 83 & 53 & 79 \\ 83 & 3 & 79 & 74 \\ 53 & 79 & 45 & 69 \\ 79 & 74 & 69 & 44 \end{bmatrix}$$


(b)

**Figura 2.3.** Reordenamiento de una imagen en bloques de  $2 \times 2$  píxeles.

- Restauración
- Clasificación
- Detección
- Desoclusión
- Superresolución, en lo que se centra este trabajo

De ahora en adelante las letras mayúsculas y en negrita también denotaran imágenes reordenadas en bloques, cuando son usadas en problemas de optimización.

## 2.2 ALGORITMOS DE REPRESENTACIÓN POCO DENSA

Varios algoritmos se han propuesto para encontrar la representación poco densa de una señal, dos enfoques han sido dominantes, uno es la utilización de métodos de búsqueda voraz y el otro, métodos de relajación convexa. Los métodos de búsqueda voraz consisten en encontrar el átomo más correlacionado con la señal, luego calcular su contribución, y restársela para encontrar el siguiente átomo, de esta manera se va construyendo paso por paso. Entre los más importantes destacan MP (*Matching Pursuit*, el cual se detalla en la referencia [5]) y OMP (*Orthogonal Matching Pursuit*).

Los métodos de relajación convexa reemplazan el problema combinatorial de aproximación poco densa, con un programa convexo relacionado, existen propuestas como en [23], y enfoques estadísticos como en [24].

### 2.2.1 Algoritmo OMP para representación poco densa

El algoritmo OMP es un algoritmo de búsqueda voraz, derivado de MP con una restricción de mínimos cuadrados que acelera su convergencia. Está diseñado para resolver, tanto el problema restringido por poca densidad (2.4), como el restringido por error (2.5). El proceder de este algoritmo consiste en encontrar el átomo que este más correlacionado con la señal, calcular su contribución resolviendo el problema de mínimos cuadrados, y cancelar esta contribución a la señal para empezar nuevamente hasta que el criterio de parada sea cumplido.

El uso de la restricción de mínimos cuadrados ortogonaliza el residuo, con respecto a la aproximación obtenida, evitando de esta manera que se vuelva a elegir el mismo átomo. OMP es uno de los algoritmos más usados para encontrar la representación poco densa de la señal, su desventaja reside en la poca robustez ante contaminaciones de tipo impulsivas, generando artefactos indeseados en la reconstrucción. En la Tabla 2.1 [21] se describe paso a paso este algoritmo.

#### 2.2.1.1 Optimizando OMP con OMP de conjunto

A pesar que OMP es un algoritmo bastante rápido, hace operaciones costosas como la pseudoinversa, esto ante una enorme cantidad de señales no es conveniente, por esta razón

Tabla 2.1. Algoritmo OMP.

ENTRADA	<ul style="list-style-type: none"> <li>• Diccionario <math>\mathbf{D} \in \mathbb{R}^{n \times m}</math></li> <li>• Señal objetivo <math>\mathbf{y} \in \mathbb{R}^n</math></li> <li>• Nivel de esparcidad <math>\tau</math> o error permitido <math>\epsilon</math></li> </ul>
INICIALIZACIÓN	<ul style="list-style-type: none"> <li>• Conjunto de índices <math>\Lambda^{(0)} = \emptyset</math></li> <li>• Residuo <math>\mathbf{r} = \mathbf{y}</math></li> <li>• Vector de coeficientes poco densos <math>\boldsymbol{\gamma} = 0</math></li> <li>• Iteración <math>t = 1</math></li> </ul>
PROCEDIMIENTO	<ul style="list-style-type: none"> <li>• Encontrar el átomo mas correlacionado con la señal, resolviendo           <math display="block">k = \arg \max  \mathbf{D}^\dagger \mathbf{r} </math> </li> <li>• Incluir el nuevo índice en el conjunto de índices ya encontrado           <math display="block">\Lambda^t = \Lambda^{t-1} \cup k</math> </li> <li>• Actualizar el vector de coeficientes resolviendo el problema de mínimos cuadrados           <math display="block">\boldsymbol{\gamma} = \arg \min_{\boldsymbol{\gamma}} \ \mathbf{D}_{\Lambda^t} \boldsymbol{\gamma} - \mathbf{s}\ _2 = \mathbf{D}_{\Lambda^t}^+ \mathbf{y}</math> </li> <li>• Actualizar el residuo <math>\mathbf{r} = \mathbf{y} - \mathbf{D}_{\Lambda} \boldsymbol{\gamma}</math></li> <li>• Incrementar la iteración <math>t = t + 1</math> y repetir hasta que se cumpla el criterio de parada</li> </ul>
SALIDA	<ul style="list-style-type: none"> <li>• El vector poco denso <math>\boldsymbol{\gamma} \in \mathbb{R}^m</math></li> <li>• El conjunto de índices <math>\Lambda</math></li> </ul>

Ron Rubenstein y col en [25], proponen una serie de equivalencias que disminuyen considerablemente el costo computacional del algoritmo ante grandes grupos de señales, y lo nombran OMP de conjunto (*BatchOMP*).

**Propiedad 2.1.** Dada la factorización Cholesky de  $\tilde{\mathbf{A}} = \tilde{\mathbf{L}}\tilde{\mathbf{L}}^\dagger \in \mathbb{R}^{(N-1) \times (N-1)}$ , la factorización Cholesky de

$$\mathbf{A} = \begin{pmatrix} \tilde{\mathbf{A}} & \mathbf{v} \\ \mathbf{v}^\dagger & c \end{pmatrix} \in \mathbb{R}^{N \times N}$$

es dada por  $\mathbf{A} = \mathbf{L}\mathbf{L}^\dagger \in \mathbb{R}^{N \times N}$ , con

$$\mathbf{L} = \begin{pmatrix} \tilde{\mathbf{L}} & \underline{0} \\ \mathbf{w}^\dagger & \sqrt{c - \mathbf{w}^\dagger \mathbf{w}} \end{pmatrix} \in \mathbb{R}^{N \times N}, \quad \mathbf{w} = \tilde{\mathbf{L}}^\dagger \mathbf{v}$$

En la Tabla 2.1 [25], el problema de mínimos cuadrados se resuelve aplicando la pseudoinversa de los átomos previamente seleccionados, que por definición, es  $\mathbf{D}_\Lambda^+ = (\mathbf{D}_\Lambda^\dagger \mathbf{D}_\Lambda)^{-1} \mathbf{D}_\Lambda^\dagger$ , debido al proceso de ortogonalización, la matriz  $\mathbf{D}_\Lambda^\dagger \mathbf{D}_\Lambda$  es definida simétrica positiva, lo que permite su factorización *Cholesky*, además ésta se va componiendo añadiendo columnas en cada iteración. Según la propiedad 2.1, conocida la factorización *Cholesky* de una submatriz de dimensiones  $(N - 1) \times (N - 1)$  solo hace falta calcular la última fila de la matriz triangular inferior ( $\mathbf{L}$ ) para encontrar la nueva factorización.

Si  $\mathbf{G}_\Lambda = \mathbf{D}_\Lambda^\dagger \mathbf{D}_\Lambda$  entonces  $\mathbf{D}_\Lambda^+ = (\mathbf{G}_\Lambda)^{-1} \mathbf{D}_\Lambda^\dagger$ , conduciendo a una solución equivalente para el problemas de mínimos cuadrados:

$$\mathbf{G}_\Lambda \boldsymbol{\gamma} = \mathbf{D}_\Lambda^\dagger \mathbf{s} \quad (2.17)$$

Aplicando la factorización de *Cholesky* a (2.17):

$$\mathbf{L} \mathbf{L}^\dagger \boldsymbol{\gamma} = \mathbf{D}_\Lambda^\dagger \mathbf{y} \quad (2.18)$$

Resolver (2.18) para encontrar  $\boldsymbol{\gamma}$ , es bastante simple, ya que,  $\mathbf{L}$  es una matriz triangular inferior y las técnicas de sustitución hacia adelante y hacia atrás son aplicables. A medida que el algoritmo itera, el término  $\mathbf{D}_\Lambda^\dagger \mathbf{y}$  se convierte en  $\mathbf{D}_\Lambda^\dagger \mathbf{r}$ , para la selección del nuevo átomo no hace falta conocer  $\mathbf{r}$  o  $\boldsymbol{\gamma}$ , pero sí  $\mathbf{D}_\Lambda^\dagger \mathbf{r}$ , entonces se puede sustituir el calculo de  $\mathbf{r}$  con una multiplicación de bajo costo. Denotando  $\boldsymbol{\alpha}^t = \mathbf{D}_\Lambda^\dagger \mathbf{r}$ , para la primera iteración se cumple que  $\boldsymbol{\alpha}^0 = \mathbf{D}_\Lambda^\dagger \mathbf{y}$ , el término  $\boldsymbol{\alpha}^1$  será

$$\begin{aligned} \boldsymbol{\alpha}^1 &= \mathbf{D}_\Lambda^\dagger (\mathbf{y} - \mathbf{D}_\Lambda (\mathbf{D}_\Lambda^\dagger) \mathbf{y}) \\ \boldsymbol{\alpha}^1 &= \boldsymbol{\alpha}^0 - \mathbf{G}_\Lambda (\mathbf{D}_\Lambda^\dagger) \mathbf{y} \\ \boldsymbol{\alpha}^1 &= \boldsymbol{\alpha}^0 - \mathbf{G}_\Lambda (\mathbf{G}_{\Lambda, \Lambda})^{-1} \boldsymbol{\alpha}^0 \end{aligned} \quad (2.19)$$

Con esto se demuestra, que conocido el valor de  $\boldsymbol{\alpha}^0$  y  $\mathbf{G}$  calcular  $\boldsymbol{\alpha}^t$  se reduce a multiplicar por la matriz  $\mathbf{G}_\Lambda$ , en vez de todo el diccionario  $\mathbf{D}$ , ya que, la inversión de  $\mathbf{G}_{\Lambda, \Lambda}$  se hace usando la factorización de *Cholesky*. Los pasos hasta ahora mencionados

resuelven el problema restringido por poca densidad, para resolver la restricción por error es necesario hacer otra equivalencia. El error se calcula como  $\|\mathbf{y} - \mathbf{D}\boldsymbol{\gamma}\|_2^2$ , la cual es la expresión del residuo en la Tabla 2.1, definiendo así la expresión del error como:

$$\epsilon^t = \|\mathbf{r}^t\|_2^2 = \|\mathbf{r}^{t-1} + \mathbf{D}(\boldsymbol{\gamma}^{t-1} - \boldsymbol{\gamma}^t)\|_2^2 \quad (2.20)$$

Una forma equivalente de expresar la norma  $l_2$  al cuadrado es  $\|\mathbf{r}^t\|_2^2 = (\mathbf{r}^t)^\dagger \mathbf{r}^t$ , aplicando esta equivalencia a la expresión (2.20) y reordenándola se tiene:

$$\begin{aligned} (\mathbf{r}^t)^\dagger \mathbf{r}^t &= (\mathbf{r}^t)^\dagger (\mathbf{r}^{t-1} + \mathbf{D}(\boldsymbol{\gamma}^{t-1} - \boldsymbol{\gamma}^t)) \\ &= (\mathbf{r}^t)^\dagger \mathbf{r}^{t-1} + (\mathbf{r}^t)^\dagger \mathbf{D}\boldsymbol{\gamma}^{t-1} - \cancel{(\mathbf{r}^t)^\dagger \mathbf{D}\boldsymbol{\gamma}^t}^0 \\ &= (\mathbf{r}^{t-1} + \mathbf{D}(\boldsymbol{\gamma}^{t-1} - \boldsymbol{\gamma}^t))^\dagger \mathbf{r}^{t-1} + (\mathbf{r}^t)^\dagger \mathbf{D}\boldsymbol{\gamma}^{t-1} \\ &= \|\mathbf{r}^{t-1}\|_2^2 + \cancel{(\mathbf{r}^{t-1})^\dagger \mathbf{D}\boldsymbol{\gamma}^{t-1}}^0 + (\mathbf{r}^t)^\dagger \mathbf{D}\boldsymbol{\gamma}^{t-1} - (\mathbf{r}^{t-1})^\dagger \mathbf{D}\boldsymbol{\gamma}^t \\ &= \|\mathbf{r}^{t-1}\|_2^2 + (\mathbf{s} - \mathbf{D}\boldsymbol{\gamma}^t)^\dagger \mathbf{D}\boldsymbol{\gamma}^{t-1} - (\mathbf{s} - \mathbf{D}\boldsymbol{\gamma}^{t-1})^\dagger \mathbf{D}\boldsymbol{\gamma}^t \\ &= \|\mathbf{r}^{t-1}\|_2^2 + \mathbf{s}^\dagger \mathbf{D}\boldsymbol{\gamma}^{t-1} - \cancel{(\mathbf{D}\boldsymbol{\gamma}^t)^\dagger \mathbf{D}\boldsymbol{\gamma}^{t-1}} - \mathbf{s}^\dagger \mathbf{D}\boldsymbol{\gamma}^t + \cancel{(\mathbf{D}\boldsymbol{\gamma}^{t-1})^\dagger \mathbf{D}\boldsymbol{\gamma}^t} \\ &= \|\mathbf{r}^{t-1}\|_2^2 + (\mathbf{r}^{t-1} + \mathbf{D}\boldsymbol{\gamma}^{t-1})^\dagger \mathbf{D}\boldsymbol{\gamma}^{t-1} - (\mathbf{r}^t + \mathbf{D}\boldsymbol{\gamma}^t)^\dagger \mathbf{D}\boldsymbol{\gamma}^t \\ &= \|\mathbf{r}^{t-1}\|_2^2 + (\boldsymbol{\gamma}^{t-1})^\dagger \mathbf{D}^\dagger \mathbf{D}\boldsymbol{\gamma}^{t-1} - (\boldsymbol{\gamma}^t)^\dagger \mathbf{D}^\dagger \mathbf{D}\boldsymbol{\gamma}^t \\ &= \|\mathbf{r}^{t-1}\|_2^2 + (\boldsymbol{\gamma}^{t-1})^\dagger \mathbf{G}\boldsymbol{\gamma}^{t-1} - (\boldsymbol{\gamma}^t)^\dagger \mathbf{G}\boldsymbol{\gamma}^t \end{aligned} \quad (2.21)$$

Por simplicidad se hace la siguiente sustitución  $\delta^t = (\boldsymbol{\gamma}^t)^\dagger \mathbf{G}\boldsymbol{\gamma}^t$  y aplicándola junto con (2.20) a (2.21) se consigue:

$$\epsilon^t = \epsilon^{t-1} + \delta^{t-1} + \delta^t \quad (2.22)$$

De esta manera se puede concluir que el nuevo error se calcula en función del anterior y del término  $\delta$ , el cual, exige un costo computacional bastante bajo, ya que,  $\mathbf{G}\boldsymbol{\gamma}^t$  es calculado previamente al actualizar  $\boldsymbol{\alpha}$  en (2.19).

Tabla 2.2. Algoritmo OMP *Batch*.

ENTRADA	<ul style="list-style-type: none"> <li>• <math>\boldsymbol{\alpha}^0 = \mathbf{D}\mathbf{y}</math>.</li> <li>• Error inicial <math>\epsilon^0 = \mathbf{y}^\dagger \mathbf{y}</math>.</li> <li>• <math>\mathbf{G} = \mathbf{D}^\dagger \mathbf{D}</math>.</li> <li>• Nivel de esparcidad <math>\tau</math> o error permitido <math>\epsilon</math>.</li> </ul>
INICIALIZACIÓN	<ul style="list-style-type: none"> <li>• Conjunto de índices <math>\Lambda^{(0)} = \emptyset</math>.</li> <li>• Matriz triangular para la factorización <math>\mathbf{L} = \mathbf{1}</math>.</li> <li>• Vector de coeficientes poco densos <math>\boldsymbol{\gamma} = \underline{\mathbf{0}}</math>.</li> <li>• <math>\boldsymbol{\alpha} = \boldsymbol{\alpha}^0</math>.</li> <li>• <math>\delta = 0</math>.</li> <li>• Iteración <math>t = 1</math>.</li> </ul>
PROCEDIMIENTO	<ul style="list-style-type: none"> <li>• Encontrar el átomo mas correlacionado con la señal, resolviendo           <math display="block">k = \arg \max  \boldsymbol{\alpha} </math> </li> <li>• Si <math>t &gt; 1</math> encontrar <math>\mathbf{w}</math> resolviendo <math>\mathbf{L}\mathbf{w} = \mathbf{G}_{\Lambda,k}</math> y actualizar <math>L</math> <math display="block">\mathbf{L} = \begin{bmatrix} \mathbf{L} &amp; \underline{\mathbf{0}} \\ \mathbf{w}^\dagger &amp; \sqrt{1 - \ \mathbf{w}\ _2^2} \end{bmatrix}</math> </li> <li>• Incluir el nuevo índice en el conjunto de índices ya encontrado           <math display="block">\Lambda^t = \Lambda^{t-1} \cup k</math> </li> <li>• Actualizar el vector de coeficientes resolviendo la simplificación del problema de mínimos cuadrados           <math display="block">\mathbf{L}\mathbf{L}^\dagger \boldsymbol{\gamma} = \boldsymbol{\alpha}_\Lambda^0</math> </li> <li>• Actualizar <math>\boldsymbol{\alpha} = \boldsymbol{\alpha}^0 - \mathbf{G}\boldsymbol{\gamma}</math></li> <li>• Si la restricción es por error actualizar <math>\delta^t = \boldsymbol{\gamma}^\dagger \mathbf{G}\boldsymbol{\gamma}</math> y calcular el error           <math display="block">\epsilon^t = \epsilon^{t-1} - \delta^t + \delta^{t-1}</math> </li> <li>• Incrementar la iteración <math>t = t + 1</math> y repetir hasta que se cumpla el criterio de parada.</li> </ul>
SALIDA	<ul style="list-style-type: none"> <li>• El vector poco denso <math>\boldsymbol{\gamma} \in \mathbb{R}^n</math>.</li> <li>• El conjunto de índices <math>\Lambda</math>.</li> </ul>

## 2.2.2 Algoritmo K-SVD para entrenamiento de diccionarios redundantes

El uso de diccionarios entrenados permite mejorar la aproximación usando representación poco densa, si este captura los detalles de la imagen, por lo tanto un buen algoritmo de entrenamiento es aquel que logra ese cometido. M. Aharon y col en [26] proponen un algoritmo de entrenamiento, usando descomposición en valores singulares (SVD *Singular Value Decomposition*) con una funcionalidad similar a *K-means* para cuantización vectorial.

### 2.2.2.1 *K-means* para cuantización vectorial

La cuantización vectorial (CV) es una técnica de procesamiento de señales, que permite modelar funciones de densidad de probabilidad usando vectores prototipo. Puede ser vista como un caso particular de la representación poco densa, en otras palabras, el objetivo de la cuantización vectorial es representar una señal con una única componente dentro de un libro de códigos. Dado un conjunto de vectores  $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n]$ , donde  $\mathbf{y}_i \in \mathbb{R}^n$ .  $\mathbf{C}$  el libro de códigos compuesto por  $[\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k]$  con  $n \gg k$ , la cuantización vectorial consiste en encontrar un conjunto de vectores  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$  con  $\mathbf{x}_i = \mathbf{e}_j$ , siendo  $\mathbf{e}_j$  un vector de ceros con 1 en la posición  $j$ , donde,  $\mathbf{C}\mathbf{x}_i$  sea el vector más cercano, en términos de la distancia euclidiana, que puede ser medida usando la norma  $l_2$ ,  $\|\mathbf{C}\mathbf{x}_i - \mathbf{y}_i\|_2$ .

El entrenamiento del libro de códigos puede ser llevado a cabo resolviendo el siguiente problema de minimización

$$\arg \min_{\mathbf{C}, \mathbf{X}} \|\mathbf{Y} - \mathbf{C}\mathbf{X}\|_F^2, \text{ tal que } \forall i, \mathbf{x}_i = \mathbf{e}_j \quad (2.23)$$

La norma Forbenius  $\|\cdot\|_F$  de una matriz, puede ser vista como la extensión de la norma  $l_2$  para vectores. Sea la matriz  $\mathbf{A} \in \mathbb{R}^{M \times N}$ , la norma Forbenius se denota como:

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^M \sum_{j=1}^N |A_{ij}|^2} \quad (2.24)$$

El algoritmo *K-means*, se divide en 2 etapas, la primera es la de codificación, que consiste, en encontrar el conjunto de vectores  $\mathbf{X}$  que resuelva (2.23), y la segunda, con  $\mathbf{X}$  conocido, actualizar el libro de códigos  $\mathbf{C}^{t-1}$  en función de disminuir el error medio

cuadrático (MSE *Mean Squared Error*), que es equivalente a resolver (2.23).

### 2.2.2.2 *K-SVD como generalización de K-means*

La cuantización vectorial puede ser vista como un caso particular de la representación poco densa, en el cual, el vector de coeficientes puede tener mas de una entrada distinta de cero, el problema de optimización (2.23) se expresaría como:

$$\arg \min_{Y,D} \|\mathbf{Y} - \mathbf{D}\mathbf{\Gamma}\|_F^2 \text{ tal que } \forall \|\gamma_i\|_0 \leq \tau \quad (2.25)$$

o también

$$\arg \min_{Y,D} \sum_i \|\gamma_i\|_0 \text{ tal que } \|\mathbf{Y} - \mathbf{D}\mathbf{\Gamma}\|_F^2 \leq \epsilon \quad (2.26)$$

Al igual que el algoritmo de *K-means*, *K-SVD* también posee 2 etapas, la primera que es encontrar la representación poco densa  $\mathbf{\Gamma}$  del conjunto de señales  $\mathbf{Y}$ , usando cualquier algoritmo de codificación, los métodos de búsqueda voraz son los mas empleados por su velocidad, luego mediante un proceso iterativo se va actualizando cada átomo del diccionario  $\mathbf{d}_k$ . En este proceso hay una cierta diferencia con respecto a *K-means*, ya que, también se actualiza el vector de coeficientes  $\mathbf{\Gamma}$  para obtener la mejor combinación que disminuya el MSE usando como herramienta SVD<sup>3</sup>. Los pasos que sigue este algoritmo se detallan en la Tabla 2.3 [3], se recomienda revisar el artículo original [26] donde los autores describen con mayor profundidad el algoritmo.

Su implementación en MATLAB esta disponible en los sitios web de *M. Elad* y *R. Rubenstein* <http://www.cs.technion.ac.il/~elad/software/>, <http://www.cs.technion.ac.il/~ronrubin/software.html>, la implementación en Python como resultado de esta investigación forma parte de la herramienta disponible en el siguiente repositorio <https://bitbucket.org/osjerick/sparsesat>.

<sup>3</sup>La matriz de rango 1 asociada al valor singular dominante minimiza el error en término de la norma forbenius  $F$ .

Tabla 2.3. Algoritmo K-SVD.

ENTRADA	<ul style="list-style-type: none"> <li>• Señales de entrenamiento <math>\mathbf{Y}</math>.</li> <li>• Nivel de poca densidad <math>\tau</math> o error <math>\epsilon</math>.</li> <li>• Diccionario <math>\mathbf{D}</math>.</li> </ul>
INICIALIZACIÓN	Iteración $t = 0$ .
PROCEDIMIENTO	<ul style="list-style-type: none"> <li>• Etapa de codificación poco densa: Encontrar los vectores de representación poco densa <math>\gamma_i</math> resolviendo           <math display="block">\arg \min_{\mathbf{Y}, \mathbf{D}} \ \mathbf{Y} - \mathbf{D}\mathbf{\Gamma}\ _F^2 \text{ tal que } \forall i, \ \gamma_i\ _0 \leq \tau</math> </li> <li>• Etapa de actualización del diccionario: Para cada átomo del diccionario <math>\mathbf{D}^{(t-1)}</math> actualizar por:           <ul style="list-style-type: none"> <li>– Definir el grupo de señales en <math>\mathbf{Y}</math> que usan el átomo               <math display="block">\omega_k = \{i   1 \leq i \leq N, \gamma_T^k(i) \neq 0\}</math> </li> <li>– Calcular la matriz de error como:               <math display="block">\mathbf{E}_k = \mathbf{Y} - \mathbf{D}\gamma_k</math> </li> <li>– Restringir <math>\mathbf{E}_k</math> a las columnas correspondientes a <math>\omega_k</math>, obteniendo <math>\mathbf{E}_k^R</math></li> <li>– Aplicar SVD a <math>\mathbf{E}_k^R = \mathbf{U}\mathbf{\Delta}\mathbf{V}^\dagger</math>. Actualizar el átomo <math>\mathbf{d}_k</math> como la primera columna de <math>\mathbf{U}</math>. Actualizar el vector de coeficiente <math>\gamma_R^k</math> como la primera de columna de <math>\mathbf{V}</math> multiplicada por <math>\mathbf{\Delta}(1, 1)</math></li> </ul> </li> <li>• Incrementar la iteración <math>t = t+1</math></li> </ul>
SALIDA	<ul style="list-style-type: none"> <li>• El diccionario <math>\mathbf{D}^{(t)}</math> actualizado.</li> <li>• El grupo de vectores poco denso <math>\mathbf{\Gamma} \in \mathbb{R}^{N \times M}</math>.</li> </ul>

## 2.3 SUPERRESOLUCIÓN DE IMÁGENES USANDO REPRESENTACIÓN POCO DENSA

Como se explicó en el capítulo anterior, una imagen de baja resolución  $\mathbf{Y}$  es una versión diezmada y desenfocada por un operador  $L$ , de una imagen de alta resolución  $\mathbf{X}$ . Usando la teoría de representación poco densa, los bloques de la imagen  $\mathbf{x} \in \mathbb{R}^N$  pueden ser representados en un diccionario sobrecompleto  $\mathbf{D} \in \mathbb{R}^{M \times N}$  con  $M < N$ , usando un grupo de vectores  $\gamma \in \mathbb{R}^M$  los bloques de la correspondiente imagen de baja resolución pueden ser encontrados mediante:

$$\mathbf{y} = L\mathbf{x} = L\mathbf{D}\boldsymbol{\gamma} \quad (2.27)$$

Este problema es aun más subdeterminado, sin embargo, bajo ciertas condiciones la solución poco densa  $\boldsymbol{\gamma}$  es única, además, si  $\mathbf{D}$  cumple ciertas condiciones de isometría cercana, para una gran cantidad de operadores  $L$ , cualquier cantidad de representaciones poco densas de los bloques  $\mathbf{x}$  en términos de  $\mathbf{D}$ , a partir, de las observaciones de baja resolución  $\mathbf{y}$  pueden ser obtenidas, como lo establece la teoría de sensado comprimido en [27].

El uso de diccionarios entrenados ha llevado al estado del arte técnicas de restauración de imágenes [3], para adaptar esta metodología al problema de superresolución se propone el uso de pares de diccionarios  $\mathbf{D}_l$  y  $\mathbf{D}_h$ , para representar los bloques de baja y de alta resolución respectivamente. El grupo de vectores de representación poco densa de los bloques de baja resolución serán utilizados para representar los bloques de alta resolución, esto se logra entrenando los diccionarios concatenados forzando a que la representación poco densa sea similar. Yang y col [20] proponen resolver el problema de optimización de la siguiente manera utilizando la restricción por error:

$$\text{mín } \|\boldsymbol{\gamma}\|_1 \text{ tal que } \|F\mathbf{D}_l\boldsymbol{\gamma} - F\mathbf{y}\|_2^2 \leq \epsilon_1, \|P\mathbf{D}_h\boldsymbol{\gamma} - \mathbf{w}\|_2^2 \leq \epsilon_2 \quad (2.28)$$

El operador  $F$  extrae los detalles para que la reconstrucción haga más énfasis en las componentes de alta frecuencia, el operador  $P$  extrae la región de solapamiento que existe entre la reconstrucción anterior y la actual, y  $\mathbf{w}$  es la región de la reconstrucción previa que se solapa con la nueva. Cabe resaltar, que es necesario aplicar un procesamiento de bloques solapados para que el segundo término de regularización haga sentido. Usando mutiplicadores de Langrange y concatenando las condiciones de optimización, el problema en (2.28) puede reformularse como:

$$\text{mín}_{\boldsymbol{\gamma}} \|\hat{\mathbf{D}}\boldsymbol{\gamma} - \hat{\mathbf{y}}\|_2^2 + \lambda\|\boldsymbol{\gamma}\|_1 \quad (2.29)$$

Donde  $\hat{\mathbf{D}} = \begin{bmatrix} F\mathbf{D}_l \\ P\mathbf{D}_h \end{bmatrix}$ ,  $\hat{\mathbf{y}} = \begin{bmatrix} F\mathbf{y} \\ \mathbf{w} \end{bmatrix}$ . Este es un problema de regresión lineal regula-

rizado con norma  $l_1$ , el cual los autores proponen resolver usando la técnica Lasso [24].

Una vez encontrado  $\gamma$  los bloques de alta resolución se reconstruyen usando el correspondiente diccionario  $\mathbf{x} = \mathbf{D}_h \gamma$ . Dado que,  $\mathbf{y} \approx \mathbf{D}_l \gamma$ , la reconstrucción de superresolución  $\mathbf{X}_0$  no obedece a un modelo de degradación, para solucionar esto, se proyecta  $\mathbf{X}_0$  sobre el espacio de solución de  $S\mathbf{H}\mathbf{X} = \mathbf{Y}$ .

$$\mathbf{X}^* = \arg \min_{\mathbf{X}} \|\mathbf{S}\mathbf{H}\mathbf{X} - \mathbf{Y}\|_2^2 + c\|\mathbf{X} - \mathbf{X}_0\|_2^2 \quad (2.30)$$

Este problema se resuelve iterativamente usando métodos de descenso por gradiente.

### 2.3.1 Diccionarios semiacoplados para superresolución de imágenes

En la metodología anterior se propone hacer un entrenamiento conjunto de los diccionarios

$\mathbf{D}_h$  y  $\mathbf{D}_l$ , al concatenarlos en un único diccionario  $\mathbf{D} = \begin{bmatrix} \mathbf{D}_h \\ \mathbf{D}_l \end{bmatrix}$ , con las señales de en-

trenamiento igualmente concatenadas  $\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$ , el problema de optimización se plantea como:

$$\arg \min_{\mathbf{D}} \|\mathbf{D}\boldsymbol{\Gamma} - \mathbf{Z}\|_F \text{ tal que } \forall i \|\boldsymbol{\gamma}_i\|_0 \leq \tau \quad (2.31)$$

Esta solución es óptima en los espacios acoplados, pero no lo es en cada espacio individual, entonces, la reconstrucción de superresolución  $\mathbf{x} = \mathbf{D}_h \gamma$  no esta garantizada a ser la mejor. Para solventar esta limitante, Yang y col en [28] proponen un método de entrenamiento de diccionarios acoplados, el cual consiste en entrenar un diccionario a la vez, considerando el otro óptimo, sin embargo, este enfoque es muy restringido para manejar la flexibilidad de la composición de las imágenes. Wang y col. [29] parten del hecho de que la imagen  $\mathbf{X}$  de alta resolución y su equivalente de baja resolución  $\mathbf{Y}$  existen en 2 espacios vectoriales, tales que, una función invertible de mapeo  $F(\cdot)$  permita ir de un espacio a otro. La principal limitante de esta suposición es que dicha función  $F(\cdot)$  es desconocida.

A fin de encontrar el par de espacios vectoriales expandidos por los diccionarios  $\mathbf{D}_h$  y  $\mathbf{D}_l$  junto con su correspondiente función de mapeo  $F(\cdot)$ , se propone minimizar la siguiente

función de energía:

$$\min_{\mathbf{D}_h, \mathbf{D}_l, F(\cdot)} E_{data}(\mathbf{D}_h, X) + E_{data}(\mathbf{D}_l, Y) + \alpha E_{map}(F(\mathbf{\Gamma}_h), \mathbf{\Gamma}_l) + \lambda E_{reg}(\mathbf{\Gamma}_h, \mathbf{\Gamma}_l, F(\cdot), \mathbf{D}_h, \mathbf{D}_l) \quad (2.32)$$

Donde  $E_{data}(\cdot, \cdot)$  representa la fidelidad de la reconstrucción en términos del error,  $E_{map}(\cdot, \cdot)$  es la fidelidad del mapeo para representar el error en la codificación poco densa,  $E_{reg}(\cdot, \cdot, \cdot, \cdot)$  es un término de regularización para balancear el error de mapeo y el error de reconstrucción,  $\mathbf{\Gamma}_h$  la representación poco densa de una imagen de alta resolución sobre  $\mathbf{D}_h$ , y  $\mathbf{\Gamma}_l$  la representación poco densa de su correspondiente imagen de baja resolución sobre  $\mathbf{D}_l$ . En el último término de (2.32) se observa que la función de mapeo  $F(\cdot)$  relaciona los vectores de representación poco densa ( $\mathbf{\Gamma}_h$  y  $\mathbf{\Gamma}_l$ ) y no las imágenes ( $\mathbf{X}$  y  $\mathbf{Y}$ ) propiamente.

Es importante que la función de mapeo sea lineal, de manera tal que,  $F(u + v) = F(u) + F(v)$  y  $F(ku) = kF(u)$  para que el problema en (2.32) se convierta un problema de entrenamiento de diccionarios junto con una regularización de *Tikhonov*<sup>4</sup> [29], definido como:

$$\min_{\mathbf{D}_h, \mathbf{D}_l, \mathbf{W}} \|\mathbf{X} - \mathbf{D}_h \mathbf{\Gamma}_h\|_F^2 + \|\mathbf{Y} - \mathbf{D}_l \mathbf{\Gamma}_l\|_F^2 + \alpha \|\mathbf{\Gamma}_l + \mathbf{W} \mathbf{\Gamma}_h\|_F^2 + \lambda_h \|\mathbf{\Gamma}_h\|_1 + \lambda_l \|\mathbf{\Gamma}_l\|_1 + \lambda_W \|\mathbf{W}\|_F^2 \quad (2.33)$$

Donde  $\lambda_h$ ,  $\lambda_l$ ,  $\lambda_W$  son términos de regularización, y  $\mathbf{W}$  es la matriz que representa la función de mapeo<sup>5</sup>. Luego de haber entrenado los diccionarios  $\mathbf{D}_h$ ,  $\mathbf{D}_l$  y haber encontrado la función de mapeo  $\mathbf{W}$ , la reconstrucción de superresolución se lleva a cabo en 2 pasos, el primer, resolver para cada bloque de la imagen el siguiente problema:

$$\min_{\gamma_{h_i}, \gamma_{l_i}} \|\mathbf{x}_i - \mathbf{D}_h \gamma_{h_i}\|_F^2 + \|\mathbf{y}_i - \mathbf{D}_l \gamma_{l_i}\|_F^2 + \alpha \|\gamma_{l_i} - \mathbf{W} \gamma_{h_i}\|_F^2 + \lambda_h \|\gamma_{h_i}\|_1 + \lambda_l \|\gamma_{l_i}\|_1 \quad (2.34)$$

Una vez encontrado los vectores de representación poco densa  $\mathbf{\Gamma}_h$  y  $\mathbf{\Gamma}_l$ , los bloques de

<sup>4</sup>El problema de regularización de *Tikhonov* también conocido como regresión de *Ridge* es un problema de mínimos cuadrados regularizado.

<sup>5</sup>En el contexto del algebra lineal una función lineal corresponde a un transformación lineal.

alta resolución se reconstruyen haciendo la siguiente operación:

$$\mathbf{x}_i = \mathbf{D}_h \mathbf{W}^+ \boldsymbol{\gamma}_{l_i}$$

### 2.3.1.1 Entrenamiento de diccionarios semiacoplados

Para resolver el problema propuesto en (2.33) éste se divide en 3 subproblemas de optimización. El primero es encontrar la representación poco densa de las señales, y la función de mapeo  $\mathbf{W}$ , la cual se inicializa con la matriz identidad  $\mathbf{I}$ .

$$\begin{aligned} \min_{\boldsymbol{\Gamma}_h} \|\mathbf{X} - \mathbf{D}_h \boldsymbol{\Gamma}_h\|_F^2 + \alpha \|\boldsymbol{\Gamma}_l - \mathbf{W}_h \boldsymbol{\Gamma}_h\|_F^2 \text{ tal que } \forall i \|\boldsymbol{\gamma}_{h,i}\|_0 \leq \tau \\ \min_{\boldsymbol{\Gamma}_l} \|\mathbf{Y} - \mathbf{D}_l \boldsymbol{\Gamma}_l\|_F^2 + \alpha \|\boldsymbol{\Gamma}_h - \mathbf{W}_l \boldsymbol{\Gamma}_l\|_F^2 \text{ tal que } \forall i \|\boldsymbol{\gamma}_{l,i}\|_0 \leq \tau \end{aligned} \quad (2.35)$$

Wang resuelve este problema usando algoritmos de optimización de norma  $l_1$  específicamente LARS [30]. Una vez conseguido los vectores de representación poco densa  $\boldsymbol{\Gamma}_h$  y  $\boldsymbol{\Gamma}_l$ , los diccionarios se entrenan resolviendo:

$$\min_{\mathbf{D}_h, \mathbf{D}_l} \|\mathbf{X} - \mathbf{D}_h \boldsymbol{\Gamma}_h\|_F^2 + \|\mathbf{Y} - \mathbf{D}_l \boldsymbol{\Gamma}_l\|_F^2 \text{ tal que } \forall i \|\boldsymbol{\gamma}_{h,i}\|_0 \leq \tau, \|\boldsymbol{\gamma}_{l,i}\|_0 \leq \tau \quad (2.36)$$

El cual es un problema de programación cuadrática restringido cuadráticamente (QCQP *Quadratically Constrained Quadratic Program*) [28]. Por último se actualiza la función de mapeo resolviendo:

$$\min_{\mathbf{W}} \|\boldsymbol{\Gamma}_l - \mathbf{W} \boldsymbol{\Gamma}_h\| + (\lambda_W / \alpha) \|\mathbf{W}\|_F^2 \quad (2.37)$$

A través de:

$$\mathbf{W} = \boldsymbol{\Gamma}_l \boldsymbol{\Gamma}_h^\dagger (\boldsymbol{\Gamma}_h \boldsymbol{\Gamma}_h^\dagger + (\lambda_W / \alpha) \mathbf{I})^{-1} \quad (2.38)$$

### 2.3.2 Superresolución de imágenes usando diccionarios de pseudoinversa

Este tercer método se basa en la propuesta de Zeyde y col [31], la cual no difiere mucho de las propuestas anteriores, en este sentido, se propone hacer un enriquecimiento de una

versión sobreescalada de la imagen de baja resolución, obtenida usando cualquier algoritmo interpolación para superresolución, esto con el fin de que la ubicación de los bloques sea la misma en ambas imágenes. Denotando  $Q$  como el operador de interpolación la imagen sobreescalada puede expresarse como:

$$\mathbf{Y} = Q(SH\mathbf{X} + \mathbf{v}) = QSH\mathbf{X} + Q\mathbf{v} = L\mathbf{X} + \hat{\mathbf{v}} = QSH\mathbf{X} + Q\mathbf{v} \quad (2.39)$$

La idea es procesar los bloques  $L\mathbf{x}_i \in \mathbb{R}^N$  para obtener  $\mathbf{x}_i \in \mathbb{R}^N$ , conservando las dimensiones entre los espacios. En la fase de entrenamiento, el diccionario para imágenes de baja resolución  $\mathbf{D}_l$ , no se adapta a la estructura de la imagen interpolada, mas bien, a las componentes de alta frecuencia de manera similiar a como lo propone Yang, usando el operador  $F$ . Los autores proponen utilizar 4 filtros para esto:  $\mathbf{f}_1 = [1, 0, -1]$ ,  $\mathbf{f}_2 = \mathbf{f}_1^\dagger$ ,  $\mathbf{f}_3 = [1, 0, -2, 0, 1]$  y  $\mathbf{f}_4 = \mathbf{f}_3^\dagger$ , este preprocesamiento arroja 3 píxeles nuevos por cada píxel original de la imagen. Para reducir la complejidad que implica el hecho de que  $F\mathbf{y} \in \mathbb{R}^{4N}$  se aplica la técnica de PCA para reducir la dimensión fr los datos, seleccionando las componentes que preserven el 99,9 % de la energía,  $\hat{\mathbf{y}} = \mathbf{B}F\mathbf{y}$ , siendo  $\mathbf{B} \in \mathbb{R}^{K \times 4N}$  el operador de proyección.

Al igual que el método anterior y a diferencia del primero, el entrenamiento del par de diccionarios se hace por separado, el diccionario para las imágenes de baja resolución se entrena usando K-SVD, resolviendo el siguiente problema de optimización:

$$\arg \min_{\mathbf{D}_l, \mathbf{\Gamma}} \|\mathbf{B}F\mathbf{Y} - \mathbf{D}_l\mathbf{\Gamma}\|_F^2 \text{ tal que } \forall i \|\gamma_i\|_0 \leq \tau \quad (2.40)$$

El diccionario de alta resolución se entrena con los detalles perdidos por el proceso de interpolación, esto significa, que las imágenes de entrenamiento se obtienen haciendo  $\hat{\mathbf{X}} = \mathbf{X} - Q\mathbf{Y}$ . El diccionario corresponde a la solución de,  $\min_{\mathbf{D}_h} \|\hat{\mathbf{X}} - \mathbf{D}_h\mathbf{\Gamma}\|_F^2$ , que es obtenida aplicando la pseudoinversa  $\mathbf{D}_h = \hat{\mathbf{X}}\mathbf{\Gamma}^+$ .

Una vez entrenados los diccionarios, y encontrado el conjunto de vectores de representación poco densa  $\mathbf{\Gamma}$  los bloques de alta resolución se componen haciendo  $\mathbf{X} = \mathbf{Y} + k\mathbf{D}_h\mathbf{\Gamma}$  donde  $k$  es una variable que controla el nivel de enriquecimiento.

# CAPÍTULO 3

## APLICACIÓN DE MÉTODOS DE SUPERRESOLUCIÓN A IMÁGENES SATELITALES

*Las técnicas de superresolución de imágenes usando representación poco densa descritas en el capítulo anterior requieren el uso de algoritmos de optimización bastante costosos en términos computacionales, específicamente las propuesta de Yang y Wang (ver secciones 2.3 y 2.3.1 respectivamente). Por eso fueron probadas en imágenes relativamente pequeñas (con dimensiones no mayores a  $256 \times 256$  píxeles). Con el objetivo de acelerarlas, se adaptaron al uso de los algoritmos OMP y K-SVD, a fin de poder aplicarlas sobre imágenes satelitales, que es el objetivo de esta investigación en cuestión. Además se explicarán, las consideraciones tomadas para reconstruir la imagen a partir de los bloques que implica filtros de restauración, y un popular algoritmo dedicado a corregir el desenfoque en la imagen.*

### 3.1 MÉTODOS DE SUPERRESOLUCIÓN DE IMÁGENES SATELITALES

A fin de mejorar la reconstrucción de los detalles perdidos, a todos los bloques de las imágenes de baja resolución  $\mathbf{y} \in \mathbb{R}^n$  se les elimina la componente continua antes de ser procesados, y posteriormente se le añade a los bloques de alta resolución reconstruidos  $\mathbf{x} \in \mathbb{R}^{kn}$ , ya que las componentes en baja frecuencia se conservan. Cuando la imagen a ser procesada es multiespectral, existen 2 alternativas, procesar cada banda por separado o seleccionar tres banda y aplicar una transformación al espacio de color YCbCr, donde el 90 % de la información espacial se encuentra en la banda de luminancia Y, las bandas de crominancia son interpoladas, para luego retornar al espacio de color RGB.

### 3.1.1 Superresolución usando entrenamiento conjunto de diccionarios adaptado a OMP

Esta técnica es una adaptación del primer trabajo de superresolución propuesto por Yang y col [20]. Para aplicarla se considera que el par de diccionarios  $\mathbf{D}_h$  y  $\mathbf{D}_l$  han sido entrenados previamente, y se procede a resolver el siguiente problema de optimización usando el algoritmo OMP.

$$\min_{\Gamma} \|\mathbf{D}_h \Gamma - \hat{\mathbf{X}}\|_F^2 + \|\mathbf{D}_l \Gamma - F\mathbf{Y}\|_F^2 \text{ tal que } \forall i \|\gamma_i\|_0 < \tau \quad (3.1)$$

Donde  $\hat{\mathbf{X}}$  es una versión sobreesalada de  $\mathbf{Y}$  obtenida por interpolación,  $F$  es la familia de filtros paso alto<sup>1</sup>, descritos en la sección anterior. Partiendo del hecho de que la imagen objetivo es  $\mathbf{Y}_{obj}$ , se denotarán las imágenes filtradas y reordenadas lexicográficamente en bloques como:

$$\hat{\mathbf{Y}} = \begin{bmatrix} \mathbf{f}_1 ** \mathbf{Y}_{obj} \\ \mathbf{f}_2 ** \mathbf{Y}_{obj} \\ \mathbf{f}_3 ** \mathbf{Y}_{obj} \\ \mathbf{f}_4 ** \mathbf{Y}_{obj} \end{bmatrix}$$

Donde el operador  $**$  denota convolución en 2 dimensiones, entonces el problema de optimización se convierte en:

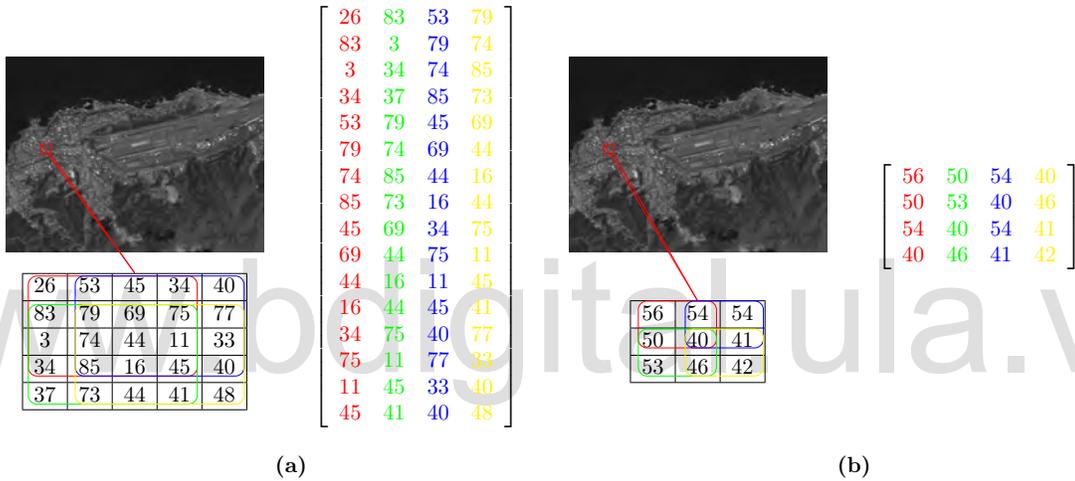
$$\min_{\Gamma} \|\mathbf{D}_{hl} \Gamma - \hat{\mathbf{Z}}\|_F^2 \text{ tal que } \forall i \|\gamma_i\|_0 < \tau, \text{ con } \hat{\mathbf{Z}} = \begin{bmatrix} \mathbf{X} \\ \hat{\mathbf{Y}} \end{bmatrix} \text{ y } \mathbf{D}_{hl} = \begin{bmatrix} \mathbf{D}_h \\ \mathbf{D}_l \end{bmatrix} \quad (3.2)$$

Una vez resuelto el problema anterior, los bloques de alta resolución son generados a partir de las proyecciones de los vectores poco densos sobre el correspondiente diccionario  $\mathbf{X} = \mathbf{D}_h \Gamma$ , las técnicas empleadas para componer la imagen a partir de los bloques serán

<sup>1</sup>El uso de los filtros es opcional, los resultados experimentales muestran resultados similares al resolver el problema de optimización

explicadas en la siguiente sección.

El entrenamiento de los diccionarios se lleva a cabo usando un conjunto de imágenes de mayor resolución  $\mathbf{X}$  con características similares a la imagen objetivo, luego son degradadas para obtener un grupo de imágenes  $\mathbf{Y}$  y divididas en bloques reordenados lexicográficamente, alineados en las matrices  $\mathbf{X} \in \mathbb{R}^{N \times K}$  y  $\mathbf{Y} \in \mathbb{R}^{M \times K}$ , de manera tal, que la ubicación de los bloques haga sentido con el factor de escalamiento y se cumpla que  $\forall i \hat{\mathbf{y}}_i = FSH\mathbf{x}_i + F\mathbf{v}_i$ . En la Fig 3.1 se ilustra la selección de estos bloques. La Fig. 3.1(a) es una imagen de  $512 \times 512$  píxeles dividida en bloques de  $4 \times 4$ . La Fig. 3.1(b) es la misma imagen redimensionada a  $256 \times 256$  píxeles dividida en bloques de  $2 \times 2$ .



**Figura 3.1. Ejemplo de la selección de bloques en la misma región de 2 imágenes.**

Una vez seleccionados los bloques de entrenamiento, se inicializan los diccionarios  $\mathbf{D}_h$ ,  $\mathbf{D}_l$  con el DCT sobrecompleto<sup>2</sup> de dimensiones  $N \times L$  y  $M \times L$  respectivamente con  $L > M + N$ . Estos diccionarios deben ser normalizados, de manera tal, que al concatenarse todas sus columnas tengan norma  $l_2$  unitaria. Definiendo el vector  $\mathbf{u} \in \mathbb{R}^L$  donde cada componente  $\mathbf{u}[i] = \sqrt{\|\mathbf{d}_{h,i}\|^2 + \|\mathbf{d}_{l,i}\|^2}$ , los correspondientes diccionarios normalizados son  $\hat{\mathbf{D}}_h = \begin{bmatrix} \mathbf{d}_{h_1} \\ \mathbf{u}[1] \end{bmatrix}, \begin{bmatrix} \mathbf{d}_{h_2} \\ \mathbf{u}[2] \end{bmatrix}, \dots, \begin{bmatrix} \mathbf{d}_{h_L} \\ \mathbf{u}[L] \end{bmatrix}$  y  $\hat{\mathbf{D}}_l = \begin{bmatrix} \mathbf{d}_{l_1} \\ \mathbf{u}[1] \end{bmatrix}, \begin{bmatrix} \mathbf{d}_{l_2} \\ \mathbf{u}[2] \end{bmatrix}, \dots, \begin{bmatrix} \mathbf{d}_{l_L} \\ \mathbf{u}[L] \end{bmatrix}$ , y entrenados usando el algoritmo K-SVD resolviendo:

$$\arg \min_{\mathbf{D}, \mathbf{\Gamma}} \|\mathbf{D}\mathbf{\Gamma} - \mathbf{Z}\|_F^2 \text{ tal que } \forall i \|\gamma_i\|_0 < \tau \quad (3.3)$$

<sup>2</sup>Otra opción de inicialización de los diccionarios es tomar L señales aleatorias de los bloques de entrenamiento.

$$\text{Donde } \mathbf{Z} = \begin{bmatrix} \mathbf{X} \\ \hat{\mathbf{Y}} \end{bmatrix}, \mathbf{D} = \begin{bmatrix} \hat{\mathbf{D}}_h \\ \hat{\mathbf{D}}_l \end{bmatrix}.$$

Al concluir el entrenamiento los átomos de los diccionarios deben ser nuevamente normalizados, de la siguiente manera:

$$\forall i \mathbf{d}_{h_i} = \frac{\mathbf{d}_i[1..N]}{\|\mathbf{d}_i[1..N]\|}, \mathbf{d}_{l_i} = \frac{\mathbf{d}_i[N+1..N+M]}{\|\mathbf{d}_i[N+1..N+M]\|}$$

### 3.1.2 Superresolución usando diccionarios semiacoplados adaptado a OMP

El problema de optimización planteado por Wang resumido en la ecuación (2.35) es bastante exigente desde el punto de vista de complejidad computacional, y no puede ser resuelto por un algoritmo como OMP directamente, por esta razón es ligeramente modificado de la siguiente manera. Asumiendo la existencia del par de diccionarios  $\mathbf{D}_h$ ,  $\mathbf{D}_l$  y la función de mapeo  $\mathbf{W}$ , primero se encuentra el grupo de vectores de representación poco densa  $\mathbf{\Gamma}$  resolviendo:

$$\arg \min_{\mathbf{\Gamma}} \|\mathbf{Y} - \mathbf{D}_l \mathbf{\Gamma}\|_F^2 \text{ tal que } \forall i \|\boldsymbol{\gamma}_i\|_0 < \tau \quad (3.4)$$

Luego el grupo de bloques de alta resolución es generado a partir de:

$$\mathbf{X} = \mathbf{D}_h \mathbf{W} \mathbf{\Gamma} \quad (3.5)$$

Para entrenar los diccionarios se resuelve el siguiente problema de optimización aplicando independientemente el algoritmo K-SVD por separado, una vez para determinar  $\mathbf{D}_h$ , y la otra para encontrar  $\mathbf{D}_l$ .

$$\arg \min_{\mathbf{D}_h, \mathbf{\Gamma}_h} \|\mathbf{D}_h \mathbf{\Gamma}_h - \mathbf{X}\|_F^2 + \arg \min_{\mathbf{D}_l, \mathbf{\Gamma}_l} \|\mathbf{D}_l \mathbf{\Gamma}_l - \mathbf{Y}\|_F^2 \text{ tal que } \forall i \|\boldsymbol{\gamma}_{h_i}\|_0 \leq \tau, \|\boldsymbol{\gamma}_{l_i}\|_0 \leq \tau \quad (3.6)$$

Siendo  $\mathbf{X}$  y  $\mathbf{Y}$  las matrices que contienen los bloques de entrenamiento, como se explicó en la técnica anterior. La ventaja de utilizar K-SVD es que no solo optimiza el diccionario sino el vector de representación poco densa facilitando la obtención de la función de mapeo,

que será determinada a partir de la solución a:

$$\min_{\mathbf{W}} \|\Gamma_h - \mathbf{W}\Gamma_l\| \quad (3.7)$$

Que corresponde a  $\mathbf{W} = \Gamma_h \Gamma_l^+$ . Cabe destacar que la existencia de  $\mathbf{W}$  esta garantizada, ya que, se cumple  $\forall i \|\gamma_{l_i}\|_0 \geq 1$ , y esto a su vez asegura que el  $\det(\Gamma_l^\dagger \Gamma_l) \neq 0$ . Otro importante hecho a resaltar a partir de los experimentos, es que para la mayoría de los casos se observó una similitud entre las normas  $l_1$  de  $\Gamma_h$  y  $\mathbf{W}\Gamma_l$ , permitiendo inferir que el segundo término está bastante cerca de la solución óptima en el espacio expandido por  $\mathbf{D}_h$ , partiendo de la demostrado por *Donoho* y col en [32], donde se demuestra que bajo ciertas condiciones, para la mayoría de los sistemas subdeterminados la minimización de la norma  $l_1$  también es la de menor densidad.

### 3.2 RECONSTRUCCIÓN DE LA IMAGEN DE ALTA RESOLUCIÓN

Las técnicas hasta ahora descritas se han centrado en la obtención de los bloques de alta resolución, para generar la imagen hay que usar filtros de reconstrucción para estimar cada píxel y tomar en cuenta ciertas consideraciones con respecto a la ubicación de los bloques, tales como, que el solapamiento disminuye al incrementarse el tamaño de la imagen en proporción al factor de escalamiento.

Supóngase que se tiene una imagen  $\mathbf{Y}$  de dimensiones  $N \times N$  píxeles, y quiere ser ampliada un factor  $k > 1$ , con  $k \in \mathbb{N}$ , utilizando las técnicas de representación poco densa descritas hasta ahora. El primer paso es dividir esta imagen en bloques de  $\sqrt{n} \times \sqrt{n}$ , considerando solapamiento con pasos de  $a$  entre bloques consecutivos, con  $0 < a \leq \sqrt{n}$  y  $a \in \mathbb{N}$ , que se traduce en  $(\sqrt{n} - a) \times \sqrt{n}$  píxeles solapados, para alinearlos en una matriz  $\mathbf{Y} \in \mathbb{R}^{n \times L}$ , donde  $L = \frac{1}{a^2}(M - \sqrt{n} + a)^2$ . Al aplicar las técnicas de superresolución se obtiene una matriz  $\mathbf{X} \in \mathbb{R}^{k^2 n \times L}$ , la nueva dimensión de los bloques es  $k\sqrt{n} \times k\sqrt{n}$ , y la dimensión de la imagen  $\mathbf{X}$  debe ser  $kN \times kN$ . Si se quisiera dividir esta imagen en bloques de  $k\sqrt{n} \times k\sqrt{n}$  con pasos de  $b$ , su correspondiente matriz de reordenamiento sería  $\mathbf{X} \in \mathbb{R}^{k^2 \times L'}$  con  $L' = \frac{1}{b^2}(kM - k\sqrt{n} + b)^2 = (\frac{k}{b})^2(M - \sqrt{n} + \frac{b}{k})^2$ , ahora si el nuevo paso entre

bloques es  $b = ka$  entonces  $L' = \frac{1}{a^2}(M - \sqrt{n} + a)^2$ , demostrando así, que el nuevo paso entre bloques debe ser el de la imagen objetivo multiplicado por el factor de escalamiento  $k$ .

Obsérvese que  $kN \times kN < k^2n \times L'$ , lo que implica que en la matriz de reordenamiento  $\mathbf{X}$  hay mayor cantidad de píxeles que los necesarios para generar la imagen  $\mathbf{X}$ , esto significa que cada píxel de la imagen es el resultado de la estimación matemática de los píxeles en los bloques cuya ubicación se solape. En la Fig. 3.2 se ilustra este principio cuando el factor de escalamiento es de 3; donde la Fig. 3.2(a) representa 3 bloques de  $3 \times 3$  solapados, considerando pasos de 1 y la Fig. 3.2 (b) representa los bloques obtenidos con pasos de 3.

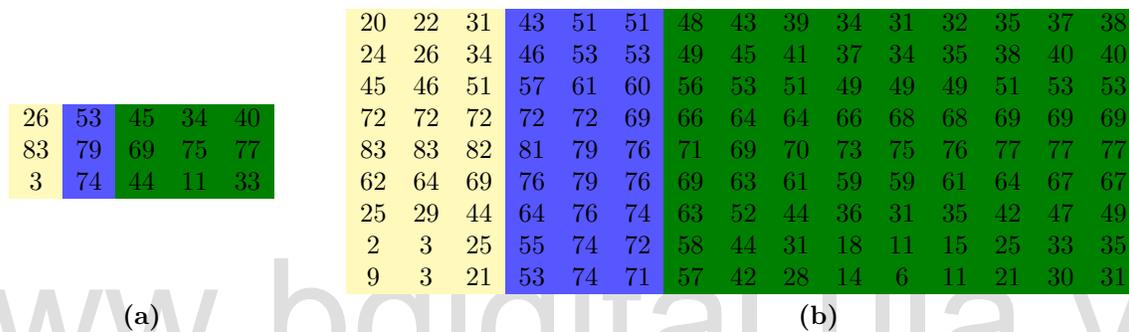


Figura 3.2. Ubicación de los bloques después de un escalamiento por un factor 3.

### 3.2.1 Filtros de reconstrucción

Como se explicó en la sección anterior, cada píxel de la imagen es el resultado de una estimación matemática de un conjunto de valores compuesto por el píxel de cada bloque que se ubica sobre la posición de interés, la dimensión de este conjunto varía dependiendo de la ubicación del bloque, su dimensión y el paso del solapamiento. Los píxeles que están en las esquinas son únicos, ya que, existe un único píxel en esa ubicación, mientras que los que están mas al centro generan conjuntos de la dimensión del bloque entre el paso del solapamiento al cuadrado. En la Fig. 3.3 se ejemplifica este principio, donde cada valor representa la cantidad de píxeles usados para la estimación; usando bloques de  $4 \times 4$  con pasos de 2 en el solapamiento.

Para encontrar el valor del píxel en función de los múltiples valores estimados se pueden usar los siguientes filtros:

1	1	2	2	...	2	...	2	2	1	1
1	1	2	2	...	2	...	2	2	1	1
2	2	4	4	...	4	...	4	4	2	2
2	2	4	4	...	4	...	4	4	2	2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2	2	4	4	...	4	...	4	4	2	2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2	2	4	4	...	4	...	4	4	2	2
2	2	4	4	...	4	...	4	4	2	2
1	1	2	2	...	2	...	2	2	1	1
1	1	2	2	...	2	...	2	2	1	1

Figura 3.3. Cantidad de píxeles usados para la estimación.

- Filtro promediador:** Es el filtro más eficiente desde el punto de vista computacional, ya que, solo requiere promediar los píxeles del conjunto  $\hat{x}_{i,j} = \sum_{z=1}^p \frac{x_{i,j,z}}{p}$ , donde  $p$  es la cantidad usada para la estimación.
- Filtro de mediana:** La operación de mediana es una estimación más robusta, pero computacionalmente más costosa, ya que, requiere un reordenamiento de datos. El píxel vendra dado por  $\hat{x}_{i,j} = \text{MEDIANA}([x_{i,j,1}, \dots, x_{i,j,p}])$ .
- Filtro de media ponderada:** La calidad de la estimación puede ser mejorada si se le da una ponderación adecuada a los píxeles dentro del conjunto,  $\hat{x}_{i,j} = \sum_{z=1}^p \frac{x_{i,j,z} \times w_z}{W}$  siendo  $W$  la sumatoria de las ponderaciones.
- Filtro de mediana ponderada:** Este filtro combina calidad con robustez, pero a su vez es la operación mas costosa de todas, la estimación viene dada por  $\hat{x}_{i,j} = \text{MEDIANA}([w_1 \diamond x_{i,j,1}, \dots, w_p \diamond x_{i,j,p}])$ . El operador  $\diamond$  denota repetición, por ejemplo,  $w \diamond x$  indica que la muestra  $x$  se repite  $w$  veces. En el apéndice A se describe como determinar la mediana ponderada.

### 3.2.1.1 Selección de las ponderaciones

Una selección adecuada de las ponderaciones del conjunto de píxeles, conlleva a una mejor reconstrucción cuando se usan filtros ponderados, ya que, permite discriminar los datos

de acuerdo a un resultado esperado, en este caso, que la información sea visualmente significativa. Cuando se reconstruye y/o procesa una señal digitalmente, los bordes de ésta no son procesados tan satisfactoriamente como la información central, ya que, no se puede usar la información aportada por los vecinos. Los bloques de la imagen que son las señales objetivo usando representación poco densa tampoco escapan de esta peculiaridad, cuando se estima un píxel la contribución de los bordes debería ser menor a la de un píxel central. En la Fig. 3.4 se resalta en rojo la posición de un píxel en las distintas ventanas de arreglos  $3 \times 3$ ; demostrando la existencia de píxeles esquinados en la estimación

3	74	44	71	43	38	41	40	23
34	85	16	86	23	81	15	73	64
37	73	44	71	44	77	45	70	70
37	77	21	85	23	84	15	55	55
38	67	40	70	42	49	44	59	48
14	44	53	57	57	43	19	37	82
35	70	43	66	43	63	45	84	54
25	52	24	56	33	26	55	24	71
69	50	41	25	53	59	73	63	35

**Figura 3.4. Composición de un píxel.**

Para determinar las ponderaciones se usó una estrategia basada en el trabajo desarrollado por *Flaig* y col en [33], donde se propone utilizar una función de afinidad  $A^{\mu, \delta}$  que varía entre 0 y 1 para estimar la fidelidad de las muestras. El objetivo de esta función de afinidad es disminuir la contribución de las muestras que estén más lejos de un parámetro conocido  $\mu \in (-\infty, \infty)$ . El término  $\delta \in [0, \infty)$  controla la fidelidad de la muestra, la función de afinidad debe ser tal que para  $\delta = 0$  sea un impulso y para  $\delta \rightarrow \infty$  se una constante donde todas las muestras tengan la misma importancia y por tanto la misma ponderación.

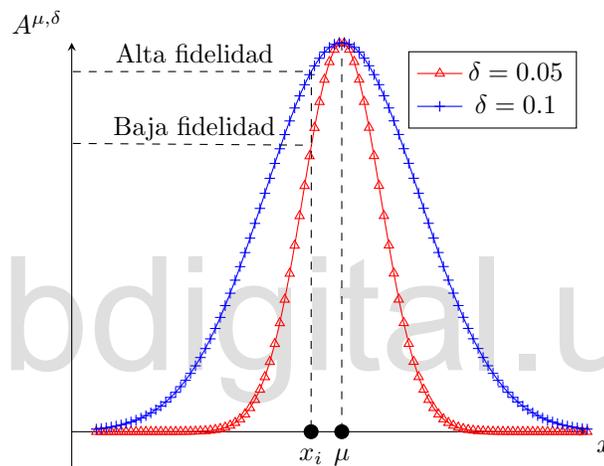
La función de afinidad gaussiana ilustrada en la Fig. 3.5, usada para ponderar las muestras, se define como:

$$A^{\mu, \delta} : w(x) = e^{-\frac{(x-\mu)^2}{\delta}} \quad (3.8)$$

Al usar esta función para determinar las ponderaciones de las muestras se conduce a un filtro de afinidad central (*center affine filter*) (3.9), y a una alteración del filtro de mediana ponderada (3.10).

$$\hat{x} = \frac{\sum_{i=1}^N A_i^\delta x_i w_i}{\sum_{i=1}^N A_i^\delta w_i} \quad (3.9)$$

$$\hat{x} = \text{MEDIANA}[(A_1^\delta w_1 \diamond x_{i,j,1}, \dots, A_p^\delta w_p \diamond x_{i,j,p})] \quad (3.10)$$



**Figura 3.5.** Función de afinidad para la ponderación de las muestras.

Variando el parámetro  $\delta$  según la ubicación del píxel en el bloque se puede ajustar su contribución (como se muestra en la Fig. 3.5), de manera tal que sea mayor en las posiciones centrales. Para determinar cual es el valor de  $\delta$  óptimo se puede resolver el siguiente problema de optimización, de manera iterativa,  $\min_{\delta} \|\mathbf{X} - \hat{\mathbf{X}}_{(\delta)}\|_F^2$ , usando descenso por gradiente. Analizando la expresión del error:

$$E = \|\mathbf{X} - \hat{\mathbf{X}}_{(\delta)}\|_F^2 = \sum_{i=1}^N \sum_{j=1}^M (x_{ij} - \hat{x}_{ij})^2 \quad (3.11)$$

Derivando con respecto a  $\delta$

$$\frac{\partial E}{\partial \delta} = \left( -2 \sum_{i=1}^N \sum_{j=1}^M x_{ij} - \hat{x}_{ij} \right) \frac{\partial \hat{\mathbf{X}}}{\partial \delta} \quad (3.12)$$

A fin de adaptar esta optimización a la metodología de procesamiento por bloque de la imagen, debe existir un  $\delta$  en cada posición del bloque denotado como  $\delta_p$ , y su respectivo gradiente vendrá dado por la siguiente expresión:

$$\frac{\partial \hat{x}_{ij}}{\partial \delta_p} = \frac{\sum_{z \in P} w_{ijz} \frac{A_{ijz}^{\delta_p}}{\partial \delta_p} (x_{ij} - \hat{x}_{ijz})}{\sum_{i=1}^K A_{ijz}^{\delta} w_{ijz}} \quad (3.13)$$

Donde P es el conjunto que contiene los píxeles que usan  $\delta_p$ . El nuevo  $\delta_p$  se obtiene haciendo:

$$\delta_p(n+1) = \delta_p(n) - \frac{\mu}{2} \frac{\partial E}{\partial \delta_p} \quad (3.14)$$

### 3.2.2 Algoritmo iterativo de *Back-Projection*

Finalizado el proceso de superresolución de una imagen queda una última tarea por hacer, la cual dependerá de su aspecto cualitativo, es decir, si la información es visualmente significativa. Este hecho se justifica porque el modelo de observación de la imagen es desconocido, y las técnicas descritas no resuelven directamente el problema inverso para  $\mathbf{X}$ ,  $\mathbf{Y} = S\mathbf{H}\mathbf{X} + \mathbf{N}$ , tal como lo explica Yang [20]. Para mejorar estos aspectos de la imagen, como los efectos de desenfoco, se propone utilizar la conocida técnica de *Back-projection* introducida por Peleg y col. en [34], basada en el proceso de generación de una imagen de tomografía. Ésta consiste en aprovechar la influencia que tiene un píxel sobre sus vecinos debido a la manera como se capta la imagen, y la respectiva función de dispersión de punto del sensor empleado.

Supóngase que se tiene una imagen  $\mathbf{Y}$  y se le aplica una técnica de superresolución para obtener  $\mathbf{X}$ , asumiendo un modelo de degradación se obtiene una imagen  $\mathbf{Y}^0$ , la diferencia  $\mathbf{Y} - \mathbf{Y}^0$  es calculada y usada para disminuir el error:

$$\epsilon = \|\mathbf{Y} - \mathbf{Y}^0\|_F \quad (3.15)$$

Considerando un vecindario de píxeles cuyo centro es  $z$ , influidos por el desenfoque del sensor  $H^{PSF}$ , cada píxel de la imagen original será  $y_{i,j} = x_{i,j} * H_z^{PSF}$ , donde  $*$  denota convolución en una dimensión. La idea es seleccionar un píxel en la imagen  $\mathbf{Y}^0$ , ver la influencia que este tiene sobre su vecindario y sumar el error ponderado por  $\frac{H^{BP}}{c}$ , donde  $H^{BP}$  es un kernel gaussiano y  $c$  un término de regularización, con el fin de que los píxeles que tengan alta influencia en  $y_{i,j}$  tengan una mayor contribución en  $x_{i,j}$ , y los de baja influencia tenga cada vez menor contribución a medida que se itera.

$$\mathbf{x}_{i,j}^{n+1} = x_{i,j}^n + \sum_{i=0}^N \sum_{j=0}^N (y_{i,j} - y_{i,j}^n) \frac{h^{BP^2}}{c \sum h^{BP}} \quad (3.16)$$

La expresión (3.16), también puede ser analizada como un proceso de enriquecimiento de imágenes.

$$s_{i,j} = x_{i,j} + \lambda F(x_{i,j})$$

Donde  $s_{i,j} = x_{i,j}^{(n+1)}$ ,  $\lambda = 1/c$ ,  $F(x_{i,j}) = \epsilon * H^{BP}$ . Esta técnica es muy útil recuperando los detalles perdidos, pero es muy sensible a contaminaciones, lo que la hace compatible con la robustez de la teoría de representación poco densa.

### 3.3 PAQUETE DE SUPERRESOLUCION DE IMÁGENES

A fin de de completar todos los objetivos propuestos, se desarrollaron una serie de funciones para utilizar las técnicas de superresolución estudiadas, éstas fueron escritas en el lenguaje de programación `Python`, debido a su portabilidad, su sintaxis limpia y su alto nivel de abstracción (orientado a objetos) permitiendo desarrollar potentes herramientas de una forma sencilla. El paquete de superresolución incluye los módulos presentados a continuación, algunos descritos en la última sección de este trabajo. En el CD que incluye una versión digital de este documento se encuentra el paquete completo junto con algunos ejemplos.

- `sparse.py`: Este módulo contiene los 3 métodos de superresolución, los cuales son llamados a través de la función `restoration`, junto con la función de entrenamiento `train`, que de igual manera se particulariza para cada uno. Además posee las fun-

ciones `imcalibrate` e `imrecalibrate`, que permiten acondicionar la imagen cuando es multibanda.

- `wm.py`: Este módulo implementa la función de interpolación polifásica basada en mediana ponderada.
- `dicctools.py`: Contiene las funciones para procesar la imagen en bloques `im2col` y luego reconvierte los bloques en imagen `col2im`.
- `omp.py`: Contiene todas las implementaciones del algoritmo OMP incluyendo su versión más veloz `BatchOMP`.
- `ksvd.py`: Este módulo implementa el algoritmo para entrenamiento de diccionarios K-SVD
- `utilities.py`: Contiene una serie de funciones que permite realizar tareas sencillas como medir el error, contaminar la imagen, aplicar los filtros de reconstrucción, y la función que implementa el algoritmo de *Back-projection*

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

# CAPÍTULO 4

## SIMULACIÓN Y RESULTADOS

*En el presente capítulo se procede a validar los resultados de forma experimental, simulando el modelo de observación para la captura de imágenes, usando distintos factores de escalamiento. Se hace un estudio cuantitativo en función de las medidas de error comunes y de la detección de bordes, variando distintos parámetros dentro de las técnicas desarrolladas. Además se hace un análisis cualitativo aprovechando las características de las distintas imágenes que puede capturar el satélite de una misma escena. Con los resultados obtenidos, se puede llegar a conclusiones sobre la calidad de los métodos desarrollados, así como la influencia que tiene la variación de parámetros*

### 4.1 DEGRADACIÓN DE LAS IMÁGENES

Para aplicar el modelo de degradación descrito en la Sección 1.3.1 es necesario aplicar una operación de diezmo y contaminación de la imagen seleccionada. Como primer paso la imagen es degradada usando un filtro promediador de ventana  $f \times f$ , donde  $f$  es el factor de escalamiento, para eliminar las componentes de alta frecuencia que se puedan solapar. En segundo lugar la imagen es diezmada haciendo una selección intercalada de los píxeles (*down-sampling*) en ambas direcciones de variación (horizontal y vertical), por último la contaminación se realiza usando ruido aditivo gaussiano con desviación estándar  $\sigma$ . La calidad de las técnicas hasta ahora mencionadas será comparada con los métodos de interpolación [8] provistos por el paquete `skimage` escrito en Python, y con la técnica de interpolación polifásica descrita en [17], en función del error y de la información visualmente significativa que la imagen pueda aportar, la cual no es una medida absoluta. La Tabla 4.1 muestra las distintas medidas de desempeño que se usarán en esta investigación, así como la relación señal a ruido.

Tabla 4.1. Medidas de desempeño

Nombre	Descripción	Ecuación
RMSE	Raíz del error medio cuadrático ( <i>Root-Mean Square Error</i> )	$\sqrt{\frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M \ y_{ij} - \hat{y}_{ij}\ ^2}$
MAE	Error medio absoluto ( <i>Mean Absolute Error</i> )	$\frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M  y_{ij} - \hat{y}_{ij} $
PSNR	Relación señal a ruido pico ( <i>Peak Signal-to-Noise Ratio</i> )	$20 \log_{10} \left  \frac{\text{máx}(Y)}{\text{RMSE}(Y, \hat{Y})} \right $
SNR	Relación señal a ruido ( <i>Signal-to-Noise Ratio</i> )	$10 \log_{10} \left  \frac{\sum_{i=1}^N \sum_{j=1}^M \ y_{ij}\ ^2}{\sum_{i=1}^N \sum_{j=1}^M \ y_{ij} - \hat{y}_{ij}\ ^2} \right $

## 4.2 ANÁLISIS CUANTITATIVO

Este análisis se lleva a cabo usando una imagen multiespectral (10 m/píxel, con un ancho de 7,2 km) mostrada en la Fig. 4.1; se utiliza la componente de luminancia (Y) del espacio de color YCbCr luego de una conversión de las bandas RGB. Esta imagen fue diezmada usando factores de escalamiento de 2, 3 y 4, y contaminada con ruido aditivo de desviación estándar  $\sigma = \{4, 8, 12, \dots, 40\}$ . Cabe destacar que la resolución radiométrica de las imágenes satelitales es de 10 bits por banda, razón por la cuál se utilizan niveles de contaminación relativamente altos comparados con imágenes comunes de 8 bits por banda. El nivel de contaminación será medido en función de la SNR, definida en la Tabla 4.1, usando la imagen diezmada como referencia y no la original, siguiendo el modelo de degradación. En este sentido, cuando la SNR es alta significa que el nivel de contaminación es bajo, y a medida que la SNR va disminuyendo la contaminación va aumentando. En cuánto a las medidas de desempeño una PSNR alta indica una buena reconstrucción mientras que para el MAE y el RMSE un valor alto indica una mala reconstrucción. Una alternativa

para medir la calidad de los métodos en función de la información visualmente significativa, es aplicar una detección de bordes a la imagen reconstruida, para valorar la cantidad de detalles que pueden ser recuperados, que es el objetivo final de la superresolución de imágenes. Para medir la calidad de las técnicas se usará el % de píxeles empleados para representar los bordes.



Figura 4.1. Imagen de prueba para el análisis cuantitativo.

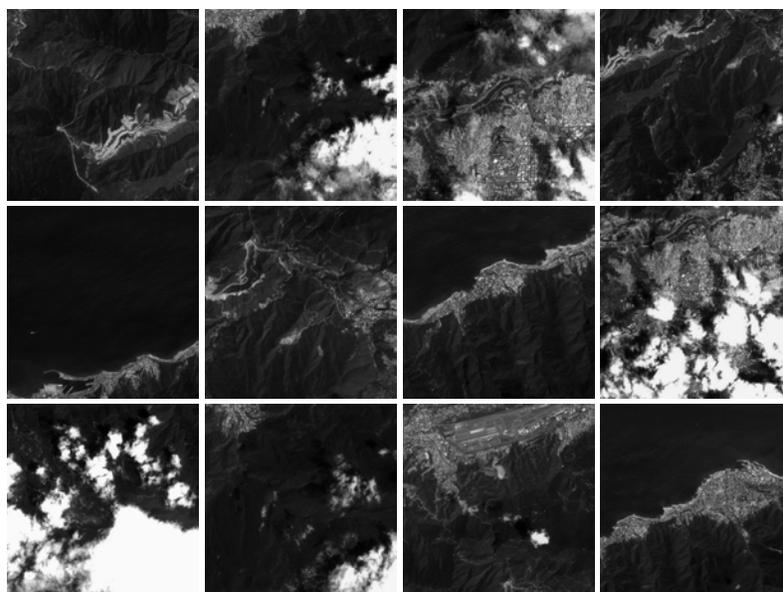


Figura 4.2. Imágenes usadas para el entrenamiento de los diccionarios empleados en el análisis cuantitativo.

### 4.2.1 Selección de Parámetros

Las técnicas expuestas utilizan una serie de parámetros además del factor de escalamiento que inciden en el resultado final como, el tamaño de bloque, el porcentaje de solapamiento, y el filtro de reconstrucción. El conjunto de imágenes mostrados en la Fig. 4.2 fueron usadas para entrenamiento, en los experimentos se usaron 2 tamaños de bloque por cada factor de escalamiento. En cuanto a los parámetros de optimización, se usó como restricción el error con un número máximo de 5 átomos, tanto para entrenamiento como para reconstrucción variando el valor del error según el nivel de ruido ( $\sigma$ ), siguiendo la metodología que usa *M. Elad* y col. [35] para disminución de ruido a través K-SVD.

### 4.2.2 Rendimiento variando el factor de escalamiento

La siguientes características mostradas en las Fig. 4.3, Fig. 4.4 y Fig.4.5 se obtuvieron al variar el factor de escalamiento usando un mismo tamaño de bloque, y un filtro promediador para estimar la imagen, usando PSNR como medida de rendimiento. Los métodos propuestos se comparan entre ellos y con los métodos de interpolación bicúbica descritos en [8], y polifásica detallados en [17].

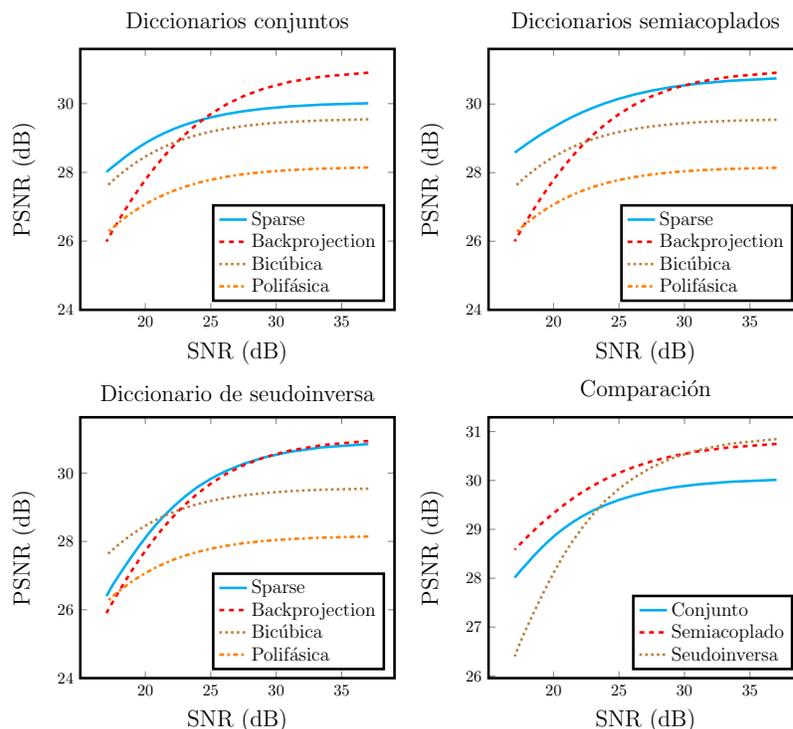


Figura 4.3. PSNR usando bloques de  $3 \times 3$  y factor de escalamiento = 2.

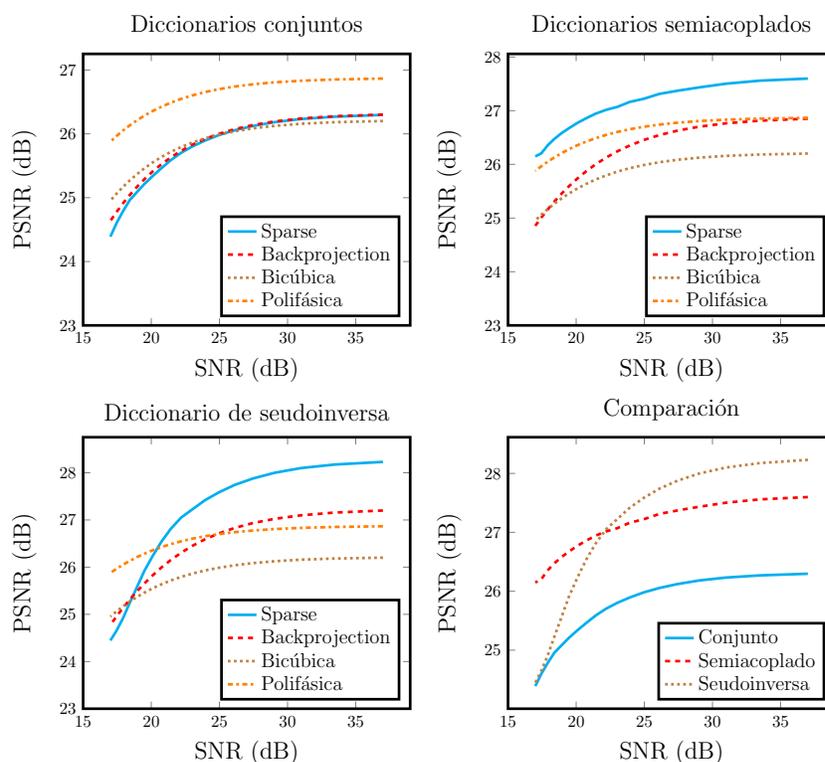


Figura 4.4. PSNR usando bloques de  $3 \times 3$  y factor de escalamiento = 3.

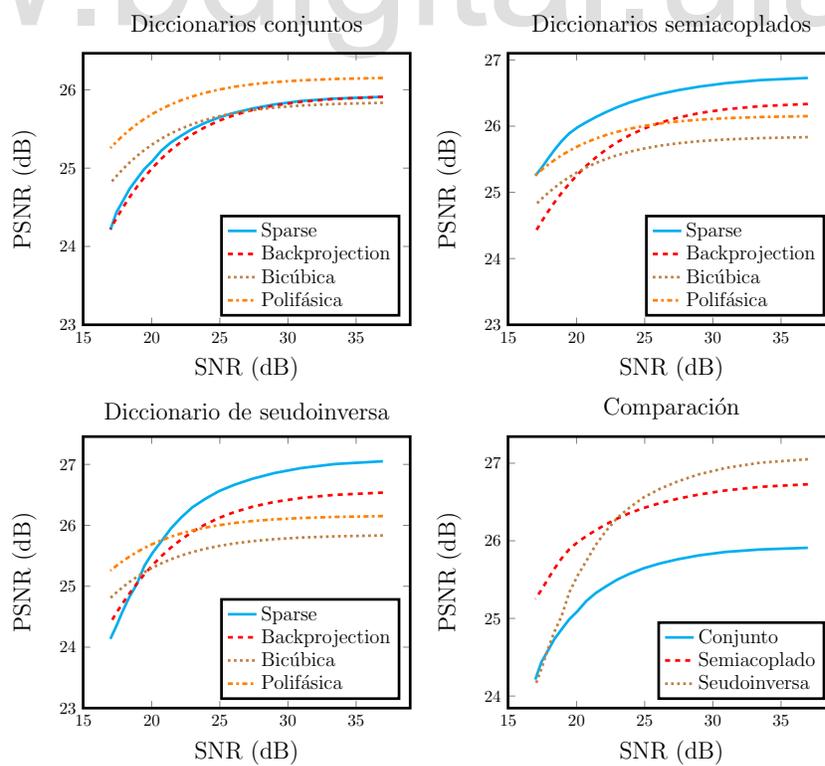


Figura 4.5. PSNR usando bloques de  $3 \times 3$  y factor de escalamiento = 4.

En las gráficas anteriores se observa el desempeño de las técnicas propuestas, para el método que emplea diccionarios conjuntos existe una mejora sobre la interpolación bicúbica, de aproximadamente 0,4 dB en términos de la PSNR, cuando la imagen se amplía por un factor de 2 lo cual hace sentido, ya que, ésta se usa como restricción para resolver el problema de optimización. Para factores de escalamiento mayores, es decir, 3 y 4, la calidad de reconstrucción es equivalente a la interpolación bicúbica e inclusive menos robusta, decayendo cerca de 0,5 dB cuando la SNR es menor a 20 dB.

Cuando se emplean diccionarios semiacoplados, se observa una mejora significativa sobre ambos métodos de interpolación y una mayor robustez, para un factor de escalamiento de 2 la mejora es de 1,8 dB cuando la imagen está limpia, va decayendo suavemente hasta 0,9 dB cuando la contaminación es la máxima aplicada, esta comparación se hace con la interpolación bicúbica que presenta un mejor desempeño. Para factores de 3 y 4 el comportamiento es similar, obteniendo mejoras más discretas que van de 1 dB con imágenes limpias a 0,2 dB con imágenes contaminadas, y de 0,8 dB a 0 dB para factores de escalamiento de 3 y 4 respectivamente.

El uso de diccionarios de pseudoinversa provee la mejor calidad de reconstrucción pero una alta sensibilidad a la contaminación, para un factor de escalamiento de 2, existe una mejora poco superior a los 1,8 dB para imágenes limpias, pero cuando la SNR es menor a 20 dB el desempeño es menor a la interpolación bicúbica generando una pérdida en la calidad de 2 dB para el peor de los casos. Para mayores factores de escalamiento la ganancia de este método es bastante elevada, con una mejora de 2 dB y 1,2 dB en imágenes limpias y un límite de contaminación cuando la SNR es de 20 dB, partir de ese valor la contaminación se propaga; este comportamiento se debe al uso de filtros paso alto a la hora de pre-acondicionar la imagen.

El algoritmo *backprojection* fue aplicado después de reconstruir la imagen, demostrando un comportamiento similar independiente del método; para un factor de escalamiento de 2 la calidad de reconstrucción va de 26 dB a 30,6 dB agregando una mejora si el desempeño del método previo está por debajo, o reduciendo la calidad si el método previo está por arriba.

### 4.2.3 Rendimiento variando tamaño de bloque

Las siguientes características mostradas en las Fig. 4.6, Fig. 4.7 y Fig. 4.8, se obtuvieron variando el tamaño de bloque para distintos factores de escalamiento evaluando el desempeño en función del MAE.

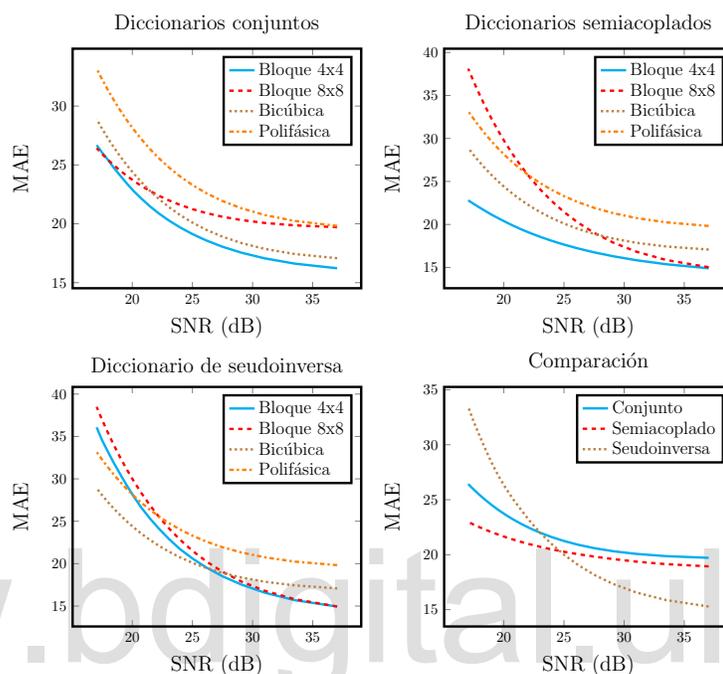


Figura 4.6. MAE Variando el tamaño de bloque y factor de escalamiento = 2.

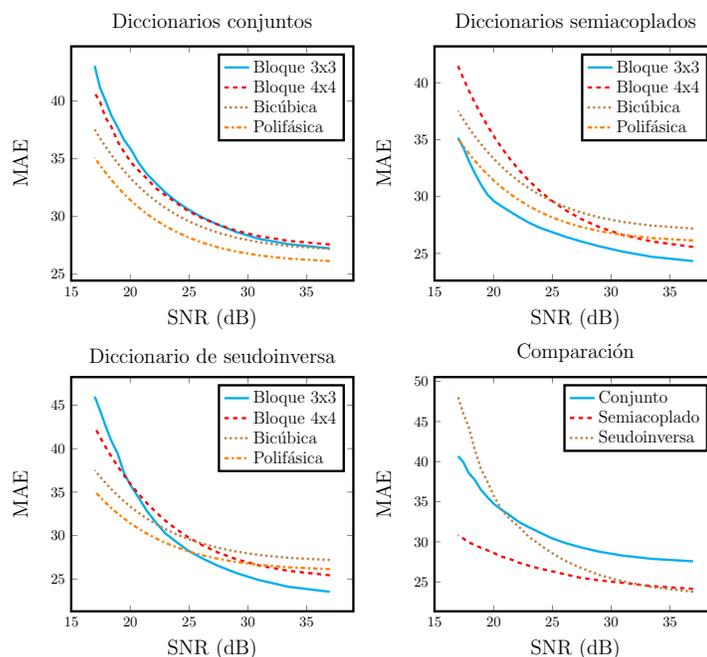
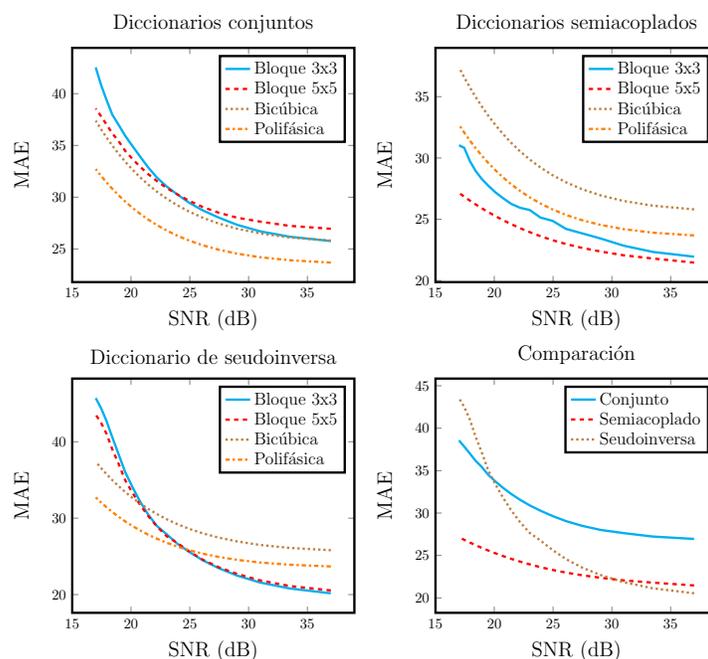


Figura 4.8. MAE Variando el tamaño de bloque y factor de escalamiento = 4.



**Figura 4.7. MAE Variando el tamaño de bloque y factor de escalamiento = 3.**

Cuando se emplea el método de diccionarios conjuntos, el uso de bloques más grandes disminuye la calidad de reconstrucción, con una diferencia de aproximadamente 5 en función del MAE entre ellos y mejorando un valor aproximado de 2 con respecto a la mejor interpolación, en cuanto a la robustez existe una tendencia a un valor común cuando la SNR es menor a 20 dB.

En el caso de diccionarios semiacoplados, es más conveniente usar un tamaño de bloque mayor cuando el factor de escalamiento es 3 presentando una mejora de 2 valores en función del MAE cuando se emplea bloques de  $5 \times 5$  con respecto a bloques  $3 \times 3$ . Para un factor de escalamiento de 2 la calidad de reconstrucción es igual cuando la imagen está limpia, pero la robustez de un tamaño de bloque de  $4 \times 4$  es bastante superior a uno de  $8 \times 8$ , conduciendo a una diferencia de 18 cuando la SNR es menor a 20 dB, siendo este el peor desempeño. Para un factor de escalamiento de 4 el uso de bloques de  $3 \times 3$  es el mejor resultado con una mejora que va de 2 a 5 valores en función del MAE con respecto a bloques de  $4 \times 4$ , siendo este último de menor desempeño a la interpolación.

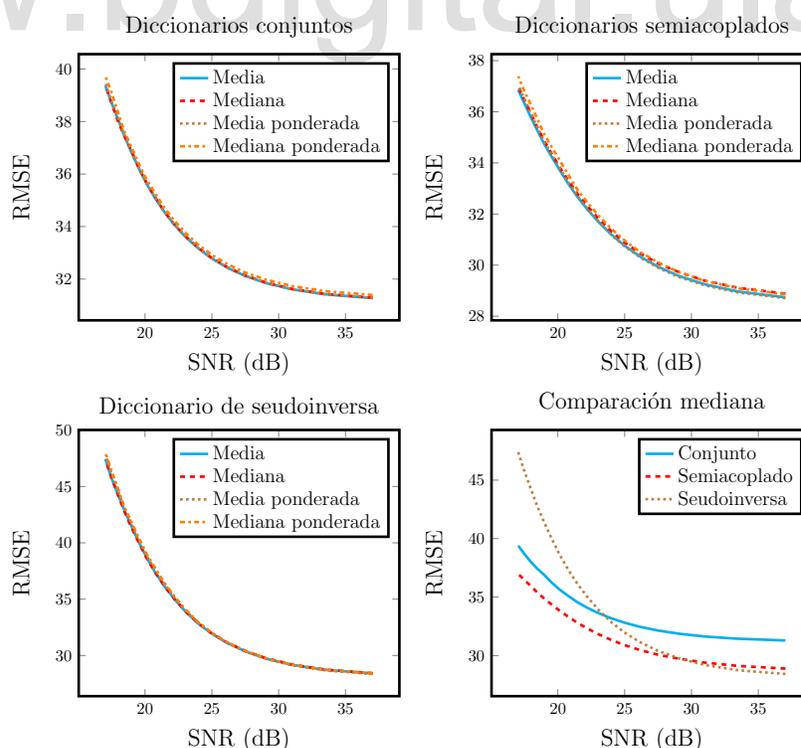
Cuando se aplicó el método de diccionarios de pseudoinversa el comportamiento no varió considerablemente al usar distintos tamaños para factores de escalamiento de 2 y 3. Para un factor de 4 se observa una mejora de 2 cuando se usan bloques de  $3 \times 3$ , con respecto

a los bloques de  $4 \times 4$  esta mejora va decreciendo junto con la SNR hasta que para 20 dB se igualan y luego el uso de bloques  $4 \times 4$  asegura una mayor robustez.

El hecho de que bloques más pequeños arrojen mejores resultados se justifica en las características de las imágenes satelitales, que concentran información con mucho detalle en pequeñas regiones, el uso de bloques pequeños permite estimar esa información de manera más satisfactoria, además, bloques más grandes conllevan a un mayor tiempo de cómputo, concluyendo así que es más factible usar dimensiones de bloque comprendidas entre  $3 \times 3$  y  $5 \times 5$ .

#### 4.2.4 Rendimiento variando el filtro de reconstrucción

Las siguientes características mostradas en las Fig. 4.9, Fig. 4.10 y Fig. 4.11 se obtuvieron usando distintos filtros de reconstrucción, usando la interpolación bicúbica para el parámetro de afinidad, evaluando su desempeño para varios factores de escalamiento en función del RMSE, y haciendo una comparación (mostrada en cada figura) entre métodos para los otros 3 tipos de filtros.



**Figura 4.9. RMSE usando bloques de  $4 \times 4$  y factor de escalamiento = 2.**

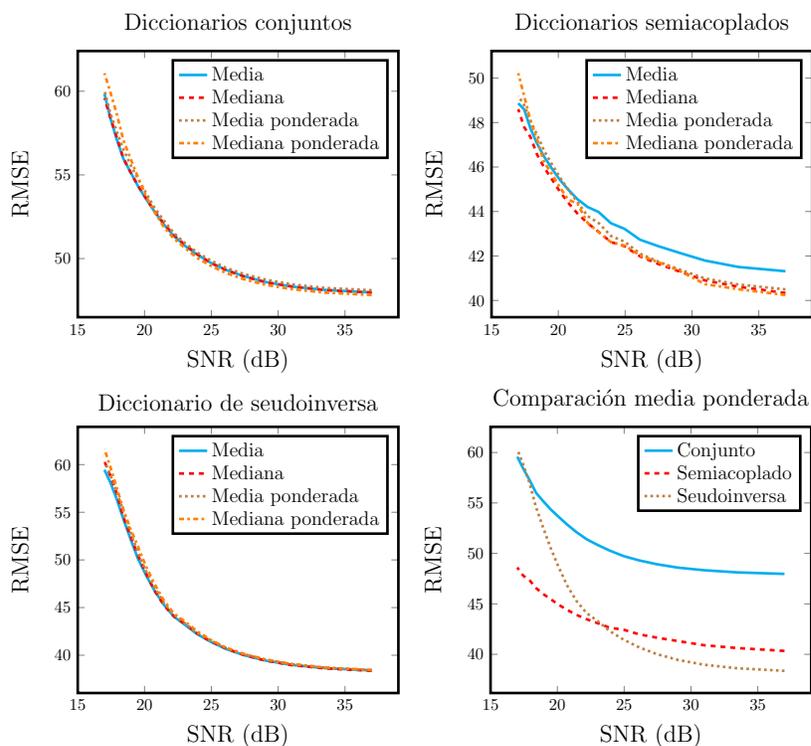


Figura 4.10. RMSE usando bloques de  $3 \times 3$  y factor de escalamiento = 3.

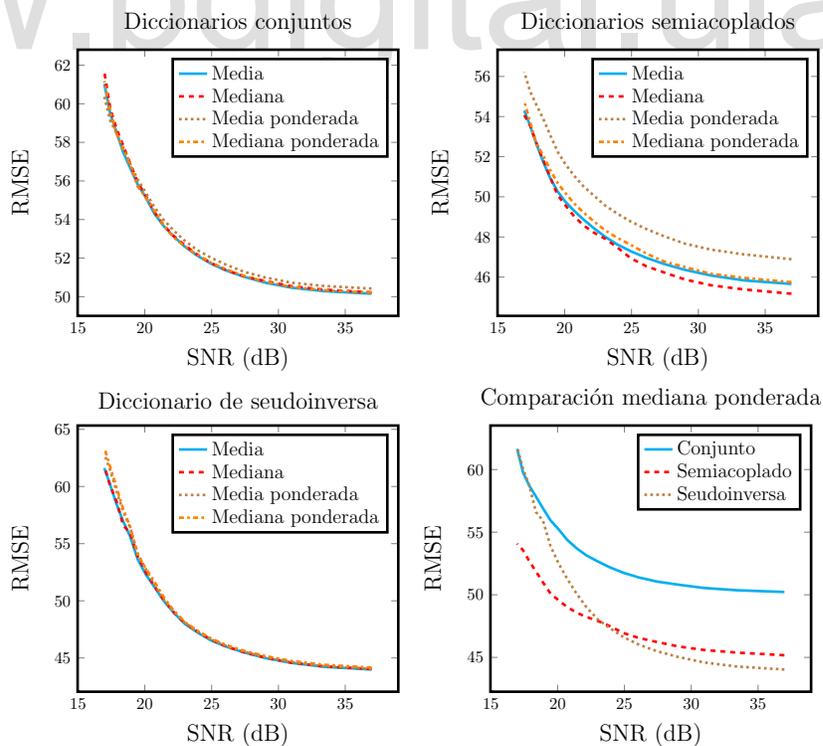


Figura 4.11. RMSE usando bloques de  $3 \times 3$  y factor de escalamiento = 4.

El uso de distintos filtros no incide significativamente en la reconstrucción de la imagen,

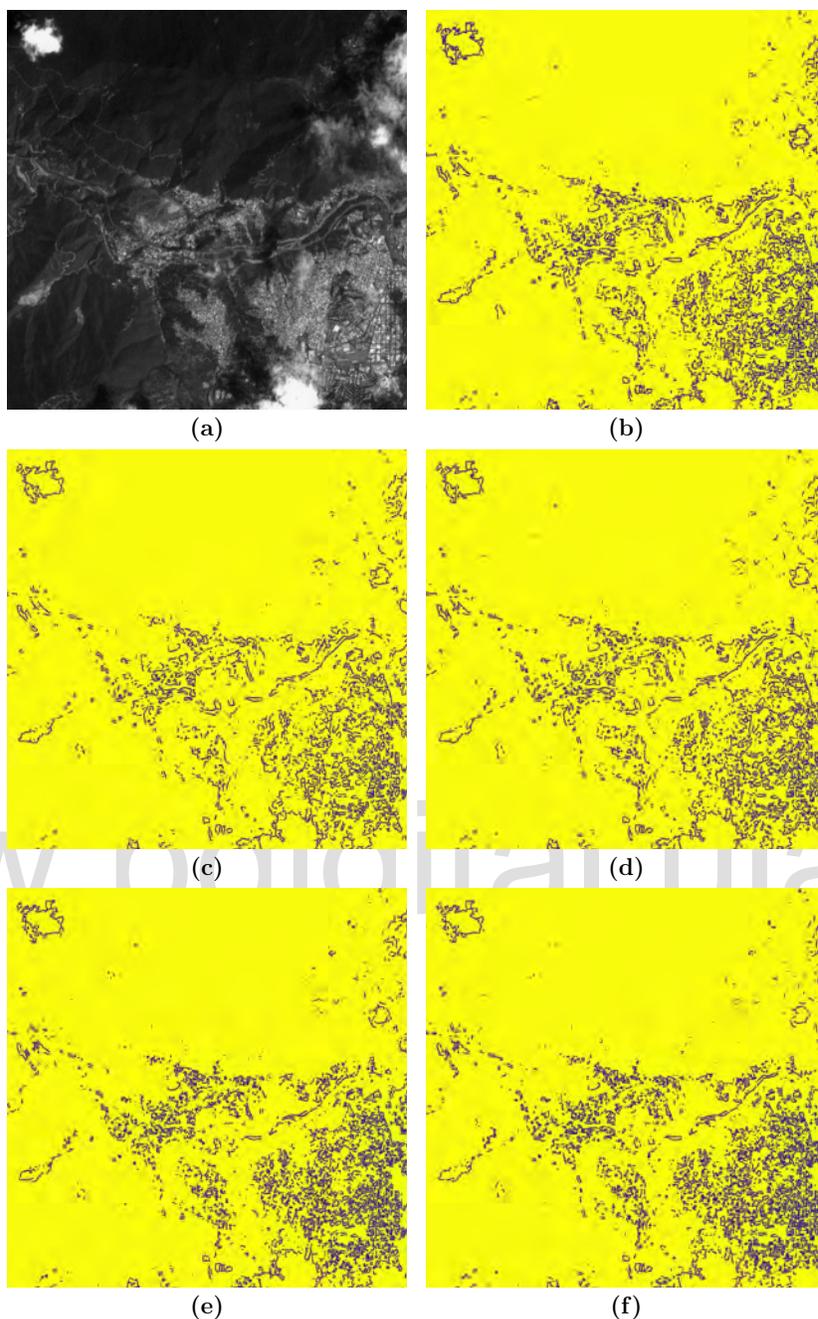
esto se debe, a que la variación entre píxeles para una posición determinada es baja, tendiendo a un único valor sin importar el estimador. Cuando se comparan los 3 métodos entre sí, el método de mayor calidad es el de diccionarios de pseudoinversa con una mejora de cercana a las 2 puntos en función del RMSE para factores de escalamiento de 3 y 4, para un factor de 2 tiene un desempeño igual al método de diccionarios semiacoplados cuando la imagen está limpia. Sin embargo este método es el menos robusto dejando generando la peor reconstrucción cuando la SNR es menor a 20 dB.

#### 4.2.5 Detección de bordes como medida de rendimiento

Al aplicar el detector de bordes (*sobel*) a las imágenes de alta resolución, se obtuvieron los resultados mostrados en la Fig. 4.12, donde la Fig 4.12(a) es la imagen de baja resolución, la Fig 4.12(b) es el resultado del usando interpolación polifásica, con un 3,62 % de bordes detectados, la Fig 4.12(c) es el resultado usando interpolación bicúbica, con un 2,81 % de bordes detectados, la Fig 4.12(d) es el resultado usando diccionarios conjuntos, con un 3,08 % de bordes detectados, la Fig 4.12(e) es el resultado usando diccionarios semiacoplados, con 3,4 % de bordes detectados y la Fig 4.12(f) es el resultado usando diccionarios de pseudoinversa, con un 3,56 % de bordes detectados. Se observa una mejora con respecto a la interpolación bicúbica cuando se emplean diccionarios semiacoplados o de pseudoinversa, mas no superan el realce que hace la interpolación polifásica, dado que el principal atributo de esta técnica es realzar los bordes [17].

### 4.3 ANÁLISIS CUALITATIVO

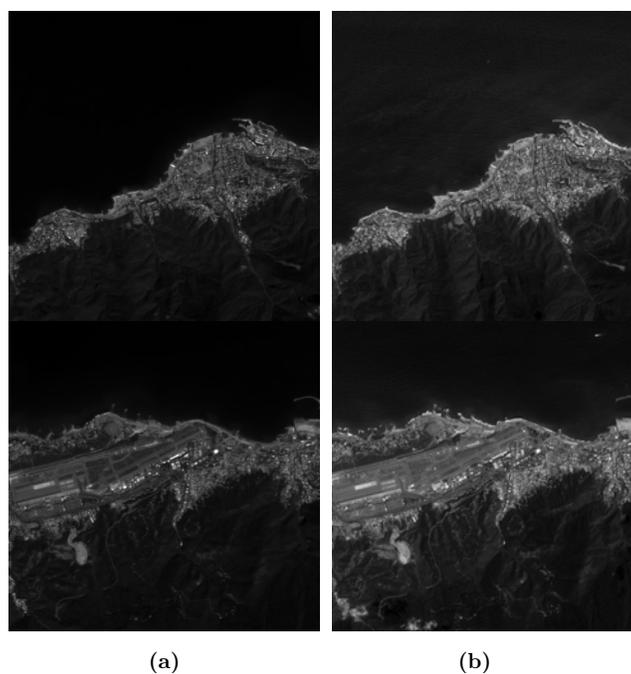
Este análisis se realizó con el fin de darle un sentido práctico a las técnicas desarrolladas hasta ahora, aprovechando las características de los datos. Se seleccionó una imagen multiespectral y una de barrido ancho que contiene la región capturada por la multiespectral, luego en esas 2 imágenes se seleccionaron 2 regiones idénticas mostradas en la Fig. 4.13; donde la Fig. 4.13(a) es una sección redimensionada de “La Guaira” y el aeropuerto internacional de Maiquetía “Simón Bolívar”, captadas por la cámara WMC el 17/11/2013 de resolución 20 m/píxel. La Fig. 4.13(b) Sección de “La Guaira” y el aeropuerto internacional de Maiquetía “Simón Bolívar”, captadas por la cámara MSS el 21/02/2014 de resolución



**Figura 4.12. Resultados de la detección de bordes.**

10 m/píxel. Dado que las resoluciones de estas imágenes son 10 m/píxel y 16 m/píxel, es necesario redimensionar la segunda de manera tal que exista un factor entero entre las resoluciones, llevándola así a 20 m/píxel.

En la Fig. 4.14 se muestran los resultados de hacer un acercamiento a 2 regiones de una imagen WMC en la que se usa la banda cercana al infrarrojo; aplicando los distintos métodos que usan representación poco densa. Las Fig. 4.14(a) y Fig. 4.14(d) corresponden



**Figura 4.13. Imágenes de entrenamiento para los diccionarios del análisis cualitativo.**

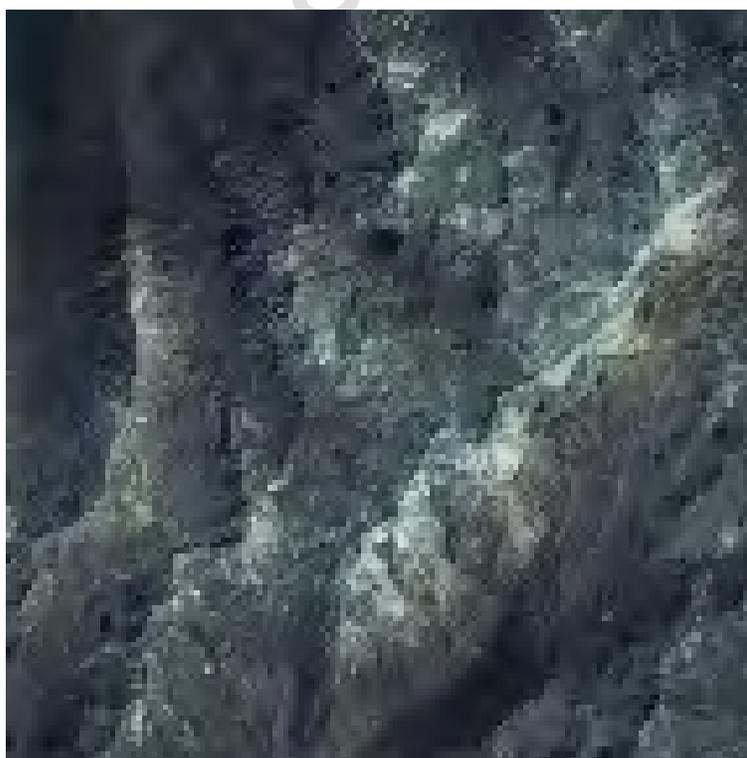
al uso de diccionarios conjuntos, las Fig. 4.14(b) y Fig. 4.14(e) corresponden al uso de diccionarios semiacoplados, las Fig. 4.14(c) y Fig. 4.14(f) corresponden al uso de diccionarios de pseudoinversa.



**Figura 4.14. Resultados del análisis cualitativo.**

Se observa una mejora significativa cuando se emplean diccionarios conjuntos contrariamente a los resultados mostrados en la sección anterior, esto se debe, a que el uso de diccionarios conjuntos permite introducir una imagen de mayor resolución como restricción al problema de optimización, convirtiéndolo en un método recursivo iterativo que debe ser supervisado para generar resultados satisfactorios y que se adecuan de mejor manera a las características de la imagen.

En un segundo experimento se usó la información de una misma región en una imagen pancromática y otra multiespectral para entrenar los diccionarios, dada la cantidad de detalles que contiene la imagen pancromática como, la textura del terreno, una definición mucho mas precisa de los detalles en los edificios, entre otras cosas, es necesario degradarla a fin de que tenga semejanza con la imagen de baja resolución. En este sentido la imagen de la Fig.4.16(a) fue degradada usando un filtro promediador de ventana 5, para usarla junto con la imagen de la Fig.4.16(b) como imagenes de entrenamiento. En la segunda etapa de este experimento se usaron los diccionarios obtenidos para ampliar la imagen de la Fig. 4.15(a) por un factor de 4. Los resultados se muestran en las Fig. 4.17, Fig. 4.18 y Fig. 4.19.



**Figura 4.15.** Sierra nevada captada por la cámara PMC 10 m/píxel.



(a)



(b)

Figura 4.16. Imágenes de “La guaira” usadas para entrenar.

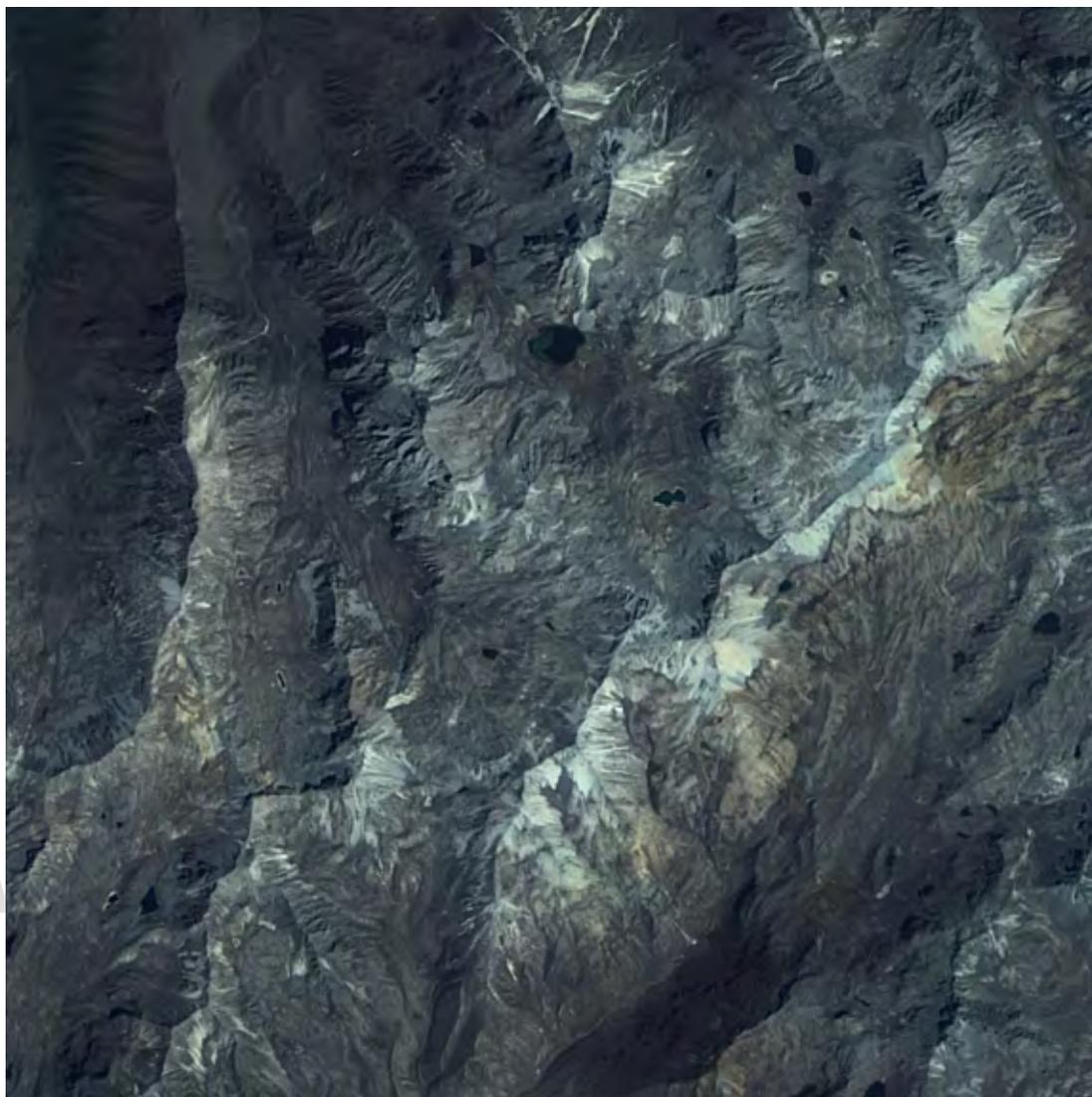
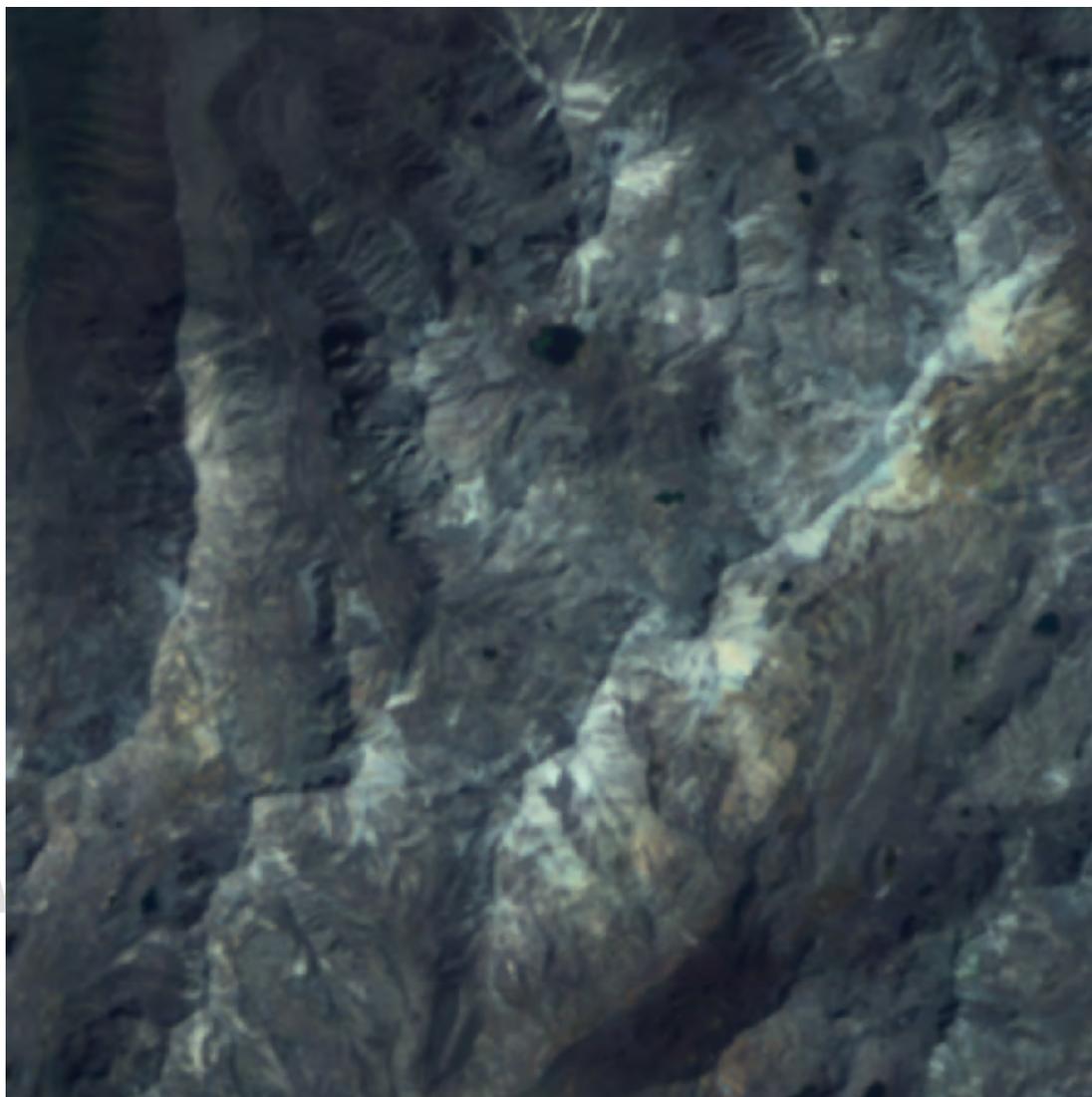
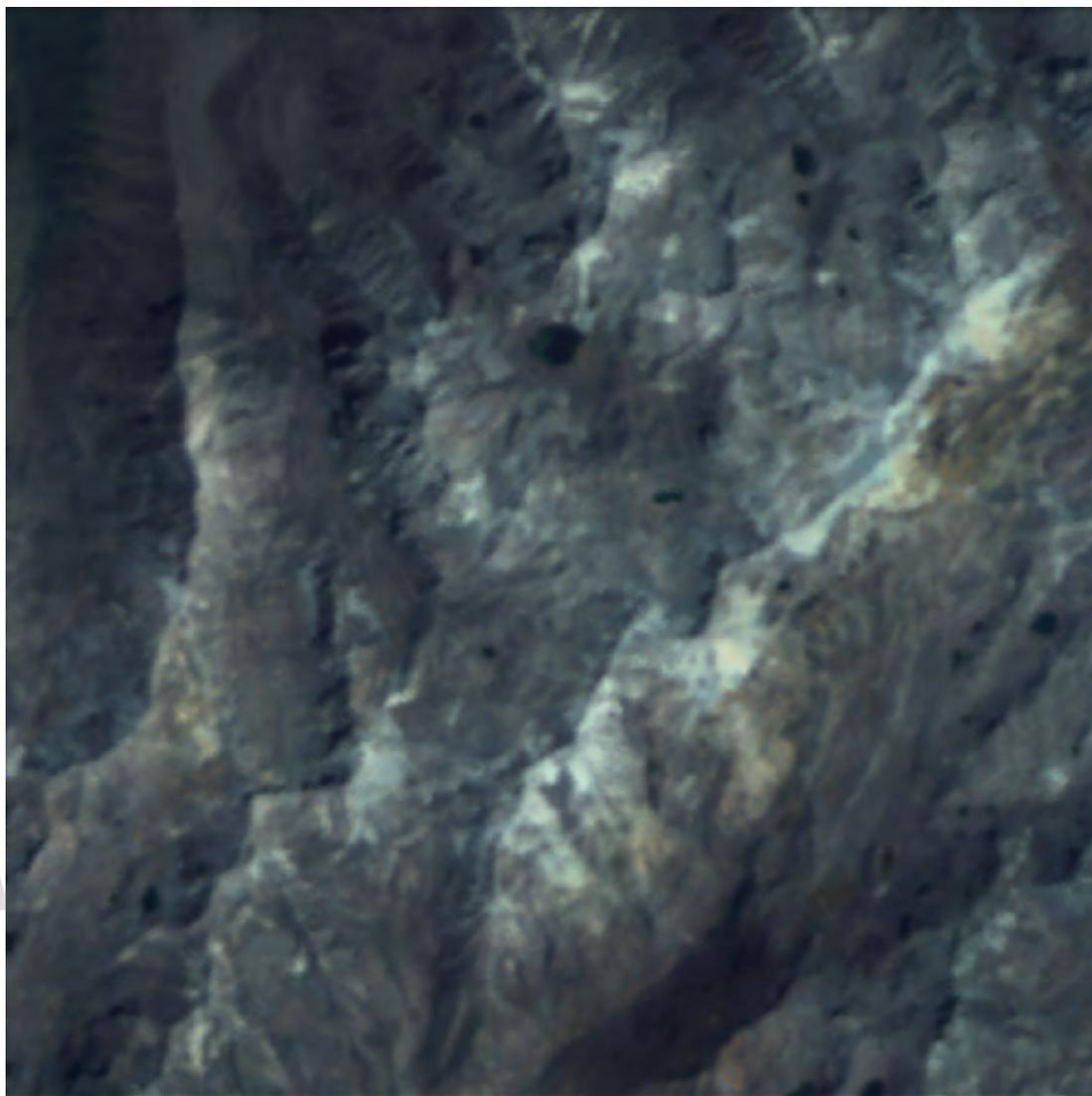


Figura 4.17. Sierra nevada ampliada por un factor de 4 usando el método de diccionarios conjuntos.



**Figura 4.18.** Sierra nevada ampliada por un factor de 4 usando el método de diccionarios semiacoplados.



**Figura 4.19. Sierra nevada ampliada por un factor de 4 usando el método de diccionarios de pseudoinversa.**

En este experimento se comprueba nuevamente el uso de diccionarios conjuntos de manera iterativa conduce a los mejores resultados, produciendo un realce en los detalles de la cordillera, superior a cuando se emplea diccionarios semiacoplados y de pseudoinversa. Además se comprueba la capacidad de generalización de los diccionarios para trabajar con distintos tipos de imágenes que no se encuentran en una misma región, y con características geográficas diferentes.

## CONCLUSIONES

A partir de los resultados obtenidos y de la información recavada a lo largo de este trabajo se llegó a las siguientes conclusiones:

- Los algoritmos presentados en esta investigación, muestran una mejora en la reconstrucción con respecto a otras técnicas de superresolución como la de interpolación de entre 0,5 dB y 2 dB en función de la PSNR, sin embargo, la calidad de reconstrucción demostró ser inversamente proporcional a la robustez, es decir, mientras mejor sea la calidad del método usado mayor será la sensibilidad a contaminaciones poniendo como valor límite una SNR de 22 dB, lo cual limita en cierta manera el alcance de las técnicas desarrolladas. Este comportamiento está estrechamente ligado al algoritmo de reconstrucción, en este caso OMP, y a las características de las imágenes satelitales que poseen muchos más detalles que imágenes convencionales, lo que implica una alta correlación con las componentes de alta frecuencia del ruido que no logran ser discriminadas por el algoritmo.
- El algoritmo K-SVD usado para entrenamiento demostró ser una buena alternativa para conseguir el par de diccionarios usados por cualquiera de las técnicas de superresolución, es conveniente usar pocos átomos como restricción de entrenamiento para facilitar el acoplamiento entre los espacios vectoriales de baja y alta resolución. Debe existir una relación entre el acoplamiento de los diccionarios y su capacidad de generalización, ya que, si las representaciones son iguales para un grupo de imágenes de entrenamiento cualquier distorsión en la imagen objetivo será magnificada en su equivalente de alta, como se demostró en los resultados cualitativos, donde la diferencia de la imagen objetivo con respecto a las de entrenamiento no pudo ser bien modelada por los diccionarios semiacoplados o de pseudoinversa, pero sí por el uso de diccionarios conjuntos.
- Para el caso particular de las imágenes tomadas por satélite “Miranda” el sistema de adquisición es bastante robusto con SNR superior a los 35 dB, permitiendo que las

técnicas desarrolladas sean convenientes, debido a que, su robustez no está comprometida. El algoritmo de *backprojection*, presenta una buena alternativa para mejorar la calidad visual de las imágenes en caso de que esta no aporte la información necesaria, debido a su capacidad de eliminar los efectos de desenfoco, ofreciendo una mejora de hasta 2 dB en función de la PSNR en el mejor de los casos.

- Los algoritmos presentados ofrecen una amplia variedad de parámetros, que permiten generar distintos resultados a partir una misma imagen, además es posible combinar métodos de varias maneras, abriendo la posibilidad de particularizar dichos parámetros para mejorar la calidad de la reconstrucción.

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

## RECOMENDACIONES

Las técnicas de superresolución y la teoría de representación poco densa son áreas de investigación que continúan creciendo, por lo tanto se hacen las siguientes recomendaciones para futuras investigaciones:

- Proponer algoritmos de entrenamiento de diccionarios que se adapten particularmente al problema de superresolución con el fin de generar acoplamientos más robustos entre los diccionarios, usando como restricción información a priori sobre el modelo de observación.
- Acelerar las técnicas de reconstrucción para hacer más atractivos los métodos de representación poco densa que han demostrado mejorar los resultados de técnicas previas.
- Estudiar el comportamiento de las técnicas desarrolladas usando algoritmos más robustos que mejoren la calidad de la reconstrucción ante contaminaciones.
- Probar con imágenes capturadas por otros satélites que poseen mejor resolución espacial a fin de mejorar la información aportada por el satélite VRSS-1.

# APÉNDICE A

## CÁLCULO DE LA MEDIANA PONDERADA

*En este apéndice se explica como calcular la mediana ponderada cuando las ponderaciones de las muestras son números reales, siguiendo la metodología desarrollada en [17].*

La operación mediana ponderada denotado como:

$$\hat{x} = \text{MEDIANA}[w_1 \diamond x_1, w_2 \diamond x_2, w_3 \diamond x_3, \dots, w_n \diamond x_n]$$

Donde se cumple,  $\forall x_i \in \mathbb{R}$  y  $\forall w_i \in \mathbb{R}$ , se puede calcular de la siguiente manera:

1. Calcular el umbral  $T_0 = \frac{1}{2} \sum_{i=1}^n |w_i|$ .
2. Ordenar las muestras multiplicadas por el signo de su ponderación,  $\text{sgn}(w_i)x_i$ .
3. Sumar la magnitud de las ponderaciones correspondientes a las muestras ordenadas, comenzado por el máximo valor y continuando en orden decreciente.
4. La salida  $\hat{x}$  es la muestra cuya sumatoria acumulada de la magnitud de las ponderaciones se hace  $\geq T_0$ .

## REFERENCIAS

- [1] J. B. Campbell, *Introduction to remote sensing*. CRC Press, 2002.
- [2] “Image resolution”, [https://en.wikipedia.org/wiki/Image\\_resolution](https://en.wikipedia.org/wiki/Image_resolution), consultado: (15/10/2015).
- [3] O. Jerick, “Restauración de imágenes usando técnicas de restauración de sensado comprimido”, Trabajo de Grado, Universidad de Los Andes, Noviembre 2012.
- [4] ABAE, “Tecnología espacial en venezuela satélite miranda (VRSS-1)”, Mayo 2014.
- [5] G. Juan, “Desarrollo de herramientas computacionales de detección de objetos de interés en imágenes satelitales usando representación poco densa de señales en diccionarios redundantes”, Trabajo de Grado, Universidad de Los Andes, Diciembre 2015.
- [6] “Venezuelan remote sensing satellite (VRSS-1)”, <https://directory.eoportal.org/web/eoportal/satellite-missions/v-w-x-y-z/vrss-1#mission-status>, consultado: (12/10/2015).
- [7] N. Waltham, “Ccd and cmos sensors”, en *Observing photons in space*. Springer, 2013, pp. 423–442.
- [8] P. Milanfar, *Super-resolution imaging*. CRC Press, 2010.
- [9] S. C. Park, M. K. Park, y M. G. Kang, “Super-resolution image reconstruction: a technical overview”, *Signal Processing Magazine*, vol. 20, no. 3, pp. 21–36, 2003.
- [10] T. R.Y. y H. T.S., “Multipleframe image restoration and registration”, *Advances in Computer Vision and Image Processing*, pp. 317–339, 1984.
- [11] S. Kim, N. K. Bose, y H. Valenzuela, “Recursive reconstruction of high resolution image from noisy undersampled multiframe”, *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 38, no. 6, pp. 1013–1027, 1990.

- [12] S. P. Kim y W.-Y. Su, "Recursive high-resolution reconstruction of blurred multiframe images", *Image Processing, IEEE Transactions on*, vol. 2, no. 4, pp. 534–539, 1993.
- [13] N. Bose, H. Kim, y H. Valenzuela, "Recursive implementation of total least squares algorithm for image reconstruction from noisy, undersampled multiframe", en *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, vol. 5. IEEE, 1993, pp. 269–272.
- [14] S. Rhee y M. G. Kang, "Discrete cosine transform based regularized high-resolution image reconstruction algorithm", *Optical Engineering*, vol. 38, no. 8, pp. 1348–1356, 1999.
- [15] M. Protter y M. Elad, "Super resolution with probabilistic motion estimation", *Image Processing, IEEE Transactions on*, vol. 18, no. 8, pp. 1899–1904, 2009.
- [16] S. Lertrattanapanich y N. K. Bose, "High resolution image formation from low resolution frames using delaunay triangulation", *Image Processing, IEEE Transactions on*, vol. 11, no. 12, pp. 1427–1441, 2002.
- [17] G. R. Arce, J. Paredes, y J. Mullan, "Nonlinear filtering for image analysis and enhancement", *The Essential Guide to Image Processing*, pp. 263–291, 2000.
- [18] W. T. Freeman, T. R. Jones, y E. C. Pasztor, "Example-based super-resolution", *Computer Graphics and Applications, IEEE*, vol. 22, no. 2, pp. 56–65, 2002.
- [19] H. Chang, D.-Y. Yeung, y Y. Xiong, "Super-resolution through neighbor embedding", en *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1. IEEE, 2004, pp. I–I.
- [20] J. Yang, J. Wright, T. S. Huang, y Y. Ma, "Image super-resolution via sparse representation", *Image Processing, IEEE Transactions on*, vol. 19, no. 11, pp. 2861–2873, 2010.
- [21] E. W. Cheney y A. K. Cline, "Topics in sparse approximation", 2004.
- [22] K. R. Rao y P. C. Yip, *The transform and data compression handbook*. CRC press, 2000.

- [23] M. Schmidt, “Least squares optimization with l1-norm regularization”, *CS542B Project Report*, 2005.
- [24] R. Tibshirani, “Regression shrinkage and selection via the lasso”, *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.
- [25] R. Rubinstein, M. Zibulevsky, y M. Elad, “Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit”, *CS Technion*, vol. 40, no. 8, pp. 1–15, 2008.
- [26] M. Aharon, M. Elad, y A. Bruckstein, “K-svd: An algorithm for designing overcomplete dictionaries for sparse representation”, *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [27] E. J. Candès *et al.*, “Compressive sampling”, en *Proceedings of the international congress of mathematicians*, vol. 3. Madrid, Spain, 2006, pp. 1433–1452.
- [28] J. Yang, Z. Wang, Z. Lin, S. Cohen, y T. Huang, “Coupled dictionary training for image super-resolution”, *Image Processing, IEEE Transactions on*, vol. 21, no. 8, pp. 3467–3478, 2012.
- [29] S. Wang, L. Zhang, Y. Liang, y Q. Pan, “Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis”, en *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2216–2223.
- [30] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani *et al.*, “Least angle regression”, *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.
- [31] R. Zeyde, M. Elad, y M. Protter, “On single image scale-up using sparse-representations”, en *Curves and Surfaces*. Springer, 2012, pp. 711–730.
- [32] D. L. Donoho, “For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution”, *Communications on pure and applied mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

- [33] A. Flaig, G. R. Arce, y K. E. Barner, “Affine order-statistic filters:”medianization” of linear fir filter”, *Signal Processing, IEEE Transactions on*, vol. 46, no. 8, pp. 2101–2112, 1998.
- [34] M. Irani y S. Peleg, “Improving resolution by image registration”, *CVGIP: Graphical models and image processing*, vol. 53, no. 3, pp. 231–239, 1991.
- [35] M. Elad y M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries”, *Image Processing, IEEE Transactions on*, vol. 15, no. 12, pp. 3736–3745, 2006.

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

## ANEXOS

### Algoritmo *Orthogonal Matching Pursuit*

Implementación computacional del algoritmo para representación poco densa OMP, la función `_omp` resuelve el problema restringido por poca densidad, y `_omp2` resuelve el problema restringido por error con la opción de limitar el número de átomos.

- Paquetes requeridos

- Python 2.7
- numpy

- Parámetros de entrada

- Señal objetivo (`x`)
- Diccionario (`dic`)
- Cantidad máxima de átomos (`maxatoms`)
- Error máximo permitido (`error`)

- Salida

- Vector de representación poco densa con los coeficientes(`coef`)

```

1 import numpy as np
2
3
4 def _omp(dic, x, maxatoms):
5     # Se crean los espacios para almacenar las posiciones y los
6     # valores de los coeficientes, ind y coef respectivamente
7     ind = np.zeros(maxatoms, 'uint64')
8     coef = np.zeros((dic.shape[1], x.shape[1]))
9     # Bucle que recorre cada columna de la señal objetivo
10    for i in range(x.shape[1]):
11        res = np.copy(x[:, i])
12    # Bucle que encuentra la representación poco densa de cada señal
13        for j in range(maxatoms):
14            k = np.argmax(np.abs(dic.T.dot(res)))
15            ind[j] = k
16            alfa = np.linalg.pinv(dic[:, ind[0:j+1]]).dot(x[:, i])
17            res = x[:, i] - dic[:, ind[0:j+1]].dot(alfa)
18        coef[ind, i] = alfa
19    return coef
20
21
```

```

22 def _omp2(dic, x, error, maxatoms=None):
23     if maxatoms is None:
24         maxatoms = x.shape[0]
25     coef = np.zeros((dic.shape[1], x.shape[1]))
26     # Bucle que recorre cada columna de la señal objetivo
27     for i in range(x.shape[1]):
28         res = np.copy(x[:, i])
29     # Bucle que encuentra la representación sparse de cada señal
30         j = 0
31         ind = np.array([], 'uint64')
32         alfa = 0
33         while(np.linalg.norm(res) > error and j < maxatoms):
34             ind = np.concatenate((ind, np.array([0], 'uint64')))
35             k = np.argmax(np.abs(dic.T.dot(res)))
36             ind[j] = k
37             alfa = np.linalg.pinv(dic[:, ind[0:j+1]]).dot(x[:, i])
38             res = x[:, i] - dic[:, ind[0:j+1]].dot(alfa)
39             j += 1
40         coef[ind, i] = alfa
41     return coef

```

### Algoritmo *K-times Singular Value Decomposition*

Implementación computacional del algoritmo para entrenamiento de diccionarios K-SVD, la función recibe como parámetro de entrada un diccionario<sup>1</sup> con las variables de entrada que requiere el algoritmo

- Paquetes requeridos

- Python 2.7
- numpy
- omp

- Parámetros de entrada

- Señales de entrenamiento (`signal`)
- Diccionario (`dict`)
- Cantidad máxima de átomos (`maxatoms`)
- Error máximo permitido (`error`)
- Opcion de retornar la matriz de coeficientes poco densos(`gamma`)
- Numero de iteraciones(`iternum`)

- Salida

- Diccionario entrenado (`dict`)
- Matriz de representación poco densa con los coeficientes(`coef`)

---

<sup>1</sup>Este diccionario es el tipo de dato de Python, donde cada campo corresponde a las variables de entrada

```

1 import numpy as np
2 import scipy.linalg as LA
3 from .omp import *
4
5
6 def ksvd(args):
7     if args['dict'].shape[1] > args['signal'].shape[1]:
8         raise TypeError('Numero de senales de entrenamiento es menor que la\
9             cantidad de atomos')
10    # Seleccion aleatoria de los atomos
11    indices = np.random.permutation(args['dict'].shape[1])
12    for i in range(args['iternum']):
13        coef = OMP(args)
14        replaced = 0
15        for j in range(args['dict'].shape[1]):
16            if np.linalg.norm(coef[indices[j], :], ord=0) == 0:
17                replaced += 1
18                perm = np.random.permutation(args['signal'].shape[1])
19                perm = np.compress(perm < 5000, perm)
20                E = np.sum(args['signal'][:, perm] - args['dict'].dot(coef[:,
21                    perm]**2), axis=0)
22                condition = np.linalg.norm(E.reshape(1, -1), axis=0) > 0.1
23                E = np.compress(E, condition)
24                d = np.argmax(-1.0 * E)
25                atom = args['signal'][:, d] /\
26                    np.linalg.norm(args['signal'][:, d])
27                args['dict'][:, indices[j]] = atom
28                coef[:, indices[j]] = np.zeros(coef[:, indices[j]].shape)
29                # Seleccion de las senales que utilizan el atomo
30                pos = np.where(coef[indices[j], :] != 0)[0]
31                if(pos.shape != (0, )):
32                    # Calculo del error sin la contribucion del atomo
33                    Ek = args['signal'][:, pos] - args['dict'].dot(coef[:, pos]) +\
34                        args['dict'][:, indices[j]:indices[j] + 1].dot(coef[
35                            indices[j]:indices[j] + 1, pos])
36                    if Ek.shape[1] > 900:
37                        U, s, V = svds(Ek)
38                        args['dict'][:, indices[j]] = U[:, 0] / LA.norm(U)
39                        coef[indices[j], pos] = V[:, 0] * s * LA.norm(U)
40                    else:
41                        U, s, V = svds2(Ek)
42                        args['dict'][:, indices[j]] = U[:, 0]
43                        coef[indices[j], pos] = V[:, 0] * s[0]
44    del(args['signal'])
45    if 'gamma' not in args:
46        return args['dict']
47    else:
48        return args['dict'], coef
49
50
51 def svds(matrix):
52    epsilon = matrix.shape[1] / 1000
53    v = np.ones((matrix.shape[1], 1))
54    v_ant = v.copy()
55    s = 1
56    y = matrix.T.dot(matrix.dot(v))
57    s = y.max()
58    v = y / s
59    error = LA.norm(matrix.T.dot(matrix.dot(v)) - s*v, ord=1)
60    for i in range(15):
61        error_ant = error.copy()
62        v_ant = v.copy()
63        y = matrix.T.dot(matrix.dot(v))
64        s = y.max()
65        v = y / s
66        error = LA.norm(matrix.T.dot(matrix.dot(v)) - s * v, ord=1)
67        if error < epsilon:
68            break

```

```
69     v = v / LA.norm(v)
70     s = np.sqrt(s)
71     u = matrix.dot(v) / s
72     s = s * LA.norm(u)
73     u = u / LA.norm(u)
74     return u, s, v
75
76
77 def svds2(a):
78     if a.shape[0] > a.shape[1]:
79         A = a.T.dot(a)
80         s, v = LA.eigh(A, eigvals=(A.shape[0] - 1, A.shape[0] - 1))
81         s = np.sqrt(s)
82         u = a.dot(v)/s
83     else:
84         A = a.dot(a.T)
85         s, u = LA.eigh(A, eigvals=(A.shape[0] - 1, A.shape[0] - 1))
86         s = np.sqrt(s)
87         v = u.T.dot(a) / s
88         v = v.T
89     return u, s, v
```

www.bdigital.ula.ve