



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA  
Instituto Universitario de Microelectrónica Aplicada  
Sistemas de información y Comunicaciones

# Máster en Tecnologías de Telecomunicación



## Trabajo Fin de Máster

### ALGORITMO DE LINEAR-UNMIXING BASADO EN LA BÚSQUEDA DE ABUNDANCIAS NEGATIVAS PARA IMÁGENES HIPERESPECTRALES

Autor: Rubén Marrero Gómez  
Tutor(es): Roberto Sarmiento Rodríguez  
Gustavo Marrero Callicó  
Fecha: Julio 2012



t +34 928 451 086 | iuma@iuma.ulpgc.es  
f +34 928 451 083 | www.iuma.ulpgc.es

Campus Universitario de Tafira  
35017 Las Palmas de Gran Canaria





UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA  
Instituto Universitario de Microelectrónica Aplicada  
Sistemas de información y Comunicaciones

# Máster en Tecnologías de Telecomunicación



## Trabajo Fin de Máster

### ALGORITMO DE LINEAR-UNMIXING BASADO EN LA BÚSQUEDA DE ABUNDANCIAS NEGATIVAS PARA IMÁGENES HIPERESPECTRALES

#### HOJA DE FIRMAS

<b>Alumno/a:</b>	Rubén Marrero Gómez	Fdo.:
<b>Tutor/a:</b>	Roberto Sarmiento Rodríguez	Fdo.:
<b>Tutor/a:</b>	Gustavo Marrero Callicó	Fdo.:

**Fecha: Julio 2012**



t +34 928 451 086 | iuma@iuma.ulpgc.es  
f +34 928 451 083 | www.iuma.ulpgc.es

Campus Universitario de Tafira  
35017 Las Palmas de Gran Canaria





# Máster en Tecnologías de Telecomunicación



## Trabajo Fin de Máster

### ALGORITMO DE LINEAR-UNMIXING BASADO EN LA BÚSQUEDA DE ABUNDANCIAS NEGATIVAS PARA IMÁGENES HIPERESPECTRALES

## HOJA DE EVALUACIÓN

Calificación: .....

**Presidente**      Javier A. García García      Fdo.:

**Secretario**      Fernando de la Puente Arrate      Fdo.:

**Vocal**      C. Javier Sosa González      Fdo.:

**Fecha: Julio 2012**





# Contenido

<b>1</b>	<b>Capítulo 1</b>	<b>9</b>
1.1	Introducción	9
1.2	Objetivos	12
<b>2</b>	<b>Capítulo 2</b>	<b>13</b>
2.1	Introducción	13
2.2	Antecedentes Teóricos	13
2.2.1	Definición de Espacio Vectorial	14
2.2.2	Definición de Combinación Lineal	14
2.2.3	Definición de Independencia Lineal	14
2.2.4	Definición de Base	15
2.2.5	Definición de Dimensión	15
2.2.6	Pseudoinversa de Moore-Penrose	15
2.2.7	Mínimos Cuadrados	15
2.2.8	Símplex	16
2.3	Imagen Hiperespectral	17
2.4	El Problema de la Mezcla	19
2.4.1	Tipos de Mezcla	19
2.4.2	Modelos	22
2.5	Algoritmos	25
2.5.1	Cadena Típica de Linear Spectral Unmixing	25
2.5.2	Estimación del número de Endmembers	26
2.5.3	Extracción de Endmembers	27
2.5.4	Cálculo de Abundancias	32

<b>3</b>	<b>Capítulo 3</b> .....	<b>35</b>
3.1	Introducción.....	35
3.2	Hipótesis .....	35
3.3	Estructuras de Datos.....	37
3.3.1	Image.....	37
3.3.2	p.....	38
3.3.3	E_index.....	38
3.3.4	is_candidate .....	38
3.3.5	next_j_index .....	39
3.3.6	max_pixels_in_simplex.....	39
3.4	Algoritmo .....	40
3.4.1	Solución frente al Ruido.....	47
<b>4</b>	<b>Capítulo 4</b> .....	<b>51</b>
4.1	Introducción.....	51
4.2	Biblioteca de imágenes sintéticas .....	51
4.3	Evolución de Píxeles Descartados Durante la Ejecución del Algoritmo Propuesto.....	77
4.4	Métricas cuantitativas.....	55
4.5	Diseño del Experimento .....	56
4.6	Comparativa.....	57
4.6.1	Número de endmembers estimado.....	59
4.6.2	Ángulo espectral entre los endmembers encontrados y los reales... ..	65
4.6.3	RMSE de las abundancias calculadas.....	69
4.6.4	Tiempo de ejecución.....	72
<b>5</b>	<b>Capítulo 5</b> .....	<b>79</b>

---

5.1	Conclusiones.....	79
5.2	Líneas Futuras .....	80
<b>6</b>	<b>Bibliografía.....</b>	<b>83</b>

## INDICE DE FIGURAS

Figura 1.- Cubo hiperespectral .....	11
Figura 2.- Mínimos Cuadrados .....	16
Figura 3.- Símpex en 3 dimensiones .....	17
Figura 4.- Cubo hiperespectral y correspondencia vectoral .....	18
Figura 5.- (a) Imagen multiespectral (6 bandas) (b) e imagen hiperespectral (224 bandas) .....	19
Figura 6.- Tipos de píxel en imágenes hiperespectrales .....	20
Figura 7.- Mezcla macroscópica.....	21
Figura 8.- Mezcla íntima.....	22
Figura 9.- Modelo lineal de mezcla.....	23
Figura 10.- Diagrama de dispersión bidimensional .....	24
Figura 11.- Modelo no lineal de mezcla.....	25
Figura 12.- Cadena típica de <i>Linear Spectral Unmixing</i> .....	26
Figura 13.- Representación bidimensional del funcionamiento del algoritmo PPI	28
Figura 14.- Posible situación en el algoritmo N-FINDR .....	30
Figura 15.- Representación Bidimensional del algoritmo VCA.....	32
Figura 16.- Esquema del algoritmo propuesto.....	37
Figura 17.- Matriz Image .....	38
Figura 18.- Vector <i>is_candidate</i> .....	39
Figura 19.- Flujograma del algoritmo desarrollado .....	41
Figura 20.- Flujograma del bloque <i>Detección de enganche</i> .....	46
Figura 21.- Espectros de los endmembers empleados .....	53
Figura 22.- Espectros de una imagen sin ruido de tamaño 100x100 con 5 endmembers .....	54

---

Figura 23.-Espectros de una imagen con ruido de tamaño 100x100 con 5 endmembers .....	55
Figura 24.- Esquema de los experimentos .....	57
Figura 25.- <i>Reference Linear Spectral Unmixing Chain</i> .....	57
Figura 26.- Estimación del número de <i>endmembers</i> (30x30).....	60
Figura 27.-Estimación del número de <i>endmembers</i> (100x100).....	61
Figura 28.- Estimación del número de <i>endmembers</i> (150x150).....	62
Figura 29.-Errores en la estimación del número de endmembers. (30x30).....	63
Figura 30.-Errores en la estimación del número de <i>endmembers</i> (100x100).....	63
Figura 31.-Errores en la estimación del número de endmembers. (150x150).....	64
Figura 32.- Media de los ángulos espectrales error (30x30) .....	65
Figura 33.-Media de los ángulos espectrales error (30x30) .....	66
Figura 34.- Media de los ángulos espectrales error (100x100) .....	67
Figura 35.- Media de los ángulos espectrales error (150x150) .....	68
Figura 36.- RMSE de las abundancias calculadas (30x30).....	69
Figura 37.- RMSE de las abundancias calculadas (100x100).....	70
Figura 38.- RMSE de las abundancias calculadas (150x150).....	71
Figura 39.- Tiempos de ejecución (30x30) .....	73
Figura 40.- Tiempos de ejecución (100x100) .....	74
Figura 41.- Tiempos de ejecución (150x150) .....	75
Figura 42.- Detalle de los tiempos de ejecución (150x150).....	76
Figura 43.- Evolución del máximo número de píxeles dentro del simplex y del número de píxeles descartados sin ruido .....	77
Figura 44.- Evolución del máximo número de píxeles dentro del simplex y del número de píxeles descartados con SNR=40dB.....	78

---

## INDICE DE TABLAS

Tabla 1.- Resumen de los errores promedio cometidos en la estimación del número de <i>endmembers</i> .....	64
Tabla 2.- Resumen de los ángulos espectrales promedio obtenidos .....	68
Tabla 3.- Resumen del RMSE promedio de las abundancias obtenidos.....	72
Tabla 4.- Resumen de los tiempos de ejecución.....	76

---

## **ACRÓNIMOS**

<b>ANC</b>	Abundance Nonnegativity Constraint
<b>APS</b>	Alternating Projected Subgradients
<b>ASC</b>	Abundance Sum-to-one Constraint
<b>ASI</b>	Agenzia Spaziale Italiana
<b>AVIRIS</b>	Airborn Visible-Infra Red Imaging Spectrometer
<b>CCA</b>	Convex Cone Analysis
<b>DLR</b>	German Aerospace Center
<b>ESA</b>	European Space Agency
<b>FCLS</b>	Fully Constrained Least Squares
<b>GFZ</b>	German Research Centre for Geosciences
<b>HSI</b>	Hyperspectral Imaging
<b>HySime</b>	Hyperspectral signal Subspace identification by minimum error
<b>ICE</b>	Iterated Constrained Endmembers
<b>JBA</b>	Joint Bayesian Approach
<b>JPL</b>	Jet Propulsion Laboratory
<b>LSU</b>	Linear Spectral Unmixing
<b>MNF</b>	Maximum Noise Fraction
<b>MVT</b>	Minimum Volume Transform
<b>NASA</b>	National Aeronautics and Space Administration
<b>NCLS</b>	Nonnegatively Constrained Least Squares
<b>N-FINDR</b>	N-finder
<b>NMF</b>	Nonnegative Matrix Factorization
<b>PCA</b>	Principal Component Analysis
<b>PPI</b>	Pixel Purity Index
<b>RMSE</b>	Root Mean Square Error
<b>SNR</b>	Signal Noise Ratio
<b>VCA</b>	Vertex Component Analysis
<b>VD</b>	Virtual Dimensionality



# Capítulo 1

## Antecedentes y Objetivos

---

### **1.1 Introducción**

En los años setenta existían sensores multiespectrales sensibles a pocas decenas de longitudes de onda distintas, permitiendo desarrollar un nuevo paradigma en la detección y clasificación de materiales.

Esta clasificación se fundamenta en el principio por el cual todos los materiales reflejan, absorben y emiten energía electromagnética, a ciertas longitudes de onda específicas, con patrones distintos relacionados con la composición molecular de los materiales [1]. Esta respuesta dependiente de la longitud de onda, se denomina firma espectral, existiendo una firma espectral única para cada material.

Ya a principios de los años ochenta, el *Jet Propulsion Laboratory* (JPL) de la *National Aeronautics and Space Administration* (NASA) desarrolló el sensor *Airborne Visible-Infra Red Imaging Spectrometer* (AVIRIS) sensible a 224 longitudes de onda distintas, estrechas y cercanas, comprendidas entre los 400 y 2500 nanómetros. Este sensor permitió además superar las dificultades que

aparecen al embarcar sensores en plataformas móviles [2], ampliando la frontera de posibilidades que ofrece esta técnica.

Entre estas posibilidades explotadas por el mismo JPL con el sensor AVIRIS se encuentra la teledetección. Actualmente, AVIRIS se encuentra embarcado en cuatro aviones, que a una altura de 20.000 metros han caracterizado gran cantidad de zonas del planeta Tierra. Otros ejemplos de teledetección son los dos sensores que se encuentran actualmente en órbita: el *Hyperion* en el satélite *Earth Observing-1* de la NASA y *CHRIS* en el satélite *Proba-1* de la *European Space Agency* (ESA) [3]. En los próximos tres años se planea colocar en órbita tres sensores más: *Prisma* de la agencia espacial italiana (ASI) en 2013 [4], *EnMAP* del *German Aerospace Center* (DLR) y del *German Research Centre for Geosciences* (GFZ) en 2015 [5] y *HypSI* de la NASA en 2015 también [6].

La información recogida por estos sensores se almacena en lo que se han denominado imágenes o cubos hiperespectrales como el que se muestra en la Figura 1. Así, las imágenes hiperespectrales se modelan como matrices tridimensionales, una dimensión que refleja la información espectral (longitudes de onda) y dos dimensiones que reflejan la información espacial, cuya resolución da lugar al tamaño de píxel. La resolución espacial de estas imágenes hiperespectrales, sobre todo en las aplicaciones de teledetección, es menor que el tamaño de los objetos, existiendo así píxeles cuya información espectral es una combinación de las firmas espectrales de diversos materiales.

Existen multitud de modelos teóricos que tratan de explicar cómo se combinan las firmas espectrales para dar lugar a la información contenida en la imagen, estando la mayoría basados en un modelo de mezcla lineal (*linear mixing*) tal que cada píxel se pueda modelar como una combinación lineal de un número finito de firmas espectrales, firmas que la literatura denomina *endmembers* [7].

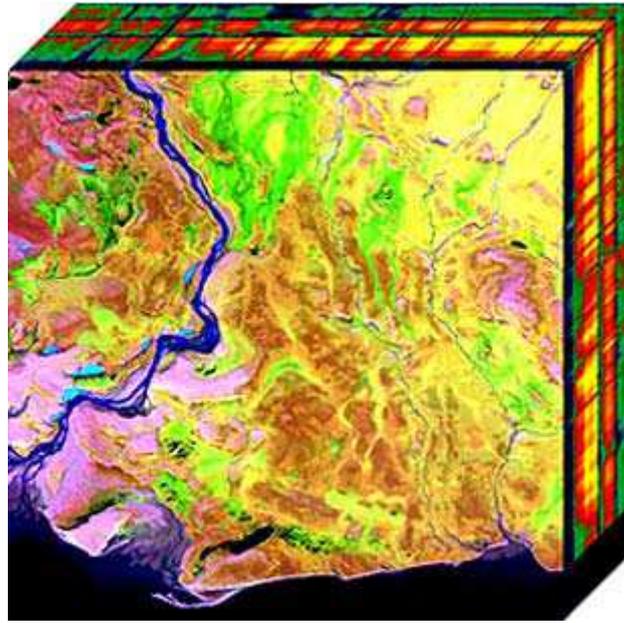


Figura 1.- Cubo hiperespectral

Conociendo el modo en que se combinan los *endmembers*, es posible descomponer cada píxel de la escena en los distintos *endmembers* y en la abundancia que pueda existir de cada *endmember* en cada píxel. Este procedimiento se denomina *hyperspectral unmixing* y es el que permite identificar los materiales que existen en la escena, siendo de una alta complejidad computacional y dividiéndose típicamente en un proceso de tres etapas: estimación del número de *endmembers* (reducción de dimensionalidad espectral), extracción de *endmembers* y estimación de abundancias.

La reducción de dimensionalidad espectral tiene como propósito reducir la complejidad computacional de cara a las subsecuentes etapas. Algoritmos ampliamente usados son *Principal Component Analysis* (PCA) [8] y *Maximum Noise Fraction* (MNF) [9], aunque *Virtual Dimensionality* (VD) [10] e *Hyperspectral signal Subspace identification by minimum error* (HySime) [11] son capaces de estimar cuántas bandas son necesarias para representar realmente el espacio de datos, valor íntimamente relacionado con el número de *endmembers* existentes en la escena cuyo valor es necesario conocer para las subsecuentes etapas.

La extracción de *endmembers* consiste en determinar qué firmas espectrales son necesarias para representar el total de la imagen hiperespectral. Entre los algoritmos existentes cabe destacar *Pixel Purity Index* (PPI) [12], *N-finder* (N-FINDR) [13] [14], *Vertex Component Analysis* (VCA) [15] y *Convex Cone Analysis* (CCA) [16].

Finalmente, la estimación de abundancias consiste en averiguar cuál es la aportación cuantitativa de cada *endmember* a cada píxel de la imagen. Así, el algoritmo más empleado es el *Fully Constrained Least Squares* (FCLS) [17].

Por otra parte, existen algoritmos que son capaces de realizar la extracción de *endmembers* y estimación de abundancias de forma simultánea, como *Alternating Projected Subgradients* (APS) [18], *Iterated Constrained Endmembers* (ICE) [19], *Nonnegative Matrix Factorization* (NMF) [20] [21], *Joint Bayesian Approach* (JBA) [22] o *Minimum Volume Transform* (MVT) [23].

## 1.2 Objetivos

El objetivo final del trabajo de investigación que aquí se describe, consiste en el desarrollo de un nuevo algoritmo que realiza conjuntamente, tanto la estimación del número de *endmembers*, como la extracción de *endmembers* y el cálculo de abundancias, formando así toda la cadena de *Linear-Unmixing* en lo que podría denominarse un algoritmo unificado *1-step*. Se implementará en el entorno *Matlab* y se acompañará de un conjunto de pruebas que demuestren el correcto funcionamiento de este procedimiento.

Las pruebas que se realizan, irán acompañadas de los resultados comparativos con una cadena de referencia estándar que se emplea en el estado del arte como cadena de *Linear-Unmixing*, permitiendo así probar si la aproximación que se plantea en este trabajo de investigación es correcta.

# Capítulo 2

## Análisis Previo

---

### **2.1 Introducción**

Antes de describir detalladamente la hipótesis y el algoritmo desarrollado en este trabajo de investigación, se hace necesario describir el contexto en el que se encuentra dicho trabajo, haciendo un repaso por la Teoría del Álgebra necesaria, detallando la problemática y mostrando algunas de las soluciones que conforman el estado del arte en el proceso de *Linear Unmixing* para imágenes hiperespectrales.

### **2.2 Antecedentes Teóricos**

En este apartado se pretende realizar un pequeño resumen de los aspectos más importantes del álgebra lineal que se emplea en el tratamiento de las imágenes hiperespectrales.

### 2.2.1 Definición de Espacio Vectorial

Un espacio vectorial es un conjunto  $V$  de elementos denominado vectores, que contiene una operación interna (suma) y una operación externa (producto por un escalar) definidas para los elementos del conjunto, que además satisfacen las siguientes condiciones; para el escalar  $c$  y los vectores  $u$ ,  $v$  y  $w$ .

#### Cuerpo cerrado

1. La suma  $u + v$  existe y es un elemento de  $V$  ( $V$  es cerrado bajo adición).
2.  $cu$  es un elemento de  $V$  ( $V$  es cerrado bajo multiplicación por un escalar).

#### Axiomas sobre la suma

3.  $u + v = v + u$  (propiedad conmutativa).
4.  $u + (v + w) = (u + v) + w$  (propiedad asociativa).
5. Existe un elemento de  $V$  denominado **vector neutro**, denotado por  $0$ , tal que  $u + 0 = u$ .
6. Para todo elemento  $u$  de  $V$  existe un elemento denominado **negativo** de  $u$ , denotado por  $-u$ , tal que  $u + (-u) = 0$ .

#### Axiomas sobre el producto por un escalar

7.  $c(u + v) = cu + cv$
8.  $(c + d)u = cu + du$
9.  $c(du) = (cd)u$
10.  $1u = u$

### 2.2.2 Definición de Combinación Lineal

Sean  $v_1, \dots, v_m$  vectores de un espacio vectorial  $V$ . El vector  $v$  en  $V$  es una **combinación lineal** de  $v_1, \dots, v_m$  si existe un conjunto de escalares  $c_1, \dots, c_m$  tal que  $v$  pueda ser expresado como

$$v = c_1v_1 + \dots + c_mv_m$$

### 2.2.3 Definición de Independencia Lineal

Un conjunto de vectores  $\{v_1, \dots, v_m\}$  es **linealmente independiente** si

$c_1v_1 + \dots + c_mv_m = 0$  sólo se puede satisfacer si  $c_i = 0 \forall i$ .

---

### 2.2.4 Definición de Base

Un conjunto finito de vectores es denominado **base** de un espacio  $V$  si son linealmente independientes y todo elemento de  $V$  se puede escribir como combinación lineal de los elementos de la base.

### 2.2.5 Definición de Dimensión

Si un espacio vectorial  $V$  tiene una base que consiste en  $n$  vectores, entonces la **dimensión** del espacio es  $n$ .

**Teorema:** El espacio de columnas y el espacio de filas de una matriz tienen la misma dimensión.

### 2.2.6 Pseudoinversa de Moore-Penrose

Sea  $A$  una matriz. La matriz  $(A^T A)^{-1} A^T$  se denomina **pseudoinversa** y se denota por  $A^+$  o  $\text{pinv}(A)$ . Se trata de una generalización de la inversa de una matriz e igual que la inversa de una matriz, la pseudoinversa no siempre existe, pues para que exista  $(A^T A)^{-1}$  debe existir.

### 2.2.7 Mínimos Cuadrados

Es sabido que un sistema  $A\mathbf{x} = \mathbf{y}$  de  $n$  ecuaciones y  $n$  incógnitas, donde  $A$  es invertible, tiene solución única  $\mathbf{x} = A^{-1}\mathbf{y}$ . Sin embargo, si  $A\mathbf{x} = \mathbf{y}$  es un sistema de  $n$  ecuaciones y  $m$  incógnitas donde  $n > m$ , el sistema no tiene solución en general y se dice que está sobredeterminado o que el sistema es incompatible. Además,  $A$  no es cuadrada para tal sistema, con lo que  $A^{-1}$  no existe.

Sin embargo, mediante la pseudoinversa de  $A$  se puede alcanzar la solución más cercana a la solución real, o incluso la solución real si existe, en lo que se denomina solución por **mínimos cuadrados**, tal que  $\mathbf{x} = A^+\mathbf{y}$ .

Así, se puede concluir:

*Sea  $A\mathbf{x} = \mathbf{y}$  un sistema de  $n$  ecuaciones lineales con  $m$  incógnitas donde  $n > m$  y  $A$  tiene rango  $m$ . Este sistema tiene una solución por mínimos cuadrados. Si el sistema tiene solución real, la solución por mínimos cuadrados es la solución real. Si el sistema no tiene solución real la solución por mínimos cuadrados es la*

solución más cercana a la solución real que podemos obtener. El sistema no puede tener más de una solución.

Gráficamente, y llevándolo a un espacio tridimensional, si se tiene un sistema  $A\mathbf{x} = \mathbf{b}$ , donde la matriz  $A$  tiene 3 filas y 2 columnas (rango 2) y se denota por  $\mathbf{x}'$  la solución por mínimos cuadrados  $\mathbf{x}' = A^+ \mathbf{b}$ , entonces el resultado es el que se muestra en la Figura 2.

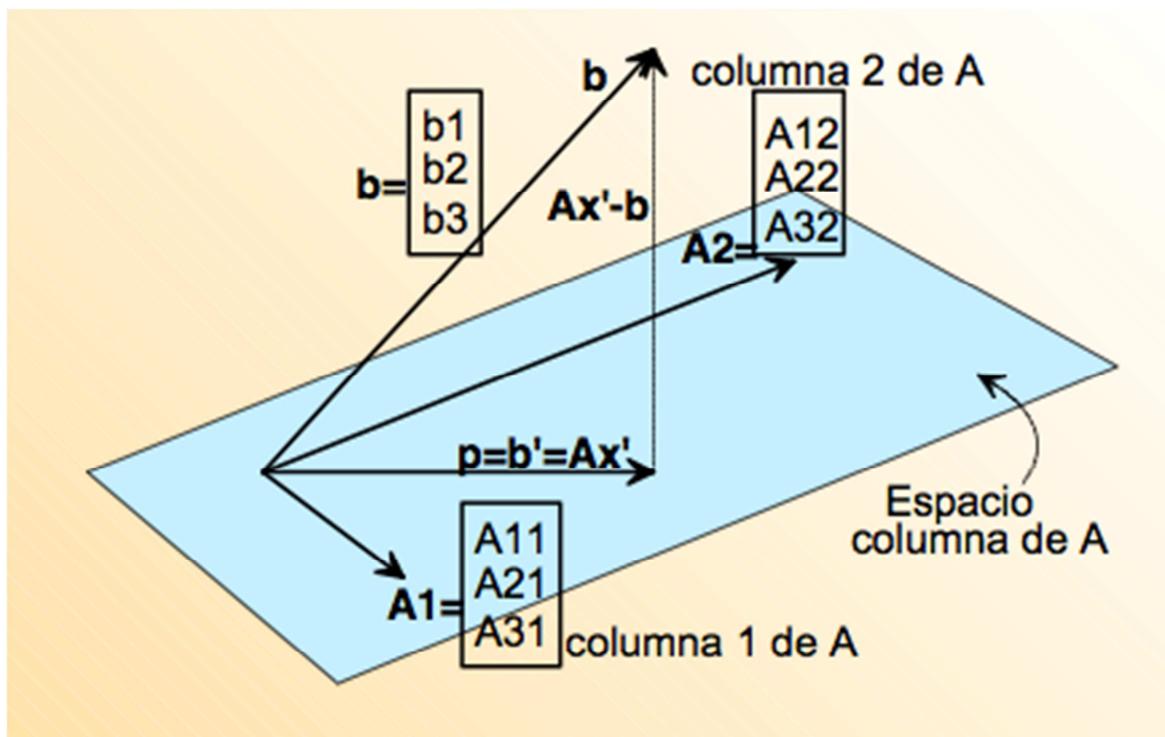


Figura 2.- Mínimos Cuadrados

Se observa que el espacio de columnas de  $A$  tiene dimensión 2, es decir, los vectores columna de  $A$  son base de un plano al que no pertenece  $\mathbf{b}$  y por tanto no existe combinación lineal de los vectores columnas de  $A$  que den como resultado  $\mathbf{b}$ . Sin embargo,  $A\mathbf{x}'$  es la proyección ortogonal de  $\mathbf{b}$  en el plano por lo que la distancia  $\|A\mathbf{x}' - \mathbf{b}\|$  es mínima.

### 2.2.8 Símplex

Un símplex o símplice se define como la envoltura convexa de un conjunto de  $n+1$  puntos independientes afines en un espacio euclídeo de dimensión  $n$  o mayor

[24]. Así, un triángulo es un símpex en 2 dimensiones y un tetraedro un símpex en 3 dimensiones (Figura 3).

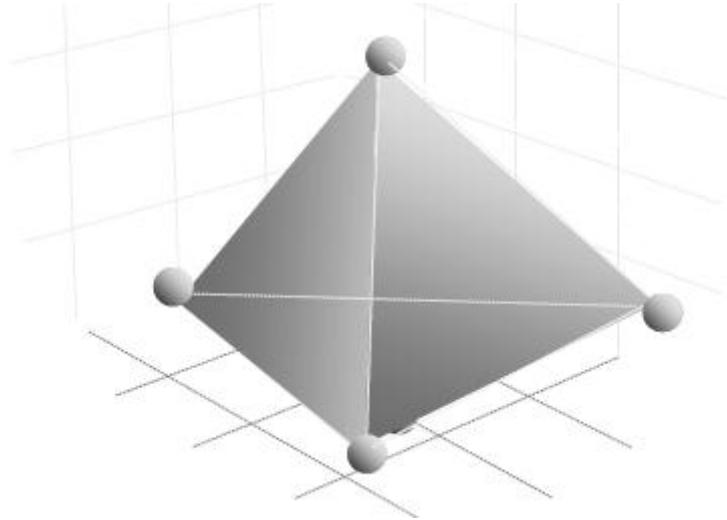


Figura 3.- Símpex en 3 dimensiones

Además, todo punto  $s$  del espacio  $n$ -dimensional está contenido en el interior del símpex sí

$$s = \sum_{i=1}^{n+1} v_i a_i \in \mathbb{R}^n \left| \sum_{i=1}^{n+1} a_i = 1 \text{ y } a_i \geq 0 \forall i \right.$$

siendo  $v_i$  el vértice  $i$ -ésimo del símpex.

### 2.3 Imagen Hiperespectral

Una imagen hiperespectral es una imagen en que cada píxel se define por un vector de varias componentes espectrales (normalmente en las centenas) que se corresponde con la reflectancia obtenida para ciertas longitudes de onda. Estas longitudes de onda se corresponden con los canales del espectro electromagnético que obtienen los sensores hiperespectrales, típicamente comienzan en el ultravioleta y terminan en el infrarrojo, abarcando un ancho de banda superior a  $10 \mu\text{m}$ .

Usualmente se suele representar los datos hiperespectrales en modo de cubo (hipercubo), compuesto por una pila de imágenes correspondientes a cada longitud de onda. En la Figura 4 se muestra un esquema del hipercubo y la

correspondencia vectorial asociada a cada píxel, lo que permite tratar las imágenes a partir de una formulación algebraica.

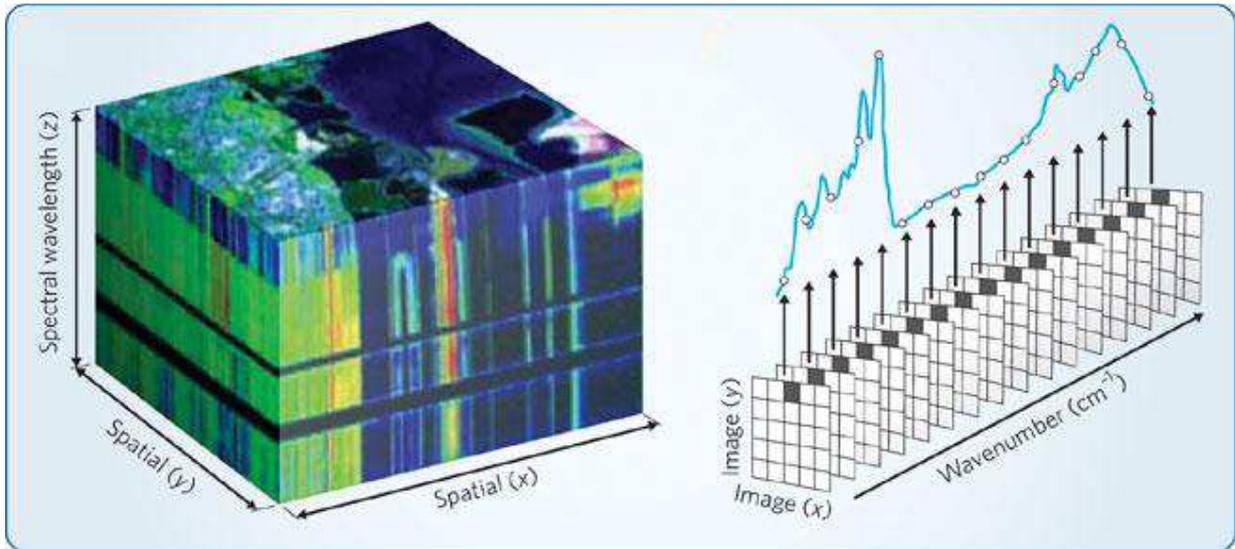


Figura 4.- Cubo hiperespectral y correspondencia vectorial

El muestreo de las diferentes longitudes de onda viene determinado por la tecnología del sensor que se emplea. A menor separación entre bandas se suele tener mejores resultados en las aplicaciones gracias a que se puede obtener una firma espectral más fidedigna de los elementos observados. Así, las imágenes hiperespectrales suponen un salto no solo cuantitativo, sino también cualitativo respecto a las imágenes multiespectrales, que sólo distinguen un número de longitudes de onda en el orden de las decenas.

En la Figura 5 se exhibe una muestra de la diferencia entre una imagen multiespectral (a) y una imagen hiperespectral (b), refiriéndose ambas gráficas a un píxel de vegetación. La imagen multiespectral fue tomada por el sensor *Landsat Thematic Mapper* [25] y la imagen hiperespectral fue tomada por el sensor *Airbone Visible/InfrarRed Imaging Spectrometer (AVIRIS)* [2].

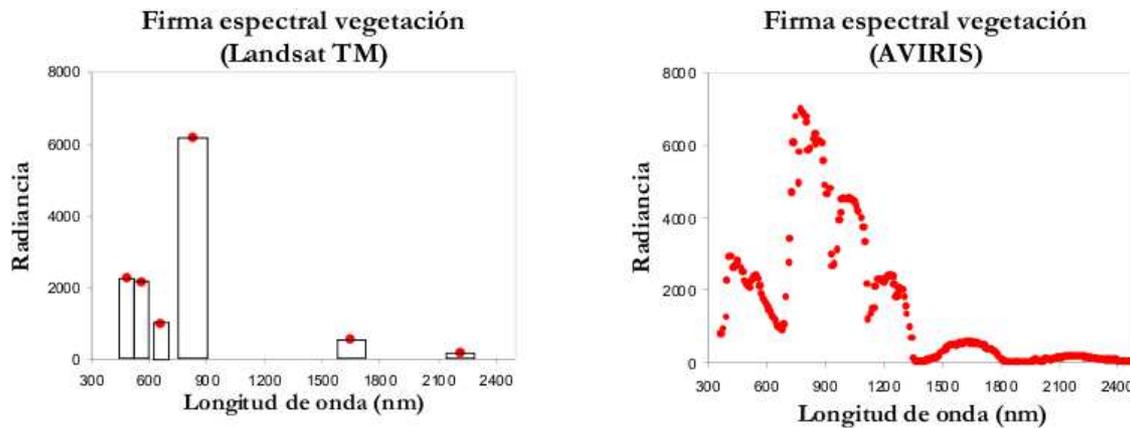


Figura 5.- (a) Imagen multiespectral (6 bandas) (b) e imagen hiperespectral (224 bandas)

## 2.4 El Problema de la Mezcla

En la práctica, la información espectral de gran parte de los píxeles de la imagen es resultado de la *mezcla* de firmas espectrales de diversos materiales, existiendo multitud de modelos teóricos que tratan de explicar cómo se combinan las firmas espectrales para dar lugar a la información contenida en la imagen.

En este apartado se explican los distintos tipos de mezcla que se producen y los dos principales modelos de mezcla que se consideran en el estado del arte.

### 2.4.1 Tipos de Mezcla

En las imágenes hiperespectrales es muy común la existencia de mezcla a nivel subpíxel, por lo que podemos encontrarnos dos tipos de píxeles: píxeles puros o píxeles mezcla.

Un píxel puro es aquel en el que todo el píxel hace referencia a un único material de forma homogénea y en un único estado de la materia. Se le denomina *endmember*.

Por otra parte, un píxel mezcla es aquel en el que existen diferentes materiales [26-28]. Este tipo de píxel es el que constituye la mayor parte de la imagen hiperespectral, ya que este fenómeno es independiente de la escala considerada pues tiene lugar incluso a niveles microscópicos.

La Figura 6 muestra un ejemplo del proceso de adquisición de píxeles puros (a nivel macroscópico) y píxeles mezcla en imágenes hiperespectrales.

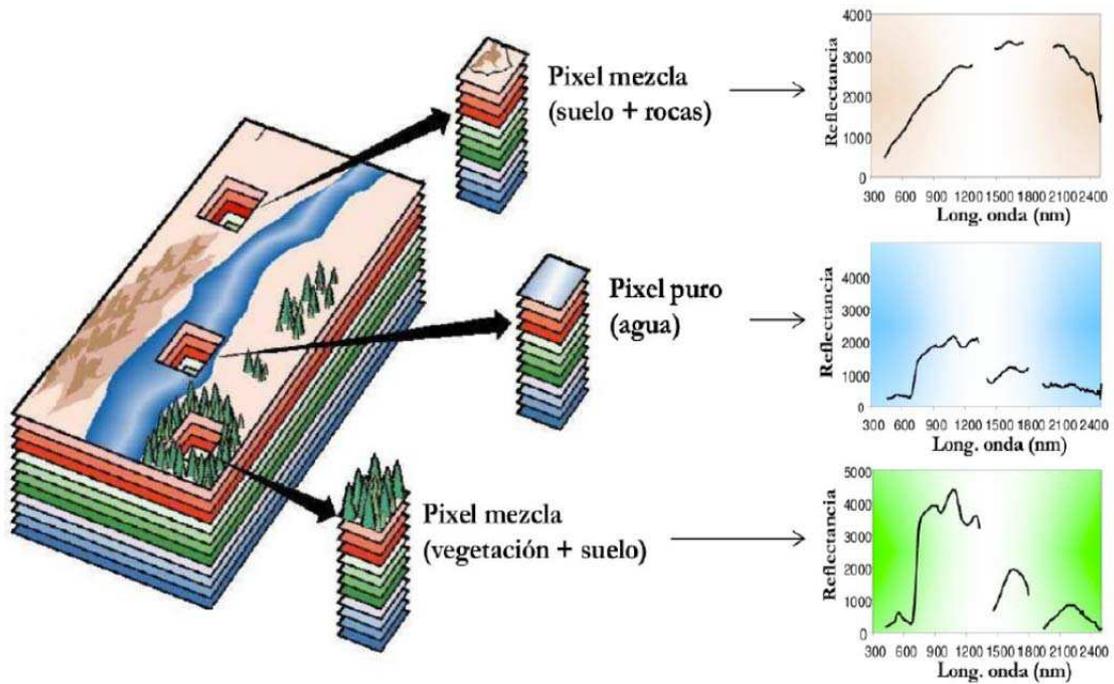


Figura 6.- Tipos de píxel en imágenes hiperespectrales

Así, se puede clasificar la mezcla en:

1. Mezcla macroscópica: es un tipo de mezcla debida a la reducida resolución espacial del píxel. Si la resolución de píxel no es lo suficientemente alta para que diferentes materiales aparezcan en diferentes píxeles, dichos materiales ocuparan el mismo espacio asignado al píxel, con lo que el espectro resultante será la suma de los espectros de los materiales contenidos en el píxel, ponderados por la cantidad de superficie del píxel que ocupe cada material [28]. Esta situación es la que se representa en la Figura 7.

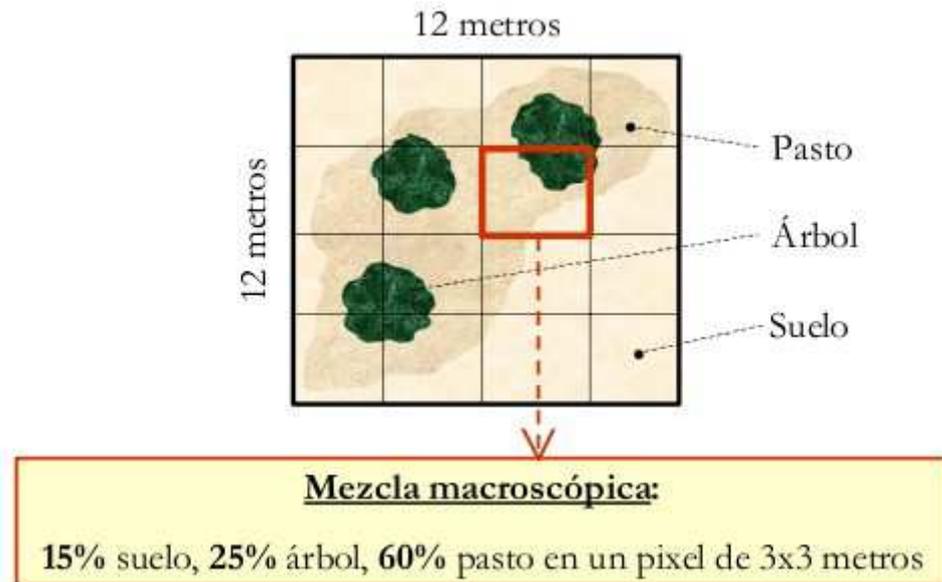


Figura 7.- Mezcla macroscópica

2. Mezcla íntima: este tipo de mezcla es independiente de la resolución del píxel ya que es una mezcla que puede llegar a ser microscópica. Se debe a la combinación de diferentes materiales que dan lugar a una mezcla que podría considerarse pura [29,30]. Esta situación viene reflejada en la Figura 8.

Un píxel mezcla puede ser descompuesto en una colección de espectros puros o característicos, ponderados cada uno de ellos por las denominados abundancias, que indican la proporción o contribución individual de cada uno de los espectros puros al píxel mezcla [31].

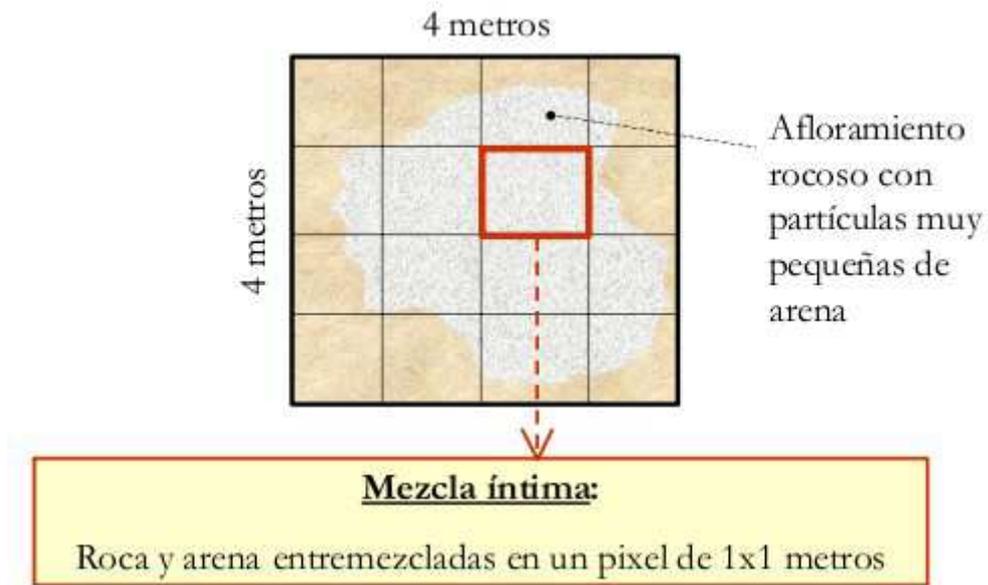


Figura 8.- Mezcla íntima

### 2.4.2 Modelos

El escenario comentado anteriormente es el denominado “problema de la mezcla” y se modela bajo la suposición de que cualquier imagen está constituida por un conjunto de *endmembers* con propiedades características y diferentes entre sí, y un conjunto de píxeles mezcla que son combinaciones de los *endmembers* de la imagen [32,33].

Así, se definen dos tipos de modelo de mezcla: lineal y no lineal.

#### **Modelo lineal**

Se trata de un modelo basado en la suposición de que cada haz de radiación de luz incidente interactúa solamente con un único material, de forma que la radiación total reflejada por un píxel mezcla se puede descomponer de forma proporcional a la abundancia de cada material en dicho píxel mezcla [34]. Este modelo es el más extendido debido a su sencillez y generalidad a la hora de analizar imágenes hiperespectrales.

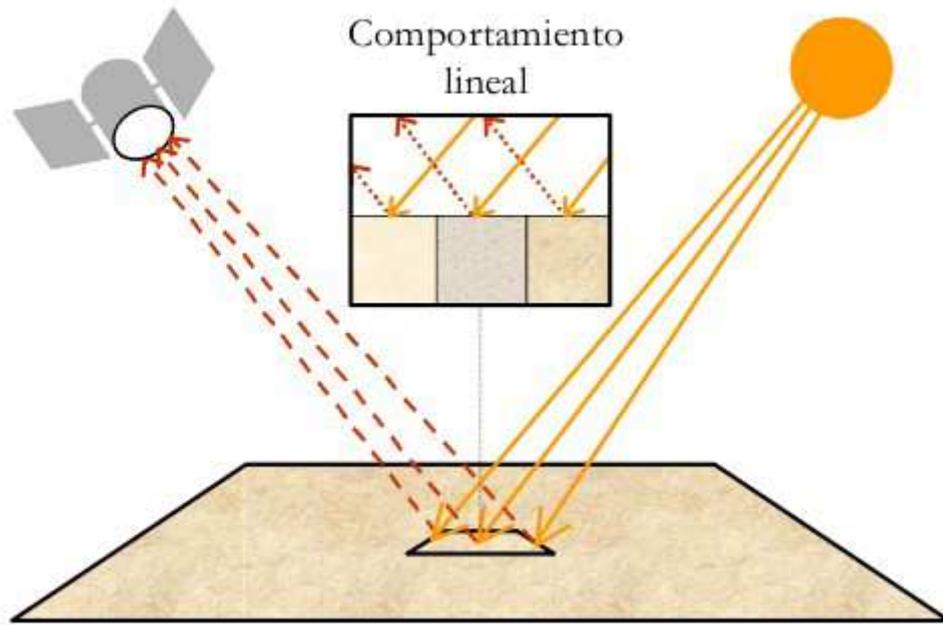


Figura 9.- Modelo lineal de mezcla

Dada la representación de la Figura 9, sea  $s_j$  la firma espectral obtenida por un sensor hiperespectral en un determinado píxel mezcla. Este espectro puede ser considerado como un vector  $N$ -dimensional, donde  $N$  es el número de bandas espectrales de la imagen y, por tanto, el píxel  $j$ -ésimo puede modelarse como combinación lineal de vectores *endmembers*  $e_i$ ,  $i=1\dots p$ , de acuerdo con la expresión:

$$s_j = \sum_{i=1}^p e_i a_{ij} + n_{ij}$$

donde  $p$  es el número total de *endmembers*,  $a_{ij}$  es el coeficiente escalar que representa la abundancia del *endmember*  $e_i$  en el píxel  $s_j$ , y  $n_{ij}$  es el ruido.

Así, si no se tiene en cuenta el ruido, podemos considerar que los  $p$  *endmembers* forman la base de un espacio de dimensión  $p$ , aunque no se puede asegurar a priori que los *endmembers* sean linealmente independientes. Además, retomando la definición de *símplex*, se observa que los *endmembers* son los vértices de un *símplex*, en cuyo interior se encuentran los píxeles  $s_j$ . Esto atiende a que las

abundancias de un material en un píxel sólo pueden tomar valores comprendidos entre 0 y 1, ya que representan la proporción que de cada material existe en el píxel, por lo que además, la suma de estas abundancias es 1.

En la Figura 10, se representa un diagrama de dispersión bidimensional, dadas dos bandas poco correlacionadas, donde los puntos más extremos son los *endmembers* y el resto de puntos son las combinaciones lineales de dichos *endmembers*, es decir, los píxeles mezcla.

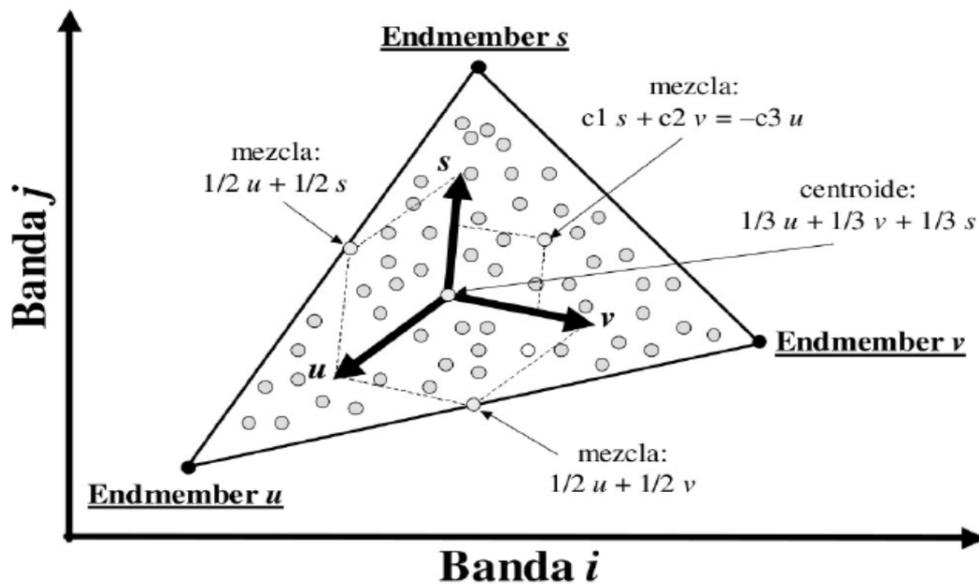


Figura 10.- Diagrama de dispersión bidimensional

### Modelo no lineal

Este modelo de mezcla supone que el haz de luz es reflejado por diversos materiales debido a efectos de dispersión [35], debidos a la constitución de los materiales o a la forma de los objetos. Un ejemplo de este fenómeno es la reflexión de la luz sobre el agua, pues una cantidad de energía es reflejada por la capa superficial del agua, pero otra es refractada, reflejándose en niveles inferiores con distintos ángulos. Este fenómeno se muestra en la Figura 11.

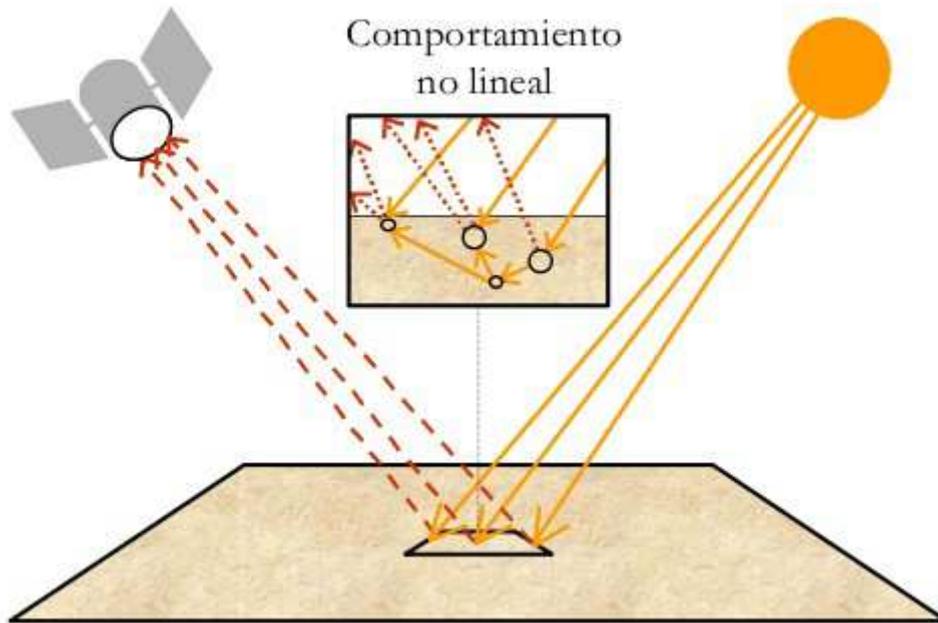


Figura 11.- Modelo no lineal de mezcla

El modelo no lineal describe el comportamiento de la luz de un modo más preciso que el modelo lineal. Sin embargo, su correcta aplicación requiere de información acerca de la geometría y propiedades físicas de los objetos observados. En general no se dispone de esta información a priori, y en caso de disponerse o poder extraerse de la imagen, dificultan la implementación de cualquier sistema.

## 2.5 Algoritmos

A continuación se realiza un breve repaso de algunos de los algoritmos más destacados que conforman la cadena típica de *Linear Spectral Unmixing* de imágenes hiperespectrales. Todos los algoritmos que aquí se presentan se basan exclusivamente en la información espectral de la imagen, es decir, no emplean información espacial, pues son algoritmos que no tienen en cuenta el orden de los píxeles.

### 2.5.1 Cadena Típica de *Linear Spectral Unmixing*

En la Figura 12 se muestra una representación de la cadena típica de algoritmos que realizan el proceso de *Linear Spectral Unmixing*.

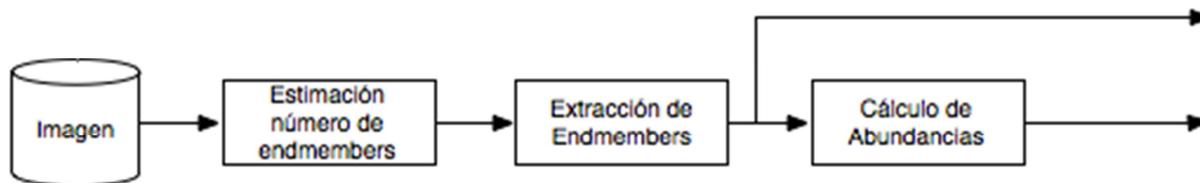


Figura 12.- Cadena típica de *Linear Spectral Unmixing*

En primer lugar, se realiza una estimación del número de *endmembers*. La estimación obtenida se pasa al algoritmo encargado de extraer los *endmembers* de la imagen, para que el siguiente algoritmo calcule las abundancias a partir de estos *endmembers*.

Es interesante notar, que este esquema es vulnerable a la propagación de errores. Así, por ejemplo, los frecuentes errores en la estimación del número de *endmembers* son propagados, afectando a las etapas posteriores.

### 2.5.2 Estimación del número de *Endmembers*

En la literatura se encuentran principalmente dos algoritmos para estimar el número  $p$  de *endmembers* que existen en la imagen hiperespectral original: *Hyperspectral signal Subspace identification by minimum error* (HySime) [11] y *Virtual Dimensionality* (VD). [10]

A pesar de las críticas a estos métodos (suelen fallar en la estimación), son los más utilizados para estimar el número de *endmembers* en la literatura. Estos métodos pueden ser brevemente resumidos del siguiente modo:

- **HySime** estima el subespacio de la imagen hiperespectral estimando primero la matriz de correlación de la señal y el ruido, y luego, seleccionando el subconjunto de autovectores que mejor represente el subespacio minimizando el error cuadrático medio.
- **VD** formula la cuestión de si una firma distinta está presente o no en cada una de las bandas espectrales como un problema de contraste de hipótesis binaria, donde se genera un detector denominado de Neyman-Pearson, para elegir la dimensión del subespacio definido por los *endmembers*,

---

basándose en la probabilidad de un parámetro denominado *false alarm rate*.

### **2.5.3 Extracción de Endmembers**

En este apartado se presentan los tres algoritmos de extracción de *endmembers* más empleados. Se basan en el modelo lineal de mezcla y están basados únicamente en la información espectral.

#### **2.5.3.1 Pixel Purity Index**

*Pixel Purity Index* (PPI) [36] es un algoritmo que ofrece resultados satisfactorios cuando los componentes que residen a nivel subpíxel aparecen espacialmente separados, situación en la cual los fenómenos de absorción y reflexión de la radiación electromagnética incidente pueden ser caracterizados siguiendo un patrón estrictamente lineal. Es un algoritmo muy utilizado por su sencillez y generalidad.

El algoritmo comienza generando un gran número de vectores aleatorios  $N$ -dimensionales llamados *skewers* y asociando un contador inicializado a 0 a cada píxel. Este contador se denomina *índice de pureza*. Cada píxel de la imagen se proyecta sobre cada *skewer* y los contadores asociados a los píxeles más extremos en la dirección de éste se incrementan. Los  $p$  píxeles con mayor valor en sus contadores (mayor *índice de pureza*) son los considerados como *endmembers*.

La gran ventaja que ofrece este algoritmo se basa en la sencillez del mismo, permitiendo la aplicación repetitiva de operaciones vectoriales regulares y predecibles.

El algoritmo se puede resumir en los pasos siguientes:

1. Asignar un índice de pureza a todos los píxeles de la imagen.
2. Generar un vector unitario aleatorio (*skewer*).
3. Proyectar todos los píxeles de la imagen sobre el vector unitario, identificando los píxeles extremos en la dirección definida por el vector unitario. Los índices de pureza de los puntos extremos se incrementa en una unidad.

4. Se repiten los pasos 2 y 3 tantas veces como defina el usuario.
5. Utilizando un valor umbral especificado como parámetro, se seleccionan los puntos de la imagen cuyo índice de pureza asociado supera dicho valor umbral. Estos píxeles se consideran *endmembers*.

La Figura 13 representa la proyección de cinco píxeles con alto índice de pureza sobre 3 *skewers* distintos, en un diagrama bidimensional, donde, intuitivamente, se comprueba que los píxeles cuyas proyecciones son más extremas sobre los *skewers*, son firmes candidatos a ser *endmembers*.

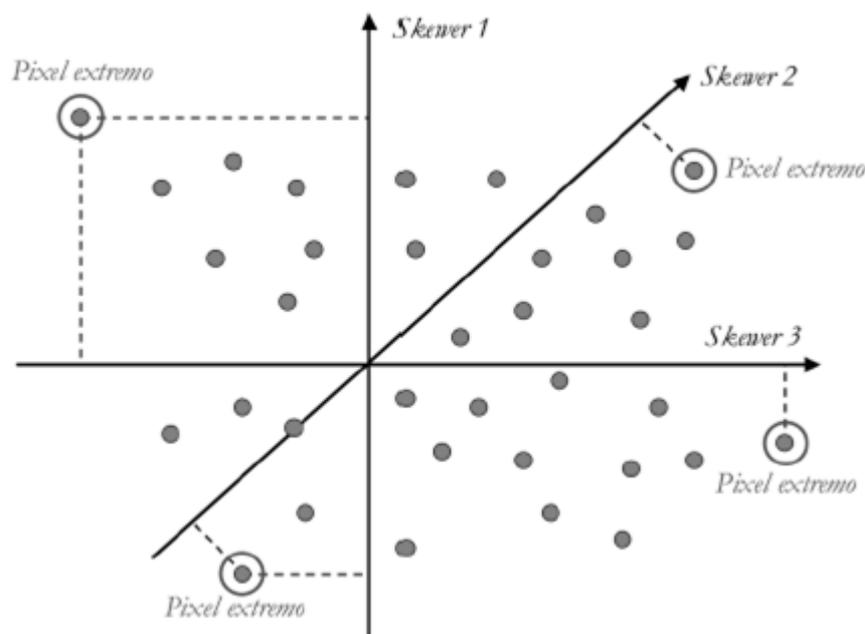


Figura 13.- Representación bidimensional del funcionamiento del algoritmo PPI

### 2.5.3.2 N-FINDR

El algoritmo N-FINDR [14,37] utiliza una técnica basada en identificar los  $p$  *endmembers* como los  $p$  vértices del simplex de mayor volumen que puede formarse a partir del conjunto de píxeles. N-FINDR no trabaja con todo el cubo de datos sino con un subespacio de dimensión  $p-1$ . Para este tipo de reducciones se suele utilizar la técnica *Principal Component Analysis* (PCA) [8,38] o *Minimum Noise Fraction* (MNF) [9,38]. Así, el único parámetro de entrada que necesita este algoritmo es el número  $p$  de *endmembers* a identificar.

El funcionamiento del algoritmo se resume en los siguientes pasos:

1. Realizar una reducción dimensional de la imagen mediante PCA o MNF a un número de dimensiones  $p - 1$ .
2. Seleccionar  $p$  píxeles de forma aleatoria que se etiquetan como *endmembers*. Esta selección inicial será refinada de forma iterativa.
3. El tercer paso consiste en seleccionar un píxel de la imagen original. Este *píxel* se va intercambiando de forma iterativa con cada uno de los considerados *endmembers* hasta el momento.
4. A medida que el píxel se va intercambiando con los *endmembers*, se realiza el cálculo del volumen del simplex formado a partir de la última sustitución.
5. Si el volumen obtenido tras la sustitución es mayor que el que había antes del intercambio, el nuevo punto trae como consecuencia un replazamiento en el conjunto de *endmembers* y el nuevo píxel pasa a formar parte del conjunto de *endmembers*. En caso contrario, se deshace la sustitución.
6. Los pasos 3-5 se repiten de forma iterativa hasta alcanzar la condición de parada. En la literatura existen diversas variantes de la condición de parada. Deben comprobarse todas las posibles combinaciones para alcanzar la solución óptima. Sin embargo, esto supone un excesivo cómputo, por lo que se suele tomar como condición de parada el probar al menos una vez todos los píxeles de la imagen. De este modo la solución obtenida no siempre es la óptima, convergiendo quizá a un máximo local, esto es, una solución subóptima [37].

El algoritmo presupone que un aumento en el volumen del simplex definido al incorporar un nuevo píxel en el conjunto de *endmembers* conlleva una mayor calidad de los mismos. Sin embargo, la Figura 14 muestra que, en situaciones con ruido, el hecho de utilizar un simplex de mayor volumen no asegura una mejor descripción del conjunto de píxeles. Un parámetro más fiable es el aumento en el número de píxeles que pueden describirse utilizando el nuevo conjunto de *endmembers*.

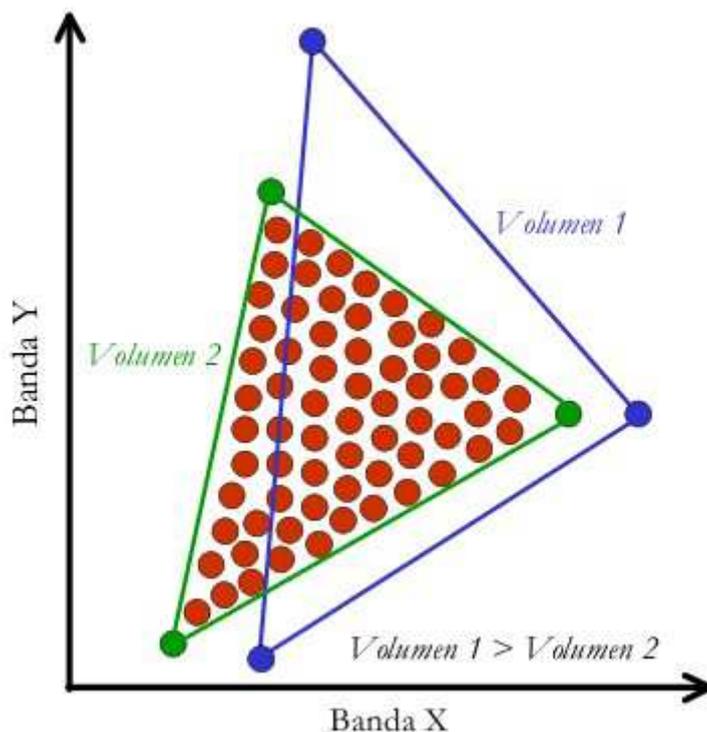


Figura 14.- Posible situación en el algoritmo N-FINDR

Para concluir la descripción de este algoritmo, es importante destacar que los *endmembers* identificados por el algoritmo N-FINDR corresponden a píxeles pertenecientes al conjunto de datos original. Utilizando este algoritmo, no es posible generar *endmembers* virtuales, pues las sustituciones se realizan siempre utilizando píxeles existentes en la imagen. Así, puede ocurrir que los *endmembers* seleccionados no sean puros, pues puede no existir en la imagen un píxel puro para alguno de los materiales que conforman la imagen.

### 2.5.3.3 Vertex Component Analysis

Se trata de un algoritmo basado en dos aspectos: por un lado los *endmembers* son los vértices de un simplex, y por otro lado, la transformación afín de un simplex es también otro simplex [15]. El algoritmo funciona tanto con imágenes proyectadas a otro espacio y reduciendo la dimensión espectral de la imagen, como con las imágenes originales que no sufren preprocesado previo. El método consiste en proyectar de manera iterativa los píxeles de la imagen sobre una dirección perpendicular al subespacio determinado por los *endmembers* ya calculados. El nuevo *endmember* se determina como el valor extremo de esta proyección. Se producen tantas iteraciones como *endmembers* se deseen calcular, es decir,  $p$  iteraciones. En la Figura 15 se muestra gráficamente el funcionamiento del algoritmo VCA.

La nube de píxeles que conforma la imagen se encuentra contenida en un cono convexo  $C_p$  que a su vez pertenece al subespacio  $E_p$  de dimensión  $p$ . El algoritmo VCA se encarga de determinar el subespacio  $E_p$  mediante *Singular Value Decomposition* (SVD) y proyectar los píxeles del cono sobre este subespacio  $y = \frac{r}{r^t u}$ , formando un simplex  $S_p$ . Este simplex se encuentra contenido en un espacio de dimensiones  $p-1$ . En caso de que la proyección se realizara sobre un subespacio  $E_d$  de mayor dimensión  $p \leq d \leq L$ , la proyección de  $C_p$  sobre  $E_d$ , seguido de una proyección adecuada es también un simplex con los mismos vértices.

Si la relación señal a ruido disminuye, el factor de escala que se aplicaba en el caso anterior,  $\frac{r}{r^t u}$ , aumentaría el ruido presente en la imagen, que en este caso es superior. Por este motivo, es preferible no usar este factor de escala, y determinar el nuevo espacio sobre el que se quiere proyectar la imagen, mediante el método PCA. En este caso, el nuevo espacio será de dimensión  $p-1$ .

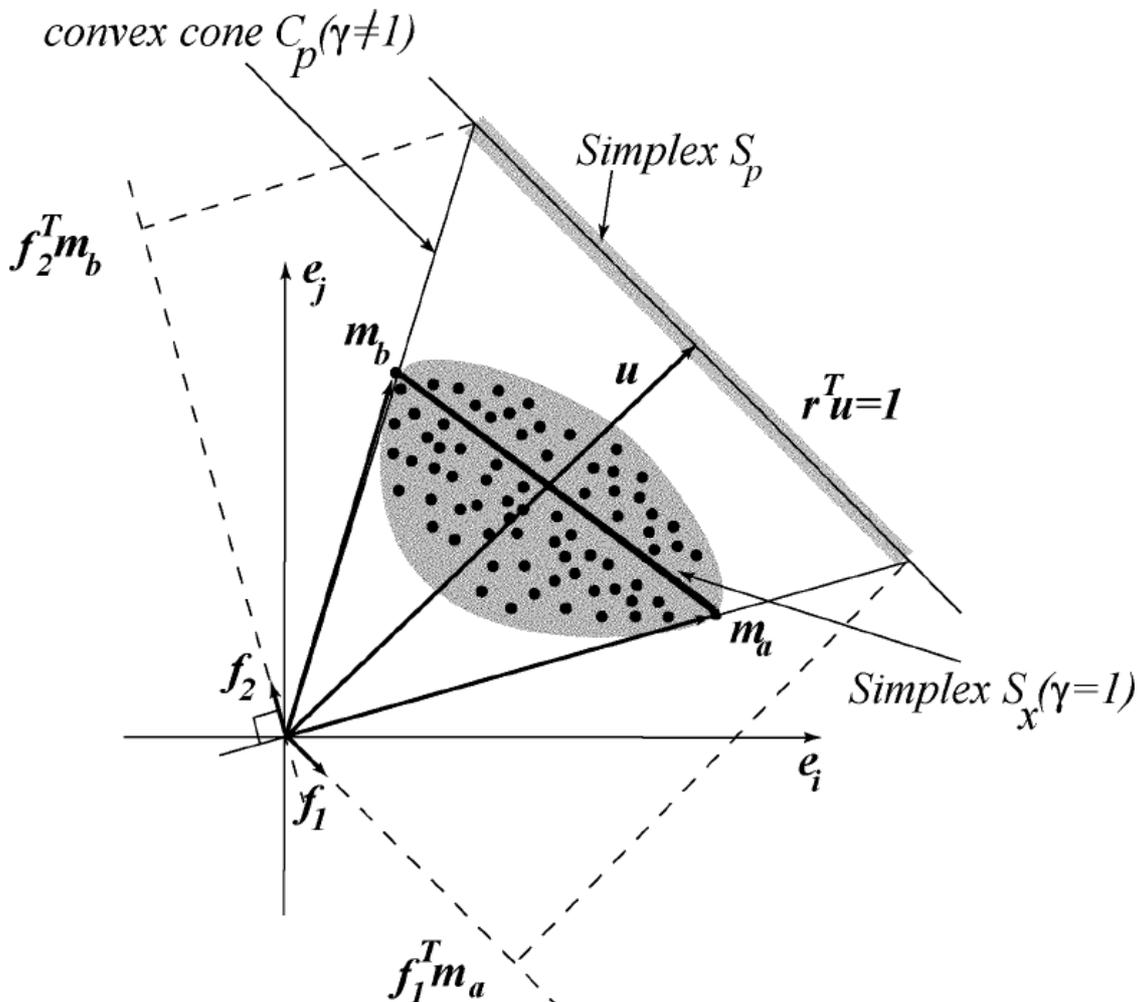


Figura 15.- Representación Bidimensional del algoritmo VCA

## 2.5.4 Cálculo de Abundancias

En este apartado se describe brevemente el algoritmo más empleado para el cálculo de las abundancias: *Fully Constrained Least Squares*.

### 2.5.4.1 Fully Constrained Least Squares

FCLS es un algoritmo basado en una aproximación por Mínimos Cuadrados para cuantificar la abundancia de los materiales en cada píxel. Se trata de un método numérico eficiente que genera soluciones óptimas, dado que no existe solución analítica. Además, el algoritmo requiere del conocimiento previo de los *endmembers* de la escena.

El método impone dos restricciones atendiendo a la naturaleza del problema:

1. Abundance sum-to-one constraint (ASC): La suma de las abundancias debe valer 1.
2. Abundance nonnegativity constraint (ANC): No existen abundancias negativas.

Para ello comienza aplicando el algoritmo *Nonnegatively Constrained Least Squares* (NCLS) que impone la restricción de no negatividad y que no tiene solución analítica, en conjunción con la restricción ASC que es más simple de implementar por tener solución analítica.

NCLS se define como un problema de optimización:

Minimizar

$$LSE = (\mathbf{M}\alpha - \mathbf{r})^T (\mathbf{M}\alpha - \mathbf{r})$$

sujeto a

$$\alpha \geq 0$$

Esto es, minimizar la distancia que existe entre la reconstrucción de la imagen  $\mathbf{M}\alpha$  y la imagen real  $\mathbf{r}$ , donde  $\mathbf{M}$  es la matriz de los *endmembers* y  $\alpha$  son las abundancias.

Para ello el algoritmo plantea un método iterativo con el que aproxima la solución de forma sucesiva.



## Capítulo 3

# Hipótesis y Algoritmo Desarrollado

---

### 3.1 Introducción

En este capítulo se describe la hipótesis sobre la que se ha realizado este trabajo de investigación, así como las estructuras de datos y el algoritmo desarrollado con el que probar la hipótesis.

### 3.2 Hipótesis

Suponiendo que en la imagen hiperespectral se encuentran los píxeles puros que constituyen el conjunto de *endmembers*, el problema consiste en averiguar qué combinación de  $p$  píxeles de la imagen forman el simplex N-dimensional que contiene al resto de píxeles en su interior, sin conocer  $p$ . Además, el problema también incluye la determinación de la abundancia de los *endmembers* en cada píxel de la imagen.

Para ello, se comienza seleccionando un conjunto de  $n$  píxeles ( $n \leq p$ ) de la imagen que inicialmente se supondrán como *endmembers*, es decir, vértices de un simplex N-Dimensional. Mediante el cálculo de las abundancias de estos

*endmembers* en el resto de píxeles por el método de Mínimos Cuadrados, se puede averiguar qué píxeles se encuentran dentro y fuera del simplex.

De este modo, la resolución que se plantea en esta investigación es sustituir y añadir de forma iterativa los píxeles que forman los vértices del simplex hasta encontrar aquella combinación que incluya a todos los píxeles mezcla en el interior. Este planteamiento es similar al que plantea N-FINDR, sustituyendo la métrica volumen por número de píxeles en el interior del simplex, aunque N-FINDR no permite aumentar el número de vértices del simplex.

La ventaja que se espera encontrar planteando la resolución de este modo consiste en poder dirigir el algoritmo hacia los mejores candidatos a sustituir los píxeles que constituyen los vértices del simplex, atendiendo al valor de las abundancias calculadas en cada iteración, pues la abundancia cuanto más negativa indica una mejor predisposición a ser *endmember*. Además, de este modo se pueden descartar de la búsqueda los píxeles que ya se ha comprobado que quedan en el interior de los sucesivos símplexes que se construyen durante el algoritmo, permitiendo también incrementar el número de *endmembers* dada la necesidad de incluir todos los píxeles en el interior del simplex final.

Así, el algoritmo será capaz de calcular el número de *endmembers*, los *endmembers* y las abundancias en un algoritmo unificado *1-step*, realimentando la información adquirida en el cálculo de las abundancias (Figura 16). Con ello, se espera mitigar la influencia que tiene la propagación de los posibles errores sobre las posteriores etapas. Se hace importante notar que se tratará de un algoritmo basado exclusivamente en la información espectral. No se tiene en cuenta la información espacial.

Por otra parte, se sabe que el ruido distorsiona la posición original de los píxeles en el espacio N-dimensional, por lo que también se debe estudiar cómo y cuánto afecta el ruido al método aquí propuesto, procurando encontrar también soluciones que permitan al algoritmo ser robusto frente al ruido.

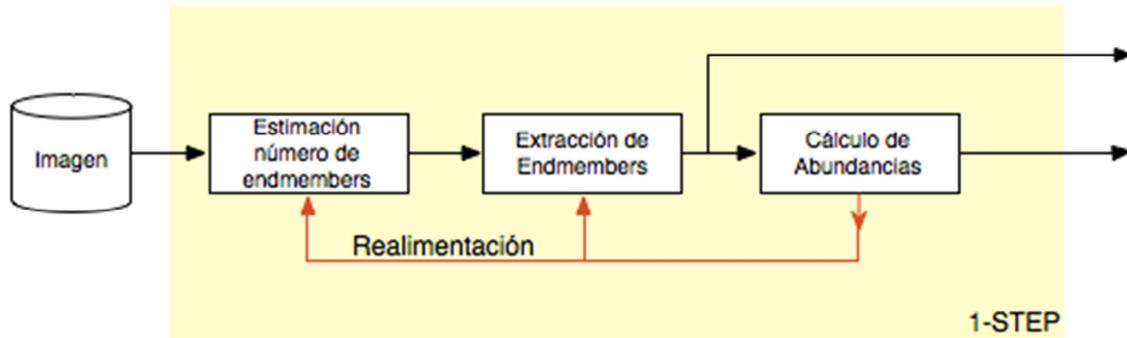


Figura 16.- Esquema del algoritmo propuesto

### 3.3 Estructuras de Datos

Antes de profundizar en la estructura algorítmica se hace necesario conocer al menos las principales estructuras de datos que contienen la información de estado del algoritmo.

#### 3.3.1 Image

Matriz de números reales (en principio tipo *double*) que representa a la imagen hiperespectral, donde cada píxel de la imagen se corresponde con una columna, y las diferentes bandas de cada píxel se representan en las diferentes filas. Con ello, se pierde la información espacial de los píxeles, pero como ya se ha indicado, el análisis que realiza este algoritmo es espectral y no se tiene en cuenta la información espacial.

El algoritmo acepta imágenes de cualquier tamaño, es decir, tolera cualquier número de píxeles y cualquier número de bandas. En la Figura 17 se muestra una representación gráfica de la matriz **Image**.

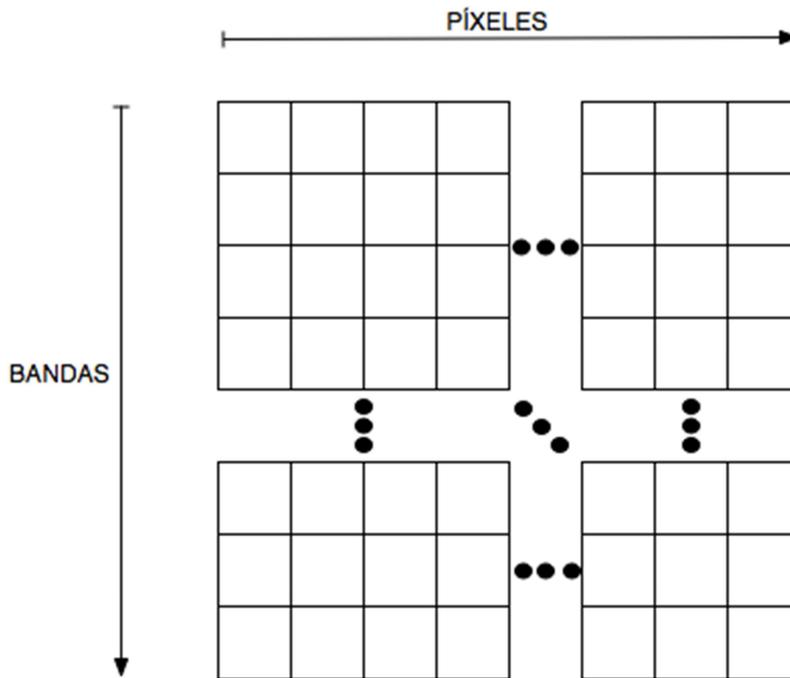


Figura 17.- Matriz Image

### 3.3.2 $p$

Variable numérica de valor entero, que representa el número de *endmembers* que el algoritmo estima que tiene la imagen en un momento dado de la ejecución.

### 3.3.3 $E\_index$

Vector de enteros de longitud  $p$  que representan el índice (columna) de los píxeles de la imagen en la matriz **Image** que el algoritmo estima que son los *endmembers* en cada instante.

### 3.3.4 $is\_candidate$

Vector binario de longitud igual al número de píxeles de la imagen, que indica si un píxel sigue o no siendo candidato a ser *endmember*. El vector se inicializa con todas las posiciones con valor 1. Estos valores son cambiados por la función **discard\_candidates()** que asigna un 0 (descarta) a aquellas posiciones que se corresponden a píxeles que se ha comprobado están dentro de alguno de los sucesivos símplexes que se construyen durante la ejecución del algoritmo.

En la Figura 18 se muestra una representación gráfica del vector  $is\_candidate$ .

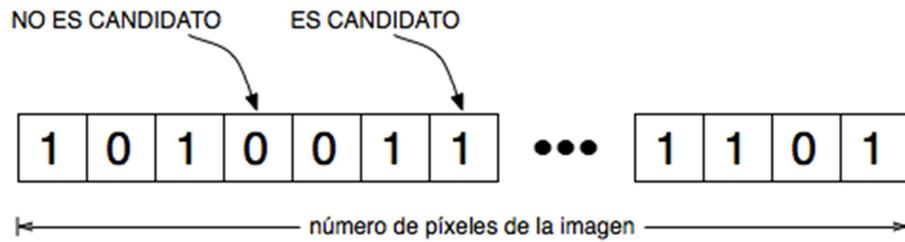


Figura 18.- Vector `is_candidate`

### 3.3.5 `next_j_index`

Vector ordenado que incluye los índices de los píxeles en función de las abundancias asociadas al mejor símplex encontrado hasta el momento dado. El índice del píxel que menor (más negativa) abundancia tiene de los supuestos *endmembers* actuales se encontrará en la primera posición; y el índice del píxel que mayor (menos negativa) abundancia tiene de los supuestos *endmembers* actuales se encontrará en la última posición.

Esta estructura es la que permite dirigir el algoritmo hacia los píxeles que se consideran mejores candidatos (abundancia más negativa) para ser *endmembers* finales, lo cual acelera mucho la convergencia del algoritmo.

Esta aceleración se debe a un efecto doble: por un lado, el hecho de realizar las sustituciones por los mejores candidatos a ser *endmembers*; y por otro lado, ya que los mejores candidatos construyen símplex capaces de encerrar mayor número de píxeles, se logra acelerar el descarte de candidatos.

### 3.3.6 `max_pixels_in_simplex`

Se trata de un valor numérico entero que representa el número máximo de píxeles que se ha logrado incluir dentro de alguno de los símplexes construidos con el número actual de vértices, es decir, con el número de *endmembers* que el algoritmo supone tiene la imagen en un momento dado. Este valor es reiniciado cada vez que varía la variable `p`.

### 3.4 Algoritmo

La estructura general del algoritmo que se ha realizado en este trabajo de investigación para probar la hipótesis de partida se muestra en la Figura 19.

Como parámetros principales de entrada, el algoritmo recibe la matriz **Image** y el número de *endmembers* **p** que el algoritmo tomará como primera aproximación al número de vértices que realmente tiene el simplex que define la imagen.

Con ello, el algoritmo comienza con la definición de la semilla, que consiste en la elección aleatoria de  $p$  píxeles de la imagen que supondremos como *endmembers*. Esta elección se hace efectiva al escoger  $p$  índices de los píxeles y almacenarlos en el vector **E\_index**. Sin embargo, dado que durante el algoritmo se realiza el cálculo de abundancias aplicando el método de Mínimos Cuadrados que requiere del cálculo de la inversa de  $\mathbf{EE}^t$ , siendo **E** la matriz que contiene los *endmembers*, es necesario asegurar que esta matriz  $\mathbf{EE}^t$  está bien condicionada, por lo que la semilla se define tantas veces como sea necesaria hasta que la matriz  $\mathbf{EE}^t$  esté bien condicionada. Normalmente el sistema queda bien condicionado con la primera elección.

Para conocer si la matriz está bien condicionada hacemos uso del comando *rcond()* de *Matlab*, que devuelve un valor entre 0 y 1, donde 1 es un sistema perfectamente condicionado y 0 un sistema en el que existen dependencias lineales y por tanto no es posible obtener una solución. En el algoritmo, se considera que la matriz está bien condicionada cuando el valor devuelto por *rcond()* es mayor o igual a  $5 \cdot 10^{-17}$  que a pesar de parecer un valor demasiado bajo, se ha comprobado experimentalmente que permite resultados muy buenos, como se demuestra en el capítulo 4.

Posteriormente, se realiza la inicialización donde se realiza el primer cálculo de abundancias y con ello se definen la variable **is\_candidate**, **next\_j\_index** y **max\_pixels\_in\_simplex**.

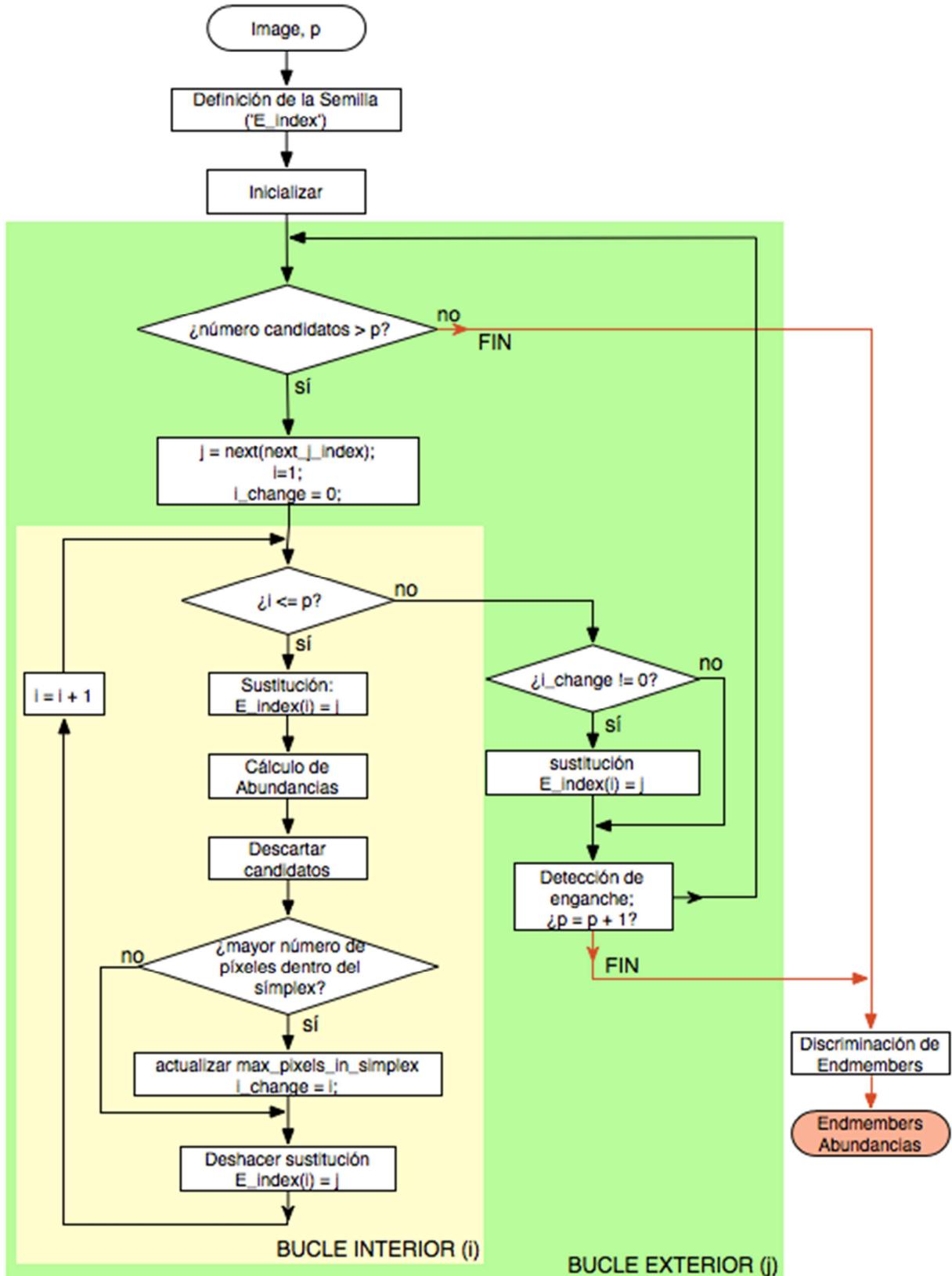


Figura 19.- Flujoograma del algoritmo desarrollado

A continuación, el algoritmo entra en la resolución iterativa que se compone de dos bucles anidados como se puede apreciar en la Figura 19. Un bucle exterior que se define sobre la variable **j** y un bucle interior definido sobre la variable **i**. La variable **j** indica el píxel candidato a ser *endmember* que se está probando en cada instante y la variable **i** indica cuál es el *endmember* que está siendo sustituido por el píxel candidato **j**. Así, el algoritmo va sustituyendo cada candidato a ser *endmember* por cada uno de los *endmembers* actuales, comprobando si, dada alguna sustitución, el simplex resultante de dicha sustitución es capaz de encerrar un mayor número de píxeles dentro del simplex, número que se almacena en la variable **max\_pixels\_in\_simplex**.

Detalladamente, el bucle exterior comienza comprobando si el número de candidatos actual es mayor al número de *endmembers* **p** que se está considerando que tiene la imagen. Esto es así porque los considerados *endmembers* en cada momento no pueden ser tachados, y por tanto, no pueden dejar de ser candidatos, por lo que si se llegara a una situación en la que el número de candidatos fuera igual o menor al valor **p**, entonces no habría más candidatos a sustituir y se podría asegurar que el algoritmo ha finalizado, saliendo del bucle.

A continuación, se escoge mediante la función **next()** el píxel que es mejor candidato en el instante dado, cuyo índice se almacena en la variable **j**; se inicializa la variable **i** a 1; y se prepara el *flag* **i\_change** con valor 0, ya que en el bucle interior este *flag* tomará el valor de la variable **i** si ha habido una mejora al realizar la sustitución con el *endmember* **i**-ésimo.

Es entonces cuando el algoritmo entra en el bucle interior donde, una vez escogido un píxel candidato, se realizan las sucesivas sustituciones de éste por los considerados *endmembers* hasta el momento, probando si con la sustitución se es capaz de encerrar un mayor número de píxeles dentro del nuevo simplex construido.

Así, el algoritmo realiza en primer lugar la sustitución y calcula las abundancias que se obtendrían con el nuevo conjunto de supuestos *endmembers*. Aunque no se ha reflejado en el diagrama de la Figura 19, previo al cálculo de las

Abundancias, se realiza la comprobación de si la matriz  $\mathbf{EE}^t$  (siendo  $\mathbf{E}$  la matriz de *endmembers*) está bien condicionada mediante la función *rcond()* (como se hizo al definir la semilla) y considerando de igual modo que la matriz está bien condicionada si el valor devuelto por *rcond()* es mayor a  $5 \cdot 10^{-17}$ , pues en caso contrario se deshace la sustitución y se incrementa el valor de  $i$  pasando a la siguiente iteración del bucle interno.

Con las abundancias calculadas, la función **discard\_candidates()** comprueba qué píxeles quedan dentro del nuevo símplex y actualiza el vector **is\_candidate** asignando un 0 (descarte) a aquellos píxeles que queden dentro.

Es importante notar que consideramos un píxel en el interior de un símplex si todas sus abundancias son positivas, sin valorar la suma de éstas.

A continuación, conociendo el número de píxeles que quedan dentro del nuevo símplex, se comprueba si este número es mayor que **max\_pixels\_in\_simplex**. En caso afirmativo, se actualiza el valor de **max\_pixels\_in\_simplex** a este número y se asigna el valor de la variable  $i$  al *flag i\_change*, para, al salir del bucle interno, saber cuál es la sustitución que ha de hacerse efectiva.

Posteriormente, se deshace la sustitución realizada al principio del bucle interno y se incrementa el valor de  $i$  para volver al principio de dicho bucle.

Una vez realizadas las  $p$  sustituciones en las  $p$  iteraciones del bucle interno, el algoritmo sale de este bucle y comprueba si ha de realizar alguna sustitución efectiva a partir del valor **i\_change**. En caso de albergar un valor distinto de 0, indica qué posición del vector **E\_index** contiene el índice del *endmember* que debe ser sustituido por el píxel con índice  $j$ . La sustitución se hace efectiva, asignando a la posición  $i$ -ésima del vector **E\_index** el valor  $j$ .

Finalmente, el bucle externo concluye con el bloque que se ha denominado *detección de enganche*. Este bloque originalmente se encargaba de comprobar si el algoritmo está *enganchado* en un bucle infinito por haber vuelto a un valor de  $j$  sin que el estado del resto del algoritmo haya sido modificado, lo que indica la posibilidad de necesitar aumentar el valor de  $p$  para crear en lo sucesivo símplices con un mayor número de vértices, es decir, suponer que la imagen tiene al menos un *endmember* más.

El diagrama de la implementación final que se ha hecho de este bloque se muestra en la Figura 20. Así, en la implementación final se ofrece la posibilidad de no necesitar volver al mismo valor de  $j$  para presuponer que se ha de aumentar el valor de  $p$ , pues experimentalmente se ha comprobado que se puede definir un contador que, definido inicialmente a un valor establecido por el usuario y disminuyendo su valor en una unidad en cada iteración en que no se descarta ningún candidato, llegado a 0 se puede concluir que es adecuado aumentar el valor de  $p$ .

Siguiendo en detalle el diagrama mostrado en la Figura 20, el bloque comienza comprobando si el número de candidatos que quedan actualmente sin descartar es igual al número de candidatos que quedaban sin descartar en la iteración anterior. El caso afirmativo implicaría que en la iteración actual no ha sido posible descartar ningún píxel como candidato, pasando a disminuir en una unidad la variable **contador** que se ha inicializado previamente con el valor que contiene la variable **init\_counter** definida por el usuario. En caso negativo, es decir, que en la iteración actual sí ha sido posible descartar al menos un píxel como candidato, se actualiza la variable **previous\_number\_candidates** al número de candidatos actual, se reinicia la variable **counter** al valor **init\_counter** y se asigna un 1 al *flag* **less\_candidates\_flag**, que nos indica que en la última iteración se ha de descartar al menos un píxel; para finalmente volver al comienzo del bucle externo.

Siguiendo el caso afirmativo anterior, una vez decrementado el valor de la variable **counter** se comprueba si el contador ha alcanzado el valor 0. Si no ha alcanzado el valor 0, se retorna al comienzo del bucle externo. Si ha llegado a 0, se comprueba el valor del *flag* **less\_candidates\_flag**. Si este *flag* está a 0, el algoritmo ha finalizado, ya que la única posibilidad de alcanzar esta zona del algoritmo con valor 0 implica no haber sido capaz de descartar ningún candidato desde la última actualización de la variable  $p$ , lo que se ha tomado como condición para finalizar el algoritmo. Esta condición se ha establecido como solución para afrontar situaciones con ruido en las que se hace imposible descartar todos los píxeles. Sin embargo, esta condición puede ser objeto de un análisis más profundo, por lo que su estudio se relega como línea futura.

En caso de que el *flag less\_candidates\_flag* sea positivo, se continúa dentro del bloque *detección de enganche* y se aumenta el valor de **p** en una unidad.

Posteriormente, se comprueba si es necesario aumentar la variable **tolerance**, variable que se explicará en el apartado “Solución frente al ruido” de este capítulo.

Finalmente, antes de retornar al comienzo del bucle externo, se actualiza la variable **previous\_number\_candidates** con valor igual al número de candidatos actuales; se asigna valor 0 a la variable **less\_candidates\_flag**; se reinicia la variable **counter** al valor **init\_counter**; y se realiza una inicialización como la que se realiza al principio del algoritmo, pero respetando el vector **is\_candidate**, es decir, los píxeles que ya han sido descartados se mantienen descartados, nunca se podrán restablecer como candidatos píxeles que ya han sido descartados.

Por último, volviendo a la Figura 19, antes de devolver los *endmembers* encontrados y las abundancias calculadas, se realiza una discriminación de *endmembers* que puedan ser similares entre sí. Esto se debe a que, atendiendo al bloque *detección de enganche*, no se puede asegurar si se ha tomado un *endmember* de más, ya que puede suceder que llegados al número *p* de *endmembers* que realmente tiene la imagen, debido al ruido, no se han descartado todos los píxeles, con lo que podemos incurrir en un error al aumentar el número de *endmembers* a  $p + 1$ . Esta discriminación se hace atendiendo a la similitud de los ángulos espectrales y la correlación existente entre estas similitudes. Esta discriminación debe ser objeto de un análisis más profundo, por lo que su estudio se relega como línea futura.

Por último, si ha habido discriminación de *endmembers*, se realiza el cálculo definitivo de abundancias.

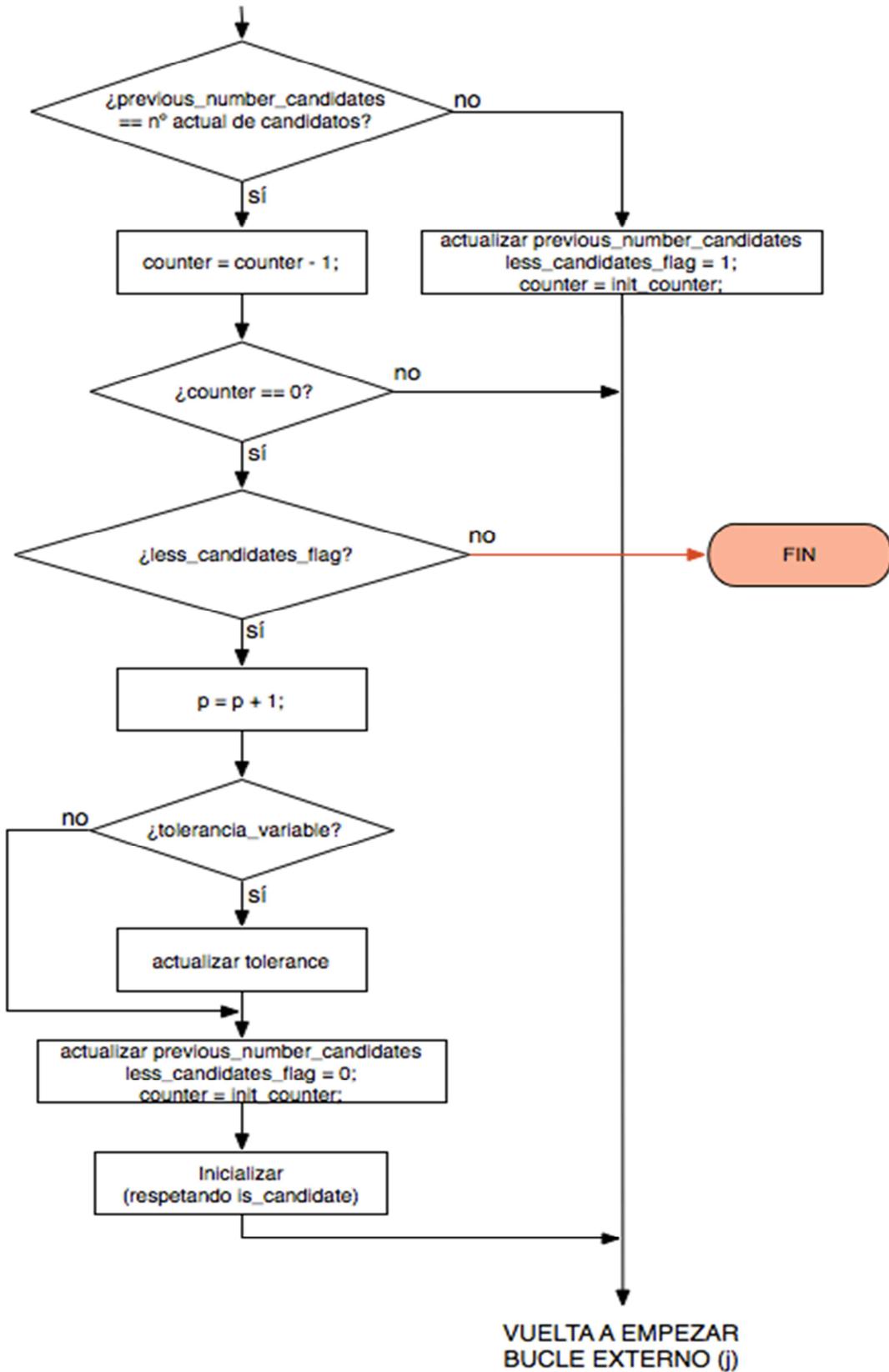


Figura 20.- Flujograma del bloque *Detección de enganche*

### 3.4.1 Solución frente al Ruido

En los casos en que la imagen está distorsionada con ruido, los vectores asociados a cada píxel quedan desplazados de su lugar original en el espacio  $N$ -dimensional. Así, los píxeles incluso pueden quedar fuera del hiperplano en el que se encuentra el simplex formado por los *endmembers*, es decir, fuera del subespacio definido por los *endmembers*.

Con la intención de controlar este efecto, se comenzó definiendo una tolerancia respecto a la suma de las abundancias obtenidas mediante el método de Mínimos Cuadrados. Se sabe que si un píxel está en el hiperplano definido por los *endmembers*, la suma de las abundancias es igual a 1, siempre que el número de píxeles tomados como *endmembers* sea el número real de *endmembers* que hay en la escena (dimensión del espacio definido por los *endmembers*) y que exista al menos la aportación de cada uno de los *endmembers* reales en alguno de los píxeles tomados como supuestos *endmembers*. Es por esto que se creó una tolerancia respecto a la suma, i.e., se calculaba la suma de las abundancias y si el resultado se encontraba en un margen preestablecido, se podía considerar que dicho píxel estaba en el hiperplano.

Sin embargo, en las imágenes reales también existe un fenómeno denominado fluctuación de luminancia, esto es, que la luminancia recibida de cada *endmember* en cada píxel queda multiplicada por un factor menor a 1, lo que implica que la suma de las abundancias de un píxel dado no suma 1.

Todo esto complicaría la implementación, razón por la cual la solución por la que se ha optado para la implementación del algoritmo se basa también en una tolerancia, pero definida sobre la distancia que existe entre un píxel y el píxel reconstruido a partir de las abundancias calculadas sobre el primero.

Como ya se explicó en el capítulo 2, el método por Mínimos Cuadrados ofrece la solución de menor distancia en caso de no existir solución exacta para la estimación de abundancias u ofrece la solución exacta en caso de existir. Así, el píxel reconstruido no es más que la proyección ortogonal del píxel en el espacio de los píxeles tomados como supuestos *endmembers*. Partiendo de esta

definición, se considera aceptable la solución si la distancia entre el píxel y el píxel reconstruido es menor a una tolerancia dada,

$$(\mathbf{E}\alpha - \mathbf{r}) < tolerance$$

siendo  $\mathbf{r}$  el píxel,  $\mathbf{E}$  la matriz de *endmembers* y  $\alpha$  el vector de tolerancias.

Con ello, se considerará que un píxel queda descartado para ser *endmember* si sus abundancias son todas positivas y si la distancia del píxel reconstruido a partir de estas abundancias es menor a la tolerancia. Esta tarea la realiza la función **discard\_candidates()**.

Es interesante comentar, que si bien es cierto que la definición de distancia  $d$  es

$$d = \sqrt{\sum_i (r_i - r'_i)^2}$$

siendo  $r$  el vector original y  $r'$  el vector reconstrucción se ha optado por realizar el cálculo de la “distancia cuadrática”, es decir,

$$d^2 = \sum_i (r_i - r'_i)^2$$

De este modo, se logra evitar realizar la raíz cuadrada, para lo cual sólo se requiere que el usuario defina la tolerancia de forma “cuadrática”.

Por otra parte, experimentalmente se ha constatado que en situaciones con ruido, a medida que aumentamos el número de *endmembers*, “contraintuitivamente” el número de píxeles que somos capaces de descartar es cada vez menor como se muestra en el capítulo 4. Ésta es la razón por la cual, en el bloque *detección de enganche* que se ha detallado en la Figura 20, se ha habilitado la modificación, variable y opcional por parte del usuario, de la tolerancia, procurando hacer el algoritmo más flexible a descartar píxeles a medida que se aumenta el valor  $\mathbf{p}$ , ya que de lo contrario, el algoritmo podría detenerse con un valor de  $\mathbf{p}$  inferior al número de *endmembers* que realmente existe en la escena.

Sería deseable conocer cuál debería ser el comportamiento óptimo de la tolerancia. Sin embargo, se trata de un aspecto relativamente complejo que puede ser objeto de un análisis más profundo, por lo que se posterga como línea futura.

En esta investigación, la variación de la tolerancia se ha definido como un incremento constante cada vez que se incrementa el valor de la variable **p**.



# Capítulo 4

## Resultados

---

### **4.1 Introducción**

Una vez se ha mostrado la hipótesis, las estructuras de datos y el algoritmo desarrollado, se hace necesario realizar los experimentos que arrojen evidencias que puedan o no probar la hipótesis.

Para ello se comienza explicando la naturaleza de las imágenes empleadas en los experimentos, cuál ha sido el esquema de los experimentos y una comparativa entre los resultados obtenidos por el algoritmo propuesto y una cadena de *Linear Unmixing* que se ha tomado como referencia.

### **4.2 Biblioteca de imágenes sintéticas**

Para la realización de los distintos experimentos se generó una biblioteca de imágenes sintéticas con la intención de que los resultados pudieran ser contrastados, ya que si se generara una imagen sintética para cada experimento, se impediría conocer si un cambio en los resultados se debe a una mejor implementación del algoritmo o a alguna particularidad de la nueva imagen

sintetizada. Además, el hecho de realizar los experimentos sobre imágenes sintéticas ofrece la posibilidad de conocer exactamente cuáles son los *endmembers* y abundancias que existen en la imagen. Esto no es posible con imágenes reales, además de que los tiempos de experimentación quedarían fuera de los plazos planteados para la realización del Trabajo de Fin de Máster.

La librería creada se divide fundamentalmente en 3 “sublibrerías”, cada una con un número de píxeles distintos: 30x30, 100x100 y 150x150.

A su vez, cada “sublibrería” se compone de 5 imágenes para cada combinación de los siguientes parámetros:

- Número de *endmembers*
  - Entre 3 y 21 *endmembers*, donde los  $n$  primeros *endmembers* siempre son los mismos.
- Relación Señal a Ruido SNR
  - 40, 60, 80 e  $\infty$  (sin ruido)

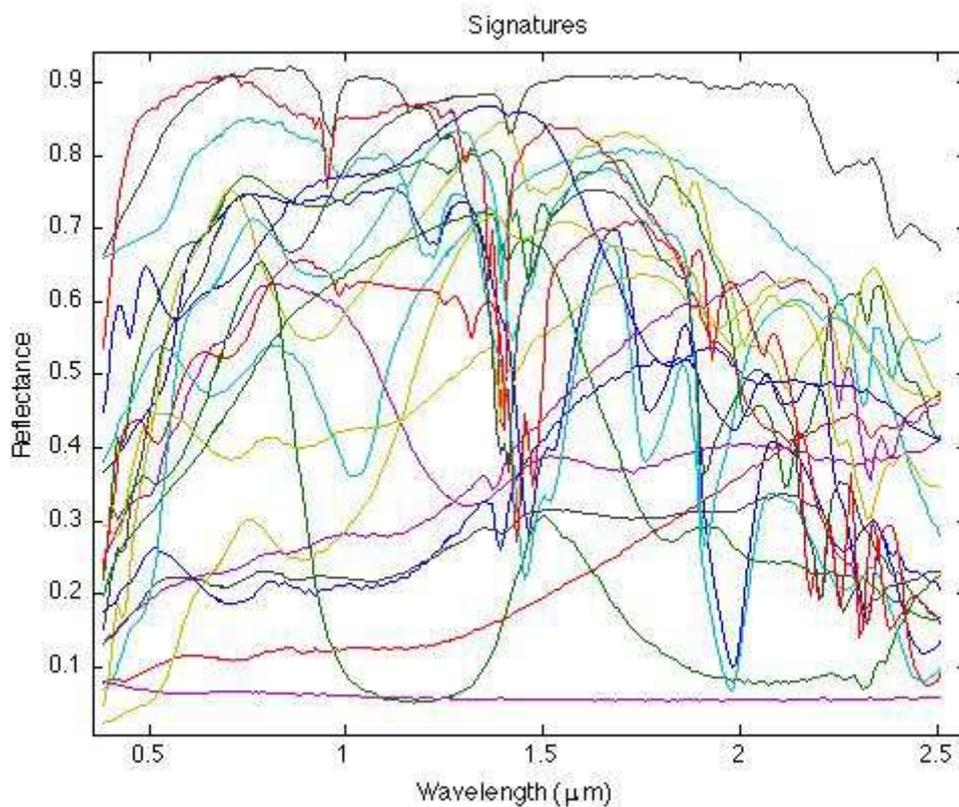
La razón por la que se generan 5 imágenes para cada combinación atiende a la intención de generar una pequeña muestra a partir de la cual generar la media de cada una de las variables de interés que se pretende estudiar.

En todas las imágenes existen píxeles puros, cuyos espectros coincide con uno de los 21 elementos seleccionados de la librería U.S. Geological Survey Library (USGS) [39], que son:

1. Carnallite NMNH98011
2. Ammonioalunite NMNH145596
3. Biotite HS28.3B
4. Actinolite NMNHR16485
5. Almandine WS478
6. Ammonio-jarosite SCR-NHJ
7. Andradite WS487
8. Antigorite NMNH96917 165u
9. Axinite HS342.3B
10. Brucite HS247.3B

11. Carnallite HS430.3B
12. Chlorite HS179.3B
13. Clinocllore NMNH83369
14. Clintonite NMNH126553
15. Corundum HS283.3B
16. Diaspore HS416.3B
17. Elbaite NMNH94217-1.a 659
18. Erionite+Merlinoit GDS144
19. Galena S26-39
20. Goethite WS220
21. Halloysite NMNH106236

En la Figura 21 se muestran las firmas espectrales de estos elementos.

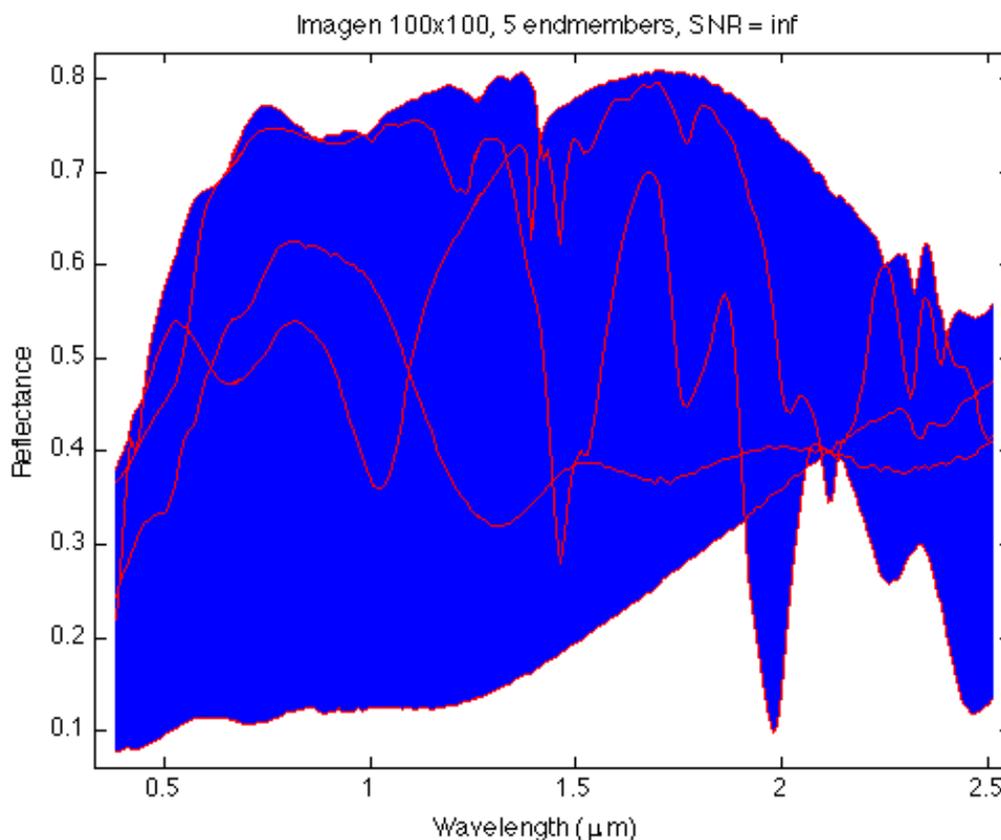


**Figura 21.- Espectros de los endmembers empleados**

El resto de píxeles se generan multiplicando el espectro de estos *endmembers* con unas abundancias generadas para cada píxel, mediante una función de

distribución tipo Dirichlet con un vector de parámetros definido como  $[ones(1, p) / p]$  donde  $p$  es el número de endmembers.

Como ejemplo, se muestra en la Figura 22 los espectros asociados a todos los píxeles de una imagen de tamaño 100x100, 5 endmembers y sin ruido, donde el espectro de los píxeles puros se muestra en color rojo.



**Figura 22.- Espectros de una imagen sin ruido de tamaño 100x100 con 5 endmembers**

Por otra parte, en la Figura 23 se muestra otra imagen también de tamaño 100x100, 5 endmembers, pero con una señal a ruido SNR de 40dB.

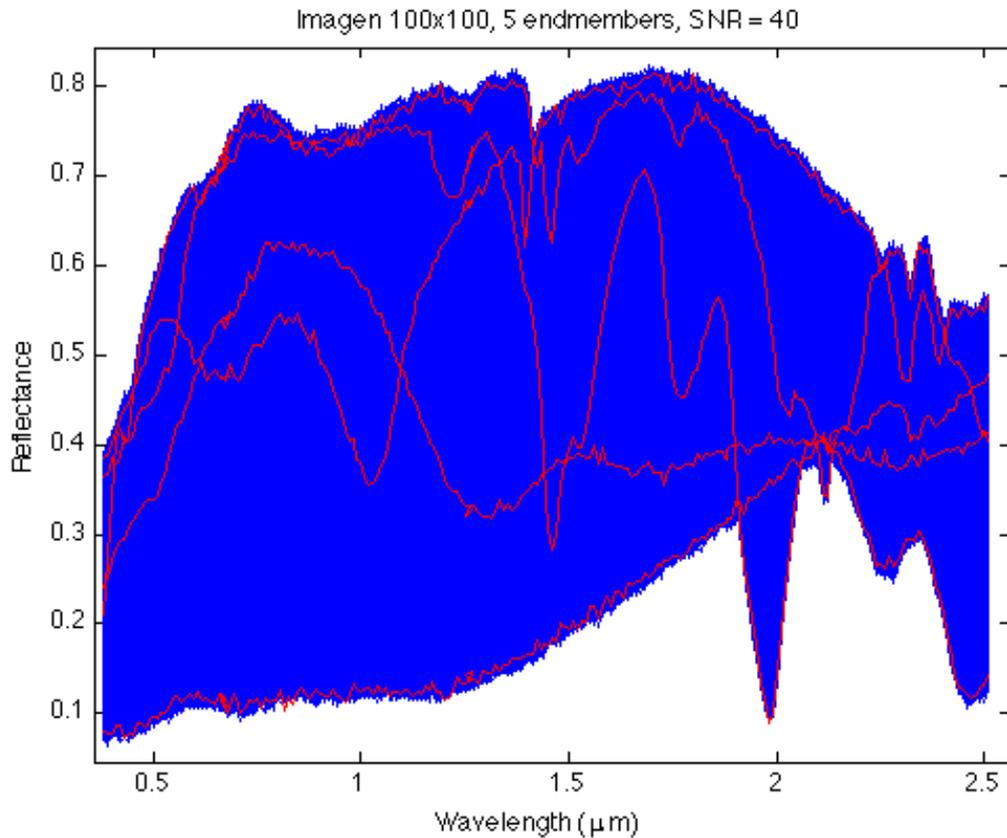


Figura 23.-Espectros de una imagen con ruido de tamaño 100x100 con 5 endmembers

Se aprecia rizado en todos los espectros. También se aprecia cómo componentes del espectro de algunos píxeles se ven desplazados llegando a salir fuera de la frontera en rojo definida por los *endmembers*.

### 4.3 Métricas cuantitativas

Las métricas empleadas para cuantificar la calidad de los resultados obtenidos son el ángulo espectral  $\theta$  entre *endmembers* y el *Root Mean Square Error* (RMSE) para las abundancias.

Se definen como:

$$\theta(\vec{a}, \vec{b}) = \arccos\left(\frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|}\right)$$

y

$$RMSE(a, b) = \sqrt{\frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M (a_{ij} - b_{ij})^2}$$

Así, dos *endmembers* se asemejan más cuanto menor es el ángulo espectral entre ambos, sin importar el módulo, ya que un píxel queda definido por su ángulo espectral y no por su módulo.

En cuanto al RMSE, dos matrices son más parecidas cuanto menor es el RMSE.

#### **4.4 Diseño del Experimento**

Las imágenes generadas son almacenadas junto a los *endmembers* y abundancias en un fichero con extensión “.mat”. Siguiendo el esquema que se muestra en la Figura 24, se realiza la ejecución a partir de la Matriz Imagen y la salida del algoritmo bajo test se compara con los *endmembers* y abundancias reales.

Al finalizar la ejecución de cada Imagen, se genera un informe que incluye el número de *endmembers* obtenido, los ángulos espectrales entre los *endmembers* obtenidos por el algoritmo y los *endmembers* reales, así como el RMSE (*Root-Mean Square Error*) entre las abundancias obtenidas por el algoritmo y las abundancias reales.

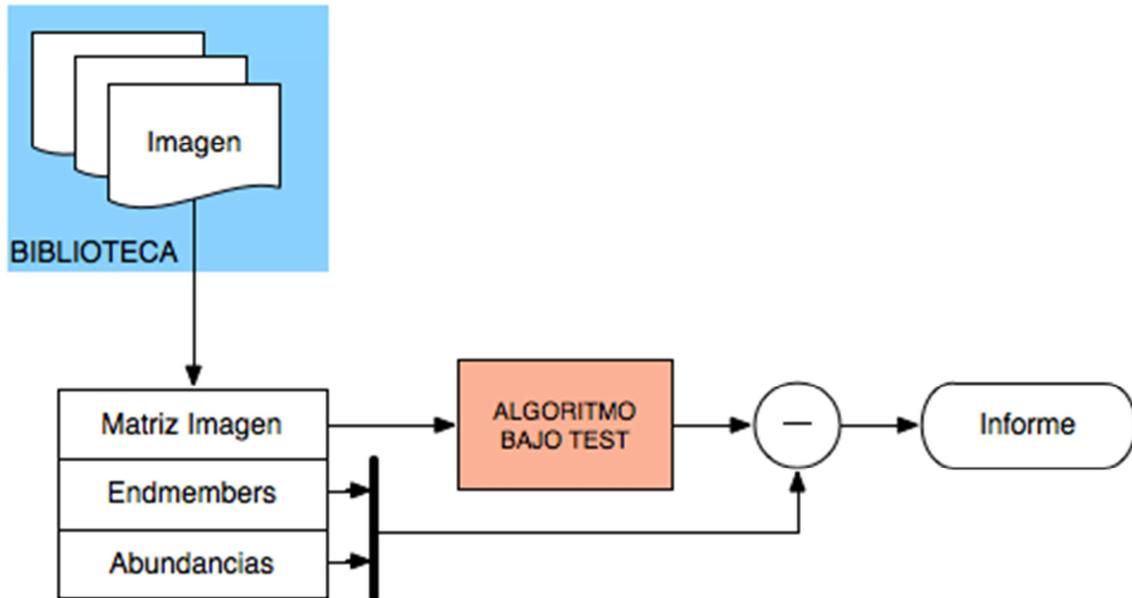


Figura 24.- Esquema de los experimentos

### 4.5 Comparativa

A continuación se realiza una descripción de varias comparativas realizadas entre el algoritmo desarrollado y una cadena de *Linear Unmixing* que se ha tomado como referencia y que se ha denominado “*Reference Linear Spectral Unmixing Chain*” (*Reference LSU Chain*). Esta cadena está compuesta por los algoritmos *Virtual Dimensionality* (VD), *Vertex Component Analysis* (VCA) y *Fully Constraint Least Squares* (FCLS). Estos algoritmos se describieron en el Capítulo 2. Con ello, la estructura de esta cadena de referencia es la que se muestra en la Figura 25.

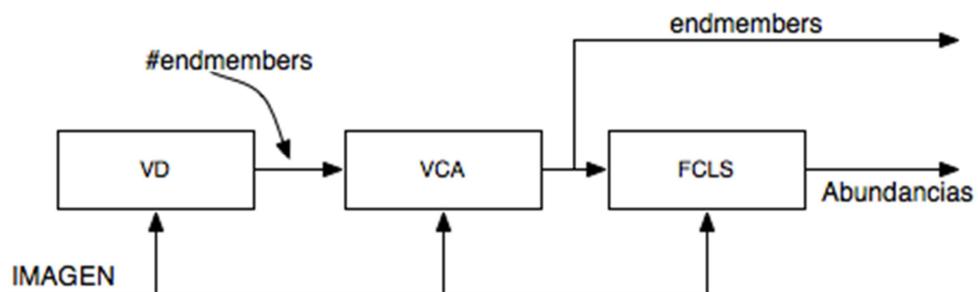


Figura 25.- Reference Linear Spectral Unmixing Chain

Así, el algoritmo VD tiene como parámetro lo que se ha definido como *false alarm rate* que se define típicamente como  $10^{-5}$ .

Por otra parte, el algoritmo desarrollado requiere la definición de algunos parámetros que se han ajustado a un valor fijo para todos los experimentos.

Los parámetros fijados son:

- **p**
  - Número de supuestos *endmembers* con que comienza el algoritmo.
    - Se define como 3, pues experimentalmente se ha comprobado que un simplex de 3 vértices es más flexible para realizar cambios, lo que abstrae al algoritmo de una mala definición de la semilla inicial. Con ello, el algoritmo evoluciona a partir de los *p* mejores *endmembers*, cuando **p** se incrementa.
- **tolerance**
  - Distancia máxima permitida entre un pixel y la reconstrucción del mismo para considerar que se encuentra en el espacio de los *endmembers*.
    - Se ha definido como 0.0025.
- **variable\_tolerance**
  - Valor binario que determina si la variable **tolerance** debe (1) o no (0) incrementarse cuando se incrementa la variable **p**.
    - Se ha definido como 1.
- **random\_seed**
  - Valor binario que determina si la semilla inicial se define de forma aleatoria o no. Se incluye para poder parametrizar el algoritmo en lo sucesivo y probar nuevas estrategias en la definición de la semilla.
    - Se ha definido como 1.
- **debug**
  - *Flag* binario que indica si se quiere reflejar por la salida estándar los valores que definen el estado instantáneo del algoritmo.
    - Se ha definido como 0.

- **debug\_plot**
  - *Flag* binario que indica si se quieren obtener por la salida estándar diferentes *plots* que muestran la evolución del algoritmo.
    - Se ha definido como 0.
- **init\_counter**
  - Valor con el que se reinicia la variable **counter** explicada en el capítulo 3.
    - Se ha definido como 1.
- **max\_iter**
  - Número máximo de iteraciones que se permite realizar al algoritmo.
    - Se ha definido como  $\infty$ .

Con ello, a continuación se muestran los resultados obtenidos para los siguientes parámetros:

- Número de *endmembers* estimado.
- Ángulo espectral entre los *endmembers* encontrados y los reales.
- RMSE de las abundancias calculadas.
- Tiempo de ejecución.

#### **4.5.1 Número de *endmembers* estimado**

A continuación se muestran los resultados obtenidos para el número de *endmembers*. Ha de recordarse que cada valor es la media de las 5 imágenes creadas en la biblioteca con igual número de *endmembers* reales e igual SNR.

Así, en la Figura 26 se muestra la media del número de *endmembers* estimados frente al número de *endmembers* reales, tanto por el algoritmo propuesto, como por la cadena de referencia, para imágenes de tamaño 30x30 píxeles.

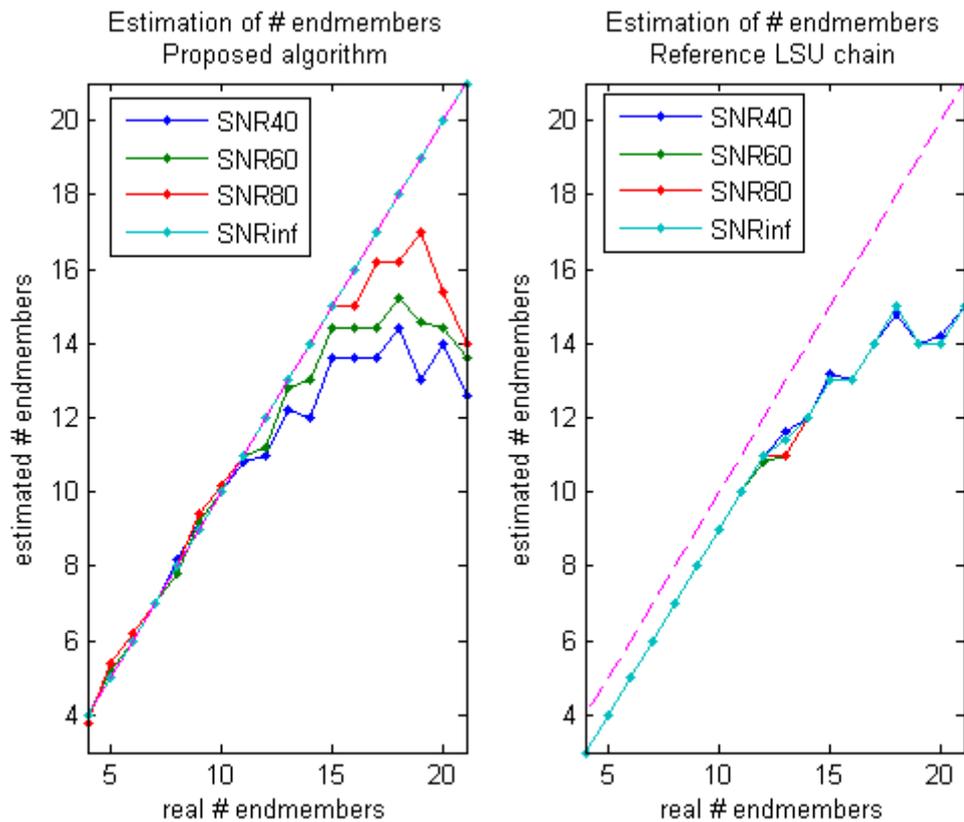


Figura 26.- Estimación del número de *endmembers* (30x30)

La línea discontinua color magenta refleja la recta de  $45^\circ$  (correspondencia 1:1) donde deberían encontrarse los resultados.

Se observa cómo el algoritmo propuesto sin ruido hace una estimación perfecta del número de *endmembers*, independientemente del número de *endmembers* real. Sin embargo, cuando el ruido y el número de *endmembers* real aumentan, se pierde la capacidad de realizar una buena estimación. Por otra parte, la cadena de referencia en ningún caso es capaz de encontrar el número de *endmembers* real, padeciendo además el mismo defecto cuando el número de *endmembers* se incrementa. Se hace interesante notar también cómo la estimación realizada por la cadena de referencia parece insensible al nivel de ruido.

En la Figura 27 se muestra también la estimación del número de *endmembers* pero en una imagen de tamaño 100x100 píxeles.

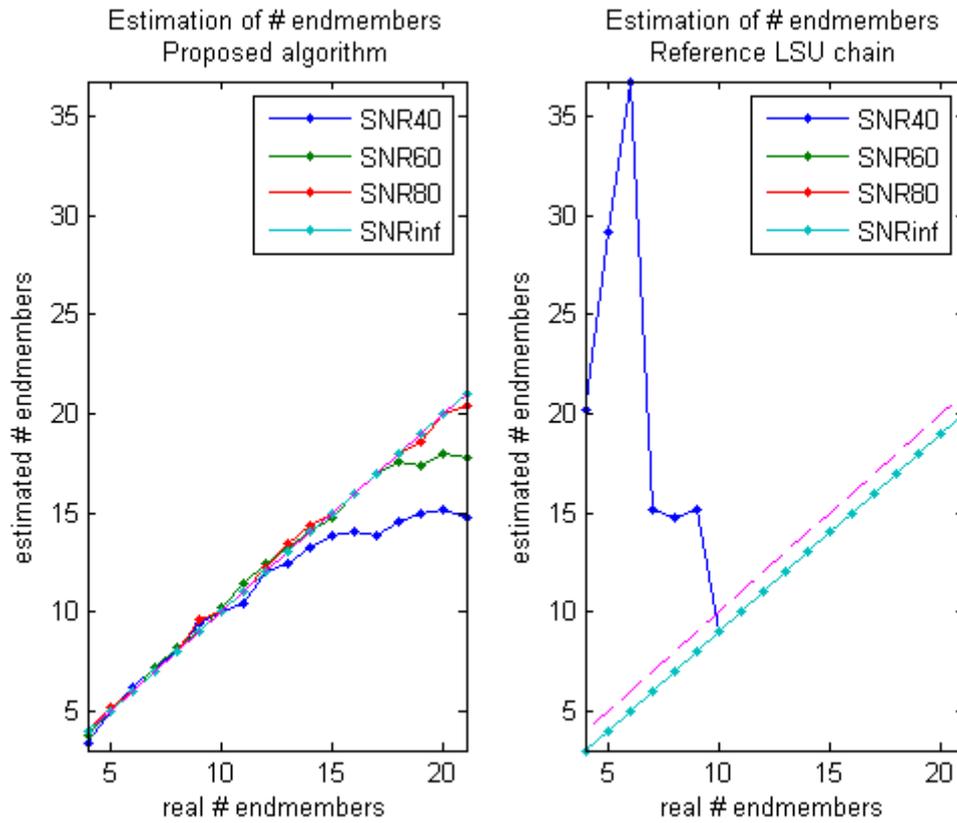


Figura 27.-Estimación del número de *endmembers* (100x100)

Ambos métodos han mejorado la estimación en los casos en que el número de *endmembers* es elevado. Sin embargo, la cadena de referencia siempre subestima en 1 *endmember* el número real de *endmembers*, salvo los casos en que el número de *endmembers* real es menor a 10 y la señal a ruido es de 40dB, donde se produce una sobrestimación exagerada.

Asimismo, en la Figura 28 se presentan los resultados obtenidos cuando la imagen es de tamaño 150x150 píxeles.

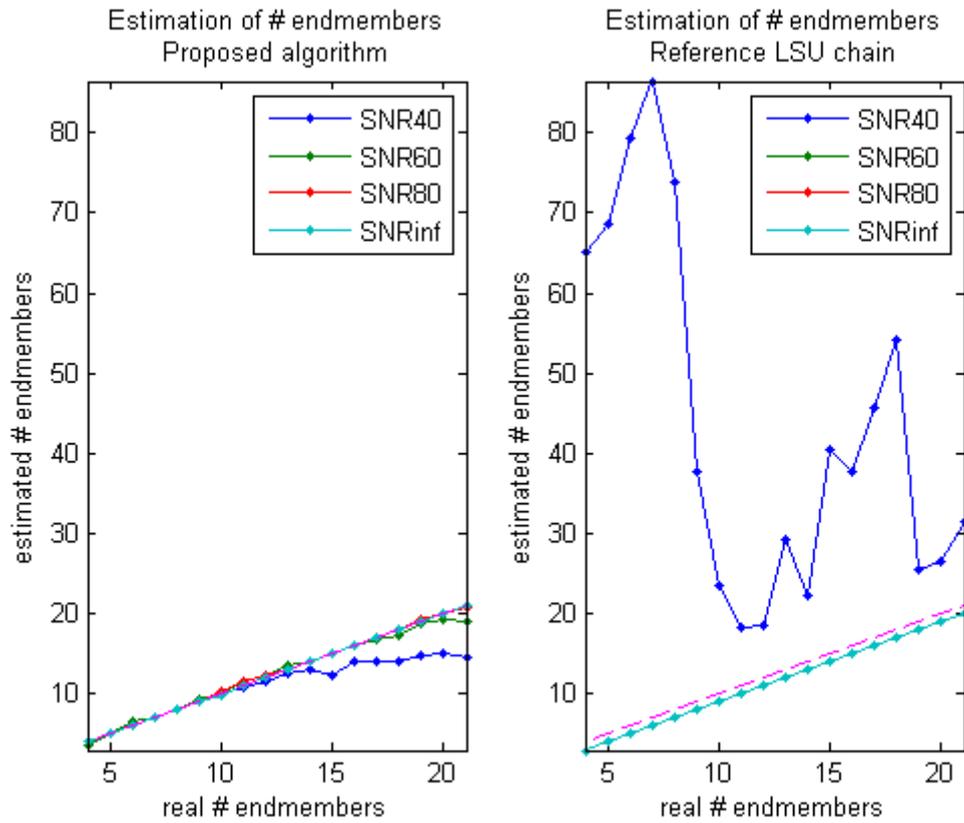


Figura 28.- Estimación del número de *endmembers* (150x150)

La estimación mejora salvo en el caso de la cadena de referencia cuando la relación señal/ruido es de 40dB, donde realiza una sobrestimación aún más exagerada que en el caso anterior. Además, la cadena de referencia sigue realizando una subestimación de 1 *endmember* para el resto de situaciones.

Esto parece indicar que la calidad de la estimación de *endmembers* es dependiente del número de píxeles, probablemente debido a la cantidad de información contenida en la imagen.

Con la intención de no pervertir las conclusiones, pues cuando realizamos media podemos estar cancelando errores que tengan en sentido opuesto, se muestra en la Figura 29, Figura 30 y Figura 31, los errores por experimento cometidos para la misma combinación de parámetros.

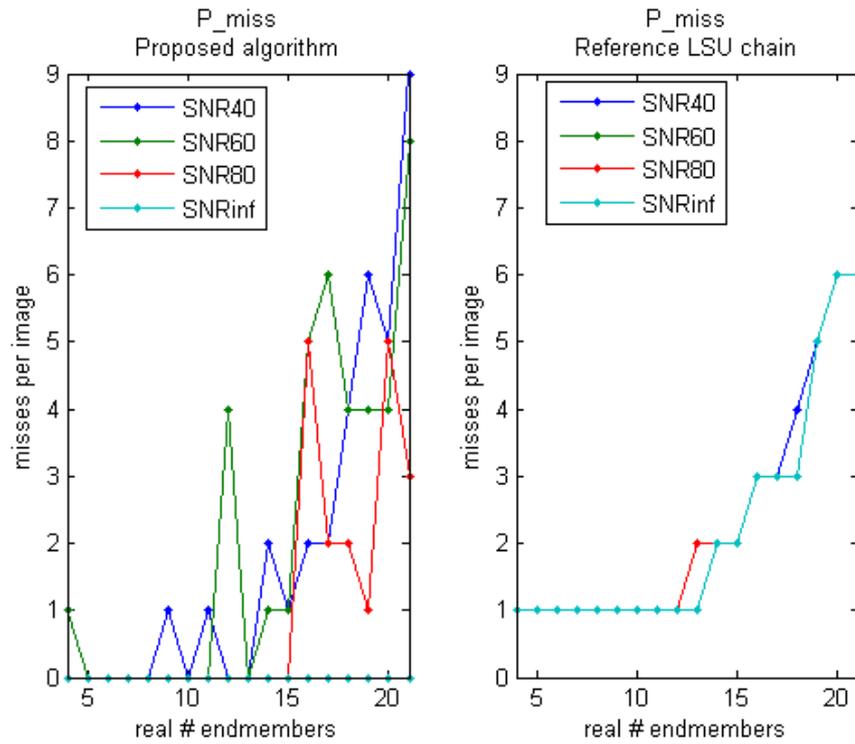


Figura 29.-Errores en la estimación del número de endmembers. (30x30)

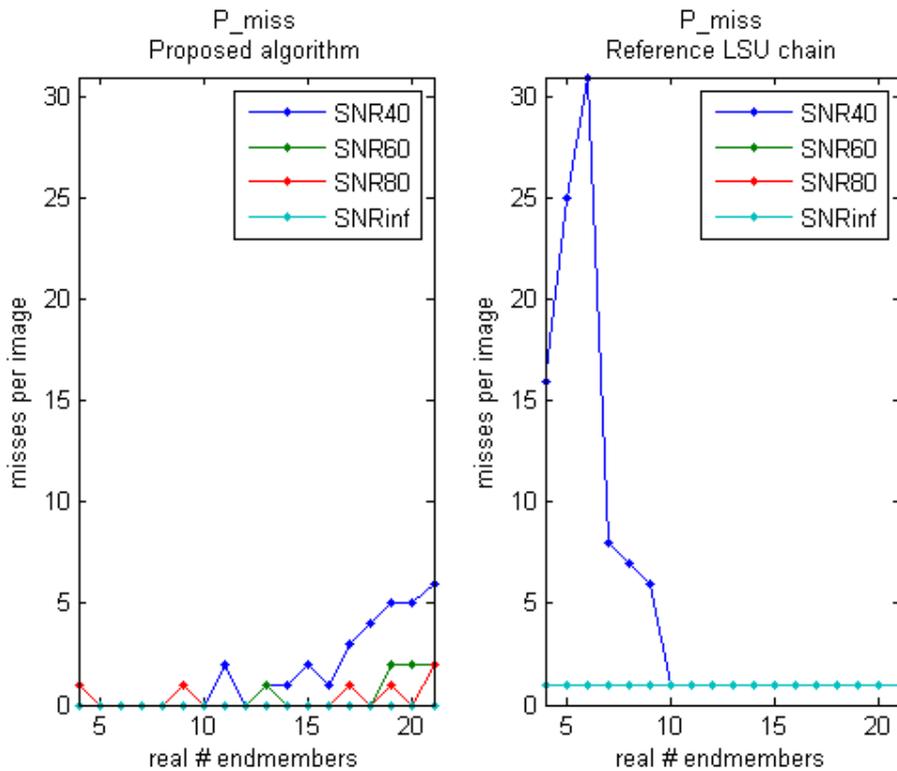


Figura 30.-Errores en la estimación del número de *endmembers* (100x100)

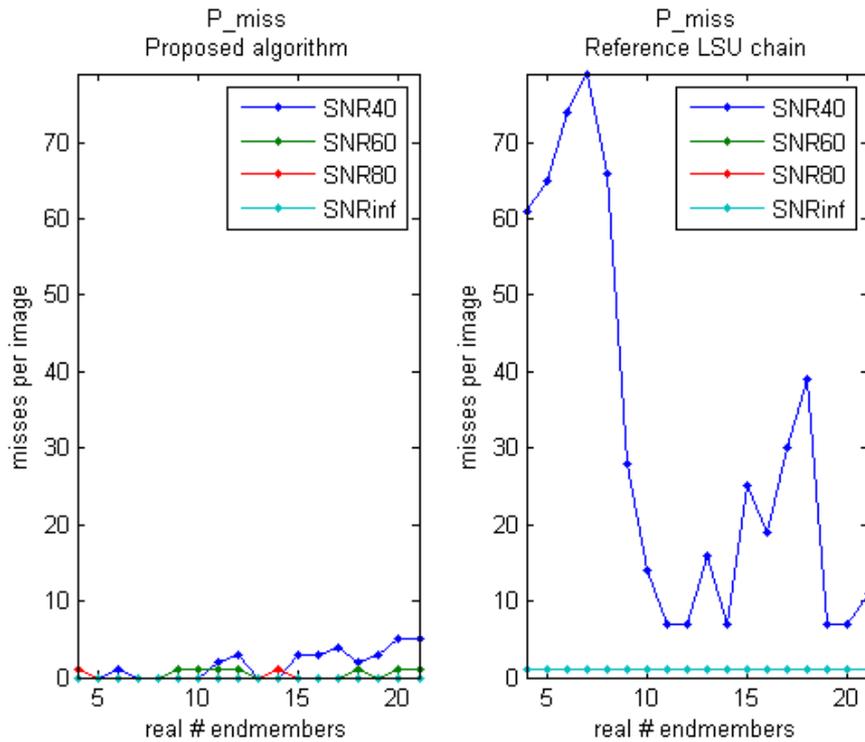


Figura 31.-Errores en la estimación del número de endmembers. (150x150)

Sin embargo, estas figuras confirman el comportamiento anterior, no aportando evidencias de comportamientos anómalos en ninguno de los experimentos.

En la Tabla 1 se resumen los errores promedio cometidos por ambos métodos para los distintos tamaños de imagen.

Tabla 1.- Resumen de los errores promedio cometidos en la estimación del número de endmembers

Tamaño Imagen	Algoritmo Propuesto	Cadena de Referencia
30x30	1.07142857142857	1.95238095238095
100x100	0.523809523809524	1.89285714285714
150x150	0.5	7.33333333333333

### 4.5.2 Ángulo espectral entre los endmembers encontrados y los reales

A continuación se muestran las medias de los ángulos espectrales entre los *endmembers* obtenidos y los *endmembers* reales. Las gráficas se refieren a la media de las 5 imágenes en el modo en que se describió previamente. Sin embargo, estos resultados se refieren a su vez a la media de los ángulos obtenidos en cada uno de los experimentos.

Por otra parte, dado que, como se ha mostrado en el apartado anterior, no siempre se obtiene el número de *endmembers* que realmente existen en la escena, los ángulos que a continuación se muestran son referidos a las  $n$  parejas más similares, donde  $n$  es el menor entre el número de *endmembers* que se estimó en el experimento y el número de *endmembers* real.

Así, en la Figura 32 se muestran los resultados obtenidos con las imágenes de tamaño 30x30.

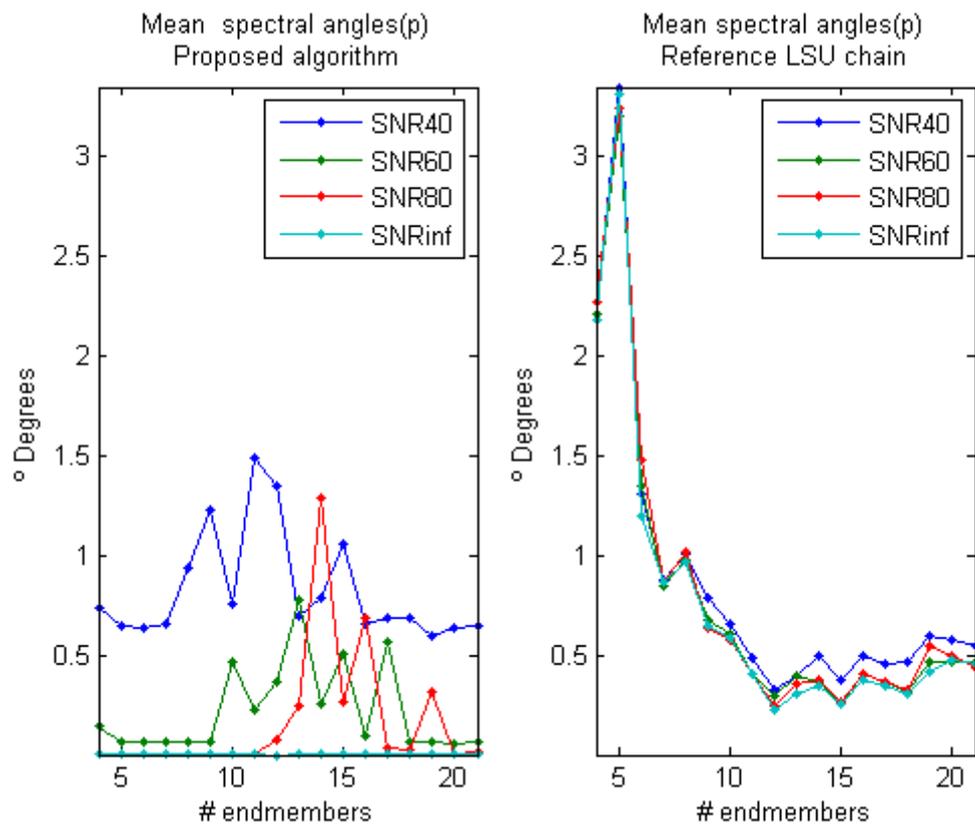


Figura 32.- Media de los ángulos espectrales error (30x30)

Se observa cómo el algoritmo propuesto es muy sensible al ruido, pues exhibe un peor comportamiento cuando la relación señal/ruido es de 40dB y un comportamiento excelente cuando no hay ruido en la imagen, pues los ángulos en este caso están en el orden de  $10^{-6}$ . Por otra parte, los ángulos espectrales en la cadena de referencia parecen insensibles al ruido, pues exhiben un comportamiento casi idéntico. Curiosamente, los ángulos son menores cuando la estimación del número de *endmembers* fue peor.

Dada la aparente sensibilidad del algoritmo propuesto al ruido, en la Figura 33 se muestran los mismos datos, pero con la relación señal/ruido en el eje de abscisas y el ángulo espectral en el eje de ordenadas, según el número de *endmembers* representado en cada una de las curvas.

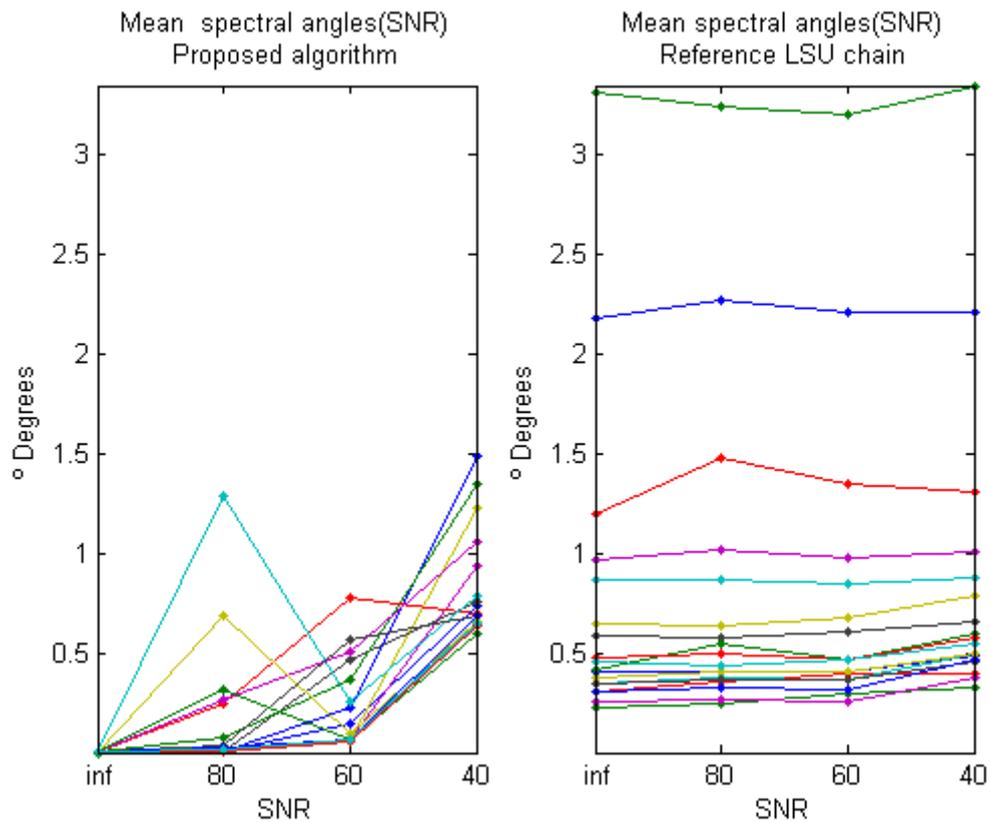


Figura 33.-Media de los ángulos espectrales error (30x30)

De este modo se comprueba la dependencia del ruido en el caso del algoritmo propuesto y la dependencia del número de *endmembers* en el que caso de la cadena de referencia.

El ángulo espectral del algoritmo propuesto depende fundamentalmente del ruido y en menor medida del número de *endmembers*, mientras que la cadena de referencia depende fundamentalmente del número de *endmembers*

En la Figura 34 se muestran los resultados para imágenes de tamaño 100x100.

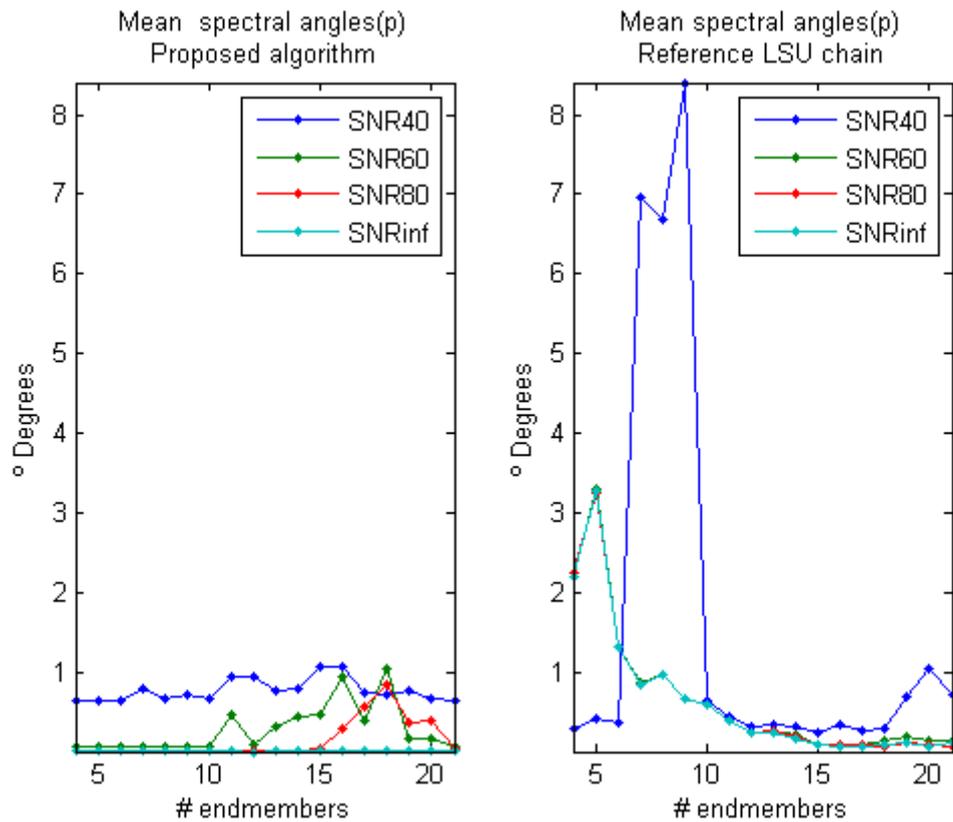


Figura 34.- Media de los ángulos espectrales error (100x100)

Se aprecia una mejora en todos los casos, salvo en la cadena de referencia cuando la relación señal/ruido es de 40dB y el número de *endmembers* es inferior a 10. Esto se debe a la sobrestimación del número de *endmembers* que realiza la cadena de referencia, como se apreció en Figura 27.

En la Figura 35 se muestran los resultados para imágenes de tamaño 150x150.

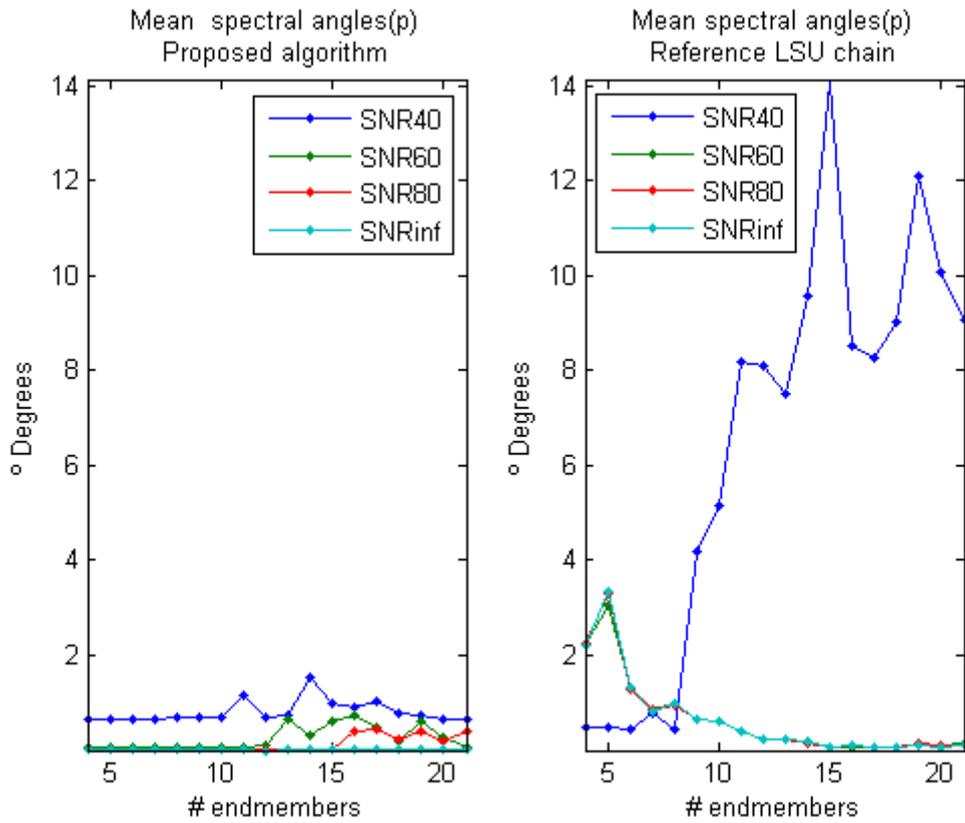


Figura 35.- Media de los ángulos espectrales error (150x150)

En el caso del algoritmo propuesto, los resultados mejoran. En el caso de la cadena de referencia, los resultados son pésimos cuando la relación señal/ruido es de 40dB y el número de *endmembers* es superior a 9, debido también a la sobrestimación exagerada que se hizo del número de *endmembers*.

En la Tabla 1 se resumen los ángulos espectrales promedio obtenidos por ambos métodos para los distintos tamaños de imagen.

Tabla 2.- Resumen de los ángulos espectrales promedio obtenidos

Tamaño Imagen	Algoritmo Propuesto	Cadena de Referencia
30x30	0.260429322528019	0.683351155205051
100x100	0.25542194098221	0.757525638812625
150x150	0.251673119148266	1.79804209185269

### 4.5.3 RMSE de las abundancias calculadas

En este apartado, se presenta la calidad de las abundancias calculadas mediante la métrica *Root-Mean Square Error* (RMSE). Como en el apartado anterior, las abundancias que se han tenido en cuenta en este apartado son aquellas que se refieren a los *endmembers* que se mostraron en el apartado anterior, que como recordamos, son los *endmembers* que más se asemejan a los reales dado que se puede estar sobrestimando o subestimando el número de estos.

Así, en la Figura 36 se muestran los resultados obtenidos para las imágenes de tamaño 30x30 píxeles.

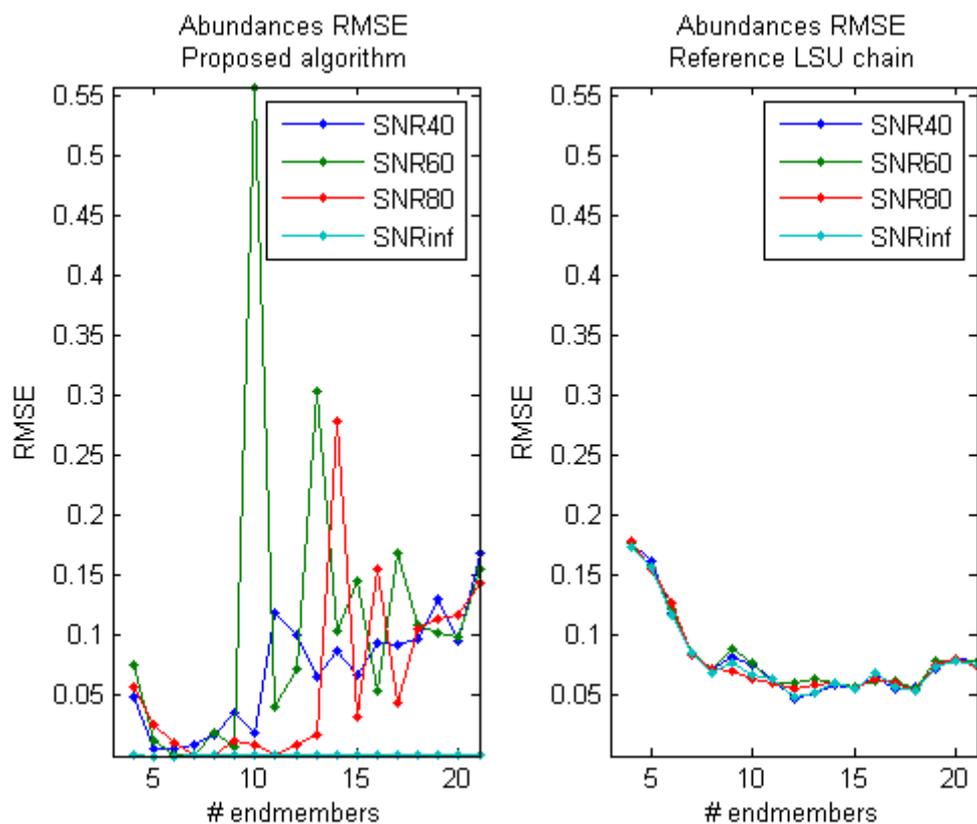


Figura 36.- RMSE de las abundancias calculadas (30x30)

Como en los apartados anteriores, el algoritmo propuesto en las situaciones en que no existe ruido, es capaz de calcular las abundancias de un modo exacto. Sin embargo, cuando la relación señal/ruido disminuye la calidad disminuye. Es interesante comprobar una vez más, la insensibilidad al ruido que exhibe el cálculo de abundancias de la cadena de referencia, además de la poca variación en cuanto al aumento en el número de *endmembers*. Además, para los casos en

que el número de *endmembers* es reducido, el algoritmo propuesto obtiene mejores abundancias y para los casos en que el número de *endmembers* es mayor, la cadena de referencia exhibe mejores resultados para imágenes con ruido.

En la Figura 37 se muestran los resultados obtenidos para las imágenes de tamaño 100x100 píxeles.

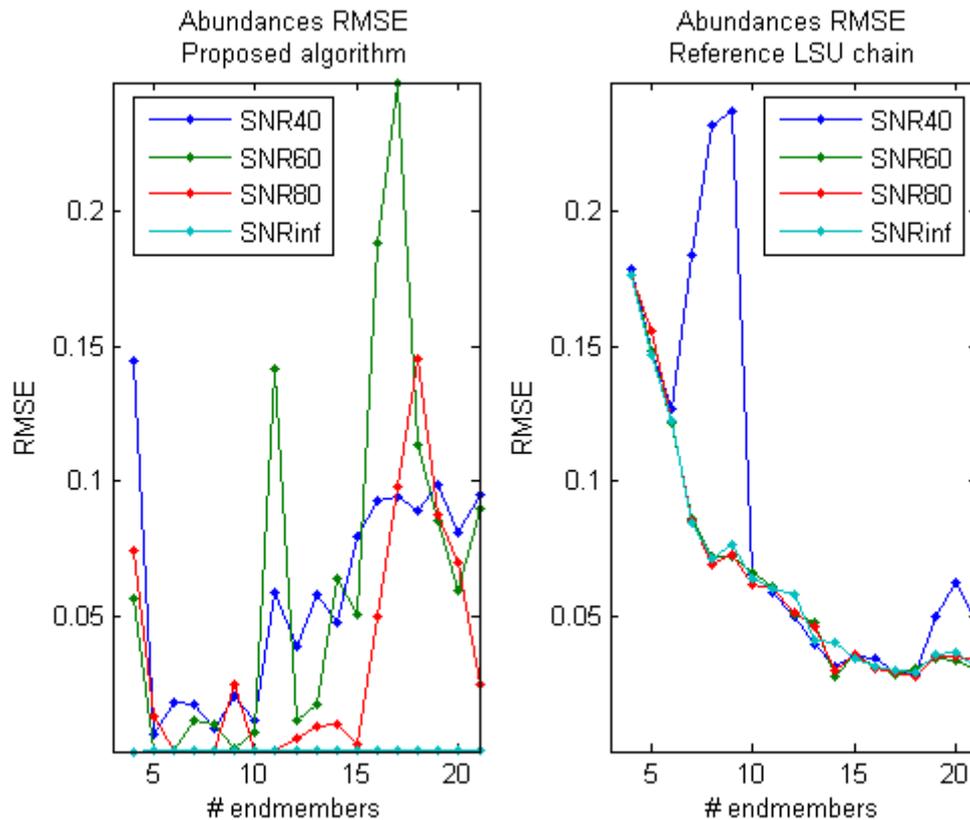


Figura 37.- RMSE de las abundancias calculadas (100x100)

Los resultados para el algoritmo propuesto han mejorado respecto al caso anterior y en la cadena de referencia se percibe un comportamiento anómalo para los casos en que la relación señal/ruido es de 40dB y el número de *endmembers* es 7, 8 y 9. Este comportamiento anómalo responde a la mala calidad de los *endmembers* calculados como se puede comprobar en la Figura 34 demostrando así la influencia evidente de la propagación de errores por la cadena de *Linear Unmixing*.

En la Figura 38 se muestran los resultados obtenidos para las imágenes de tamaño 150x150.

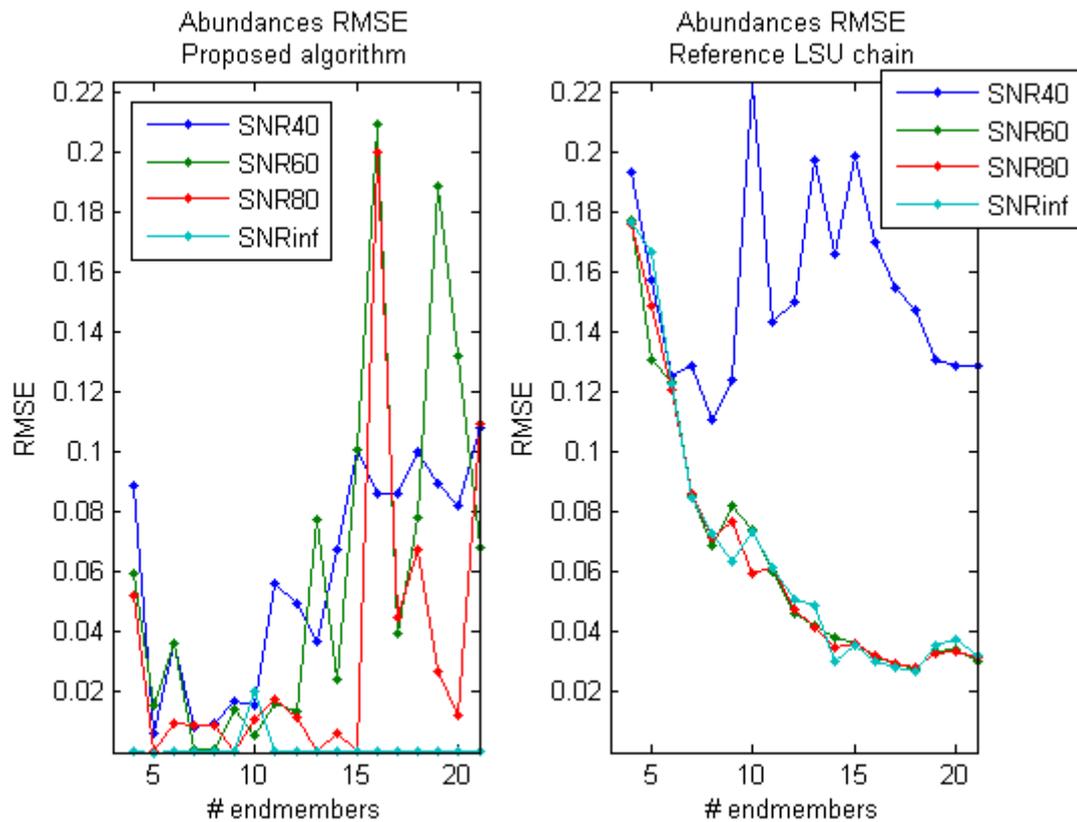


Figura 38.- RMSE de las abundancias calculadas (150x150)

En este caso, la cadena de referencia exhibe iguales resultados, salvo cuando la relación señal/ruido es de 40dB, debido a la propagación de la influencia de la mala estimación del número de *endmembers* que también afectó al cálculo de *endmembers*. Además, este efecto se suma a lo mal condicionado que está el sistema para resolver las abundancias cuando se realiza una sobrestimación del número de *endmembers*. Esto es así, porque entre los considerados *endmembers* por el algoritmo VCA en la cadena de referencia, deben de existir *endmembers* que son cuasi-linealmente dependientes, lo que provoca que la inversa de la matriz  $\mathbf{E}\mathbf{E}^T$  sea cuasi-singular, siendo  $\mathbf{E}$  la matriz de *endmembers*, provocando a su vez cálculos de poca precisión. De hecho, durante la ejecución del algoritmo FCLS, los mensajes de *warning* se sucedían sin tregua, dado que la función *rcond()* de *Matlab* devolvía valores del orden de  $10^{-19}$ .

El algoritmo propuesto sigue exhibiendo unos resultados excelentes para el caso en que no existe ruido en la imagen y resultados similares en el resto de situaciones, aunque la variabilidad parece haber aumentado.

En términos generales, cuando el número de *endmembers* aumenta y existe ruido, la cadena de referencia obtiene mejores resultados, excepto en situaciones de mucho ruido. Esto puede deberse a que el cálculo de abundancias que realiza el algoritmo propuesto es *unconstrained*, a diferencia del cálculo realizado por el algoritmo FCLS, que añade información al proceso al restringir las abundancias a valores positivos y forzando a que la suma de estas valga 1 en cada píxel.

En la Tabla 3 se resumen el RMSE promedio obtenidos por ambos métodos para los distintos tamaños de imagen.

**Tabla 3.- Resumen del RMSE promedio de las abundancias obtenidos**

<b>Tamaño Imagen</b>	<b>Algoritmo Propuesto</b>	<b>Cadena de Referencia</b>
30x30	0.0526430233175443	0.069735833243107
100x100	0.0337443344739889	0.0609918458424999
150x150	0.032561350884799	0.0745664176299564

#### **4.5.4 Tiempo de ejecución**

En este apartado se muestran los tiempos de ejecución del algoritmo propuesto frente a la cadena de referencia.

Así, en la Figura 39 se muestran los tiempos para realizar el desmezclado en una imagen de tamaño 30x30 píxeles.

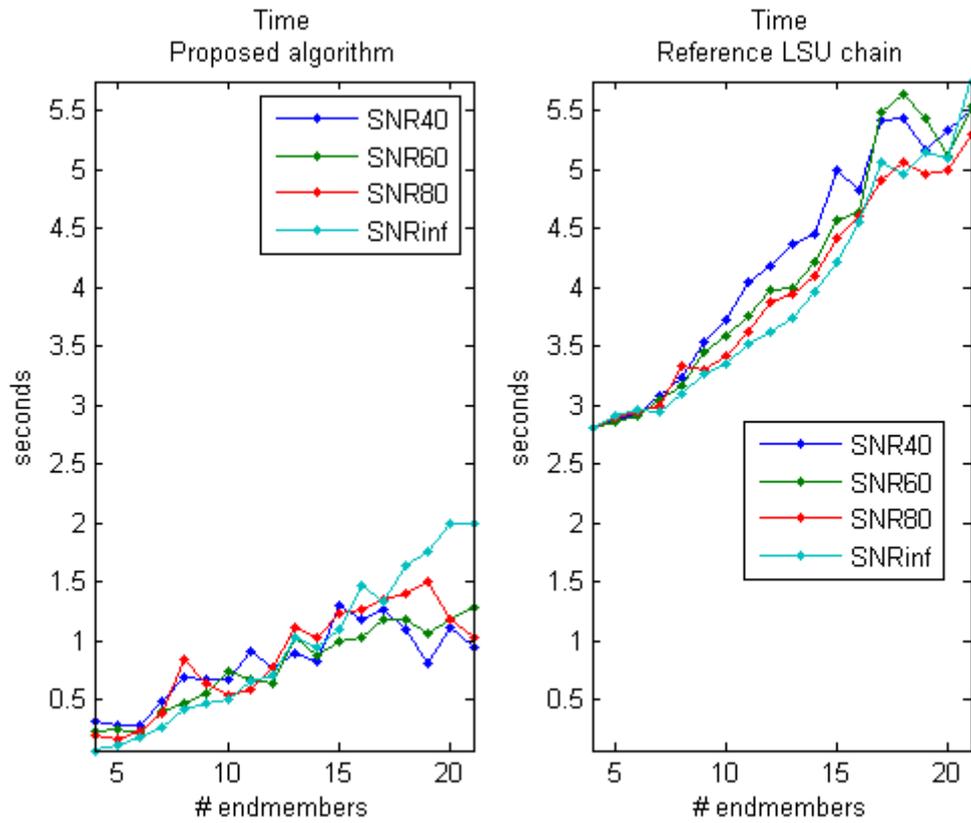


Figura 39.- Tiempos de ejecución (30x30)

Se comprueba cómo el algoritmo propuesto es mucho más rápido que la cadena de referencia, con un *speedup* promedio de  $\times 4.801$ . Además, aunque parezca a priori contradictorio, los tiempos de ejecución son menores para niveles de ruido crecientes en el caso del algoritmo propuesto. Esto se debe a que el número de *endmembers* estimados fue menor, lo que implica una menor carga computacional.

En la Figura 40 se muestran los tiempos obtenidos en los casos en que las imágenes son de tamaño 100x100 píxeles.

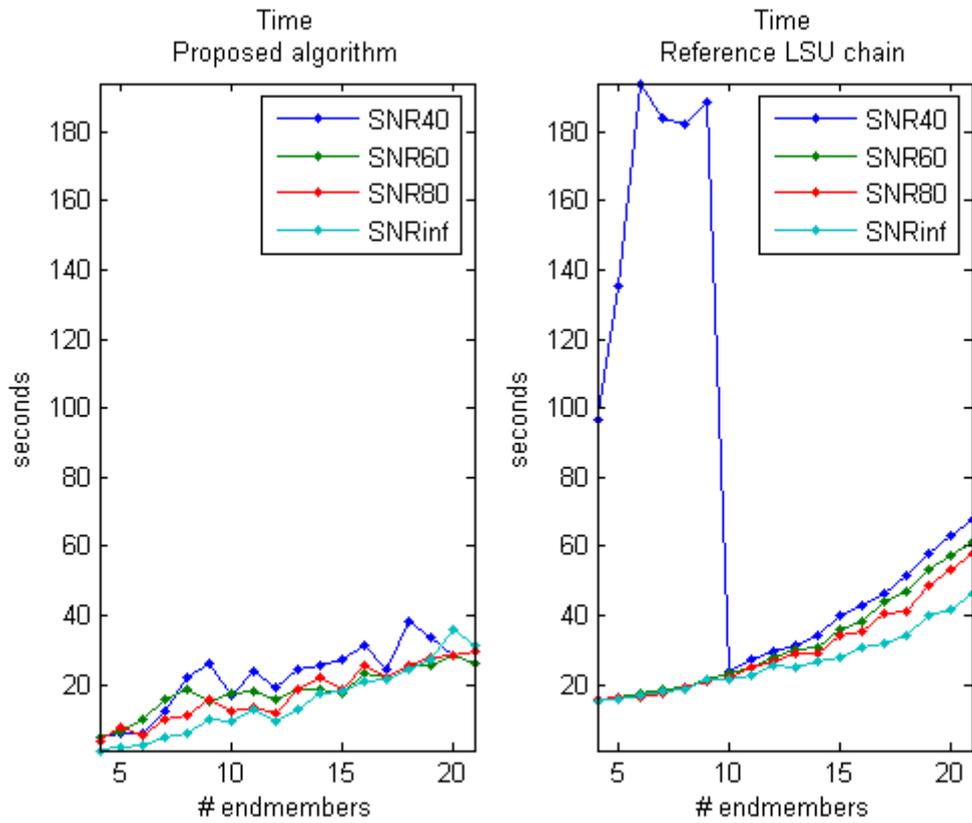


Figura 40.- Tiempos de ejecución (100x100)

Los tiempos aumentan, pero siguen siendo menores los del algoritmo propuesto, con un *speedup* promedio de  $\times 2.387$ . Además, la cadena de referencia exhibe unos tiempos muy elevados para los casos en que se sobrestimó el número de *endmembers* debido a la sobrestimación del número de *endmembers* que a su vez provocaba matrices cuasi-singulares que producen falta de precisión en los resultados, lo que puede perjudicar a la convergencia de los métodos iterativos empleados en el algoritmo FCLS.

Por otra parte, en la Figura 41 se muestran los resultados obtenidos cuando el tamaño de la imagen es de 150x150 píxeles.

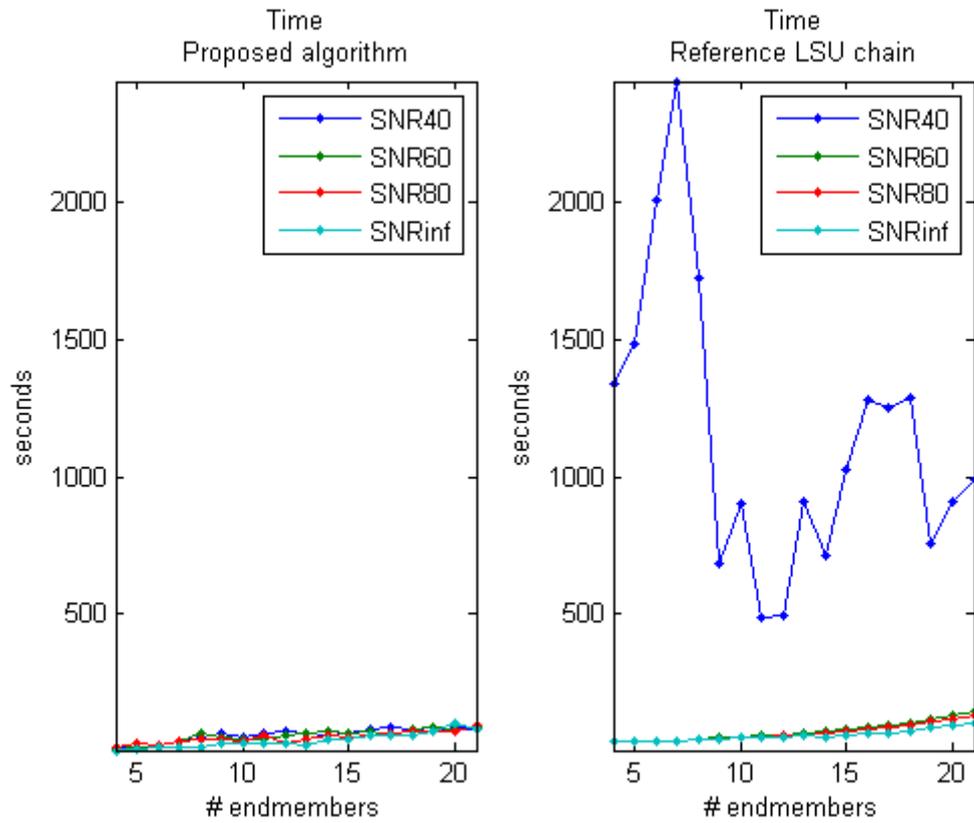


Figura 41.- Tiempos de ejecución (150x150)

Se observa un incremento súbito del tiempo necesario para los casos en que la relación señal/ruido es de 40dB en la cadena de referencia. Esto se debe al efecto comentado anteriormente. El *speedup* promedio logrado por el algoritmo propuesto es de x6.695.

En cualquier caso, los tiempos aumentan en el resto de situaciones. Así, en la Figura 42 se muestra en detalle los resultados anteriores dado que en la Figura 41 la escala se ve afectada por los valores de tiempo tan anómalos. Se observa cómo los tiempos son menores para el caso del algoritmo de referencia.

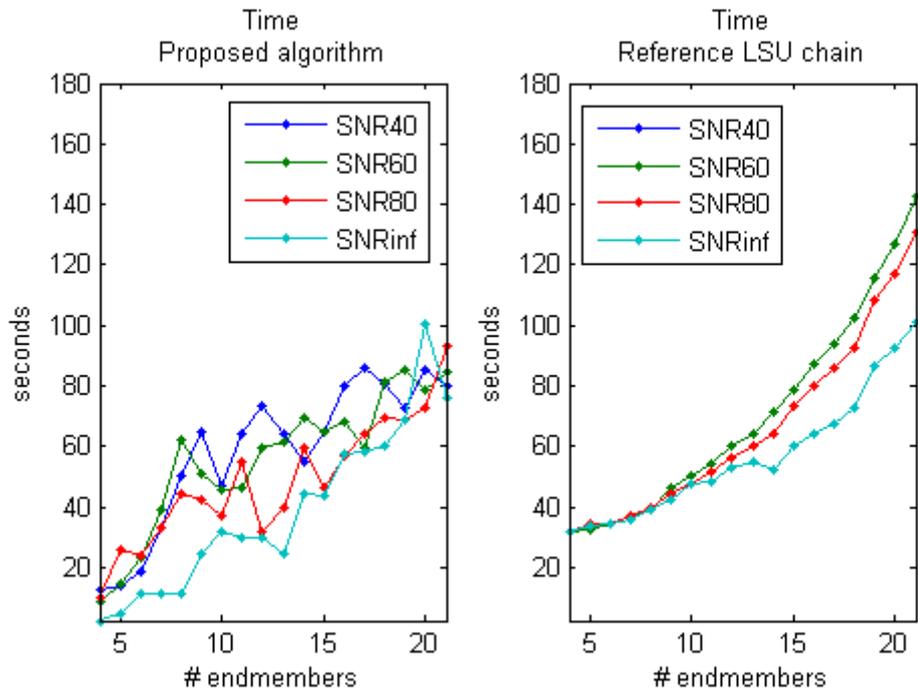


Figura 42.- Detalle de los tiempos de ejecución (150x150)

En la Tabla 4 se resumen los tiempos de ejecución promedios obtenidos por ambos métodos para los distintos tamaños de imagen.

Tabla 4.- Resumen de los tiempos de ejecución

Tamaño Imagen	Algoritmo Propuesto	Cadena de Referencia
30x30	0.725987192393214	3.48552448008358
100x100	15.5184571738366	37.0415347247611
150x150	43.0005489482965	287.880858410194

## 4.6 Evolución de Píxeles Descartados Durante la Ejecución del Algoritmo Propuesto

Se hace interesante mostrar cómo evoluciona el número de píxeles que se es capaz de encerrar en el interior de los sucesivos símplexes que se construyen, en función del número de *endmembers* que el algoritmo considera que existen en cada instante, esto es, el valor de la variable  $p$ .

Así, en la Figura 43 se muestra el comportamiento para una imagen sin ruido que contiene 12 *endmembers*.

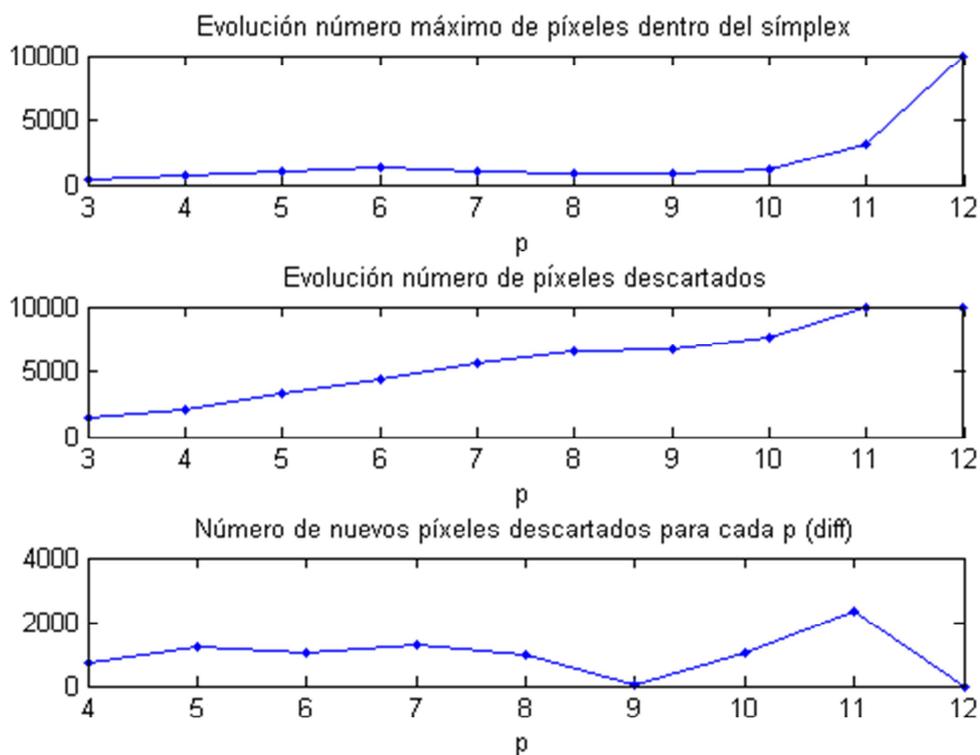


Figura 43.- Evolución del máximo número de píxeles dentro del símplex y del número de píxeles descartados sin ruido

Se observa cómo llegado el algoritmo a un número de *endmembers* próximo al número de *endmembers* real que existen en la imagen, se produce una especie de efecto avalancha como cabría esperar, esto es, se es capaz de incluir en el interior de los símplexes construídos un número cada vez mayor de píxeles, siguiendo un perfil exponencial.

Es incluso interesante notar cómo cuando el valor de  $p$  es igual a 11, ya se consideran descartados todos los píxeles aunque no se haya construido todavía ningún simplex que contenga todos los píxeles en su interior.

Se puede concluir que en la situación sin ruido, la solución obtenida es óptima.

Sin embargo, el comportamiento que exhibe el algoritmo cuando existe ruido en la imagen es “contraintuitivo”, como se muestra en la Figura 44.

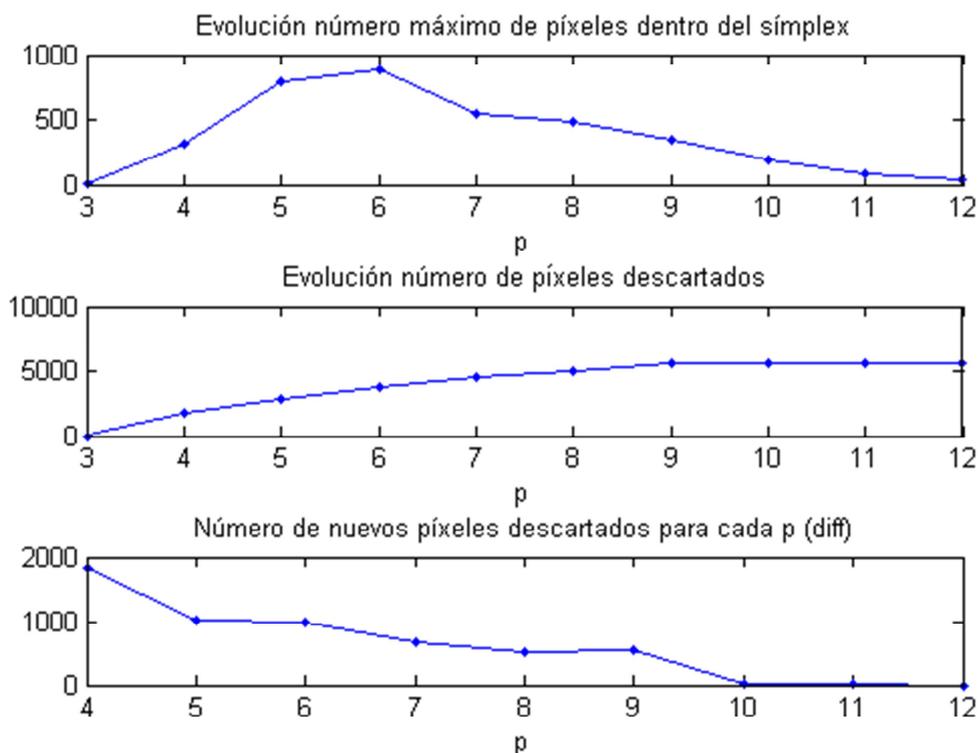


Figura 44.- Evolución del máximo número de píxeles dentro del simplex y del número de píxeles descartados con SNR=40dB

Se observa cómo el algoritmo es cada vez menos capaz de encerrar todos los píxeles dentro de los simples, cuando se aumenta el número de vértices de estos.

Además, no se llegan a descartar todos los píxeles, lo que indica que no se puede concluir que la solución a la que converge el algoritmo sea óptima.

Esto explica por qué se puede llegar a subestimar el número de *endmembers* como se puede apreciar en el apartado 4.5 y la importancia de estudiar el valor óptimo de la variable **tolerance**.

# Capítulo 5

## Conclusiones y Líneas Futuras

---

### **5.1 Conclusiones**

El objetivo principal de este trabajo de investigación es probar la hipótesis presentada en el capítulo 3. Vistos los resultados mostrados en el capítulo 4, se puede considerar que se han aportado suficientes evidencias experimentales para concluir que la hipótesis queda probada, esto es, el método propuesto es funcional, mostrando unos resultados comparables a una cadena de desmezclado estándar contrastada.

Sin embargo, se puede concluir también que, si bien los resultados son excelentes para situaciones en las que no existe ruido, ha de hacerse un estudio más profundo de los valores de las tolerancias que parametrizan el algoritmo para las situaciones en las que existe ruido. Esto responde al hecho de que la hipótesis surge de una idealización sin ruido para comprobar a posteriori si podía ser extrapolable a una situación con ruido.

Por otra parte, no deja de ser interesante notar que los tiempos de ejecución son comparativamente buenos. Por ello, se abre la posibilidad de replantear el código

buscando una eficiencia computacional que no se trató durante el desarrollo de la aplicación.

Por otra parte, el método presenta ciertas debilidades, además de las tolerancias ya mencionadas. Un ejemplo sería lo mal condicionado que suele quedar el sistema cuando tratamos de averiguar si un píxel se encuentra en el interior de un simplex, debido a que los *endmembers* distan mucho de ser ortogonales entre sí, como se mostró en el capítulo 3, donde se supone que un sistema está bien condicionado cuando la función *rcond()* de *Matlab* devuelve típicamente valores del orden de  $10^{-17}$ . Asimismo, la evolución del algoritmo mostrada en el apartado 4.6 explica la dificultad que existe en escenarios ruidosos para concluir que los píxeles están dentro del simplex, lo que explica por qué se subestima el número de *endmembers* al aumentar el número real de estos.

## 5.2 Líneas Futuras

Una vez probada la hipótesis, se cree oportuno continuar el trabajo desarrollado. Por lo tanto, se lista a continuación una sucesión de ideas que se han planteado durante la realización del trabajo y que sería interesante probar en la continuación del mismo.

- Realizar el estudio en imágenes en que el índice de pureza de los píxeles sea menor, esto es, no existan píxeles con distancias tan cercanas a los *endmembers*.
- Estudio de los parámetros óptimos del algoritmo. Comprobar si atienden a información que pueda ser extraída de la imagen, como pueda ser el ruido. Estudiar la posibilidad de hacer que los parámetros sean adaptativos.
- Aplicar restricciones a las abundancias, pues por la naturaleza del problema se sabe que las abundancias no pueden ser negativas y deben sumar 1.
- Probar técnicas de estimación o eliminación de ruido.
- Conocer la evolución del RMSE entre la imagen reconstruida a partir de los *endmembers* y abundancias que se tiene en cada momento y la imagen original. Puede ofrecer un criterio de parada.

- Estudiar la condición de parada del algoritmo.
- Diseñar una estrategia para la definición de la semilla inicial. Por ejemplo, tomar como *endmember* los píxeles más y menos brillantes de la imagen, esto es, el más y menos lejano al origen.
- Probar técnicas de regularización de matrices ya que se puede lograr que siempre exista la inversa de una matriz regularizada de un modo óptimo. Esto depende de si existe o no la posibilidad de realizar la desregularización y adaptar el resto de consecuencias.
- Desarrollar una aplicación lineal que proyecte los píxeles en el espacio de los *endmembers* con un ángulo espectral igual al que tiene cada píxel, no de forma ortogonal al espacio de los *endmembers* como realiza el método por Mínimos Cuadrados. Estudiar si puede provocar una mejor estimación de abundancias ya que un píxel contiene la misma información si su módulo varía y su ángulo espectral se mantiene constante.
- Realizar reducción dimensional de la imagen como preprocesado previo a la ejecución del algoritmo. Esto implicaría realizar una estimación del número de *endmembers* por lo que ha de estudiarse su adecuación a este algoritmo.
- Definir una tolerancia variable dependiente de la relación señal/ruido que pueda existir en cada banda. Así, existen curvas que caracterizan la relación señal/ruido que existe en los sensores hiperespectrales, que podrían ser usados en la tolerancia del algoritmo propuesto.



## Bibliografía

---

- [1] D. MANOLAKIS Y G. SHAW, "Detection Algorithms for Hyperspectral Imaging Applications", *IEEE Signal Processing Magazine*, January 2002.
- [2] Disponible en: <http://aviris.jpl.nasa.gov/> [online, última visita mayo de 2012].
- [3] Disponible en: [http://www.esa.int/esaEO/SEMFVBIK97G\\_index\\_0.html](http://www.esa.int/esaEO/SEMFVBIK97G_index_0.html) [online, última visita junio de 2012].
- [4] Disponible en: [http://www.asi.it/en/activity/earth\\_observation/prisma\\_](http://www.asi.it/en/activity/earth_observation/prisma_) [online, última visita junio de 2012].
- [5] Disponible en: <http://www.enmap.org/> [online, última visita mayo de 2012].
- [6] Disponible en: <http://hyspirc.jpl.nasa.gov/> [online, última visita mayo de 2012].
- [7] C.-I. CHANG, "Hyperspectral Imaging: Techniques for spectral detection and classification", *Kluwer Academic / Plenum Publishers*, 2003.
- [8] M. O. SMITH, P. E. JOHNSON Y J. B. ADAMS, "Quantitative determination of mineral types and abundances from reflectance spectra using principal component analysis", *J. Geophys. Res.*, vol. 90, no.2, pp. C797-C804, octubre de 1985.
- [9] A. A. GREEN, "A transformation for ordering multispectral data in terms of image quality with implications for noise removal", *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 1, pp. 65-74, mayo de 1988.
- [10] C.-I. CHANG Y Q. DU, "Estimation of number of spectrally distinct signal sources in hyperspectral imagery", *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 3, pp. 608-619, marzo de 2004.
- [11] J. M. BIOUSCAS-DIAS Y J. M. P. NASCIMENTO, "Hyperspectral subspace identification", *IEEE Trans. Geosci. Rem. Sens.*, vol. 46, no. 8, pp. 2435-2445, 2008.
- [12] J. W. BOARDMAN, F. A. KRUSE Y R. O. GREEN, "Mapping target signatures via partial unmixing of AVIRIS data", in *Proc. Summ. Jpl Air-borne Earth Sci. Workshop*, Pasadena, CA, 9-14 diciembre de 1995, vol. 1, pp. 23-26.
- [13] M. E. WINTER, "N-findr: An algorithm for fast autonomous spectral end-member determination in hyperspectral data", in *Proc. SPIE Conf. Imaging Spectrometry*, Pasadena, CA, Octubre de 1999, pp. 266-275.

- [14] W. XIONG, C.-I. CHANG, C.-C. WU, K. KALPAKIS Y H. M. CHEN, "Fast algorithms to implement N-FINDR for hyperspectral endmember extraction", *IEEE Remote Sens*, vol. 4, no. 3, pp. 545-564, Septiembre de 2011
- [15] J. M. P. NASCIMENTO Y J. M. BIOUSCAS-DIAS, "Vertex Component Analysis: A fast algorithm to unmix hyperspectral data", *IEEE Trans. Geosci. Remote Sens.*, vol 43, no. 4, pp. 898-910, abril de 2005.
- [16] A. IFARRAGUERRI Y C.-I. CHANG, "Multispectral and hyperspectral image analysis with convex cones", *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 2, pp. 756-770, marzo de 1999.
- [17] D. HEINZ Y C.-I. CHANG, "Fully constrained least squares linear mixture analysis for material quantification in hyperspectral imagery", *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 3, pp. 529-545, marzo de 2001.
- [18] A. ZYMNIS, S.-J. KIM, J. SKAF, M. PARENTE Y S. BOYD, "Hyperspectral image unmixing via alternating projected subgradients", in *Proc. 41st Asilomar Conf. Signals, Syst.*, Pacific Grove, CA, 4-7 de noviembre de 2007.
- [19] M. BERMAN, H. KIIVERI, R. LAGERSTROM, A. ERNST, R. DUNNE Y J.F. HUNTINGTON, "ICE: A statistical approach to identifying endmembers in hyperspectral images", *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 10, pp. 2085-2095, octubre de 2004.
- [20] D. LEE Y H. S. SEUNG, "Learning the parts of objects by non-negative matrix factorization", *Nature*, vol. 401, pp. 788-791, octubre de 1999.
- [21] V. P. PAUCA, J. PIPER Y R. J. PLEMMONS, "Nonnegative matrix factorization for spectral data analysis", *Linear Algebra Appl.*, vol. 1, no. 416. Pp. 29-47, 2006.
- [22] N. DOBIGEON, S. MOUSSAOUI, M. COULON, J.-Y. TOURNERET Y A. O. HERO, "Joint Bayesian endmember extraction and linear unmixing for hyperspectral imagery", *IEEE Trans. Signal Process.*, vol. 57, no. 11, noviembre de 2009.
- [23] M. D. CRAIG, "Minimum-volume transforms for remotely sensed data", *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 3, pp. 542-552, mayo de 1994.
- [24] Disponible en: <http://es.wikipedia.org/wiki/S%C3%ADmplex> [online, última visita junio de 2012]
- [25] Disponible en: <http://landsat.gsfc.nasa.gov/about/tm.html> [online, última visita mayo de 2012]
- [26] M. FARAKLIOTI, M. PETROU, "Illumination invariant unmixing of sets of mixed pixels". *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, pp. 2227- 2234, 2001.
- [27] T.M. TU, H.C. SHYU, C.H.LEE,C.-I CHANG, "An oblique subspace projection approach for mixed pixel classification in hyperspectral images", *Pattern Recognition*, vol. 32, pp. 1399-1408, 1999.
- [28] J.M. CHEN, "Spatial Scaling of a Remotely Sensed Surface Parameter by Contexture", *Remote Sensing of Environment*, vol. 69, pp. 30-42, 1999.
- [29] N. KESHAVA, J.F. MUSTARD, "Spectral unmixing", *IEEE Signal Processing Magazine*, vol. 19, pp. 44-57, 2002.
- [30] P.E. JOHNSON, M.O. SMITH, S. TAYLOR-GEORGE, J.B. ADAMS, "A Semi-Empirical Method for Analysis of the Reflectance Spectra of Binary Mineral Mixtures". *Journal of Geophysical Research*, vol. 99, pp. 3557-3561, 1983.
- [31] J. W. BOARDMAN, F. A. KRUSE, and R. O. GREEN, "Mapping Target Signatures Via Partial Unmixing of Aviris Data", *Proc. VII NASA/JPL Airborne Earth Science Workshop*, pp. 23-26, 1995.

- 
- [32] F.A. KRUSE, "Spectral Identification of Image Endmembers Determined from AVIRIS Data", *Proc. VII NASA/JPL Airborne Earth Science Workshop*, 1998.
- [33] J.J. Settle, "On the Relationship between Spectral Unmixing and Subspace Projection", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, pp. 1045–1046, 1996.
- [34] R.B. SINGER Y T.B. McCORD, "Mars: Large scale mixing of bright and dark surface materials and implications for analysis of spectral reflectance", in *Proc. 10<sup>th</sup> Lunar Planetary Science Conf.*, 1979, pp.1835-1848.
- [35] B. HAPKE, "Bidirection reflectance spectroscopy I. Theory", *J. Geophys. Res.*, vol. 86, pp. 3039-3054, 1981.
- [36] D. LAVENIER, E. FABIANI, E. DERRIEN, C. WAGNER, "Systolic array for computing the pixel purity index (PPI) algorithm on hyper spectral images". Campus de Beaulieu, France. <http://www.irisa.fr/cosi/Lavenier/Publications/Lav01cf.pdf>
- [37] M.E. WINTER, "A proof of the N-FINDR algorithm for the automated detection of endmembers in a hyperspectral image," *Algorithms and Technologies for Multispectral, Hyperspectral and Ultraspectral Imagery X*, Proc. SPIE 5425, pp. 31- 31, 2004.
- [38] HEALEY, G., & SLATER, D.. "Models and methods for automated material identification in hyperspectral imagery acquired under unknown illumination and atmospheric conditions", *IEEE Transactions on Geoscience and Remote Sensing*, 37, 2706–2717, 1999.
- [39] Disponible en <http://library.usgs.gov> [online, última visita mayo de 2012].