



ESTIMACIÓN DEL NÚMERO DE MATERIALES EN IMÁGENES HIPERESPECTRALES

Andrés Varea Tarazón

Tutor: José Prades Nebot

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2019-20

Valencia, 6 de Julio de 2020

Resumen

En este trabajo fin de grado se ha creado una aplicación que permite estimar el número de materiales que hay en una imagen hiperspectral utilizando una serie de métodos muy conocidos. La aplicación, realizada con la herramienta App Designer de MATLAB, permite a un usuario elegir la imagen hiperspectral y los valores de los parámetros de los métodos de estima. La imagen hiperspectral de entrada puede ser sintética (en este caso el usuario elige los parámetros de síntesis) o puede seleccionarse de entre un conjunto de imágenes reales. Aparte del número de materiales obtenido con cada método, la aplicación también proporciona como resultado otra información de interés. La memoria de este trabajo incluye una explicación de los métodos utilizados, una breve descripción del software implementado y un manual de usuario de la aplicación. También incluye las estimaciones de los algoritmos frente a un conjunto de imágenes sintéticas, el número de materiales que estima cada método en cada imagen real y una discusión sobre los resultados obtenidos.

Resum

En aquest treball fi de grau s'ha creat una aplicació que permet estimar el nombre de materials que hi ha en una imatge hiperspectral utilitzant una sèrie de mètodes molt coneguts. L'aplicació, realitzada amb l'eina App Designer de MATLAB, permet a un usuari escollir l'imatge hiperspectral i els valors dels paràmetres dels mètodes d'estima. L'imatge hiperspectral d'entrada pot ser sintètica (en aquest cas l'usuari escolleix els paràmetres de síntesi) o pot seleccionar d'entre un conjunt d'imatges reials. A banda del nombre de materials obtinguts amb cada mètode, l'aplicació també proporciona com a resultat altra informació d'interès. La memòria d'aquest treball inclou una explicació dels mètodes utilitzats, una breu descripció del software implantat i un manual d'usuari de la aplicació. També inclou les estimacions dels algorismes respecte a un conjunt d'imatges sintètiques, el nombre de materials que estima cada mètode en cada imatge real i una discussió sobre els resultats obtinguts.

Abstract

This bachelor's thesis revolves around the design of an application that allows the user to estimate the number of endmembers that appear on a particular hyperspectral image with the use of some well known methods. The application, programmed on MATLAB's App Designer tool, allows an user to select a hyperspectral image from a library and the methods inputs or create a synthetic hyperspectral image (in this case the user picks the synthesis parameters). The application then shows the user the number of endmembers predicted by each method alongside other relevant information. This document includes an explanation on how the used methods work, a brief explanation of the used software and a user's manual. It also includes the methods estimations for a set of synthetic images, the number of endmembers predicted by each method for every single real image and a brief discussion on the obtained results.

Índice general

| | |
|---|-----------|
| Acrónimos | -1 |
| I Memoria | |
| 1. Introducción | 3 |
| 1.1. Objetivos | 3 |
| 1.2. Metodología empleada | 4 |
| 2. Fundamentos teóricos | 7 |
| 2.1. Imágenes hiperespectrales | 7 |
| 2.2. El modelo de mezcla lineal | 9 |
| 2.3. PCA | 11 |
| 2.4. HFC y NWHFC | 12 |
| 2.5. HySime | 13 |
| 3. Guía de usuario | 15 |
| 3.1. Vista general | 15 |
| 3.2. Pestañas de entradas y opciones | 16 |
| 3.2.1. Pestaña de entradas | 16 |
| 3.2.2. Imagen sintética | 17 |
| 3.2.3. Selector de imagen | 19 |
| 3.2.4. Selector de gráfica | 19 |
| 3.3. Tablas de datos | 20 |
| 3.4. Imagen y gráfica | 21 |
| 4. Resultados | 23 |
| 4.1. Resultados con imágenes sintéticas | 23 |
| 4.2. Resultados con imágenes reales | 25 |
| 4.2.1. Indian Pines | 26 |
| 4.2.2. Salinas | 28 |
| 4.2.3. Salinas-A | 30 |
| 4.2.4. Pavia University | 31 |
| 4.2.5. Urban | 33 |
| 4.2.6. Cuprite | 35 |
| 4.2.7. Jasper Ridge | 37 |
| 4.2.8. Samson | 39 |

| | |
|---|-----------|
| 4.2.9. Washington DC Mall | 40 |
| 4.2.10. Conclusiones | 41 |
| 5. Conclusiones y líneas futuras | 43 |
| 5.1. Conclusiones | 43 |
| 5.2. Líneas futuras | 44 |
| Bibliografía | 45 |

Índice de figuras

| | |
|---|----|
| 1.1. Diagrama temporal de actividades por semana. | 5 |
| 2.1. Representación de una imagen hiperespectral: (a) cubo; (b) matriz. | 7 |
| 2.2. Descomposición de una imagen bajo el modelo LMM. | 10 |
| 3.1. Aplicación al iniciar | 15 |
| 3.2. Aplicación en funcionamiento | 16 |
| 3.3. Pestaña Required inputs | 17 |
| 3.4. Pestaña Synth image | 18 |
| 3.5. Ejemplo de imagen sintética | 18 |
| 3.6. Pestaña image | 19 |
| 3.7. Pestaña Graph | 19 |
| 3.8. Tabla con datos de la imagen | 20 |
| 3.9. Tabla con los resultados de los cuatro algoritmos | 21 |
| 3.10. Imagen y gráfica en la sección derecha de la aplicación | 21 |
| 4.1. Promedio de las bandas de Indian Pines. | 26 |
| 4.2. Gráficas de valores propios de Indian Pines | 27 |
| 4.3. Salinas | 28 |
| 4.4. Varianza explicada en Salinas | 29 |
| 4.5. Gráfica de valores propios de la imagen Salinas | 29 |
| 4.6. Salinas-A | 30 |
| 4.7. Gráfica de valores propios de la imagen Salinas-A | 30 |
| 4.8. Pavia University | 31 |
| 4.9. Varianza explicada Pavia University | 32 |
| 4.10. Gráfica de valores propios de la imagen Pavia University | 32 |
| 4.11. Urban | 33 |
| 4.12. Varianza explicada Urban | 34 |
| 4.13. Gráfica de valores propios de la imagen Urban | 34 |
| 4.14. Cuprite | 35 |
| 4.15. Valores propios de Cuprite | 36 |
| 4.16. Jasper Ridge | 37 |
| 4.17. Varianza explicada Jasper Ridge | 38 |
| 4.18. Gráfica de valores propios de la imagen Jasper Ridge | 38 |
| 4.19. Samson | 39 |
| 4.20. Los recortes de Washington DC Mall utilizados | 40 |

Índice de tablas

| | |
|---|----|
| 4.1. Resultados con imágenes sintéticas y $\eta = 0$ | 24 |
| 4.2. Resultados con imágenes sintéticas y $\eta = 1/18$ | 25 |
| 4.3. Número de endmembers en la imagen Indian Pines 1. | 26 |
| 4.4. Número de endmembers en la imagen Indian Pines 2. | 27 |
| 4.5. Número de endmembers en la imagen Salinas. | 28 |
| 4.6. Número de endmembers en la imagen Salinas-A. | 30 |
| 4.7. Número de endmembers en la imagen Pavia University. | 31 |
| 4.8. Número de endmembers en la imagen Urban. | 33 |
| 4.9. Número de endmembers en la imagen Cuprite 1. | 35 |
| 4.10. Número de endmembers en la imagen Cuprite 2. | 36 |
| 4.11. Número de endmembers en la imagen Jasper Ridge. | 37 |
| 4.12. Número de endmembers en la imagen Samson. | 39 |
| 4.13. Número de endmembers en la imagen WDM1. | 40 |
| 4.14. Número de endmembers en la imagen WDM2. | 40 |

Acrónimos

| | |
|--------|--|
| LMM | Linear Mixing Model |
| HS | HyperSpectral |
| HFC | Harsanyi-Farrand-Chang |
| NWHFC | Noise Withening Harsanyi-Farrand-Chang |
| PCA | Principal Component Analysis |
| HySime | Hyperspectral Signal Identification by Minimum Error |

Parte I

Memoria

Capítulo 1

Introducción

Las imágenes hiperespectrales son imágenes tomadas por sensores especiales. En estas imágenes, la longitud de los píxeles es mucho mayor a la de una imagen corriente, llegando a tener cientos de valores. Estos valores representan la radiación que capta el sensor en bandas del espectro electromagnético muy estrechas. Por tanto, cada uno de los valores de un píxel es la radiación que capta el sensor en esa banda mencionada en el espacio geográfico que representa dicho píxel.

La escena que representa la imagen hiperespectral tiene un número de materiales, a priori desconocidos, que es interesante conocer. Esto es debido a que, en muchas aplicaciones, es necesario conocer el número de materiales en la escena en una imagen hiperespectral, como en la clasificación de imágenes por ejemplo.

Por esta razón en los últimos años se han propuesto una serie de algoritmos que tienen por objetivo determinar el número de materiales de una escena a partir de su imagen hiperespectral.

1.1. Objetivos

La finalidad de este trabajo fin de grado es la comparación y el análisis de los resultados obtenidos por los algoritmos más importantes de estimación del número de materiales en imágenes hiperespectrales.

Para ello, se utiliza un amplio abanico de imágenes hiperespectrales reales en las que, a priori, se conoce el número de materiales que las forman; además de un generador de imágenes hiperespectrales sintéticas, en el que el usuario introduce los parámetros de la imagen, y servirá para poder estudiar el comportamiento de los diferentes algoritmos frente a distintas condiciones como el ruido, el número de endmembers o el color del ruido.

Para conseguir la finalidad perseguida se plantearon tres objetivos principales. El primer objetivo consiste en el estudio de los distintos algoritmos de estimación de materiales de imágenes hiperespectrales disponibles. En el caso de este trabajo de fin de grado, este estudio se centra en los algoritmos PCA (Principal Component Analysis), HFC (Harsanyi-Farrand-Chang), HFC con un algoritmo de blanqueo de ruido previo NWHFC (Noise Whitening Harsanyi-Farrand-Chang) e HySime (Hyperspectral Signal Identification by Minimum Error). Estos algoritmos serán posteriormente explicados a lo largo de la siguiente memoria.

El segundo objetivo consiste en la recopilación de la implementación de los mencionados algoritmos en MATLAB, así como un conjunto de imágenes hiperespectrales conocidas y un algoritmo para generar una imagen hiperespectral sintética con unos valores introducidos por el usuario. Esto permitirá comparar los resultados de los algoritmos previamente mencionados con respecto a cada una de las imágenes reales o artificiales al conocer de antemano su número de materiales.

El objetivo final es la realización de una aplicación de MATLAB que permita seleccionar una imagen hiperespectral, o en su defecto, crearla usando el algoritmo para generar una imagen sintética (el cual permite que el usuario introduzca los parámetros de síntesis) y analice la imagen usando los algoritmos de estimación de materiales para observar sus resultados y compararlos con los datos reales de la imagen conocidos de anterioridad.

Estos objetivos generales se han plasmado en los siguientes objetivos específicos:

- Estudio de los principales métodos de estimación de materiales en imágenes hiperespectrales.
- Recopilación de los principales métodos de estimación de materiales en imágenes hiperespectrales implementados en MATLAB.
- Recopilación de un conjunto de imágenes hiperespectrales de parámetros conocidos en formato MATLAB.
- Selección de un algoritmo generador de imágenes hiperespectrales sintéticas implementado en MATLAB.
- Implementación de una aplicación en MATLAB que permita seleccionar una imagen hiperespectral y analizarla con los diferentes algoritmos.
- Análisis de los resultados y conclusiones.

1.2. Metodología empleada

A grandes rasgos, la realización de este trabajo de fin de grado se ha dividido en tres grandes tareas: estudio de los fundamentos de imágenes hiperespectrales y los algoritmos utilizados para determinar el número de materiales en ellas; recopilación de algoritmos en MATLAB para su implementación en una aplicación; y la redacción de la memoria.

Más concretamente, se han realizado las siguientes actividades, cuya distribución temporal se puede ver en Diagrama temporal de actividades por semana. (Figura 1.1) (se enumeran en orden cronológico y se indica el tiempo invertido en la realización de cada una de ellas):

1. Estudiar los fundamentos de las imágenes hiperespectrales. (8 horas)
2. Estudiar los diferentes algoritmos de estimación de materiales en imágenes hiperespectrales. (15 horas)
3. Elegir un conjunto de imágenes hiperespectrales a estudiar. (2 horas)
4. Elegir los algoritmos de estimación de número de materiales a estudiar. (1 hora)

5. Recopilar la implementación de los algoritmos en el lenguaje de programación M. (3 hora)
6. Elegir una aplicación de generación de imágenes hiperespectrales sintéticas en lenguaje de programación M. (1 hora)
7. Implementar la aplicación en el App Designer de MATLAB. (60 horas)
8. Estudiar los resultados obtenidos y sacar conclusiones (10 horas)
9. Redactar la presente memoria (40 horas)

Como puede observarse, gran parte del tiempo se ha empleado en el estudio de los algoritmos, en la creación de la aplicación y en la redacción de la memoria. Además, algunas de las actividades han sido transversales a la realización del trabajo, como el aprendizaje de la aplicación App Designer de MATLAB, la profundización en el lenguaje de programación M, la profundización y repaso de álgebra o el aprendizaje del lenguaje de programación Latex.

| CRONOGRAMA | Mayo | | | | Junio | | | | | Julio |
|---|--------|---------|---------|---------|-------|-------|---------|---------|---------|-------|
| | 4 a 10 | 11 a 18 | 19 a 26 | 27 a 31 | 1 a 2 | 3 a 9 | 10 a 17 | 18 a 25 | 26 a 30 | 1 a 6 |
| Estudio de fundamentos | | | | | | | | | | |
| Estudio de algoritmos estimación materiales | | | | | | | | | | |
| Elección de imágenes hiperespectrales | | | | | | | | | | |
| Elección de algoritmos | | | | | | | | | | |
| Recopilación de algoritmos | | | | | | | | | | |
| Programación de aplicación | | | | | | | | | | |
| Recopilación generador imágenes sintéticas | | | | | | | | | | |
| Extracción de resultados y conclusiones | | | | | | | | | | |
| Redacción de memoria | | | | | | | | | | |

Figura 1.1: Diagrama temporal de actividades por semana.

Capítulo 2

Fundamentos teóricos

En este capítulo se exponen los conceptos básicos sobre imágenes hiperespectrales y se describen brevemente algunos de los algoritmos más utilizados para la determinación del número de materiales en este tipo de imágenes.

2.1. Imágenes hiperespectrales

Un sensor hiperespectral mide la energía recibida por cada una de sus celdas sensoras en un gran número de bandas espectrales. Esta característica hace que el procesamiento de imágenes hiperespectrales se utilice en un gran número de aplicaciones como la minería, la geología, la ecología o la vigilancia [1].

Podemos representar la información de una imagen hiperespectral de L bandas como un array tridimensional, o *cubo*, en el que las dos primeras coordenadas, n_1 y n_2 , representan los índices espaciales y la tercera coordenada, n_3 , representa el índice frecuencial o banda. Esta es la representación que se muestra en la figura 2.1(a) (en la práctica, los valores de N_1 , N_2 y L suelen ser mucho mayores a los mostrados en esta figura).

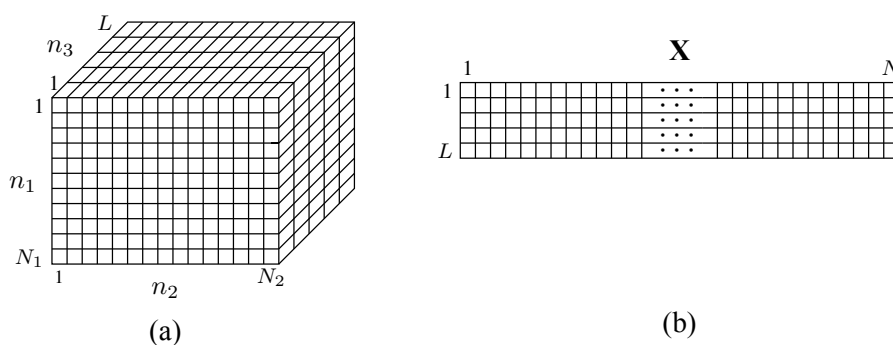


Figura 2.1: Representación de una imagen hiperespectral: (a) cubo; (b) matriz.

Alternativamente podemos representar la información de la imagen hiperespectral mediante una matriz de L filas y $N = N_1 \times N_2$ columnas:

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$$

en la que cada columna, \mathbf{x}_j , representa un *pixel* de la imagen, esto es, un vector en el que la componente i -ésima es la energía de la banda i -ésima en ese píxel. Este tipo de representación suele usarse cuando el algoritmo utilizado para procesar la imagen no tiene en cuenta la posición que cada píxel ocupa en la imagen. Dado que ese suele ser el caso en los algoritmos considerados en este trabajo, a partir de ahora se hará referencia a la matriz \mathbf{X} cuando se haga alusión a la imagen hiperespectral a procesar.

La escena que captura un sensor hiperespectral está formada por un conjunto de sustancias puras o materiales. El número de materiales que hay en la escena, K , suele ser mucho más pequeño que el número de bandas del sensor ($K \ll L$).

Cada material de la escena captada por una imagen tiene un espectro representativo de L bandas llamado *endmember*. En muchas aplicaciones es necesario saber cuántos materiales hay en la escena representada por una imagen hiperespectral. Ejemplos de este tipo de aplicaciones son la clasificación de imágenes o la determinación de sus endmembers. Esta es la razón por la que en los últimos años han aparecido una serie de algoritmos que tienen como objetivo determinar el número de materiales que hay en una escena a partir de una imagen \mathbf{X} de dicha escena [2, 3, 4, 5]. Algunos de estos algoritmos estiman el valor de K únicamente a partir de \mathbf{X} [3, 4, 5], mientras que en otros es necesario alguna información adicional más [2].

Notación. En el resto de este trabajo utilizaremos la siguiente notación. Los vectores se presentarán con letras minúsculas en negrita (por ejemplo \mathbf{x}) y las matrices con letras mayúsculas en negrita (por ejemplo \mathbf{A}). Denotaremos el operador esperanza matemática con $\mathbb{E}\{\cdot\}$. La matriz identidad de $N \times N$ se denota con \mathbf{I}_L . Denotamos con $\langle \mathbf{A} \rangle$ el espacio vectorial expandido por las columnas de la matriz \mathbf{A} .

2.2. El modelo de mezcla lineal

A la hora de determinar el número de endmembers es necesario asumir un cierto modelo para la generación de una imagen hiperespectral. El modelo de mezcla lineal (LMM) es el modelo más utilizado a tal efecto. Este modelo considera que cada píxel de una imagen hiperespectral se puede expresar como

$$\mathbf{x} = \mathbf{A} \mathbf{s} + \mathbf{e} = \sum_{i=1}^K s_i \mathbf{a}_i + \mathbf{n} \quad (2.1)$$

donde:

- K es el número de endmembers de la imagen (o equivalentemente, el número de materiales que hay en la escena).
- $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_K]$ es una matriz $L \times K$ determinista en la que \mathbf{a}_j (vector $L \times 1$) es el j -ésimo endmember ($1 \leq j \leq K$).
- $\mathbf{s} = [s_1, \dots, s_K]^T$ es un vector aleatorio cuyas componentes, llamadas *fracciones* o *abundancias*, cumplen

$$0 \leq s_i \leq 1, \quad i = 1, \dots, K \quad (2.2)$$

$$\sum_{i=1}^K s_i = 1. \quad (2.3)$$

Las restricciones (2.2) y (2.3) garantizan que las abundancias son físicamente significativas, esto es, que s_i representa la fracción del i -ésimo endmember en el píxel.

- \mathbf{n} es un vector aleatorio o *ruido*. Este vector modela cualquier desajuste entre el vector observado en un píxel y la mezcla pura de los endmembers que lo constituyen.

Podemos escribir (2.1) de forma más compacta como

$$\mathbf{x} = \mathbf{y} + \mathbf{n} \quad (2.4)$$

donde \mathbf{y} ($\mathbf{y} = \mathbf{A} \mathbf{s}$) representa la componente de señal de \mathbf{x} , y \mathbf{n} representa la componente de ruido. En general se asume que \mathbf{n} es ortogonal a \mathbf{s} y, por tanto, también lo son \mathbf{n} y \mathbf{x} :

$$\mathbb{E}\{\mathbf{x} \mathbf{n}^T\} = \mathbb{E}\{\mathbf{s} \mathbf{n}^T\} = \mathbf{0}. \quad (2.5)$$

Definimos la matriz de autocorrelación de un vector aleatorio \mathbf{z} como $\mathbf{R}_z = \mathbb{E}\{\mathbf{z} \mathbf{z}^T\}$. Teniendo en cuenta (2.5), la matriz de autocorrelación de \mathbf{x} se puede expresar como

$$\mathbf{R}_x = \mathbf{R}_y + \mathbf{R}_n \quad (2.6)$$

siendo $\mathbf{R}_y = \mathbf{A} \mathbf{R}_s \mathbf{A}^T$. La matriz \mathbf{R}_y tiene K valores propios positivos y $L - K$ valores propios nulos.

Consideremos una imagen en concreto \mathbf{X} . Sean $\mathbf{S} = [\mathbf{s}_1, \dots, \mathbf{s}_N]$ y $\mathbf{N} = [\mathbf{n}_1, \dots, \mathbf{n}_N]$ las matrices cuyas columnas representan las realizaciones de \mathbf{s} y \mathbf{n} del modelo LMM para \mathbf{X} , respectivamente. De acuerdo con el modelo LMM, podemos escribir (figura 2.2)

$$\mathbf{X} = \mathbf{A} \mathbf{S} + \mathbf{N}. \quad (2.7)$$

Por tanto, \mathbf{s}_j contiene las abundancias del píxel \mathbf{x}_j , y \mathbf{n}_j es el ruido presente en dicho píxel.

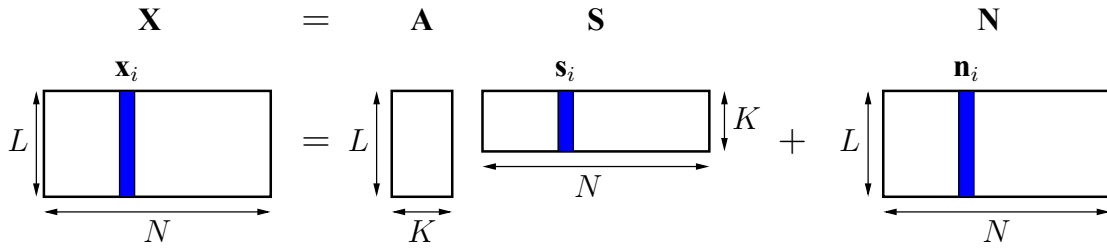


Figura 2.2: Descomposición de una imagen bajo el modelo LMM.

De manera similar a (2.4), podemos descomponer la imagen \mathbf{X} como

$$\mathbf{X} = \mathbf{Y} + \mathbf{N} \quad (2.8)$$

donde $\mathbf{Y} = \mathbf{A}\mathbf{S}$ es la componente de señal de \mathbf{X} , y \mathbf{N} es la componente de ruido de \mathbf{X} .

Supongamos que las columnas de \mathbf{X} son realizaciones de un vector aleatorio \mathbf{x} que sigue el modelo LMM. Definimos la *matriz de autocorrelación muestral* de \mathbf{X} como

$$\hat{\mathbf{R}}_{\mathbf{x}}(N) = \frac{1}{N} \mathbf{X}\mathbf{X}^T.$$

La matriz $\hat{\mathbf{R}}_{\mathbf{x}}(N)$ es una estima de $\mathbf{R}_{\mathbf{x}}$ y se tiene que

$$\lim_{N \rightarrow \infty} \hat{\mathbf{R}}_{\mathbf{x}}(N) = \mathbf{R}_{\mathbf{x}}. \quad (2.9)$$

De manera similar a como hemos definido las matrices $\mathbf{R}_{\mathbf{x}}$ y $\hat{\mathbf{R}}_{\mathbf{x}}(N)$, podemos definir las correspondientes matrices de covarianza. Así, definimos la *matriz de covarianza* de \mathbf{x} como

$$\mathbf{K}_{\mathbf{x}} = \mathbb{E}\{(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}})(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}})^T\} \quad (2.10)$$

donde $\boldsymbol{\mu}_{\mathbf{x}} = \mathbb{E}\{\mathbf{x}\}$. Definimos la *matriz de covarianza muestral* de \mathbf{X} como

$$\hat{\mathbf{K}}_{\mathbf{x}}(N) = \frac{1}{N} \sum_{j=1}^N (\mathbf{x}_j - \mathbf{m})(\mathbf{x}_j - \mathbf{m})^T \quad (2.11)$$

donde \mathbf{x}_j es la j -ésima columna de \mathbf{X} y $\mathbf{m} = \frac{1}{N} \sum_{j=1}^N \mathbf{x}_j$. La matriz $\hat{\mathbf{K}}_{\mathbf{x}}(N)$ es una estima de $\mathbf{K}_{\mathbf{x}}$ y se cumple

$$\lim_{N \rightarrow \infty} \hat{\mathbf{K}}_{\mathbf{x}}(N) = \mathbf{K}_{\mathbf{x}}.$$

2.3. PCA

En esta sección veremos que el análisis de los valores propios de \mathbf{R}_x puede permitir determinar K bajo ciertas condiciones. Sean l_1, \dots, l_L los valores propios de \mathbf{R}_y ordenados de mayor a menor ($l_1 \geq l_2 \geq \dots \geq l_L$). Supongamos que el ruido \mathbf{n} es *blanco*, esto es, \mathbf{n} es de media nula ($E\{\mathbf{n}\} = \mathbf{0}$) y su matriz de autocorrelación es de la forma

$$\mathbf{R}_n = \sigma^2 \mathbf{I}_L$$

para algún $\sigma^2 > 0$. En este caso, los L valores propios de \mathbf{R}_n son iguales a σ^2 y los valores propios de \mathbf{R}_x son

$$\lambda_k = \begin{cases} l_k + \sigma^2, & k = 1, \dots, K \\ \sigma^2, & k = K + 1, \dots, L \end{cases} \quad (2.12)$$

Por tanto, los K valores propios de mayor valor corresponden a señal (y ruido), mientras que los $L - K$ valores propios de menor valor corresponden únicamente a ruido. Según (2.12), podemos determinar K calculando los valores propios de \mathbf{R}_x , ordenándolos de mayor a menor, e identificando en la secuencia resultante un *codo* en $k = K$ y un *suelo* en $k = K + 1, \dots, L$ (esto es, la secuencia es decreciente hasta $k = K$ y es constante para $k > K$).

Sin embargo, en imágenes hiperespectrales el ruido no es blanco, lo que hace que no exista un codo fácilmente identificable en la secuencia λ_k . Una solución a este problema consiste elegir K como el menor entero que cumple

$$\frac{\sum_{k=1}^K \lambda_k}{\sum_{k=1}^L \lambda_k} \times 100 \geq t \quad (2.13)$$

donde t es un porcentaje umbral. Sin embargo, el valor de t que permite identificar K es desconocido.

La selección de los valores propios anteriormente descrita es idéntica a la realizada en *análisis de componentes principales* (PCA), razón por la cual en este trabajo nos referiremos al método que decide el valor K de acuerdo con (2.13) como PCA. PCA es la técnica más utilizada en estadística para reducir la dimensionalidad de los datos. Un valor de t usado frecuentemente en PCA es 95%. Sin embargo, no hay ninguna garantía de que dicho valor sea válido cuando PCA se utiliza para determinar K en imágenes hiperespectrales. En la práctica, PCA suele considerar la secuencia de valores propios de la matriz de covarianzas \mathbf{K}_x . Si usamos dicha secuencia para determinar el valor de K , debemos sumar uno al valor obtenido con la regla (2.13).

Una dificultad añadida de PCA es que no conocemos \mathbf{R}_x o \mathbf{K}_x por lo que en su lugar se utilizan sus versiones muestrales ($\hat{\mathbf{R}}_x$ o $\hat{\mathbf{K}}_x$).

Al igual que en PCA, muchos de los métodos para determinar K en imágenes hiperespectrales están basados en un análisis de los valores propios de $\hat{\mathbf{R}}_x$ o $\hat{\mathbf{K}}_x$ [2, 3, 4, 5]. Utilizando distintas aproximaciones, todos estos métodos intentan determinar el umbral que permita seleccionar aquellos valores propios debidos exclusivamente al ruido, y así obtener K .

2.4. HFC y NWHFC

Hay varios algoritmos que determinan el valor de K utilizando una secuencia de test de hipótesis. Un ejemplo es el algoritmo Harsanyi-Farrand-Chang (HFC) [2].

HFC asume que \mathbf{y} es determinista y que \mathbf{n} es blanco y Gaussiano. Bajo estas hipótesis, si $\lambda_1, \dots, \lambda_L$ es la secuencia ordenada de valores propios de \mathbf{x} y $\hat{\lambda}_1, \dots, \hat{\lambda}_L$ es la secuencia ordenada de valores propios de \mathbf{K}_x , cabe esperar que

$$\begin{cases} \lambda_k > \hat{\lambda}_k > \sigma^2, & \text{para } k = 1, \dots, K \\ \lambda_k = \hat{\lambda}_k = \sigma^2, & \text{para } k = K + 1, \dots, L \end{cases} \quad (2.14)$$

Equivalentemente, si definimos $\delta_k = \lambda_k - \hat{\lambda}_k$, cabe esperar que

$$\begin{cases} \delta_k > 0, & \text{para } k = 1, \dots, K \\ \delta_k = 0, & \text{para } k = K + 1, \dots, L \end{cases} \quad (2.15)$$

Por tanto, a partir de \mathbf{R}_x y \mathbf{K}_x es posible determinar K buscando un codo en la secuencia δ_k . En la práctica, esta búsqueda debe realizarse a partir de los valores propios de las matrices $\hat{\mathbf{R}}_x$ y $\hat{\mathbf{K}}_x$. En este caso, la secuencia δ_k será generalmente distinta de 0 en $k > K$ (aunque generalmente serán valores pequeños). Basándose en esto, HFC estima el valor de K realizando L tests de hipótesis binarios y en cada uno de estos tests se decide si un valor propio está generado por señal y ruido o está generado exclusivamente por ruido.

HFC asume que el ruido es blanco, pero está hipótesis dista de ser verdadera en imágenes hiperespectrales. Para solucionar este problema, hay una versión de HFC, llamada Noise-Whitened HFC (NWHFC), que incluye un paso en el que se blanquea el ruido de la imagen [2]. Para ello el algoritmo debe primero estimar la matriz de autocorrelación del ruido. En [2], el algoritmo utilizado para estimar la matriz de autocorrelación es el descrito en [6], aunque pueden usarse otros algoritmos (cuanto mayor sea su precisión, mejor será la estima de K).

En HFC y NWHFC, el usuario debe proporcionar el valor de la probabilidad de falsa alarma, P_{FA} , que se usará en los tests de hipótesis. Dependiendo del valor P_{FA} proporcionado, el valor de K puede cambiar. En [2], $P_{FA} = 10^{-3}$ proporciona buenos resultados en las dos imágenes analizadas.

2.5. HySime

Al igual que NWHFC, el algoritmo *Hyperspectral Signal identification by minimum error* (HySime) estima el ruido que hay en la imagen, pero la determinación de K se realiza de forma completamente distinta [3].

HySime primero estima la componente de ruido \mathbf{N} de \mathbf{X} utilizando regresión múltiple. Esta técnica es eficaz puesto que en imágenes hiperespectrales hay un alto grado de correlación entre bandas espectrales adyacentes. En este algoritmo cada fila (o banda) de \mathbf{X} se aproxima con la combinación lineal del resto de filas (o bandas) que proporciona una mínima suma de errores (de aproximación) cuadráticos. Con la aproximación obtenida en cada fila se forma una nueva matriz $\hat{\mathbf{Y}}$ que es la estima de \mathbf{Y} . A partir de ellas, HySime estima el ruido como

$$\hat{\mathbf{N}} = \mathbf{X} - \hat{\mathbf{Y}}. \quad (2.16)$$

y su matriz de autocorrelación \mathbf{R}_n como

$$\hat{\mathbf{R}}_n = \frac{1}{N} \hat{\mathbf{N}}\hat{\mathbf{N}}^T \quad (2.17)$$

Los valores propios de la matriz de autocorrelación de $\hat{\mathbf{Y}}$ generalmente no permite identificar claramente K porque el algoritmo de estimación de ruido no es perfecto y se ha obtenido a partir de un número de muestras finito.

Para obtener K , HySime identifica el subespacio vectorial de \mathbb{R}^L, Y , en el que está $\hat{\mathbf{Y}}$ e identifica K como la dimensión de dicho subespacio. Concretamente, HySime considera que Y es el subespacio expandido por el subconjunto de k vectores propios de $\hat{\mathbf{R}}_y$ que minimiza el error cuadrático medio (MSE) entre \mathbf{y} y la proyección de \mathbf{x} sobre dicho subespacio. Veamos a continuación cómo obtener Y y su dimensión K

La matriz $\hat{\mathbf{R}}_y$ puede descomponerse como

$$\hat{\mathbf{R}}_y = \mathbf{E} \mathbf{E}^T$$

donde $\mathbf{E} = [\mathbf{e}_1, \dots, \mathbf{e}_L]$ es la matriz de vectores propios de $\hat{\mathbf{R}}_y$. Dada una permutación $\pi = \{i_1, \dots, i_L\}$ de los enteros $\{1, \dots, L\}$, tenemos la descomposición ortogonal

$$\mathbb{R}^L = \langle \mathbf{E}_k \rangle \oplus \langle \mathbf{E}_k^\perp \rangle$$

donde $\langle \mathbf{E}_k \rangle$ es el subespacio expandido por $\mathbf{E}_k = [\mathbf{e}_{i_1}, \dots, \mathbf{e}_{i_k}]$ y $\langle \mathbf{E}_k^\perp \rangle$ el subespacio expandido por $\mathbf{E}_k^\perp = [\mathbf{e}_{i_{k+1}}, \dots, \mathbf{e}_{i_L}]$. Sea

$$\mathbf{U}_k = \mathbf{E}_k \mathbf{E}_k^T$$

la matriz proyección sobre $\langle \mathbf{E}_k \rangle$ y sea

$$\tilde{\mathbf{y}}_k = \mathbf{U}_k \mathbf{y}$$

la proyección de \mathbf{x} sobre $\langle \mathbf{E}_k \rangle$. Entonces el error cuadrático medio entre \mathbf{y} e $\tilde{\mathbf{y}}_k$

$$\text{mse}(k) := \mathbb{E} \{ (\mathbf{y} - \tilde{\mathbf{y}}_k)^T (\mathbf{y} - \tilde{\mathbf{y}}_k) \} \quad (2.18)$$

$$= \text{tr}(\mathbf{U}_k^\perp \mathbf{R}_y) + 2 \text{tr}(\mathbf{U}_k \hat{\mathbf{R}}_n) + c \quad (2.19)$$

donde c es una constante.

El criterio para estimar el subespacio de señal, Y , es minimizar $\text{mse}(k)$ con respecto a todas las permutaciones π hasta k . Dado que \mathbf{R}_x es desconocida, la sustituiremos por su versión muestral $\hat{\mathbf{R}}_x = (\hat{\mathbf{X}}\hat{\mathbf{X}}^T)/N$. Por tanto, HySime resuelve el problema

$$\hat{Y} = \langle [\mathbf{e}_{\hat{i}_1}, \dots, \mathbf{e}_{\hat{i}_k}] \rangle \quad (2.20)$$

$$(\hat{k}, \hat{\pi}) = \arg \min_{k, \pi} \left\{ \text{tr}(\mathbf{U}_k^\perp \hat{\mathbf{R}}_x) + 2 \text{tr}(\mathbf{U}_k \hat{\mathbf{R}}_n) \right\}. \quad (2.21)$$

Dada una permutación π , el primer término de (2.21) refleja la potencia del error de proyección y decrece con k , y el segundo término refleja la potencia de ruido y crece con k . Se puede reescribir (2.21) como

$$(\hat{k}, \hat{\pi}) = \arg \min_{k, \pi} \left\{ c + \sum_{j=1}^k \underbrace{-p_{i_j} + 2\sigma_{i_j}^2}_{\delta_{i_j}} \right\} \quad (2.22)$$

donde c es una constante, y p_{i_j} y $\sigma_{i_j}^2$ son

$$p_{i_j} = \mathbf{e}_{i_j}^T \hat{\mathbf{R}}_x \mathbf{e}_{i_j} \quad (2.23)$$

$$\sigma_{i_j}^2 = \mathbf{e}_{i_j}^T \hat{\mathbf{R}}_n \mathbf{e}_{i_j}. \quad (2.24)$$

Como (2.23) y (2.24) son formas cuadráticas, p_{i_j} y $\sigma_{i_j}^2$ son cantidades no negativas. Teniendo en cuenta esto y que queremos *minimizar* la suma de (2.22) debemos incluir en dicha suma los términos negativos de δ_i ($i = 1, \dots, L$). Resumiendo todo lo anterior obtenemos el algoritmo 1.

Algoritmo 1: HySime.

A partir de la imagen \mathbf{X} :

Obtén $\hat{\mathbf{R}}_x = (\hat{\mathbf{X}}\hat{\mathbf{X}}^T)/N$

Obtén $\hat{\mathbf{N}}$ (utilizando regresión múltiple) y $\hat{\mathbf{R}}_n$

Obtén $\hat{\mathbf{Y}}$ y $\hat{\mathbf{R}}_y$

Obtén los vectores propios de $\hat{\mathbf{R}}_y$: $\mathbf{e}_1, \dots, \mathbf{e}_L$

Obtén $\delta_1, \dots, \delta_L$. Ordénalos y guarda $\hat{\pi}$

Fija al \hat{K} al número de δ_i negativos

Capítulo 3

Guía de usuario

En este capítulo se ofrece una guía detallada sobre la aplicación, como usarla y como interpretar lo que aparece en pantalla, proporcionando imágenes explicativas y una descripción de cada una de sus funciones y opciones.

3.1. Vista general

En esta sección se describe el aspecto general de la aplicación, tanto al iniciarla como al trabajar con ella.

La figura Aplicación al iniciar (Figura 3.1) es lo que se muestra al ejecutar la aplicación. En este momento lo único que se puede ver es una serie de pestañas en la parte superior izquierda, que serán las entradas que permiten al usuario introducir valores de entrada o seleccionar diversas opciones para el funcionamiento de la aplicación. Debajo de las pestañas podemos encontrar dos tablas, por ahora vacías, que, tras la ejecución, mostrarán los datos y resultados que necesitamos de las imágenes. Además, aparece una gráfica vacía en la que visualizará una de las cuatro opciones de las que dispone la aplicación.

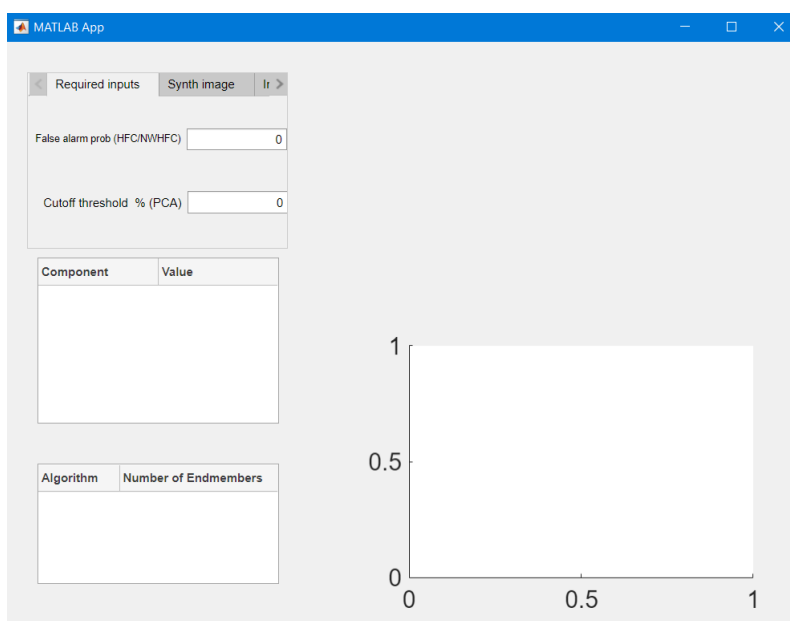


Figura 3.1: Aplicación al iniciar

La figura Aplicación en funcionamiento (Figura 3.2) muestra una imagen de la aplicación en funcionamiento. Ahora las tablas muestran información. En la tabla de arriba se muestran los datos reales de la imagen y en la tabla de abajo se muestran los resultados de cada uno de los algoritmos que analizan dicha imagen hiperespectral. La parte derecha de la aplicación muestra una representación de la imagen y un gráfico con información que suplementa la obtenida por la tabla de resultados.

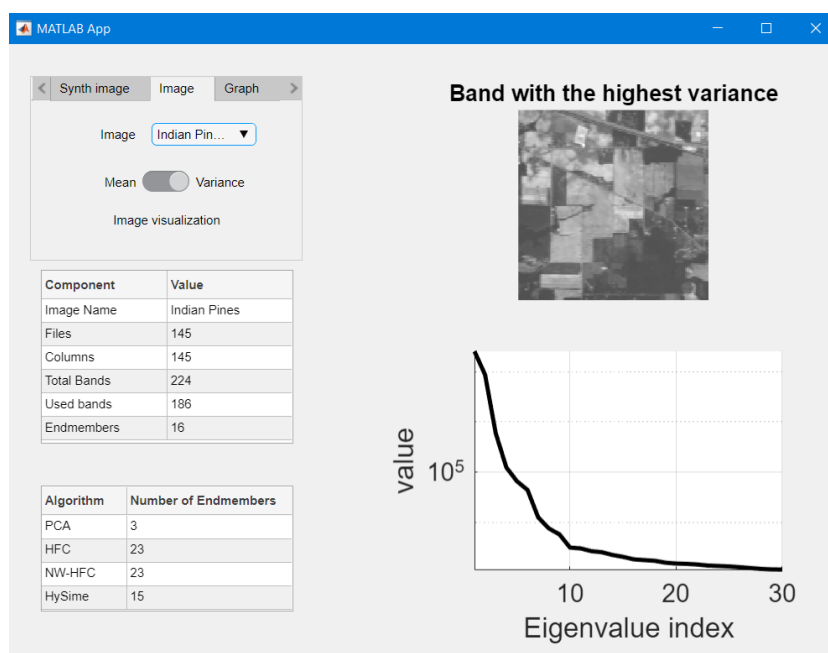


Figura 3.2: Aplicación en funcionamiento

3.2. Pestañas de entradas y opciones

En primer lugar, lo que se observa en la aplicación es la sección de pestañas que permite elegir el funcionamiento de la aplicación y condiciona lo que vemos en pantalla. En esta sección se describe cada una de las distintas pestañas, sus entradas y opciones.

3.2.1. Pestaña de entradas

La primera de las pestañas, Pestaña Required inputs (Figura 3.3), permite seleccionar las variables de entrada de los algoritmos que utiliza la aplicación. Los argumentos de entrada de los algoritmos de estimación del número de materiales son los siguientes:

- False alarm prob. (HFC/NWHFC). Valor de la probabilidad de falsa alarma que se utiliza en los test de hipótesis realizados en los algoritmos HFC y NWHFC.
- Cutoff thershold % (PCA). Porcentaje de la varianza de la imagen que debe retenerse tras eliminar algunos de sus valores propios en PCA.

| < Required inputs | Synth image | Ir > |
|------------------------------|-------------------------------------|------|
| False alarm prob (HFC/NWHFC) | <input type="text" value="0.0001"/> | |
| Cutoff threshold % (PCA) | <input type="text" value="99"/> | |

Figura 3.3: Pestaña Required inputs

3.2.2. Imagen sintética

En esta sección se describe la pestaña de creación de imágenes sintéticas, Pestaña Synth image (Figura 3.4). La ventaja de las imágenes sintéticas es que el usuario tiene un control absoluto sobre las características de las imágenes generadas (por ejemplo, el número de endmembers de la imagen).

La aplicación crea una imagen hiperespectral sintética siguiendo el procedimiento descrito en [3]. Para crear una imagen de este tipo, primero se elige al azar un conjunto de endmembers de una librería; después se generan las abundancias y el vector ruido de cada píxel; finalmente, a partir de los endmembers, las abundancias y los vectores ruido, se generan los píxeles de acuerdo con el modelo LMM.

Las abundancias de cada píxel se obtienen muestreando una distribución Dirichlet. Esta distribución es adecuada porque las abundancias generadas cumplen las restricciones (2.2) y (2.3). Los parámetros de la distribución, $\alpha_1, \alpha_2, \dots, \alpha_K$, cumplen

$$\alpha_1 = \alpha_2 = \dots = \alpha_K = \frac{1}{K}$$

por lo que la función de densidad de probabilidad es simétrica. El vector ruido de cada píxel se obtiene a partir de una distribución Gaussiana multivariada L -dimensional de media nula y matriz de covarianza diagonal. La varianza de la i -ésima componente del ruido es

$$\sigma_i^2 = \sigma^2 \frac{e^{-\frac{(i-L/2)^2}{2/\eta^2}}}{\sum_{j=1}^L e^{-\frac{(j-L/2)^2}{2/\eta^2}}} \quad (3.1)$$

donde σ^2 es la potencia media de ruido. La forma de la secuencia de varianzas $\sigma_1^2, \dots, \sigma_L^2$ depende del valor del parámetro η . Así, cuando $\eta = 0$ tenemos ruido blanco puesto que $\sigma_1^2 = \sigma_2^2 = \dots = \sigma_L^2$. Cuando $\eta > 0$, la secuencia de varianzas tiene forma de campana de Gauss (cuanto menor es η , más estrecha es la campana).

La relación señal a ruido es

$$\text{SNR} = 10 \log \frac{\frac{1}{NL} \sum_{j=1}^N \mathbf{x}_j^T \mathbf{x}_j}{\frac{1}{NL} \sum_{j=1}^N \mathbf{n}_j^T \mathbf{n}_j} \quad (3.2)$$

donde \mathbf{x}_j y \mathbf{n}_j representan el j ésimo píxel y el j ésimo vector ruido, respectivamente.

Los argumentos de entrada del algoritmo de generación de imágenes sintéticas son los siguientes (Figura 3.4):

- Number of endmembers. Número de endmembers de la imagen (K en el modelo LMM). Los endmembers se seleccionan al azar de la librería *U. S. Geological Survey* (archivo USG_1995_Library.mat).
- Number of pixels. Número de píxeles de la imagen generada (N en el modelo LMM).
- Noise colour (eta). Valor del parámetro η en (3.1).
- Signal to Noise Ratio (dB). Valor de la relación señal a ruido en definida como (3.2).

| Required inputs | | Synth image | lr |
|--------------------------------------|------------------------------------|-------------|----|
| Synthetic image parameters if needed | | | |
| Number of endmembers | <input type="text" value="15"/> | | |
| Number of pixels | <input type="text" value="1e+04"/> | | |
| Noise colour (eta) | <input type="text" value="0.055"/> | | |
| Signal to Noise Ratio (dB) | <input type="text" value="15"/> | | |

Figura 3.4: Pestaña Synth image

Las imágenes sintéticas generadas con esta opción contienen píxeles que no están correlados espacialmente. Esto se puede verificar en la figura 3.5 que muestra el promedio de las bandas de una imagen sintética. Por tanto, las imágenes sintéticas generadas por la aplicación no son adecuadas para comprobar la eficiencia de aquellos algoritmos que, en parte, basan la estima de K en la correlación que hay entre píxeles adyacentes.

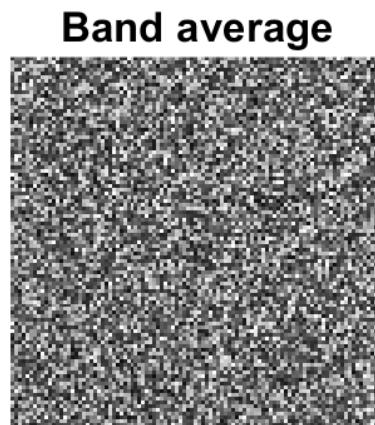


Figura 3.5: Ejemplo de imagen sintética

3.2.3. Selector de imagen

En esta sección se presenta el selector de imagen de la aplicación que permite seleccionar la imagen a analizar y como visualizarla. La Pestaña image (figura 3.6) permite elegir una imagen real de la librería o, en su defecto, generar una imagen sintética con los parámetros escogidos. Además, permite seleccionar entre una de las dos formas de representar la imagen escogida; bien mostrando una imagen con la media de cada píxel (si se elige la opción Mean) o, en su lugar, mostrando la banda de mayor varianza (si se elige la opción Variance). En ambos casos, la imagen se muestra en escala de grises.

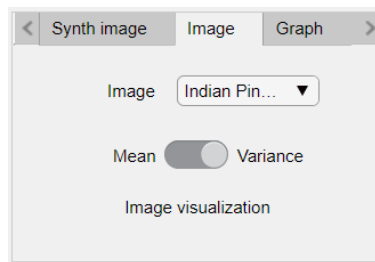


Figura 3.6: Pestaña image

3.2.4. Selector de gráfica

La Pestaña Graph (Figura 3.7) permite seleccionar uno de los tres tipos de gráfica que puede mostrar la aplicación. Las gráficas complementan la información proporcionada por los algoritmos (esto es, el número de endmembers).

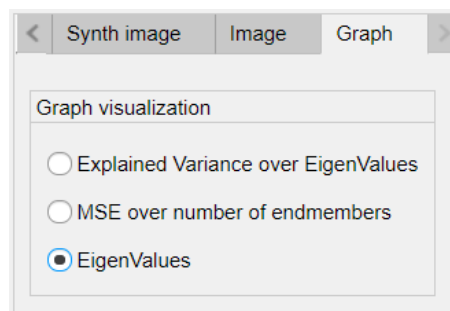


Figura 3.7: Pestaña Graph

Las tres opciones de gráfica a escoger son las siguientes:

- Explained Variance. Esta gráfica representa el porcentaje de varianza sobre el total que se obtiene al seleccionar los k valores propios de mayor valor. En concreto, si λ_k es la secuencia ordenada de valores propios de la matriz de covarianza, la gráfica muestra la cantidad

$$\frac{\sum_{k=1}^K \lambda_k}{\sum_{k=1}^L \lambda_k} \times 100$$

para $K = 1, 2, \dots, 30$. Esta gráfica complementa de forma visual al algoritmo PCA, ya que puede mostrar un codo que permita seleccionar el número de valores propios más representativos.

- MSE over number of endmembers. Esta gráfica complementa visualmente la información que proporciona el algoritmo HySime. La gráfica muestra los tres errores que aparecen en (2.19) en función de k (número de endmembers). En concreto se muestra el error cuadrático medio (representado en negro) como la suma entre el error de proyección (representado en rojo) y la potencia de ruido (representado en azul).
- EigenValues. Esta gráfica muestra los valores de los treinta primeros valores propios de la matriz de autocovarianza de la imagen, ordenados de mayor a menor.

3.3. Tablas de datos

En esta sección se describe las dos tablas que aparecen en el lado izquierdo de la aplicación, debajo de Pestañas de entradas y opciones.

La tabla con datos de la imagen (Figura 3.8) es la tabla superior del programa y muestra las características de la imagen hiperespectral seleccionada. En concreto, esta tabla indica el nombre de la imagen, su número de filas y columnas, el número de bandas (captadas por el sensor), el número de bandas que se usan en el análisis y el número de endmembers (de acuerdo con estudios publicados). Este último dato es vital a la hora de valorar la eficacia de cada algoritmo. El número de bandas utilizado en el análisis suele ser inferior al número de bandas de la imagen porque hay bandas en las que la potencia de ruido es muy grande y su inclusión podría empeorar la estima de K .

| Component | Value |
|-------------|--------------|
| Image Name | Indian Pines |
| Files | 145 |
| Columns | 145 |
| Total Bands | 224 |
| Used bands | 200 |
| Endmembers | 16 |

Figura 3.8: Tabla con datos de la imagen

Cuando la imagen es sintética, la Tabla con datos de la imagen muestra: el nombre de la imagen como *Synthetic*, el número de bandas, el número de endmembers, el número de píxeles, la relación señal a ruido y el valor de η .

La tabla de resultados (Figura 3.9) muestra el número de endmembers que obtiene cada algoritmo (PCA, HFC, NW-HFC e HySime) en la imagen seleccionada a partir de los valores introducidos en Pestaña de entradas (sección 3.2.1). Recordemos que el número real de materiales que se ha identificado en la imagen analizada (de acuerdo con algunos estudios) se muestra en la Tabla con datos de la imagen (Figura 3.8).

| Algorithm | Number of Endmembers |
|-----------|----------------------|
| PCA | 25 |
| HFC | 37 |
| NW-HFC | 37 |
| HySime | 19 |

Figura 3.9: Tabla con los resultados de los cuatro algoritmos

3.4. Imagen y gráfica

La última parte que compone el programa, la parte derecha de la aplicación, muestra la imagen analizada y la gráfica. Cómo se muestra la imagen y la función que en concreto se representa en la gráfica dependen de la selección realizada en Pestañas de entradas y opciones (Sección 3.2) y cambian según la opción elegida. El eje de ordenadas de la gráfica EigenValues está en escala logarítmica debido a la gran diferencia que suele haber entre los primeros y últimos valores propios representados.

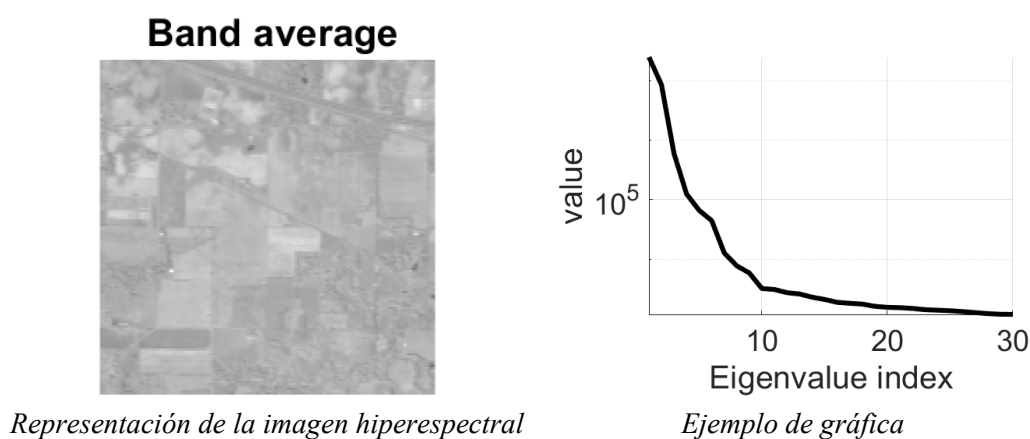


Figura 3.10: Imagen y gráfica en la sección derecha de la aplicación

Capítulo 4

Resultados

En este capítulo se exponen y comentan los resultados obtenidos al estimar el número de end-members de un conjunto de imágenes, tanto sintéticas como reales, utilizando los algoritmos PCA, HFC, NWHFC e HySime. Las implementaciones de los algoritmos HFC y NWHFC son las que se incluyen en [1] mientras que la implementación del algoritmo HySime es la proporcionada en <http://www.lx.it.pt/~bioucas/code.htm>.

4.1. Resultados con imágenes sintéticas

En esta sección se muestran y discuten los resultados obtenidos al estimar el número de end-members en imágenes sintéticas. La ventaja de usar este tipo de imágenes es que siguen el modelo LMM de forma exacta y podemos fijar en ellas el valor de K .

Al igual que en [3] y [5], se han generado imágenes sintéticas con cuatro valores de K (3, 5, 10 y 15), cuatro valores de SNR (50, 35, 25 y 15 dB) y dos valores de η (0 y 1/18). Para cada combinación de K , SNR y η , se han generado 51 imágenes de 10000 píxeles y se ha obtenido \hat{K} en cada una de ellas utilizando PCA, HFC, NWHFC e HySime. En PCA se han considerado tres valores de t (95 %, 99 % y 99.9 %) mientras que en HFC y NWHFC se han considerado tres posibles valores de P_{FA} (10^{-3} , 10^{-4} y 10^{-5}). Las tablas 4.1 y 4.2 muestran la mediana de los 51 valores de \hat{K} y el porcentaje de veces que se ha obtenido dicho valor (mostrado entre paréntesis) para cada combinación de K , SNR, η y algoritmo. La tabla 4.1 muestra los resultados obtenidos cuando $\eta = 0$ (ruido blanco) y la tabla 4.2 muestra los resultados con $\eta = 1/18$.

Analicemos primero los resultados de la tabla 4.1. En todos los algoritmos, al disminuir el SNR o aumentar K , la estima \hat{K} es generalmente menos precisa y tiene una mayor variabilidad (esto es, hay una mayor dispersión entre los 51 valores de \hat{K} obtenidos). HySime proporciona los resultados más precisos y menos variables (únicamente con SNR = 15 dB y $K = 15$ hay grandes diferencias entre \hat{K} y K). La precisión y variabilidad empeoran significativamente en HFC y NWHFC al aumentar K o disminuir la SNR (por ejemplo, con SNR = 15 dB, la mediana de \hat{K} es 3 o 4, independientemente del valor de K). Para cada valor de SNR y K , los algoritmos HFC y NWHFC proporcionan resultados similares y, además, dichos resultados no cambian apreciablemente con P_{FA} . Con PCA, el umbral $t = 99\%$ da buenos resultados cuando SNR = 50 dB; con el resto de valores de SNR, \hat{K} está muy alejado de K y es enorme cuando t es alto o el SNR es bajo (en estos casos, PCA decide que la mayor parte de los valores propios son debidos a componentes de señal).

| SNR | Método | \widehat{K} | | | |
|-------|------------------------------|---------------|------------|------------|------------|
| | | $K = 3$ | $K = 5$ | $K = 10$ | $K = 15$ |
| 50 dB | PCA (95 %) | 3 (55 %) | 3 (69 %) | 3 (61 %) | 4 (55 %) |
| | PCA (99 %) | 3 (92 %) | 4 (71 %) | 6 (45 %) | 6 (37 %) |
| | PCA (99.9 %) | 3 (88 %) | 5 (88 %) | 8 (37 %) | 11 (23 %) |
| | HFC ($P_{FA} = 10^{-3}$) | 3 (80 %) | 4 (43 %) | 7 (31 %) | 9 (18 %) |
| | HFC ($P_{FA} = 10^{-4}$) | 3 (82 %) | 4 (45 %) | 7 (25 %) | 7 (31 %) |
| | HFC ($P_{FA} = 10^{-5}$) | 3 (73 %) | 4 (27 %) | 6 (27 %) | 7 (20 %) |
| | NWHFC ($P_{FA} = 10^{-3}$) | 3 (80 %) | 4 (43 %) | 7 (33 %) | 8 (14 %) |
| | NWHFC ($P_{FA} = 10^{-4}$) | 3 (82 %) | 4 (45 %) | 7 (27 %) | 8 (20 %) |
| | NWHFC ($P_{FA} = 10^{-5}$) | 3 (73 %) | 4 (27 %) | 6 (31 %) | 7 (20 %) |
| | HySime | 3 (100 %) | 5 (100 %) | 10 (98 %) | 15 (92 %) |
| 35 dB | PCA (95 %) | 3 (80 %) | 3 (67 %) | 4 (59 %) | 4 (67 %) |
| | PCA (99 %) | 3 (80 %) | 4 (53 %) | 6 (37 %) | 7 (33 %) |
| | PCA (99.9 %) | 159 (2 %) | 144 (2 %) | 128 (4 %) | 133 (6 %) |
| | HFC ($P_{FA} = 10^{-3}$) | 3 (80 %) | 4 (43 %) | 7 (22 %) | 8 (18 %) |
| | HFC ($P_{FA} = 10^{-4}$) | 3 (78 %) | 4 (49 %) | 7 (24 %) | 7 (14 %) |
| | HFC ($P_{FA} = 10^{-5}$) | 3 (80 %) | 4 (37 %) | 6 (24 %) | 7 (12 %) |
| | NWHFC ($P_{FA} = 10^{-3}$) | 3 (80 %) | 4 (41 %) | 7 (22 %) | 7 (14 %) |
| | NWHFC ($P_{FA} = 10^{-4}$) | 3 (78 %) | 4 (49 %) | 6 (20 %) | 7 (12 %) |
| | NWHFC ($P_{FA} = 10^{-5}$) | 3 (82 %) | 4 (37 %) | 6 (25 %) | 7 (16 %) |
| | HySime | 3 (100 %) | 5 (100 %) | 10 (80 %) | 14 (45 %) |
| 25 dB | PCA (95 %) | 3 (49 %) | 4 (35 %) | 5 (33 %) | 5 (39 %) |
| | PCA (99 %) | 170 (2 %) | 138 (2 %) | 136 (6 %) | 129 (2 %) |
| | PCA (99.9 %) | 218 (16 %) | 217 (6 %) | 215 (14 %) | 214 (18 %) |
| | HFC ($P_{FA} = 10^{-3}$) | 3 (80 %) | 4 (47 %) | 6 (24 %) | 6 (25 %) |
| | HFC ($P_{FA} = 10^{-4}$) | 3 (84 %) | 4 (39 %) | 6 (18 %) | 5 (25 %) |
| | HFC ($P_{FA} = 10^{-5}$) | 3 (90 %) | 5 (51 %) | 6 (20 %) | 5 (14 %) |
| | NWHFC ($P_{FA} = 10^{-3}$) | 3 (80 %) | 4 (45 %) | 6 (25 %) | 6 (25 %) |
| | NWHFC ($P_{FA} = 10^{-4}$) | 3 (86 %) | 4 (39 %) | 6 (18 %) | 5 (22 %) |
| | NWHFC ($P_{FA} = 10^{-5}$) | 3 (90 %) | 5 (51 %) | 6 (22 %) | 5 (16 %) |
| | HySime | 3 (98 %) | 5 (84 %) | 9 (51 %) | 12 (45 %) |
| 15 dB | PCA (95 %) | 182 (2 %) | 167 (2 %) | 166 (4 %) | 166 (4 %) |
| | PCA (99 %) | 216 (6 %) | 215 (12 %) | 214 (10 %) | 222 (16 %) |
| | PCA (99.9 %) | 224 (39 %) | 224 (60 %) | 224 (76 %) | 224 (77 %) |
| | HFC ($P_{FA} = 10^{-3}$) | 3 (80 %) | 4 (31 %) | 4 (27 %) | 4 (33 %) |
| | HFC ($P_{FA} = 10^{-4}$) | 3 (90 %) | 4 (33 %) | 4 (29 %) | 3 (29 %) |
| | HFC ($P_{FA} = 10^{-5}$) | 3 (84 %) | 4 (31 %) | 4 (22 %) | 3 (37 %) |
| | NWHFC ($P_{FA} = 10^{-3}$) | 3 (80 %) | 4 (33 %) | 4 (25 %) | 4 (35 %) |
| | NWHFC ($P_{FA} = 10^{-4}$) | 3 (90 %) | 4 (35 %) | 4 (25 %) | 3 (27 %) |
| | NWHFC ($P_{FA} = 10^{-5}$) | 3 (84 %) | 4 (31 %) | 4 (25 %) | 3 (35 %) |
| | HySime | 3 (96 %) | 4 (49 %) | 7 (59 %) | 8 (47 %) |

Tabla 4.1: Resultados con imágenes sintéticas y $\eta = 0$.

A continuación se comentan los resultados obtenidos con $\eta = 1/18$ (tabla 4.2). La tabla 4.2 no incluye los resultados obtenidos con HFC porque el valor de \widehat{K} que proporciona dicho algoritmo, independientemente de los valores de K y SNR usados al generar la imagen, siempre están muy alejados del valor real. Además, existe una gran variabilidad entre los 51 resultados obtenidos para cada valor de SNR y K . Por ejemplo, con $K = 10$, SNR = 35 dB se obtuvieron valores de \widehat{K} entre 45 y 83 (sin que ninguno de ellos apareciera más de cinco veces). La razón de estos malos resultados es que HFC asume que el ruido es blanco.

Gran parte de los comentarios sobre la tabla 4.1 son aplicables a la tabla 4.2. Así, HySime proporciona resultados generalmente mejores y más estables que NWHFC y PCA. Curiosamente, NWHFC proporciona mejores resultados con $\eta = 1/18$ que con $\eta = 0$. Como ocurría con $\eta = 0$, PCA también proporciona valores enormes de \hat{K} cuando t es grande o la SNR es bajo (aunque dichos valores no son tan grandes como los obtenidos con $\eta = 0$).

| SNR | Método | \hat{K} | | | |
|-------|------------------------------|-----------|-----------|-----------|----------|
| | | $K = 3$ | $K = 5$ | $K = 10$ | $K = 15$ |
| 50 dB | PCA (95%) | 3 (65%) | 3 (69%) | 3 (49%) | 4 (73%) |
| | PCA (99%) | 3 (92%) | 4 (67%) | 6 (41%) | 7 (37%) |
| | PCA (99.9%) | 3 (96%) | 5 (86%) | 8 (45%) | 11 (37%) |
| | NWHFC ($P_{FA} = 10^{-3}$) | 3 (86%) | 4 (47%) | 8 (33%) | 9 (25%) |
| | NWHFC ($P_{FA} = 10^{-4}$) | 3 (88%) | 4 (47%) | 7 (35%) | 9 (25%) |
| | NWHFC ($P_{FA} = 10^{-5}$) | 3 (88%) | 4 (39%) | 6 (29%) | 9 (20%) |
| | HySime | 3 (100%) | 5 (100%) | 10 (100%) | 15 (90%) |
| 35 dB | PCA (95%) | 3 (75%) | 3 (57%) | 4 (53%) | 4 (53%) |
| | PCA (99%) | 3 (67%) | 5 (47%) | 6 (49%) | 7 (35%) |
| | PCA (99.9%) | 47 (6%) | 40 (2%) | 40 (2%) | 45 (12%) |
| | NWHFC ($P_{FA} = 10^{-3}$) | 3 (92%) | 5 (55%) | 7 (25%) | 10 (27%) |
| | NWHFC ($P_{FA} = 10^{-4}$) | 3 (88%) | 5 (55%) | 7 (31%) | 9 (20%) |
| | NWHFC ($P_{FA} = 10^{-5}$) | 3 (86%) | 4 (43%) | 7 (16%) | 8 (16%) |
| | HySime | 3 (100%) | 5 (100%) | 10 (78%) | 14 (37%) |
| 25 dB | PCA (95%) | 3 (57%) | 4 (43%) | 5 (31%) | 5 (33%) |
| | PCA (99%) | 40 (8%) | 43 (6%) | 41 (8%) | 44 (10%) |
| | PCA (99.9%) | 83 (6%) | 82 (6%) | 85 (16%) | 89 (12%) |
| | NWHFC ($P_{FA} = 10^{-3}$) | 3 (84%) | 5 (47%) | 7 (29%) | 9 (16%) |
| | NWHFC ($P_{FA} = 10^{-4}$) | 3 (90%) | 4 (37%) | 7 (24%) | 9 (24%) |
| | NWHFC ($P_{FA} = 10^{-5}$) | 3 (92%) | 4 (51%) | 7 (24%) | 9 (18%) |
| | HySime | 3 (100%) | 5 (89%) | 9 (51%) | 12 (40%) |
| 15 dB | PCA (95%) | 52 (8%) | 48 (6%) | 51 (14%) | 52 (16%) |
| | PCA (99%) | 81 (8%) | 80 (8%) | 80 (8%) | 82 (14%) |
| | PCA (99.9%) | 110 (4%) | 109 (10%) | 111 (12%) | 114 (8%) |
| | NWHFC ($P_{FA} = 10^{-3}$) | 3 (84%) | 4 (35%) | 8 (25%) | 9 (22%) |
| | NWHFC ($P_{FA} = 10^{-4}$) | 3 (88%) | 5 (49%) | 7 (27%) | 9 (18%) |
| | NWHFC ($P_{FA} = 10^{-5}$) | 3 (88%) | 4 (29%) | 7 (25%) | 8 (20%) |
| | HySime | 3 (92%) | 5 (55%) | 6 (45%) | 8 (41%) |

Tabla 4.2: Resultados con imágenes sintéticas y $\eta = 1/18$.

4.2. Resultados con imágenes reales

A continuación se presentan los resultados obtenidos con la aplicación para un conjunto de imágenes hiperespectrales reales de parámetros conocidos. Estas imágenes son públicas y están disponibles para descarga en Internet. En concreto, en este proyecto se han obtenido las imágenes de [3, 7] y se han seleccionado dos subimágenes de la imagen Washington DC Mall de acuerdo al recorte utilizado en [8].

4.2.1. Indian Pines

La imagen Indian Pines es una imagen captada por el sensor AVIRIS al noroeste de Indiana. Se trata de una imagen 145×145 píxeles y 224 bandas en el rango de entre $0,4 \mu\text{m}$ y $2,5 \mu\text{m}$. En esta imagen se han identificado 16 materiales. Los materiales identificados son: alfalfa, maíz sin labrar, maíz poco labrado, maíz, pastos de hierba, hierba-árboles, pastos de hierba cortada, paja secando, avena, soja sin labrar, soja poco labrada, soja, trigo, árboles, una mezcla de edificios-hierba-arboles-vehículos y una mezcla de piedra-acero-torres. La figura 4.1 muestra en escala de grises la imagen generada al promediar todas las bandas de Indian Pines.

Se han analizado dos versiones de Indian Pines:

- Indian Pines 1. En esta versión se han eliminado las bandas [104-108], [150-163] y 220 de la imagen original debido a que son bandas muy ruidosas por hallarse en la región de absorción del agua.
- Indian Pines 2. Además de las bandas eliminadas en Indian Pines 1, en Indian Pines 2 también se han eliminado otras 14 bandas ruidosas.



Figura 4.1: Promedio de las bandas de Indian Pines.

Las tablas 4.3 y 4.4 muestran los resultados en Indian Pines 1 e Indian Pines 2, respectivamente.

| | | | |
|--------|---------------------------|---------------------------|---------------------------|
| PCA | 95 % | 99 % | 99.9 % |
| | 6 | 26 | 70 |
| HFC | $P_{\text{FA}} = 10^{-3}$ | $P_{\text{FA}} = 10^{-4}$ | $P_{\text{FA}} = 10^{-5}$ |
| | 48 | 43 | 37 |
| NWHFC | $P_{\text{FA}} = 10^{-3}$ | $P_{\text{FA}} = 10^{-4}$ | $P_{\text{FA}} = 10^{-5}$ |
| | 26 | 19 | 16 |
| HySime | 18 | | |

Tabla 4.3: Número de endmembers en la imagen Indian Pines 1.

| | | | |
|--------|--------------------|--------------------|--------------------|
| PCA | 95 % | 99 % | 99.9 % |
| | 3 | 4 | 28 |
| HFC | $P_{FA} = 10^{-3}$ | $P_{FA} = 10^{-4}$ | $P_{FA} = 10^{-5}$ |
| | 26 | 23 | 22 |
| NWHFC | $P_{FA} = 10^{-3}$ | $P_{FA} = 10^{-4}$ | $P_{FA} = 10^{-5}$ |
| | 18 | 18 | 18 |
| HySime | 14 | | |

Tabla 4.4: Número de endmembers en la imagen Indian Pines 2.

Los resultados obtenidos en las dos imágenes son muy diferentes y generalmente son mejores en Indian Pines 2 que en Indian Pines 1. De hecho, como muestra la figura 4.2, la secuencia ordenada de valores propios es muy distinta en las dos imágenes. Esto es debido a que la imagen Indian Pines 2 se ha obtenido eliminando 14 bandas muy ruidosas de Indian Pines 1.

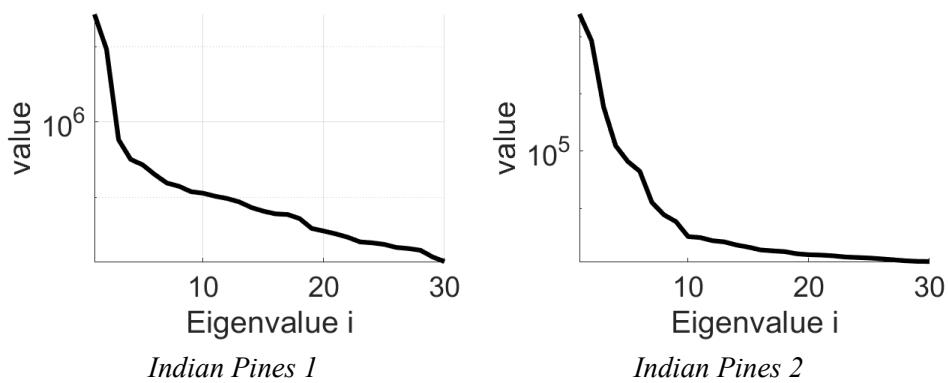


Figura 4.2: Gráficas de valores propios de Indian Pines

4.2.2. Salinas

La imagen Salinas es una imagen tomada por el sensor AVIRIS en el Valle Salinas, en California. Originalmente de 224 bandas, se han eliminado las bandas que coinciden con el ancho de banda de la absorción del agua, siendo las siguientes las bandas [108-112], [154-167] y 224. La imagen posee 16 materiales, incluyendo campos de vegetales, campos vacíos y viñedos.

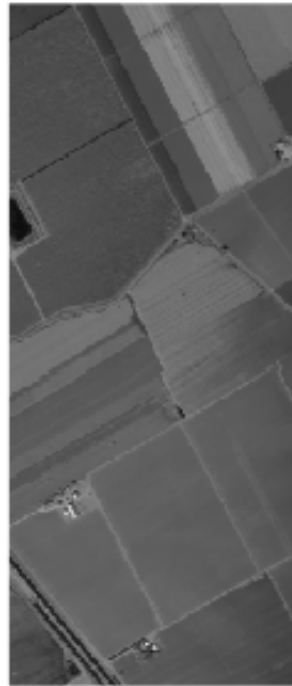


Figura 4.3: Salinas

Analizando la imagen con la aplicación obtenemos los siguientes resultados:

| | | | |
|--------|--------------------|--------------------|--------------------|
| PCA | 95 % | 99 % | 99.9 % |
| | 3 | 4 | 7 |
| HFC | $P_{FA} = 10^{-3}$ | $P_{FA} = 10^{-4}$ | $P_{FA} = 10^{-5}$ |
| | 24 | 23 | 22 |
| NWHFC | $P_{FA} = 10^{-3}$ | $P_{FA} = 10^{-4}$ | $P_{FA} = 10^{-5}$ |
| | 24 | 24 | 22 |
| HySime | 20 | | |

Tabla 4.5: Número de endmembers en la imagen Salinas.

El resultado que arroja el algoritmo PCA es un resultado tan reducido debido a que la inmensa mayoría de su varianza es explicada con los primeros 4 valores propios, como se puede apreciar en la gráfica Varianza explicada en Salinas

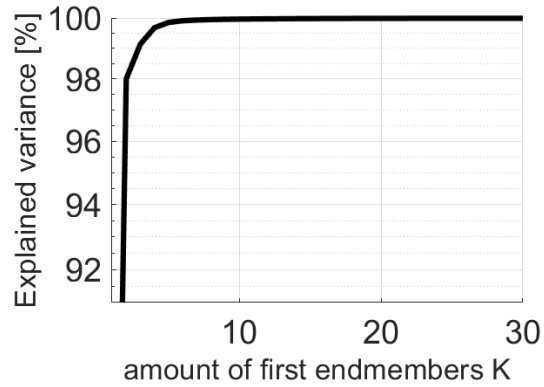


Figura 4.4: Varianza explicada en Salinas

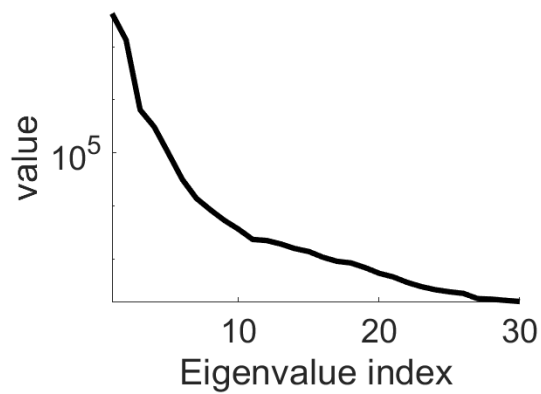


Figura 4.5: Gráfica de valores propios de la imagen Salinas

4.2.3. Salinas-A

La imagen Salinas-A es una subimagen de la imagen Salinas. Originalmente de 224 bandas, se han eliminado las bandas que coinciden con el ancho de banda de la absorción del agua, siendo las siguientes las bandas [108-112], [154-167] y 224. La siguiente imagen posee 6 materiales.

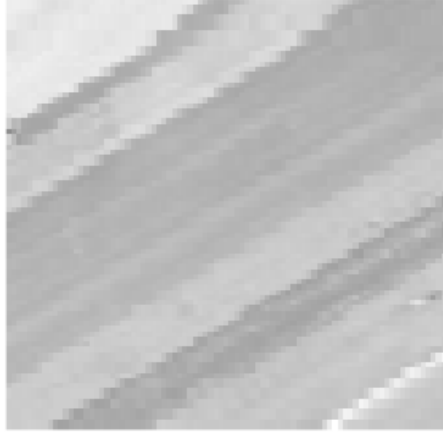


Figura 4.6: Salinas-A

Analizando la imagen con la aplicación obtenemos los siguientes resultados:

| | | | |
|--------|--------------------|--------------------|--------------------|
| PCA | 95% | 99% | 99.9% |
| | 3 | 4 | 7 |
| HFC | $P_{FA} = 10^{-3}$ | $P_{FA} = 10^{-4}$ | $P_{FA} = 10^{-5}$ |
| | 12 | 10 | 8 |
| NWHFC | $P_{FA} = 10^{-3}$ | $P_{FA} = 10^{-4}$ | $P_{FA} = 10^{-5}$ |
| | 11 | 11 | 10 |
| HySime | 12 | | |

Tabla 4.6: Número de endmembers en la imagen Salinas-A.

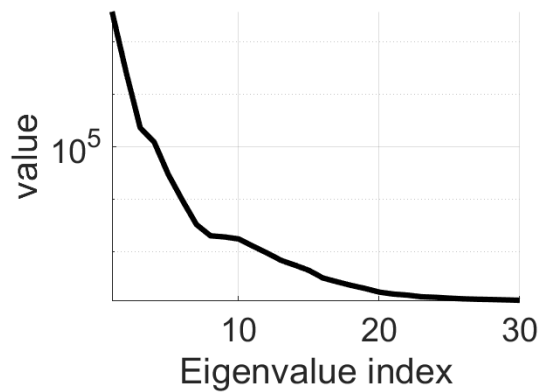


Figura 4.7: Gráfica de valores propios de la imagen Salinas-A

4.2.4. Pavia University

La imagen Pavia University es una imagen tomada por el sensor ROSIS en la región de Pavia, en el Norte de Italia. La imagen posee 103 bandas, de las cuales ninguna ha sido eliminada y en la que se distinguen 9 materiales diferentes.



Figura 4.8: Pavia University

Analizando la imagen con la aplicación obtenemos los siguientes resultados:

| | | | |
|--------|--------------------|--------------------|--------------------|
| PCA | 95 % | 99 % | 99.9 % |
| | 4 | 5 | 17 |
| HFC | $P_{FA} = 10^{-3}$ | $P_{FA} = 10^{-4}$ | $P_{FA} = 10^{-5}$ |
| | 16 | 14 | 13 |
| NWHFC | $P_{FA} = 10^{-3}$ | $P_{FA} = 10^{-4}$ | $P_{FA} = 10^{-5}$ |
| | 17 | 14 | 12 |
| HySime | 60 | | |

Tabla 4.7: Número de endmembers en la imagen Pavia University.

Como se observa en los resultados, hay diferencias muy dispares entre los endmembers que detecta PCA e HySime. Por el lado de PCA, el resultado se debe al codo que se observa en la gráfica Varianza explicada Pavia University (Figura 4.9). Por el lado de HySime, el resultado tan elevado se debe a que detecta muchos espectros diferentes que, aunque algunos se asemejen, siguen siendo lo suficientemente diferenciables como para que los clasifique como diferentes. Un ejemplo pueden ser los diferentes tejados de la imagen.

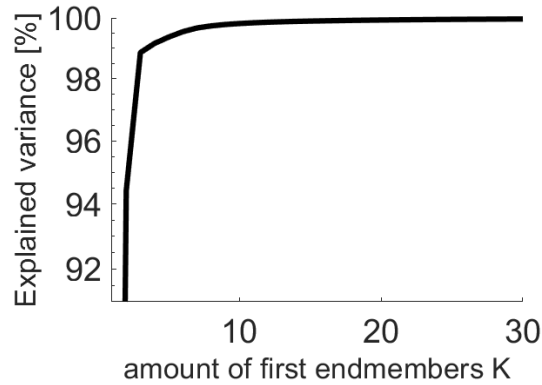


Figura 4.9: Varianza explicada Pavia University

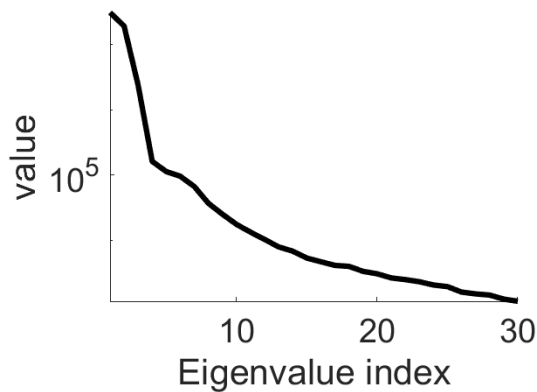


Figura 4.10: Gráfica de valores propios de la imagen Pavia University

4.2.5. Urban

La imagen Urban es una de las imágenes hiperespectrales más utilizadas. La siguiente imagen posee 210 bandas entre las amplitudes $0,4 \mu\text{m}$ y $2,5 \mu\text{m}$. Las bandas [1-4], 76, 87, [101-111], [136-153] y [198-210] han sido eliminadas debido a las interferencias de los efectos atmosféricos y el vapor de agua. El número de materiales encontrados en la imagen se halla en el rango de 4 a 6, siendo éstos asfalto, hierba, arboles, tejado en todas las versiones e incluyendo, en algunas, tierra como quinto material y, en otras, metal como sexto.



Figura 4.11: Urban

Analizando la imagen con la aplicación obtenemos los siguientes resultados:

| | | | |
|--------|---------------------------|---------------------------|---------------------------|
| PCA | 95 % | 99 % | 99.9 % |
| | 4 | 5 | 16 |
| HFC | $P_{\text{FA}} = 10^{-3}$ | $P_{\text{FA}} = 10^{-4}$ | $P_{\text{FA}} = 10^{-5}$ |
| | 50 | 37 | 29 |
| NWHFC | $P_{\text{FA}} = 10^{-3}$ | $P_{\text{FA}} = 10^{-4}$ | $P_{\text{FA}} = 10^{-5}$ |
| | 37 | 35 | 32 |
| HySime | 27 | | |

Tabla 4.8: Número de endmembers en la imagen Urban.

Destacan el número de endmembers que reconocen los algoritmos HFC, NWHFC y HySime. Todos ellos muy superiores al rango de 4 a 6 materiales que en un principio posee la imagen. El motivo es que hay materiales con diferencias entre sus espectros. Por ejemplo, es posible que el material «tejado» este formado por diferentes tipos de tejados, con espectros que, aunque sean similares, son suficientemente distintos como para que los reconozcan por separado los algoritmos.

Por otro lado, el algoritmo PCA sí que muestra unos resultados que coinciden con el número de materiales previamente comentados. Este resultado se puede observar en la gráfica Varianza explicada Urban (Figura 4.12) al observarse el codo en la región de 5 endmembers previamente mencionada.

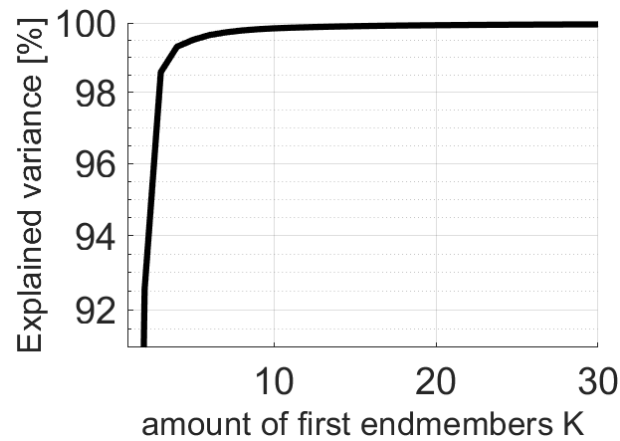


Figura 4.12: Varianza explicada Urban

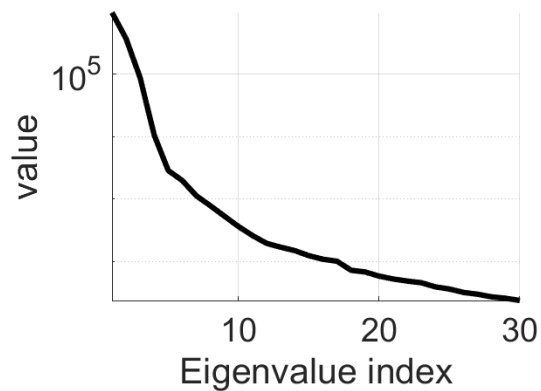


Figura 4.13: Gráfica de valores propios de la imagen Urban

4.2.6. Cuprite

La imagen Cuprite es una imagen captada por el sensor AVIRIS de las Cuprite Hills en Nevada, Estados Unidos. Inicialmente de 224 bandas en el rango $0,37 \mu\text{m}$ a $2,48 \mu\text{m}$, se eliminan las bandas [1-2] y [221-224] por ser muy ruidosas y las bandas [104-113] y [148-167] por ser las bandas de absorción del agua. La imagen posee 14 materiales distintos pero debido a las pequeñas diferencias entre variantes del mismo material ese número se reduce a 12.

En el caso de nuestro estudio, tenemos Cuprite 1 y Cuprite 2. La diferencia entre las imágenes es el recorte utilizado, siendo ligeramente diferente entre ambos, como se puede apreciar en las imágenes Cuprite (Figura 4.14).

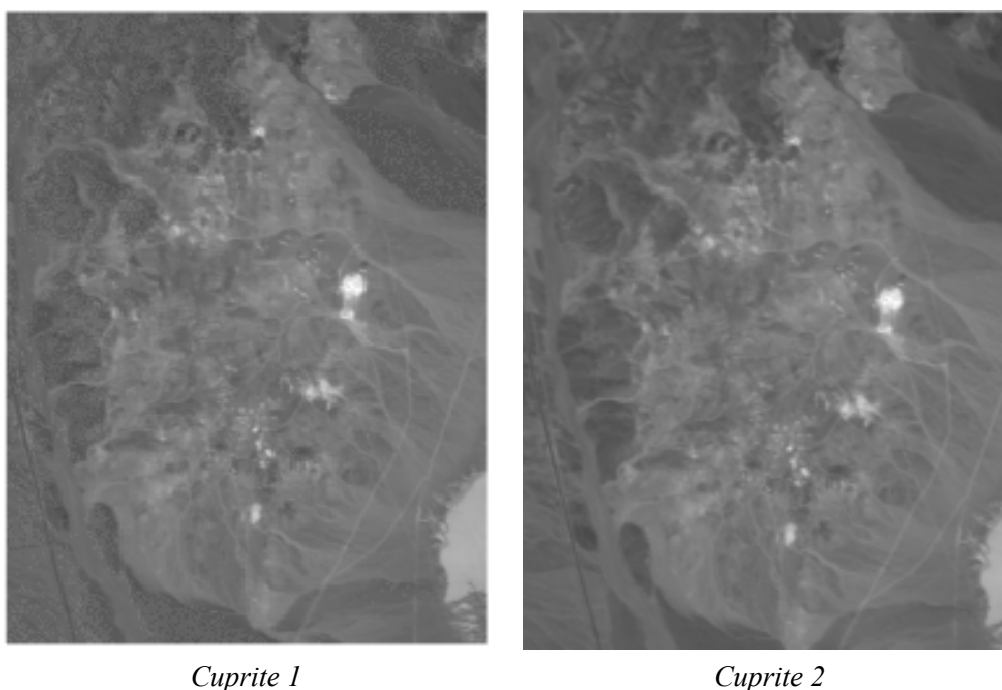


Figura 4.14: Cuprite

Analizando las imágenes con la aplicación obtenemos los siguientes resultados:

| | | | |
|--------|---------------------------|---------------------------|---------------------------|
| PCA | 95 % | 99 % | 99.9 % |
| | 3 | 4 | 6 |
| HFC | $P_{\text{FA}} = 10^{-3}$ | $P_{\text{FA}} = 10^{-4}$ | $P_{\text{FA}} = 10^{-5}$ |
| | 22 | 17 | 17 |
| NWHFC | $P_{\text{FA}} = 10^{-3}$ | $P_{\text{FA}} = 10^{-4}$ | $P_{\text{FA}} = 10^{-5}$ |
| | 17 | 16 | 15 |
| HySime | 13 | | |

Tabla 4.9: Número de endmembers en la imagen Cuprite 1.

| | | | |
|--------|--------------------|--------------------|--------------------|
| PCA | 95 % | 99 % | 99.9 % |
| | 3 | 5 | 18 |
| HFC | $P_{FA} = 10^{-3}$ | $P_{FA} = 10^{-4}$ | $P_{FA} = 10^{-5}$ |
| | 20 | 17 | 16 |
| NWHFC | $P_{FA} = 10^{-3}$ | $P_{FA} = 10^{-4}$ | $P_{FA} = 10^{-5}$ |
| | 17 | 15 | 14 |
| HySime | 18 | | |

Tabla 4.10: Número de endmembers en la imagen Cuprite 2.

La diferencia entre los resultados es, como se puede observar en la imagen Cuprite, debido a que no estamos analizando la misma imagen, con lo que es posible que aparezcan nuevos materiales en la segunda imagen; además de un cambio en los valores propios entre ambas imágenes como se observa en Valores propios de Cuprite (Figura 4.15).

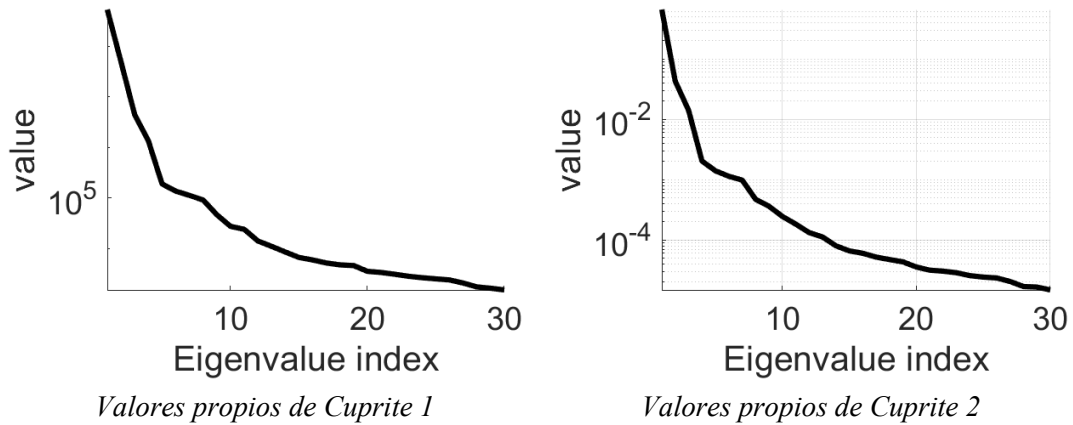


Figura 4.15: Valores propios de Cuprite

4.2.7. Jasper Ridge

Esta imagen es una pequeña porción de la imagen original de Jasper Ridge, una reserva natural en las montañas de Santa Cruz, en Estados Unidos, de 512×614 píxeles de la que se ha seleccionado una sección de 100×100 píxeles debido a que la imagen completa es demasiado compleja como para obtener el número correcto de materiales. Originalmente de 224 bandas entre los $0,38 \mu\text{m}$ y $2,5 \mu\text{m}$, se han eliminado las bandas [1-3], [108-112], [154-166] y [220-224] por ser bandas ruidosas debido al vapor de agua y los efectos atmosféricos. En la imagen existen cuatro materiales: carretera, tierra, agua y árbol.

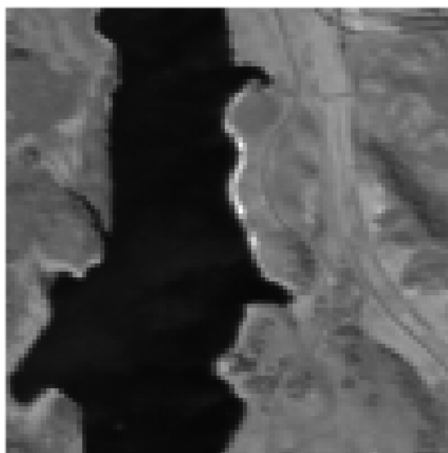


Figura 4.16: Jasper Ridge

Analizando la imagen con la aplicación obtenemos los siguientes resultados:

| | | | |
|--------|---------------------------|---------------------------|---------------------------|
| PCA | 95 % | 99 % | 99.9 % |
| | 3 | 4 | 9 |
| HFC | $P_{\text{FA}} = 10^{-3}$ | $P_{\text{FA}} = 10^{-4}$ | $P_{\text{FA}} = 10^{-5}$ |
| | 9 | 9 | 7 |
| NWHFC | $P_{\text{FA}} = 10^{-3}$ | $P_{\text{FA}} = 10^{-4}$ | $P_{\text{FA}} = 10^{-5}$ |
| | 12 | 10 | 9 |
| HySime | 18 | | |

Tabla 4.11: Número de endmembers en la imagen Jasper Ridge.

Los resultados muestran que PCA hace una buena labor prediciendo el número de materiales. Se puede comprobar con la gráfica Varianza explicada Jasper Ridge (Figura 4.17), ya que aparece un codo alrededor del valor 4. Por otro lado los resultados tan altos de HySime se pueden deber a la diferencia entre espectros de diferentes variedades de un mismo material que, aunque reducidas, siguen siendo suficientes como para que el algoritmo los detecte como endmembers diferentes.

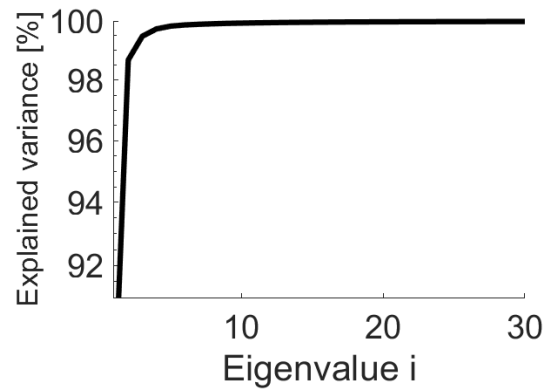


Figura 4.17: Varianza explicada Jasper Ridge

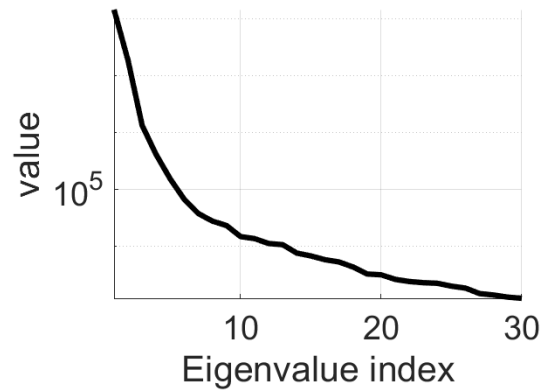


Figura 4.18: Gráfica de valores propios de la imagen Jasper Ridge

4.2.8. Samson

La imagen Samson es una región de 95×95 píxeles de la imagen hiperespectral original que es demasiado grande como para poder extraer el número exacto de materiales. La imagen posee 156 bandas, las cuales se han utilizado todas. En la imagen se aprecian tres materiales: tierra, árbol y agua.

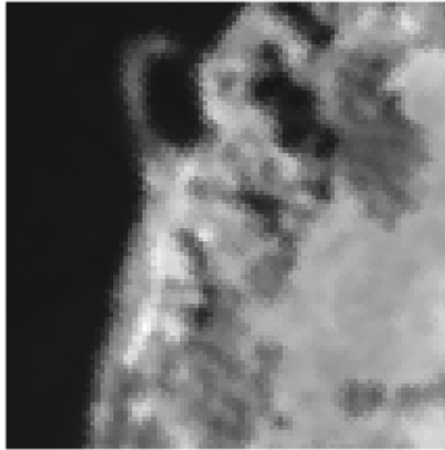


Figura 4.19: Samson

Analizando la imagen con la aplicación obtenemos los siguientes resultados:

| | | | |
|--------|--------------------|--------------------|--------------------|
| PCA | 95 % | 99 % | 99.9 % |
| | 3 | 3 | 5 |
| HFC | $P_{FA} = 10^{-3}$ | $P_{FA} = 10^{-4}$ | $P_{FA} = 10^{-5}$ |
| | 9 | 8 | 8 |
| NWHFC | $P_{FA} = 10^{-3}$ | $P_{FA} = 10^{-4}$ | $P_{FA} = 10^{-5}$ |
| | 8 | 8 | 6 |
| HySime | 43 | | |

Tabla 4.12: Número de endmembers en la imagen Samson.

En los resultados obtenidos destaca el número de endmembers que detecta HySime. Esto puede ser debido a la diferencia entre espectros de diferentes variedades de un mismo material que, aunque reducidas, siguen siendo suficientes como para que el algoritmo los detecte como endmembers diferentes.

4.2.9. Washington DC Mall

La imagen Washington DC Mall es una imagen del national mall de Washington DC. Capital de los Estados Unidos. Se analizan dos secciones de la imagen original llamados WDM1 (de dimensiones 150×150) y WDM2 (de dimensiones 180×160). La imagen original posee 210 bandas, de las cuales se eliminan 19 bandas ruidosas para un total de 191 bandas diferentes a analizar. En ambas imágenes se detectan 6 materiales diferentes.

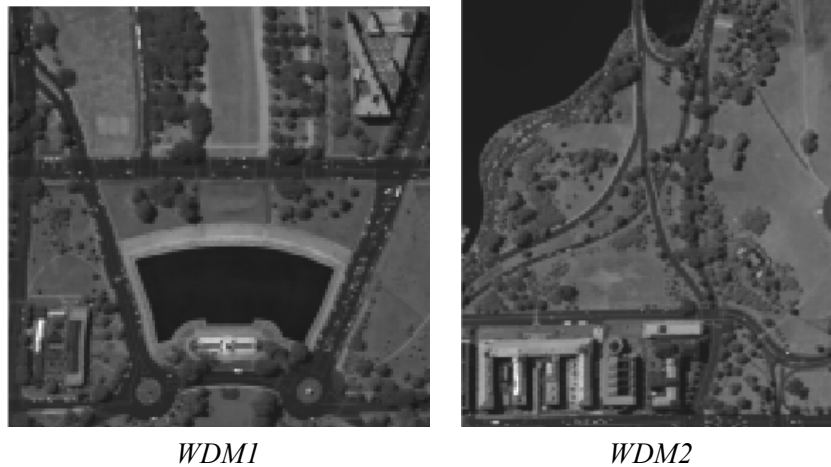


Figura 4.20: Los recortes de Washington DC Mall utilizados

Analizando la imagen con la aplicación obtenemos los siguientes resultados:

| | | | |
|--------|--------------------|--------------------|--------------------|
| PCA | 95 % | 99 % | 99.9 % |
| | 3 | 4 | 7 |
| HFC | $P_{FA} = 10^{-3}$ | $P_{FA} = 10^{-4}$ | $P_{FA} = 10^{-5}$ |
| | 13 | 10 | 9 |
| NWHFC | $P_{FA} = 10^{-3}$ | $P_{FA} = 10^{-4}$ | $P_{FA} = 10^{-5}$ |
| | 21 | 18 | 18 |
| HySime | 21 | | |

Tabla 4.13: Número de endmembers en la imagen WDM1.

| | | | |
|--------|--------------------|--------------------|--------------------|
| PCA | 95 % | 99 % | 99.9 % |
| | 3 | 4 | 8 |
| HFC | $P_{FA} = 10^{-3}$ | $P_{FA} = 10^{-4}$ | $P_{FA} = 10^{-5}$ |
| | 17 | 15 | 15 |
| NWHFC | $P_{FA} = 10^{-3}$ | $P_{FA} = 10^{-4}$ | $P_{FA} = 10^{-5}$ |
| | 29 | 26 | 25 |
| HySime | 21 | | |

Tabla 4.14: Número de endmembers en la imagen WDM2.

4.2.10. Conclusiones

En las secciones anteriores se han mostrado los resultados obtenidos al aplicar los algoritmos a cada una de las imágenes reales consideradas en este trabajo. En esta sección, se comentan dichos resultados y se sacan conclusiones sobre la eficacia de los algoritmos en cada imagen.

Cabe destacar que, aunque en las imágenes sintéticas el algoritmo que mejores resultados daba era HySime, en las imágenes reales se observa que NWHFC consigue resultados mejores o muy parecidos a esta última. Por tanto, en la práctica no hay un algoritmo superior al otro a la hora de analizar las imágenes.

También es necesario comentar que, aunque la probabilidad de falsa alarma P_{FA} es introducida por el usuario, un valor de $P_{FA} = 10^{-5}$ es el valor que mejores resultados aporta tanto en HFC como en NWHFC cuando se analizan imágenes reales.

Relacionado con el anterior punto, es destacable que, mientras que HySime, HFC y NWHFC proporcionan valores más bajos en el número de materiales detectados a los reales en imágenes sintéticas, ocurre lo contrario con imágenes reales, mostrando valores superiores a los propuestos.

Además, comentar que una explicación posible al número de endmembers detectados por los algoritmos puede deberse a la existencia de distintas variedades en un mismo material. Estas variaciones tendrían espectros similares pero lo suficientemente distintos como para que los algoritmos los detectaran como endmembers diferentes.

En la imagen Indian Pines (Sección 4.2.1) se observa que los algoritmos que mejor se aproximan a los materiales propuestos son HySime y NWHFC. Mientras que HFC y PCA obtienen resultados bastante alejados a los propuestos. El problema de PCA a la hora de determinar el número de endmembers se observa en las Gráficas de valores propios de Indian Pines (Figura 4.2) ya que no aparece un codo fácilmente distinguible, con lo que no podemos seleccionar un buen corte. Por parte de HFC, al observar que NWHFC sí que consigue resultados cercanos a los materiales determinados, podemos suponer que se debe a que el ruido que aparece en la imagen no es cercano a un ruido blanco.

En la imagen Salinas (Sección 4.2.2) observamos resultados aceptables con HFC, NWHFC e HySime. El ruido de la imagen, por tanto, se presupone cercano a blanco. Por el lado de PCA, como se ha explicado en la propia subsección, se debe a que casi toda la varianza de la imagen es explicada con los primeros valores propios.

En la imagen Salinas-A (Sección 4.2.3) se observa que los resultados de todos los algoritmos, especialmente PCA y HFC. Las diferencias de NWHFC e HySime con los materiales propuestos pueden ser debidas a sus mecanismos de blanqueo de ruido.

En la imagen Pavia University (Sección 4.2.4) se observa que PCA, HFC y NWHFC proporcionan resultados similares mientras que HySime proporciona un número de endmembers mucho mayor. Comparando estos resultados con los obtenidos en la sección Resultados con imágenes sintéticas (Sección 4.1), puede deberse a un número de endmembers elevado junto a un SNR moderado. Los valores superiores a los propuestos se pueden explicar por las distintas variaciones de un mismo material.

En la imagen Urban (Sección 4.2.5) se observa que HFC, NWHFC e HySime obtienen resultados similares y todos lo hacen por encima del margen de materiales propuestos para la imagen. Como se ha comentado en la propia sección, dichos valores elevados pueden ser debidos a las variedades del mismo material. Aun así, PCA si que hace un buen trabajo a la hora de detectar el número de materiales, lo cual se observa en la gráfica Varianza explicada Pavia University (Figura 4.9) por la aparición de un codo distinguible en esa región.

En la imagen Cuprite (Sección 4.2.6) se observa que todos los algoritmos dan un resultado similar y aproximado a los materiales propuestos en la subimagen Cuprite 2. Por otro lado, pasa algo similar en la subimagen Cuprite 1 con la excepción de PCA con un margen de 99.9%. Aun así, observando en Valores propios de Cuprite (Figura 4.15) que ambas gráficas tienen una forma muy parecida, es posible que el resultado obtenido sea debido al margen escogido.

En la imagen Jasper Ridge (Sección 4.2.7) se observa que el algoritmo que mejor se aproxima al número de materiales propuestos es PCA con un margen del 99%. Este valor se observa con claridad en la gráfica Varianza explicada Jasper Ridge (Figura 4.17) con el codo que aparece en el valor 4. Los otros resultados, por tanto, pueden ser explicados por las distintas variedades de un mismo material.

En la imagen Samson (Sección 4.2.8) se observa que PCA es el algoritmo que mejor se aproxima al número de materiales propuestos. Por otro lado, NWHFC y HFC consiguen resultados algo elevados pero cercanos a los materiales propuestos. La diferencia significativa se observa en HySime, que detecta 43 endmembers, número muy superior a los 3 materiales propuestos.

En las imágenes de Washington DC Mall (Sección 4.2.9) se observa que el algoritmo que mejor se aproxima a los materiales propuestos se trata de PCA. La gran diferencia entre los resultados de HFC y NWHFC puede indicar que el ruido de la imagen no es aproximadamente ruido blanco y, por tanto, es posible que el resultado de HFC no muestre bien el número verdadero de endmembers que componen la imagen. HySime por otro lado muestra que hay 21 endmembers diferentes en ambas imágenes, las cuales comparten materiales como agua, suelo, arboles, asfalto y tejados. Los valores similares proporcionados por tanto HySime como NWHFC puede indicar que el verdadero número de endmembers en la imagen sea cercano a esos valores y que se deba a las variaciones existentes en un mismo material.

Capítulo 5

Conclusiones y líneas futuras

En este capítulo se resume de forma breve el contenido de la presente memoria y se proporcionan unas conclusiones generales sobre el proyecto. Además, se comentarán posibles líneas futuras a realizar.

5.1. Conclusiones

A través de los distintos capítulos de este documento se ha proporcionado una pequeña iniciación al tema de las imágenes hiperespectrales en Fundamentos teóricos (Capítulo 2). A demás, se introduce al lector al modelo LMM, que es la base de los algoritmos utilizados en el proyecto.

A continuación, esta memoria describe de forma breve cada uno de los algoritmos utilizados durante este proyecto, siendo estos PCA (Sección 2.3), HFC y NWHFC (Sección 2.4), e HySime (Sección 2.5).

A partir de estos algoritmos se ha programado una aplicación en lenguaje M utilizando la aplicación de App Designer de MATLAB, cuya guía de usuario se encuentra en Guía de usuario (Capítulo 3).

Con el uso de esa aplicación se ha analizado el funcionamiento de cada uno de los algoritmos frente a determinadas imágenes hiperespectrales sintéticas en Resultados con imágenes sintéticas (Sección 4.1), para comprobar cómo afecta cada parámetro de la imagen a los resultados predichos por cada algoritmo. Después de analizar cómo se comporta cada uno de los distintos algoritmos frente a los cambios de los parámetros se ha analizado una librería de imágenes hiperespectrales reales cuyos resultados se encuentran en Resultados con imágenes reales (Sección 4.2), así como una pequeña introducción a cada una de las imágenes y, si cabe, algunos comentarios de interés. Finalmente se han analizado los resultados obtenidos por los algoritmos en las imágenes reales y se ha intentado explicar los paralelismos con los resultados de las imágenes sintéticas y una posible explicación a las discrepancias entre resultados, ya sea entre los propios algoritmos o con el número de materiales propuestos en la imagen.

5.2. Líneas futuras

Otras mejoras que se podrían haber realizado en este proyecto son:

- Buscar y comparar más métodos de blanqueo de ruido para el algoritmo NWHFC y escoger el que mejor resultados proporcione.
- Incrementar la librería de imágenes reales disponibles.
- Dar al usuario la posibilidad de introducir sus propias imágenes hiperespectrales para analizarlas con la aplicación.
- Optimizar la aplicación para que se actualice automáticamente el resultado al introducir nuevos valores de P_{FA} y margen de corte.
- Mejorar el algoritmo que permite mostrar las imágenes hiperespectrales sintéticas.
- Intentar conocer por qué en ciertas imágenes los resultados de los algoritmos son muy superiores a los propuestos.

Bibliografía

- [1] Chein-I Chang. *Hyperspectral data processing. Algorithm design and analysis*. Wiley, 2013.
- [2] Chein-I Chang y Qian Du. “Estimation of number of spectrally distinct signal sources in hyperspectral imagery”. En: *IEEE Transactions on Geoscience and Remote Sensing* 42.3 (2004), págs. 608-619.
- [3] José M. Bioucas-Dias y José M. P. Nascimento. “Hyperspectral subspace identification”. En: *IEEE Transactions on Geoscience and Remote Sensing* 46.8 (2008), págs. 2435-2445.
- [4] Kerry Cawse-Nicholson y col. “Determining the intrinsic dimension of a hyperspectral image using random matrix theory”. En: *IEEE Transactions on Image Processing* 22.4 (2013), págs. 1301-1310.
- [5] Abderrahim Halimi y col. “Estimating the intrinsic dimension of hyperspectral images using noise-whitened eigengap approach”. En: *IEEE Transactions on Geoscience and Remote Sensing* 54.7 (2016), págs. 3811-3820.
- [6] R. Roger. “Principal components transform with simple, automatic noise adjustment”. En: *International Journal of Remote Sensing* 17.14 (1996), págs. 2719-2727.
- [7] Remote Sensing Laboratory. *Remoting sensing dataset*. L imagenes estan disponibles en: <https://rslab.ut.ac.ir/data>. Accessed May 27th, 2020. 2015.
- [8] Feiyun Zhu. *Hyperspectral Unmixing: Ground Truth Labeling, Datasets, Benchmark Performances and Survey*. 2017. eprint: arXiv:1708.05125.